The Incremental Unknowns - a multilevel scheme for the simulation of turbulent channel flows

By M. Chen[1], H. Choi[2], T. Dubois[3], J. Shen[1] AND R. Temam[4]

In numerical simulation of complex flows, it is important to identify different length scales of the flow and treat them differently. In this report, we introduce a new multilevel scheme for simulating turbulent channel flows. Two different versions of the scheme, namely the spectral and finite difference versions, are presented. The spectral version of the scheme is based on a spectral-Galerkin formulation which provides a natural decomposition of the flow into small and large wavelength parts, and which leads to linear systems that can be solved with quasi-optimal computational complexity. In the finite difference version, the "Incremental Unknown" (IU) is used to separate the length scales. Preliminary numerical results indicate that the scheme is well suited for turbulence computations and provides results which are comparable to that by Direct Numerical Simulation but with significantly less CPU time.

1. Motivation

The numerical simulation of turbulent flows is an extremely challenging task for both the numerical analysts and computational fluid dynamicists. The computing power required to resolve the enormous number of degrees of freedom and their nonlinear interactions involved in a turbulent flow is often near or beyond reach of the current computer capacity so that conventional numerical schemes are often impractical for turbulence simulations.

The aim of this paper is to introduce a new multilevel scheme which is based on a differentiated treatment for small and large wavelength parts. It is well known in turbulence theory that the large number of small wavelengths only carry a small part of the total kinetic energy of the flow, however, the effect of their nonlinear interactions with large wavelengths over a long term integration can not be neglected and must be adequately resolved. Nevertheless, the small wavelength part, especially their nonlinear interactions do not need to be represented in the same accuracy as the large wavelength part. Our multilevel scheme is specially designed such that it would produce results comparable to that by Direct Numerical Simulation (DNS) but at significantly less cost so that one can simulate more complicated

1 Department of mathematics, Penn State University, University Park, PA 16802
2 Department of Mechanical Engineering, Seoul National University, Seoul 151-742, Korea
3 Lab. de Mathématiques Appliquées, Université Blaise Pascal & CNRS, 63177 Aubière, France
4 Lab. d'Analyse Numérique, Université de Paris-Sud, 91405 Orsay, France; and Institute of Scientific Computing and Applied Mathematics, Indiana University, Bloomington, IN 47405

flows with limited capability of the computer. The method can be applied to a class of dissipative equations, and can be combined with a large number of existing numerical methods.

The method starts with separating the length scales of the solution u as

$$u = f + g + r$$

where f is the large length scale, g is the intermediate length scale and r is the small scale. Then the different scales of the solution are treated differently which could involve (a) neglecting some higher-order terms involving the small scales, (b) updating small scales with lager time interval. The effect of these further approximations would, if done correctly, reduce the CPU time for each time step, improve the stability (the CFL condition will be only related to the large wavelengths) — allow larger time steps.

There are two ways to look at this method. One is that we neglect some effect of the small scale terms. Another way is we think that large scale approximation is not enough, so we are taking account the effect of small scale terms in an efficient way, instead of simply adding more mesh points.

This method has been applied to the simulation of 2D and 3D forced homogeneous turbulence (see Dubois, Jauberteau and Temam 1995a, 1995b, 1996 and the references therein). In the 3D case, it has been shown that the main statistical properties of homogeneous turbulence is well predicted with multilevel schemes. Indeed, while a saving in CPU time of 50 to 75 percent versus a classical Galerkin method is obtained, the energy and enstrophy spectra as well as the high-order moments of the velocity and its derivatives are accurately computed. The comparison of these results has been done with the results of direct simulations.

In the case of homogeneous turbulence, when Fourier expansion of the velocity is used, the separation of the flow into large and small scales is trivial. However, this is not obvious for the channel flow problem because of the no-slip boundary conditions at the walls. In particular, the popular spectral-tau (Gottlieb & Orszag 1977) method is not suitable for this purpose. We shall use the spectral-Galerkin method developed in Shen (1994, 1995) for the non-homogeneous direction. This spectral-Galerkin formulation not only provides a natural decomposition of the flow into small and large wavelength parts, but also leads to linear systems that can be solved with quasi-optimal computational complexity.

In the finite difference case, we will use the Incremental Unknowns (IU) developed by Chen and Temam (1991). The Incremental unknown method has been used for steady equations, and the result is similar to preconditioning the associated matrix. The scheme was shown theoretically convergent and has an improved efficiency (Chen and Temam 1993). Here for the first time, the IU method is applied to unsteady problems.

This report is an interim report: more detailed results using the new scheme for the turbulent channel flows will be reported later.

2. Incremental unknowns in the spectral case

2.1 Formulation of the equations

We consider the Navier-Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} - \nu \Delta \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \frac{1}{\rho}\nabla P = 0, \tag{2.1}$$

$$\text{div } \boldsymbol{u} = 0, \tag{2.2}$$

in a channel $\Omega = (0, L_x) \times (-1, 1) \times (0, L_z)$ with the boundary conditions: $\boldsymbol{u} = (u, v, w)$ is periodic in x and z, and no slip on the walls. For this channel flow, we assume that the pressure P takes the form $P = \tilde{P} + K_P\ x$, where $\tilde{P}$ is periodic in directions x and z and K_P is a given constant.

Following Kim, Moin and Moser (1987), we set

$$\begin{aligned}
\Lambda &= (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = (\Lambda_x, \Lambda_y, \Lambda_z), \\
f &= \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}, \\
g &= \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \\
h_v(\boldsymbol{u}, \boldsymbol{u}) &= \frac{\partial}{\partial x}\left(\frac{\partial \Lambda_x}{\partial y} - \frac{\partial \Lambda_y}{\partial x}\right) - \frac{\partial}{\partial z}\left(\frac{\partial \Lambda_y}{\partial z} - \frac{\partial \Lambda_z}{\partial y}\right), \\
h_g(\boldsymbol{u}, \boldsymbol{u}) &= -\left(\frac{\partial \Lambda_x}{\partial z} - \frac{\partial \Lambda_z}{\partial x}\right) = -(\boldsymbol{u} \cdot \nabla) g + g\ \frac{\partial v}{\partial y} + \frac{\partial v}{\partial x}\ \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\ \frac{\partial u}{\partial y},
\end{aligned} \tag{2.3}$$

then, (2.1)–(2.2) are equivalent to the following equations (cf. Kim et al. 1987):

$$\begin{aligned}
&\frac{\partial}{\partial t}\Delta v - \nu \Delta^2 v = h_v(\boldsymbol{u}, \boldsymbol{u}), \\
&\frac{\partial g}{\partial t} - \nu \Delta g = h_g(\boldsymbol{u}, \boldsymbol{u}), \\
&f + \frac{\partial v}{\partial y} = 0.
\end{aligned} \tag{2.4}$$

From the boundary conditions of $\boldsymbol{u}$ and the continuity equation (2.2) we deduce boundary conditions for v and g:

$$v(x, \pm 1, z, t) = \frac{\partial}{\partial y} v(x, \pm 1, z, t) = 0,$$

$$g(x, \pm 1, z, t) = 0.$$

We emphasize that $h_v(\cdot, \cdot)$ and $h_g(\cdot, \cdot)$ are indeed bilinear forms since they are derived from the original bilinear form by linear differential operations.

Writing the Fourier expansion in directions x and z for $\boldsymbol{u}$

$$\boldsymbol{u}(\boldsymbol{x},t)=\sum_{\boldsymbol{k}\in\mathcal{Z}^2}\hat{\boldsymbol{u}}_{\boldsymbol{k}}(y,t)e^{i\left(k_x\frac{2\pi}{L_x}x+k_z\frac{2\pi}{L_z}z\right)},\ \boldsymbol{k}=(k_x,k_z),$$

where $\hat{\boldsymbol{u}}_{\boldsymbol{k}}=(\hat{u}_{\boldsymbol{k}},\hat{v}_{\boldsymbol{k}},\hat{w}_{\boldsymbol{k}})$, and similarly for f, g and h_v, h_g, we derive from (2.4) that

$$\begin{aligned}&\frac{\partial}{\partial t}(k^2-\frac{\partial^2}{\partial y^2})\hat{v}_{\boldsymbol{k}}+\nu\left(k^4-2k^2\frac{\partial^2}{\partial y^2}+\frac{\partial^4}{\partial y^4}\right)\hat{v}_{\boldsymbol{k}}=\hat{h}_{v,\boldsymbol{k}}(\boldsymbol{u},\boldsymbol{u}),\\&\hat{v}_{\boldsymbol{k}}(\pm1)=\frac{\partial\hat{v}_{\boldsymbol{k}}}{\partial y}(\pm1)=0,\end{aligned}\tag{2.5}$$

and

$$\begin{aligned}&\frac{\partial\hat{g}_{\boldsymbol{k}}}{\partial t}+\nu(k^2-\frac{\partial^2}{\partial y^2})\hat{g}_{\boldsymbol{k}}=\hat{h}_{g,\boldsymbol{k}}(\boldsymbol{u},\boldsymbol{u}),\\&\hat{g}_{\boldsymbol{k}}(\pm1)=0,\end{aligned}\tag{2.6}$$

where $k^2=\left(\frac{2\pi}{L_x}\right)^2k_x^2+\left(\frac{2\pi}{L_z}\right)^2k_z^2$.

From the equations relating the velocity components u and w to f and g in (2.3) we derive

$$\left.\begin{aligned}&ik_x\frac{2\pi}{L_x}\hat{u}_{\boldsymbol{k}}+ik_z\frac{2\pi}{L_z}\hat{w}_{\boldsymbol{k}}=\hat{f}_{\boldsymbol{k}},\\&ik_z\frac{2\pi}{L_z}\hat{u}_{\boldsymbol{k}}-ik_x\frac{2\pi}{L_x}\hat{w}_{\boldsymbol{k}}=\hat{g}_{\boldsymbol{k}},\end{aligned}\right.\quad\text{for all }\ (k_x,k_z)\neq(0,0).\tag{2.7}$$

For $(k_x,k_z)\neq(0,0)$, the relations (2.7) can be used to determine $\hat{u}_{\boldsymbol{k}}(y,t)$ and $\hat{w}_{\boldsymbol{k}}(y,t)$ in terms of $\hat{f}_{\boldsymbol{k}}(y,t)$ and $\hat{g}_{\boldsymbol{k}}(y,t)$. Hence, to complete the system, we still need additional relations for $\hat{u}_0(y,t)$ and $\hat{w}_0(y,t)$. To this end, we integrate the first and last components of the Navier-Stokes equations with respect to x and z to obtain

$$\begin{aligned}\frac{\partial\hat{u}_0}{\partial t}-\nu\frac{\partial^2\hat{u}_0}{\partial y^2}+\frac{1}{L_xL_z}\int_0^{L_x}dx\int_0^{L_z}v(\boldsymbol{x})\frac{\partial u}{\partial y}(\boldsymbol{x})dz+K_P=0,\\\frac{\partial\hat{w}_0}{\partial t}-\nu\frac{\partial^2\hat{w}_0}{\partial y^2}+\frac{1}{L_xL_z}\int_0^{L_x}dx\int_0^{L_z}v(\boldsymbol{x})\frac{\partial w}{\partial y}(\boldsymbol{x})dz=0.\end{aligned}\tag{2.8}$$

The time discretization of (2.5), (2.6) and (2.8) is achieved by using a semi-implicit scheme with the second-order Crank-Nicolson for the linear terms and a third order explicit Runge-Kutta scheme for the nonlinear terms. Hence, we only have to solve a sequence of one-dimensional second-order equations for $\hat{g}_{\boldsymbol{k}}(y,t)$ and fourth-order equations for $\hat{v}_{\boldsymbol{k}}(y,t)$.

Kim, Moin and Moser (1987) applied a Chebyshev-tau approximation to the $y-$direction. Since the direct application of tau method to fourth-order equations is unstable (Gottlieb and Orszag, 1977), they proposed a time splitting scheme which consists of solving several successive second-order problems to enforce the boundary conditions on v by using a technique similar to the influence matrix method.

Based on a sequence of recent work by Shen (1994, 1995 and 1996), we present below a spectral-Galerkin scheme for these second-order and fourth-order equations. Using this method, the system (2.5)–(2.6) can be directly solved.

2.2. A spectral-Galerkin approximation of the Kim-Moin-Moser formulation

A Fourier-Galerkin approximation in the x and z directions is first applied to the problems (2.5) and (2.6), i.e. we look for

$$\boldsymbol{u}_N(\boldsymbol{x},t) = \sum_{\boldsymbol{k}\in S_N} \hat{\boldsymbol{u}}_{\boldsymbol{k}}(y,t) e^{i\left(k_x \frac{2\pi}{L_x} x + k_z \frac{2\pi}{L_z} z\right)}, \tag{2.9}$$

(where $N = (N_x, N_z)$ and $S_N = \left\{\boldsymbol{k} \in \mathcal{Z}^2/(k_x,k_z) \in [1-\frac{N_x}{2}, \frac{N_x}{2}] \times [1-\frac{N_z}{2}, \frac{N_z}{2}]\right\}$) as a solution of the system of

$$\begin{aligned} &\frac{\partial}{\partial t}(k^2 - \frac{\partial^2}{\partial y^2})\hat{v}_{\boldsymbol{k}} + \nu\left(k^4 - 2k^2\frac{\partial^2}{\partial y^2} + \frac{\partial^4}{\partial y^4}\right)\hat{v}_{\boldsymbol{k}} = \hat{h}_{v,\boldsymbol{k}}(\boldsymbol{u}_N, \boldsymbol{u}_N), \\ &\hat{v}_{\boldsymbol{k}}(\pm 1) = \frac{\partial \hat{v}_{\boldsymbol{k}}}{\partial y}(\pm 1) = 0, \end{aligned} \tag{2.10}$$

and

$$\begin{aligned} &\frac{\partial \hat{g}_{\boldsymbol{k}}}{\partial t} + \nu(k^2 - \frac{\partial^2}{\partial y^2})\hat{g}_{\boldsymbol{k}} = \hat{h}_{g,\boldsymbol{k}}(\boldsymbol{u}_N, \boldsymbol{u}_N), \\ &\hat{g}_{\boldsymbol{k}}(\pm 1) = 0, \end{aligned} \tag{2.11}$$

for all $\boldsymbol{k} \in S_N$.

We now describe Galerkin approximations of (2.10) and (2.11) in the $y-$direction. Let us denote

- P_M: the space of polynomials of degree less than or equal to M,
- $V_M = \mathrm{span}\{\varphi(y) \in P_M : \varphi(\pm 1) = 0\}$,
- $W_M = \mathrm{span}\{\varphi(y) \in P_M : \varphi(\pm 1) = 0,\ \frac{\partial \varphi}{\partial y}(\pm 1) = 0\}$.

Let $p_j(y)$ be either the Legendre or Chebyshev polynomial of degree j, then

$$V_M = \mathrm{span}\{\phi_0, \phi_1, \ldots, \phi_{M-2}\}$$

with $\phi_j(y) = p_j(y) - p_{j+2}(y)$. Moreover, following Shen (1996), we can determine (a_j, b_j) such that

$$\psi_j(y) = p_j(y) + a_j p_{j+2}(y) + b_j p_{j+4}(y)$$

satisfies the boundary conditions $\psi_j(\pm 1) = \frac{\partial \psi_j}{\partial y}(\pm 1) = 0$, i.e. $\psi_j \in W_M$. Therefore

$$W_M = \mathrm{span}\{\psi_0, \psi_1, \ldots, \psi_{M-4}\}.$$

The spectral-Galerkin scheme in the $y-$direction for (2.10) and (2.11) is to find $v_{N,M}(\boldsymbol{x},t)$ such that $\hat{v}_{\boldsymbol{k},M}(y,t) \in W_M$, and $u_{N,M}(\boldsymbol{x},t)$ (similarly for w and g) such that $\hat{u}_{\boldsymbol{k},M}(y,t) \in V_M$, for all $\boldsymbol{k} \in S_N$, such that

$$\begin{aligned} \frac{\partial}{\partial t}\left((k^2 - \frac{\partial^2}{\partial y^2})\hat{v}_{\boldsymbol{k}}, \psi_j\right)_\omega + \nu\left(\left(k^4 - 2k^2\frac{\partial^2}{\partial y^2} + \frac{\partial^4}{\partial y^4}\right)\hat{v}_{\boldsymbol{k}}, \psi_j\right)_\omega \\ = \left(\hat{h}_{v,\boldsymbol{k}}(\boldsymbol{u}_{N,M}, \boldsymbol{u}_{N,M}), \psi_j\right)_\omega, \end{aligned} \tag{2.12}$$

for all $j = 0, \ldots, M-4$,

and

$$\frac{\partial}{\partial t}\left(\hat{g}_{\boldsymbol{k}}, \phi_j\right)_\omega + \nu\left((k^2 - \frac{\partial^2}{\partial y^2})\hat{g}_{\boldsymbol{k}}, \phi_j\right)_\omega = \left(\hat{h}_{g,\boldsymbol{k}}(\boldsymbol{u}_{N,M}, \boldsymbol{u}_{N,M}), \phi_j\right)_\omega, \tag{2.13}$$
for all $j = 0, \ldots, M-2$,

where $(\varphi, \psi)_\omega = \int_{-1}^{1} \varphi(y)\,\psi(y)\;\omega\,dy$ with $\omega(y) \equiv 1$ in the Legendre case and $\omega(y) = (1-y^2)^{-\frac{1}{2}}$ in the Chebyshev case.

It is easy to see that, in (2.13), the mass matrix $\mathcal{M}$ with entries $m_{jl} = (\phi_l, \phi_j)_\omega$ is a sparse symmetric matrix with three nonzero diagonals, and that the stiffness matrix $\mathcal{S}$ with entries $s_{jl} = (\frac{\partial^2}{\partial y^2}\phi_l, \phi_j)_\omega$ is diagonal in the Legendre case, and is a special upper triangular matrix in the Chebyshev case such that the linear system $(\alpha\mathcal{M} + \mathcal{S})\boldsymbol{x} = \boldsymbol{b}$ associated with (2.13) can be solved in $O(M)$ operations (Shen 1995). Similarly, the linear systems in (2.12) can be solved in $O(M)$ operations, see Shen (1994, 1995). We emphasize that the above spectral-Galerkin scheme is superior, in both efficiency and accuracy, to the tau-method used in Kim, Moin and Moser (1987), and is, in particular, suitable for multilevel decomposition.

The Legendre-Galerkin method has been implemented and tested. In this code, the pseudo-spectral computation of the nonlinear terms is done at the Chebyshev-Gauss-Lobatto points in the normal direction (see Shen 1996). A $128 \times 129 \times 128$ simulation at the Reynolds number of 180 has been conducted. The statistics have been compared to the one presented by Kim, Moin and Moser (1987).

2.3. A multilevel spectral-Galerkin scheme

We now describe a multilevel scheme for the time integration of (2.12) and (2.13). For the sake of simplicity, we will only present a scheme based on a first-order semi-implicit scheme for the time discretization. However, one can easily generalize it to higher-order semi-implicit scheme.

The basic idea of the multilevel scheme is to decompose the solution into several length scales and treat them differently in order to improve the efficiency and stability of the classical Galerkin approximation. The special basis functions $\{\phi_j, \psi_j\}$ provide a natural decomposition of small and large wavelengths for this purpose. Furthermore, the small and large wavelengths are quasi-orthogonal in the following sense:

$$\begin{aligned}
&(\phi_l, \phi_j)_\omega = 0, \quad \text{for} \;\; j \neq l, l \pm 2, \\
&(\frac{\partial^2 \phi_l}{\partial y^2}, \phi_j)_\omega = 0, \quad \text{for} \;\; l \neq j \text{ (Legendre case)}, \\
&(\frac{\partial^2 \phi_l}{\partial y^2}, \phi_j)_\omega = 0, \quad \text{for} \;\; l < j \text{ or } l + j \text{ odd} \;\; \text{(Chebyshev case)},
\end{aligned} \tag{2.14}$$

and

$$
\begin{aligned}
&(\psi_l, \psi_j)_\omega = 0, \quad \text{for } \ j \neq l, l \pm 2, l \pm 4, \\
&(\frac{\partial^2 \psi_l}{\partial y^2}, \psi_j)_\omega = 0, \quad \text{for } \ l \neq j, l \neq j \pm 2, \\
&(\frac{\partial^4 \psi_l}{\partial y^4}, \psi_j)_\omega = 0, \quad \text{for } \ l \neq j \text{ (Legendre case)}, \\
&(\frac{\partial^4 \psi_l}{\partial y^4}, \psi_j)_\omega = 0, \quad \text{for } \ l < j \text{ or } l + j \text{ odd \ (Chebyshev case)}.
\end{aligned}
\tag{2.15}
$$

Given two appropriate cut-off numbers M_p, M_q such that $0 < M_p < M_q < M$, we may decompose $\hat{u}_{\boldsymbol{k},M}(y,t) \in V_M$ as follows

$$
\begin{aligned}
\hat{u}_{\boldsymbol{k},M}(y,t) &= \sum_{j=0}^{M-2} \hat{u}_{\boldsymbol{k},j}(t) \phi_j(y) \\
&= \sum_{j=0}^{M_p-2} \hat{u}_{\boldsymbol{k},j} \phi_j(y) + \sum_{j=M_p-1}^{M_q-2} \hat{u}_{\boldsymbol{k},j} \phi_j(y) + \sum_{j=M_q-1}^{M-2} \hat{u}_{\boldsymbol{k},j} \phi_j(y) \\
&= p_u(y,t) + q_u(y,t) + r_u(y,t),
\end{aligned}
\tag{2.16}
$$

and similarly for $\hat{w}_{\boldsymbol{k},M}(y,t)$ and then for $\hat{g}_{\boldsymbol{k},M}(y,t)$, for all $\boldsymbol{k} \in S_N$. Note that for the sake of simplicity, the dependence of p_u, q_u and r_u in $\boldsymbol{k}$ is omitted. We may also decompose $\hat{v}_{\boldsymbol{k},M}(y,t) \in W_M$ as

$$
\begin{aligned}
\hat{v}_{\boldsymbol{k},M}(y,t) &= \sum_{j=0}^{M-4} \hat{v}_{\boldsymbol{k},j} \psi_j(y) \\
&= \sum_{j=0}^{M_p-4} \hat{v}_{\boldsymbol{k},j} \psi_j(y) + \sum_{j=M_p-3}^{M_q-4} \hat{v}_{\boldsymbol{k},j} \psi_j(y) + \sum_{j=M_q-3}^{M-4} \hat{v}_{\boldsymbol{k},j} \psi_j(y) \\
&= p_v(y,t) + q_v(y,t) + r_v(y,t).
\end{aligned}
\tag{2.17}
$$

We finally obtain the following decomposition for $\hat{\boldsymbol{u}}_{\boldsymbol{k},M}$:

$$
\hat{\boldsymbol{u}}_{\boldsymbol{k},M} = \boldsymbol{p} + \boldsymbol{q} + \boldsymbol{r},
$$

where $\boldsymbol{p} = (p_u, p_v, p_w)$ and similarly for $\boldsymbol{q}$ and $\boldsymbol{r}$. The decomposition (2.16) on $\hat{u}_{\boldsymbol{k},M}$ and $\hat{w}_{\boldsymbol{k},M}$ induces a decomposition of $\hat{g}_{\boldsymbol{k},M}$ into

$$
\hat{g}_{\boldsymbol{k},M}(y,t) = p_g + q_g + r_g.
$$

Then, thanks to (2.15) (resp. (2.14)), we can *approximate* the system (2.12) (resp. (2.13)) in W_{M_p} (resp. V_{M_p}) as follows

$$
\begin{aligned}
\frac{\partial}{\partial t} \left((k^2 - \frac{\partial^2}{\partial y^2}) p_v, \psi_j \right)_\omega &+ \nu \left(\left(k^4 - 2k^2 \frac{\partial^2}{\partial y^2} + \frac{\partial^4}{\partial y^4} \right) p_v, \psi_j \right)_\omega \\
&= (\hat{h}_{v,\boldsymbol{k}}(\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r}, \boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r}), \psi_j)_\omega, \\
&\qquad \text{for } j = 0, \ldots, M_p - 4,
\end{aligned}
\tag{2.18}
$$

$$\frac{\partial}{\partial t}(p_g,\phi_j)_\omega + \nu k^2(p_g,\phi_j)_\omega - \nu\left(\frac{\partial^2 p_g}{\partial y^2},\phi_j\right)_\omega = (\hat{h}_{g,\boldsymbol{k}}(\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r},\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r}),\phi_j)_\omega, \quad \text{for } j=0,\dots,M_p-2, \tag{2.19}$$

and in W_{M_q} (resp. V_{M_q}) as follows

$$\frac{\partial}{\partial t}\left((k^2-\frac{\partial^2}{\partial y^2})(p_v+q_v),\psi_j\right)_\omega + \nu\left(\left(k^4-2k^2\frac{\partial^2}{\partial y^2}+\frac{\partial^4}{\partial y^4}\right)(p_v+q_v),\psi_j\right)_\omega = (\hat{h}_{v,\boldsymbol{k}}(\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r},\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r}),\psi_j)_\omega, \quad \text{for } j=0,\dots,M_q-4, \tag{2.20}$$

$$\frac{\partial}{\partial t}(p_g+q_g,\phi_j)_\omega + \nu k^2(p_g+q_g,\phi_j)_\omega - \nu(\frac{\partial^2}{\partial y^2}(p_g+q_g),\phi_j)_\omega = (\hat{h}_{g,\boldsymbol{k}}(\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r},\boldsymbol{p}+\boldsymbol{q}+\boldsymbol{r}),\phi_j)_\omega, \quad \text{for } j=0,\dots,M_q-2. \tag{2.21}$$

Note that in (2.18)–(2.19) and (2.20)–(2.21) linear interaction terms coming from $(p_g,\phi_j)_\omega$ (resp. $(q_v,\phi_j)_\omega$) and similarly for r_g (resp. r_v) are neglected. Until numerical tests are performed, it is not clear whether or not these terms have to be neglected. However, for the sake of simplicity we do not take them into account in the large or intermediate scale equations.

By projecting (2.12) (resp. (2.13)) onto the space $W_M\backslash W_{M_q}$ (resp. $V_M\backslash V_{M_q}$) we obtain the small scale equation

$$\begin{aligned}\frac{\partial}{\partial t}\left((k^2-\frac{\partial^2}{\partial y^2})r_v,\psi_j\right)_\omega &+ \nu\left((k^4-2k^2\frac{\partial^2}{\partial y^2}+\frac{\partial^4}{\partial y^4})r_v,\psi_j\right)_\omega \\ = (\hat{h}_{v,\boldsymbol{k}}(\boldsymbol{p}+\boldsymbol{q},\boldsymbol{p}+\boldsymbol{q}),\psi_j)_\omega & \\ -\frac{\partial}{\partial t}\left((k^2-\frac{\partial^2}{\partial y^2})q_v,\psi_j\right)_\omega &- \nu\left((k^4-2k^2\frac{\partial^2}{\partial y^2})q_v,\psi_j\right)_\omega, \\ \text{for } j=M_q-3,\dots,M-4,&\end{aligned} \tag{2.22}$$

$$\begin{aligned}\frac{\partial}{\partial t}(r_g,\phi_j)_\omega + \nu k^2(r_g,\phi_j)_\omega - \nu(\frac{\partial^2 r_g}{\partial y^2},\phi_j)_\omega & \\ = (\hat{h}_{g,\boldsymbol{k}}(\boldsymbol{p}+\boldsymbol{q},\boldsymbol{p}+\boldsymbol{q}),\phi_j)_\omega - \frac{\partial}{\partial t}(q_g,\phi_j)_\omega - \nu k^2(q_g,\phi_j)_\omega,& \\ \text{for } j=M_q-1,\dots,M-2.&\end{aligned} \tag{2.23}$$

We note that in (2.22)–(2.23) the nonlinear interaction between the small wavelength part $\boldsymbol{r}$ and the larger wavelength parts $(\boldsymbol{p}+\boldsymbol{q})$ is neglected.

Since $h_g(\cdot,\cdot)$ is a bilinear form, we can write

$$\begin{aligned}h_g(\boldsymbol{\varphi}+\boldsymbol{\psi},\boldsymbol{\varphi}+\boldsymbol{\psi}) &= h_g(\boldsymbol{\varphi},\boldsymbol{\varphi}) + (h_g(\boldsymbol{\varphi},\boldsymbol{\psi}) + h_g(\boldsymbol{\psi},\boldsymbol{\varphi}) + h_g(\boldsymbol{\psi},\boldsymbol{\psi})) \\ &= h_g(\boldsymbol{\varphi},\boldsymbol{\varphi}) + h_{g,\mathrm{int}}(\boldsymbol{\varphi},\boldsymbol{\psi}),\end{aligned}$$

and similarly for $h_v(\cdot,\cdot)$.

We may now define the multilevel scheme based on the approximation (2.18) – (2.23).

Given $\boldsymbol{U}^n_{N,M} = \boldsymbol{p}^n + \boldsymbol{q}^n + \boldsymbol{r}^n$, the approximation of $\boldsymbol{u}_{N,M}(n\Delta t)$, and an integer n_u, we define $\boldsymbol{U}^{n+2n_u}_{N,M} = \boldsymbol{p}^{n+2n_u} + \boldsymbol{q}^{n+2n_u} + \boldsymbol{r}^{n+2n_u}$ by using the following multilevel scheme:

For $j = 0, 1, \dots, n_u - 1$,

$$\begin{aligned} k^2(1+\nu k^2\Delta t)(p_v^{n+2j+1},\psi_l)_\omega &- (1+\nu k^2\Delta t)(\frac{\partial^2}{\partial y^2}p_v^{n+2j+1},\psi_l)_\omega \\ + \nu\Delta t\left(\frac{\partial^4}{\partial y^4}p_v^{n+2j+1},\psi_l\right)_\omega &= \left((k^2-\frac{\partial^2}{\partial y^2})p_v^{n+2j},\psi_l\right)_\omega \\ &+ \Delta t(\hat{h}_v(\boldsymbol{p}^{n+2j},\boldsymbol{p}^{n+2j},\psi_l)_\omega + \Delta t(\hat{h}_{v,int}(\boldsymbol{p}^n,\boldsymbol{q}^n+\boldsymbol{r}^n),\psi_l)_\omega, \\ &\qquad \text{for } l = 0,\dots,M_p-4, \end{aligned} \tag{2.24}$$

$$\begin{aligned} q_v^{n+2j+1} &= q_v^{n+2j}, \\ r_v^{n+2j+1} &= r_v^{n+2j} = r_v^n; \end{aligned}$$

$$\begin{aligned} (1+\nu k^2\Delta t)(p_g^{n+2j+1},\phi_l)_\omega - \nu\Delta t(\frac{\partial^2}{\partial y^2}p_g^{n+2j+1},\phi_l)_\omega &= (p_g^{n+2j},\phi_l)_\omega \\ + \Delta t(\hat{h}_g(\boldsymbol{p}^{n+2j},\boldsymbol{p}^{n+2j}),\phi_l)_\omega + \Delta t(\hat{h}_{g,int}(\boldsymbol{p}^n,(\boldsymbol{q}+\boldsymbol{r})^n),\phi_l)_\omega&, \\ \text{for } l = 0,\dots,M_p-2&, \end{aligned} \tag{2.25}$$

$$\begin{aligned} q_g^{n+2j+1} &= q_g^{n+2j}, \\ r_g^{n+2j+1} &= r_g^{n+2j} = r_g^n; \end{aligned}$$

$$\begin{aligned} & k^2(1+\nu k^2\Delta t)((p_v+q_v)^{n+2j+2},\psi_l)_\omega - (1+\nu k^2\Delta t)(\frac{\partial^2}{\partial y^2}(p_v+q_v)^{n+2j+2},\psi_l)_\omega \\ & + \nu\Delta t\left(\frac{\partial^4}{\partial y^4}(p_v+q_v)^{n+2j+2},\psi_l\right)_\omega = \left((k^2-\frac{\partial^2}{\partial y^2})(p_v+q_v)^{n+2j+1},\psi_l\right)_\omega \\ & \qquad + \Delta t(\hat{h}_v((\boldsymbol{p}+\boldsymbol{q})^{n+2j+1},(\boldsymbol{p}+\boldsymbol{q})^{n+2j+1}),\psi_l)_\omega \\ & \qquad + \Delta t(\hat{h}_{v,int}(\boldsymbol{p}^n+\boldsymbol{q}^n,\boldsymbol{r}^n),\psi_l)_\omega, \\ & \qquad\qquad \text{for } l = 0,\dots,M_q-4, \\ & r_v^{n+2j+2} = r_v^{n+2j+1} = r_v^n; \end{aligned} \tag{2.26}$$

$$\begin{aligned} (1+\nu k^2\Delta t)((p_g+q_g)^{n+2j+2},\phi_l)_\omega &- \nu\Delta t(\frac{\partial^2}{\partial y^2}(p_g+q_g)^{n+2j+2},\phi_l)_\omega \\ &= ((p_g+q_g)^{n+2j+1},\phi_l)_\omega \\ &+ \Delta t(\hat{h}_g((\boldsymbol{p}+\boldsymbol{q})^{n+2j+1},\boldsymbol{p}^{n+2j+1}),\phi_l)_\omega \\ &+ \Delta t(\hat{h}_{g,int}(\boldsymbol{p}^n+\boldsymbol{q}^n,\boldsymbol{r}^n),\phi_l)_\omega, \\ &\qquad \text{for } l = 0,\dots,M_q-2, \end{aligned} \tag{2.27}$$

$$r_g^{n+2j+2} = r_g^{n+2j+1} = r_g^n.$$

Once we obtain $\boldsymbol{p}^{n+2n_u}$ and $\boldsymbol{q}^{n+2n_u}$ from above, we compute $\boldsymbol{r}^{n+2n_u}$ as follows

$$
\begin{aligned}
k^2(1+2n_u\nu k^2\Delta t)(r_v^{n+2n_u},\psi_l)_\omega &- (1+2n_u\nu k^2\Delta t)(\frac{\partial^2}{\partial y^2}r_v^{n+2n_u},\psi_l)_\omega \\
+2n_u\nu\Delta t\left(\frac{\partial^4}{\partial y^4}r_v^{n+2n_u},\psi_l\right)_\omega &= \left((k^2-\frac{\partial^2}{\partial y^2})r_v^n,\psi_l\right)_\omega \\
&+2n_u\Delta t(\hat{h}_v((\boldsymbol{p}+\boldsymbol{q})^{n+2n_u},(\boldsymbol{p}+\boldsymbol{q})^{n+2n_u}),\psi_l)_\omega \\
&-\left((k^2-\frac{\partial^2}{\partial y^2})(q_v^{n+2n_u}-q_v^n),\psi_l\right)_\omega \\
&-2n_u\Delta t\left((k^4-2k^2\frac{\partial^2}{\partial y^2})q_v^{n+2n_u},\psi_l\right)_\omega \\
&\qquad \text{for } l=M_q-3,\dots,M-4,
\end{aligned}
\tag{2.28}
$$

and

$$
\begin{aligned}
(1+2n_u\nu k^2\Delta t)(r_g^{n+2n_u},\phi_l)_\omega &- 2n_u\nu\Delta t(\frac{\partial^2}{\partial y^2}r_g^{n+2n_u},\phi_l)_\omega \\
&= (r_g^{n+2n_u},\phi_l)_\omega \\
&+2n_u\Delta t(\hat{h}_g((\boldsymbol{p}+\boldsymbol{q})^{n+2n_u},(\boldsymbol{p}+\boldsymbol{q})^{n+2n_u}),\phi_l)_\omega \\
&-\left((q_g^{n+2n_u}-q_v^n),\phi_l\right)_\omega - 2n_uk^2\Delta t(q_g^{n+2n_u},\phi_l)_\omega \\
&\qquad \text{for } l=M_q-1,\dots,M-2.
\end{aligned}
\tag{2.29}
$$

Note that the computation of the right-hand side of (2.24)–(2.25) (resp. (2.26)–(2.27)) requires only fast Chebyshev transforms (FCT) with $O(M_p log_2(M_p))$ (resp. $O(M_p log_2(M_p))$) operations in the normal direction. The nonlinear interaction terms $h_{v,int}$ and $h_{g,int}$ are computed once at the time iteration $j=n$. Hence, during the $2n_u$ time iterations described above, FCT with $O(Mlog_2(M))$ are required only at $j=n$ and $j=n+2n_u$. Compared to a classical Galerkin (or tau) approximation the multilevel scheme proposed here allows to significantly reduce the CPU time needed for channel flow simulations. In the case of forced homogeneous turbulence, savings of the order of 50 to 70 percent have been obtained (see Dubois, Jauberteau and Temam 1995b and 1996).

3. Incremental unknowns in the finite difference case

The main idea of the multilevel scheme is to treat the large and small scales differently in numerical simulation. Therefore, it is important to have an appropriate decomposition of the flow into different length scales. In Section 3.1, we describe a procedure to decompose the solution into large and small scales in finite difference method. To illustrate the method, we start by applying the incremental unknowns to the Burger's equation. In Section 3.2, we test the method of separating scales using turbulent channel flow database. In Section 3.3, we suggest an algorithm to implement the scheme for the turbulent channel flow.

3.1. Incremental unknowns on Burger's equation

In this section, we start with the two-level incremental unknowns, namely, we decompose the solution u into

$$u = y + z.$$

The second-order incremental unknowns in one-dimensional case are defined as in Chen and Temam (1991) by

$$\begin{aligned} y_{2j} &= u_{2j}, \\ z_{2j+1} &= u_{2j+1} - \frac{1}{2}(u_{2j+2} + u_{2j}). \end{aligned}$$

Multilevel incremental unknowns can be defined recursively. Three-level incremental unknowns will be defined in Section 3.2.

Let us consider the Burger's equation,

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(\frac{u^2}{2}) = \nu \frac{\partial^2 u}{\partial x^2} + \mathcal{X}(x,t), \quad u(0,t) = u(1,t) = 0.$$

When the second order central difference scheme is used for the space derivatives and the explicit Euler is used for the time advancing, the finite difference scheme reads

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{4\Delta x}[(u_{i+1}^n)^2 - (u_{i-1}^n)^2] = \frac{\nu}{\Delta x^2}[u_{i+1}^n - 2u_i^n + u_{i-1}^n] + \mathcal{X}(x_i, t^n).$$

Writing y and z components separately, one finds that y satisfies

$$\begin{aligned} u_{2j+1}^n &= z_{2j+1}^n + \frac{1}{2}(y_{2j+2}^n + y_{2j}^n), \\ \frac{y_{2j}^{n+1} - y_{2j}^n}{\Delta t} &+ \frac{1}{4\Delta x}[(u_{2j+1}^n)^2 - (u_{2j-1}^n)^2] \\ &= \frac{\nu}{\Delta x^2}[u_{2j+1}^n - 2y_{2j}^n + u_{2j-1}^n] + \mathcal{X}(x_{2j}, t^n), \end{aligned}$$

and z satisfies

$$\begin{aligned} &\frac{z_{2j+1}^{n+1} - z_{2j+1}^n}{\Delta t} + \frac{1}{2\Delta t}[(y_{2j+2}^{n+1} + y_{2j}^{n+1}) - (y_{2j+2}^n + y_{2j}^n)] \\ &+ \frac{1}{4\Delta x}[(y_{2j+2}^n)^2 - (y_{2j}^n)^2] = \frac{\nu}{\Delta x^2}[-2z_{2j+1}^n] + \mathcal{X}(x_{2j+1}, t^n). \end{aligned}$$

Instead of evaluating z at each time step, we propose to fix z for m steps and then evaluate once to save the CPU time and memory. Therefore, as m increases, so does the the saving of the CPU time. On the other hand, we are also at the risk of losing accuracy as m increases. It is clear that when $m = 0$, the scheme is the same as the original standard method with the fine mesh, while if we never update

z and let it be 0, the scheme is simply the original standard method in the coarse mesh and $u = y$.

To illustrate how much savings one could obtain by freezing z systematically, we list in table 1 the ratio of the work with freezing z m times vs. 0 times, with the assumption that the work per time step per grid point is independent of the mesh size. As an example, if one freezes z for one time step in a three-dimensional problem, the work by using incremental unknowns is only 56.25% of that by using the standard finite difference method.

Table 1. Ratio of the work with freezing z m times vs. 0 times

m	0	1	2	3	$\cdots$	N_t
1-D	1	0.75	0.67	0.625	$\cdots$	0.50
2-D	1	0.625	0.5	0.44	$\cdots$	0.25
3-D	1	0.5625	0.42	0.34	$\cdots$	0.125

We now test this scheme on a model problem, in which we try to recover the steady solution $u_s(x) = f(20x) - f(0) + (f(0) - f(20))x$ of Burger's equation, where

$$f(t) = \sum_{k=1}^{15} \exp(\cos(k\sqrt{k}(2.5 + 0.5t)\pi/10) - 0.3\sin(0.8k\sqrt{k}t\pi/10)).$$

The forcing function $\mathcal{X}(x,t) = \mathcal{X}(x)$ is calculated by substituting $u_s(x)$ into the equation. Initial condition is taken as $u(x,0) = \sin(2x)$ with the boundary conditions $u(0,t) = u(1,t) = 0$. By comparing the graphs of $u_s(x)$ with $N_x = 512$, $N_x = 256$ and $N_x = 128$, one finds that $N_x = 256$ is approximately the minimum number of grid points required to adequately resolve $u_s(x)$.

The numerical results using the original scheme and the proposed scheme with different m are compared (figure 1). For $m = 1$ to 4, the results are almost identical. However, for $m = 5$, the approximate solution is significantly less accurate. Therefore, the proposed scheme has to be used with caution and m can not be too large.

3.2. Small scales in IU

In the multilevel scheme given in section 2, a spectral method is used to decompose scales. However, it is not easy to define small scales in finite difference methods. In Section 3.1, the small scale component of the flow is defined in the context of incremental unknowns. In this section, we examine this concept.

For simplicity, we will only treat the three-level IU. As is done in Section 2, the method starts with separating the length scales of the flow into

$$u = f + g + r, \tag{3.1}$$

where u is the velocity in the streamwise direction, f, g and r are respectively the large, intermediate and small scales. The definitions of f, g and r are given below (see figure 2):

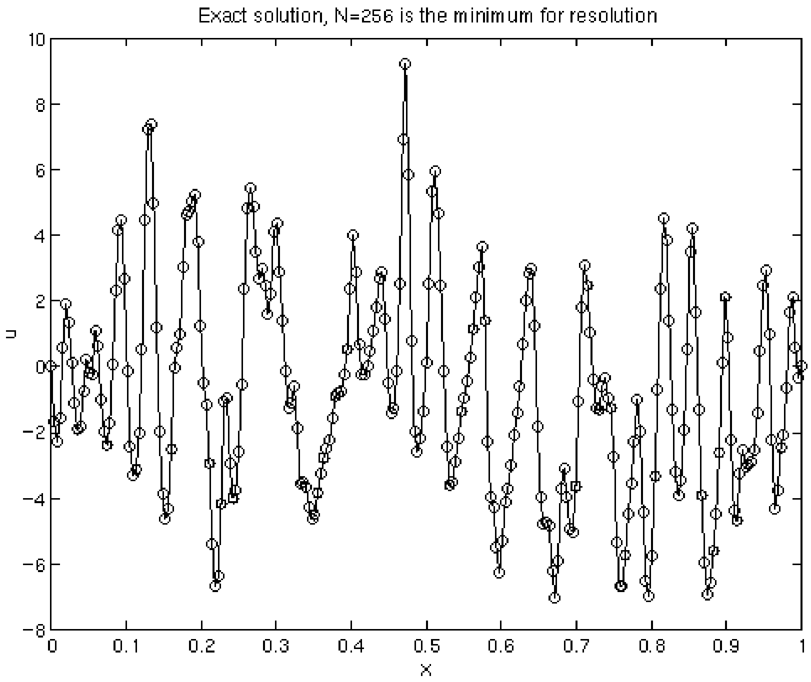

FIGURE 1. An exact steady state solution of the Burger's equation.

$$\begin{aligned} f_{4i} &= u_{4i} \\ g_{4i+2} &= u_{4i+2} - \frac{1}{2}(u_{4i+4} + u_{4i}) \\ r_{4i+1} &= u_{4i+1} - \frac{1}{2}(u_{4i+2} + u_{4i}), \end{aligned} \tag{3.2}$$

where i is the index for the streamwise (or wall-normal, or spanwise) direction $(i = 0, 1, 2, ..., N_x/4)$. The wall-normal and spanwise velocities can be defined in a similar way. We require the condition

$$|f| > |g| > |r| \tag{3.3}$$

in order to validate the assumption of separating length scales.

In the present study, the magnitudes of f, g and r are estimated using the database of turbulent channel flow. Turbulent flow in a channel is simulated using the Direct Numerical Simulation (DNS) technique. The convection and diffusion terms are integrated in time using a third-order Runge-Kutta method and the Crank-Nicolson method, respectively. A second-order central difference is used in space. A fractional step method is used to decouple the pressure from the velocity. The Reynolds number used is $Re_\tau = u_\tau\delta/\nu = 180$ and the computational domain is $4\pi\delta\ (x) \times 2\delta\ (y) \times 4\pi/3\delta\ (z)$, where u_τ is the wall shear velocity, δ is the channel

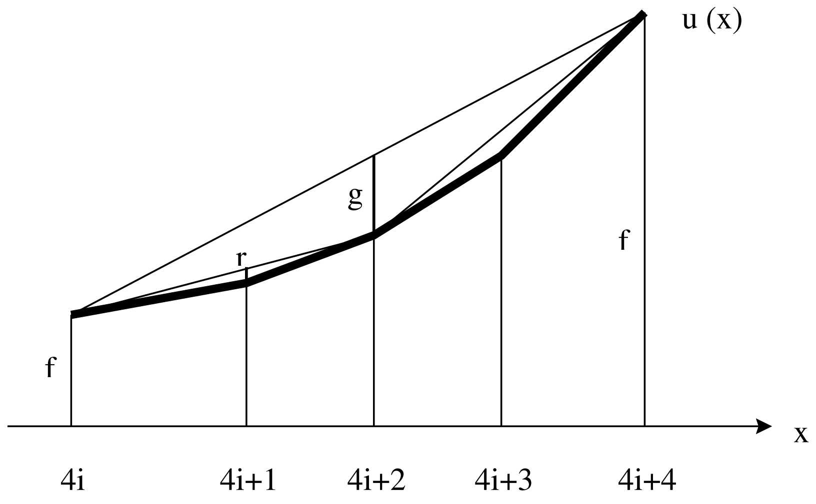

FIGURE 2. Schematic diagram of separating scales in a finite difference method

half width, and ν is the kinematic viscosity. The number of grid points used is $128\ (x)\ \times\ 129\ (y)\ \times\ 128\ (z)$.

Figure 3 shows the energy spectra of the velocity components in the streamwise and spanwise directions, where $E(f_i)$, $E(g_i)$ and $E(r_i)$ are shown at three y-locations ($y^+ = 6, 33, 177$). It is clear that r_i's have the energy of small scales, while g_i's have the energy of intermediate scales. Both g_i and r_i have orders of magnitude smaller energies in small wavenumbers as compared to f_i. Therefore, the incremental unknowns defined in (3.2) properly describe the small and intermediate scales of the velocity.

3.3. Implementation of IU in turbulent channel flow

Implementation of IU for the Navier-Stokes equations is very similar to that of IU for the Burgers equation (see Section 3.1), once the approximating factorization scheme is used (see below). The only difference is the coupling between the velocity and the pressure.

The governing equations for an incompressible flow are

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} u_i u_j = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\frac{\partial u_i}{\partial x_j}, \tag{3.4}$$

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{3.5}$$

The integration method used to solve (3.4) and (3.5) is based on a semi-implicit fractional step method, i.e., third-order Runge-Kutta method for the convection terms and Crank-Nicolson method for the diffusion terms:

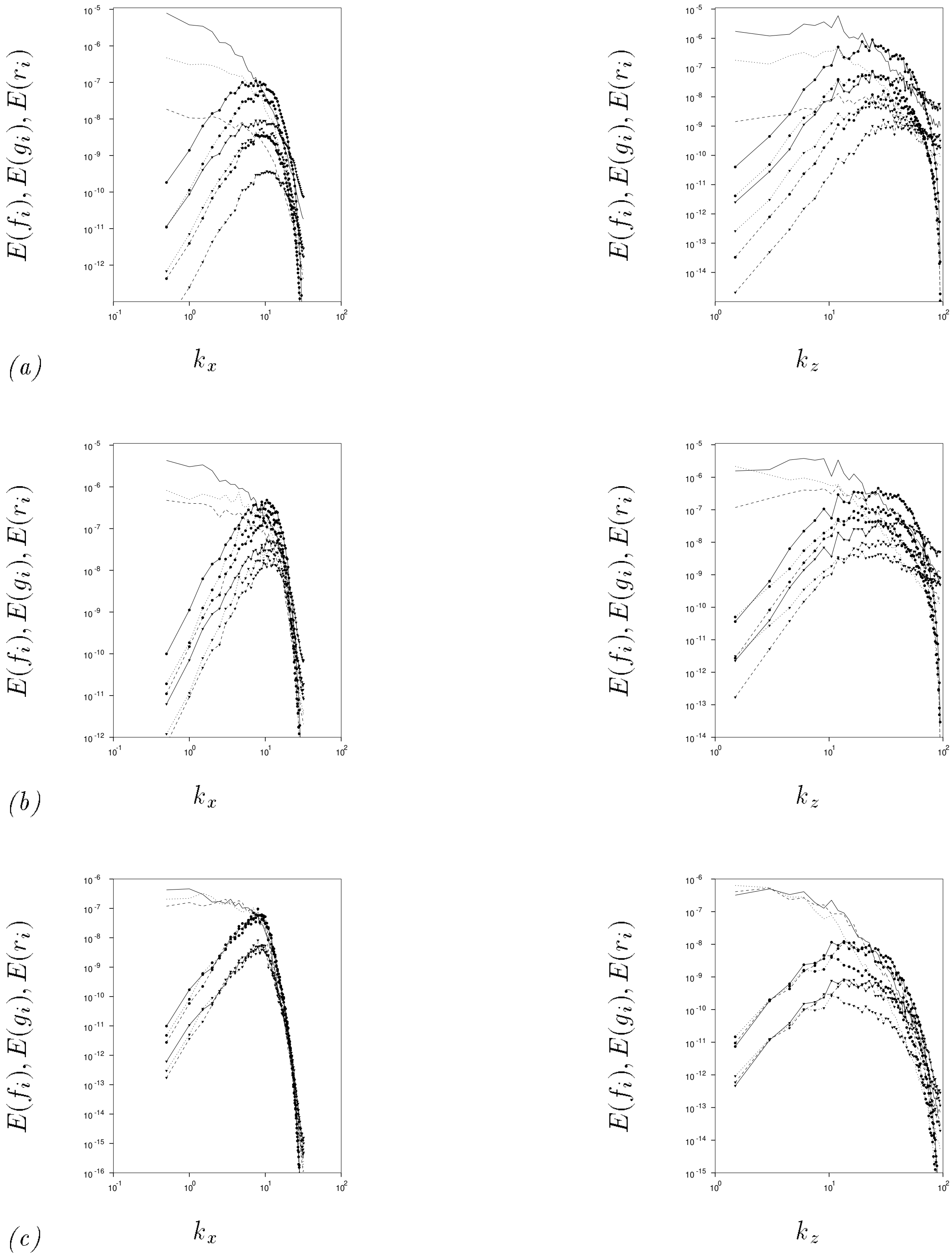

FIGURE 3. Energy spectra of the velocities, u_1 (———), u_2 (- - - -), u_3 (········): no symbol, f_i; •, g_i; ▼, r_i. (*a*) $y^+ = 6$; (*b*) $y^+ = 33$; (*c*) $y^+ = 177$.

$$\begin{aligned}\frac{\hat{u}_i^k - u_i^{k-1}}{\Delta t} =&(\alpha_k + \beta_k) L_i(\boldsymbol{u}^{k-1}) + \beta_k L_i(\hat{\boldsymbol{u}}^k - \boldsymbol{u}^{k-1}) \\ & - \gamma_k N_i(\boldsymbol{u}^{k-1}) - \zeta_k N_i(\boldsymbol{u}^{k-2}),\end{aligned} \tag{3.6}$$

$$\nabla^2 \phi^k = \frac{1}{\Delta t}\frac{\partial \hat{u}_i^k}{\partial x_i}, \tag{3.7}$$

$$\frac{u_i^k - \hat{u}_i^k}{\Delta t} = -\frac{\partial \phi^k}{\partial x_i}, \tag{3.8}$$

where L_i and N_i are the diffusion and convection terms of (3.4), $k = 1, 2, 3$, and

$$\begin{aligned} &\alpha_1 = \beta_1 = \frac{4}{15}, \qquad \gamma_1 = \frac{8}{15}, \qquad \zeta_1 = 0 \\ &\alpha_2 = \beta_2 = \frac{1}{15}, \qquad \gamma_2 = \frac{5}{12}, \qquad \zeta_2 = -\frac{17}{60} \\ &\alpha_3 = \beta_3 = \frac{1}{6}, \qquad \gamma_3 = \frac{3}{4}, \qquad \zeta_3 = -\frac{5}{12}. \end{aligned}$$

Here, $(\alpha_k + \beta_k) p^k = \phi^k - (\Delta t \beta_k / Re) \nabla^2 \phi^k$.

Rearranging (3.6) in delta form ($\delta \hat{u}_i^k = \hat{u}_i^k - u_i^{k-1}$) gives

$$(1 - \Delta t \beta_k \frac{1}{Re} \nabla^2) \delta \hat{u}_i^k = \Delta t \big[(\alpha_k + \beta_k) L_i(\boldsymbol{u}^{k-1}) - \gamma_k N_i(\boldsymbol{u}^{k-1}) - \zeta_k N_i(\boldsymbol{u}^{k-2}) \big].$$

Approximating factorization of this equation gives

$$\begin{aligned} &(1 - \Delta t \beta_k \frac{1}{Re}\frac{\partial^2}{\partial x^2})(1 - \Delta t \beta_k \frac{1}{Re}\frac{\partial^2}{\partial y^2})(1 - \Delta t \beta_k \frac{1}{Re}\frac{\partial^2}{\partial z^2}) \delta \hat{u}_i^k \\ &= \Delta t \big[(\alpha_k + \beta_k) L_i(\boldsymbol{u}^{k-1}) - \gamma_k N_i(\boldsymbol{u}^{k-1}) - \zeta_k N_i(\boldsymbol{u}^{k-2}) \big] \\ &\equiv R_i(\boldsymbol{u}^{k-1}, \boldsymbol{u}^{k-2}). \end{aligned} \tag{3.9}$$

Let us define χ_i as

$$\chi_i \equiv (1 - \Delta t \beta_k \frac{1}{Re}\frac{\partial^2}{\partial y^2})(1 - \Delta t \beta_k \frac{1}{Re}\frac{\partial^2}{\partial z^2}) \delta \hat{u}_i. \tag{3.10}$$

Then, (3.9) becomes

$$(1 - \Delta t \beta_k \frac{1}{Re}\frac{\partial^2}{\partial x^2}) \chi_i = R_i. \tag{3.11}$$

For simplicity, we only focus on the velocity in the streamwise component. Note that in turbulent channel flow the periodic boundary conditions are applied in the streamwise and spanwise directions (x, z) and the no-slip condition is applied in the wall-normal direction (y).

Now, let us decompose χ (streamwise component of χ_i) into three different scales as was introduced in section 3.2:

$$\chi \equiv f + g + r \tag{3.12}$$

As a first step, (3.11) is approximated at each fourth grid point using a second-order central difference scheme:

$$\chi_{4i} - \Gamma(\chi_{4i+1} - 2\chi_{4i} + \chi_{4i-1}) = R_{1_{4i}}, \tag{3.13}$$

where $\Gamma = \Delta t \beta_k/(Re\Delta x^2)$.

Using a similar relation to (3.2), it can be easily shown that (3.13) becomes

$$\begin{aligned} &- \frac{\Gamma}{4}\chi_{4i+4} + (1 + \frac{\Gamma}{2})\chi_{4i} - \frac{\Gamma}{4}\chi_{4i-4} \\ &= R_{1_{4i}} + \Gamma(\frac{1}{2}g_{4i+2} + \frac{1}{2}g_{4i-2} + r_{4i+1} + r_{4i-1}). \end{aligned} \tag{3.14}$$

The χ at every fourth grid point is obtained by solving (3.14). The $\chi_{4i\pm1}$ and $\chi_{4i\pm2}$ are updated with the newly obtained χ_{4i} from (3.14): e.g.,

$$\begin{aligned} \chi_{4i+2} &= g_{4i+2} + \frac{1}{2}(\chi_{4i} + \chi_{4i+4}) \\ \chi_{4i+1} &= r_{4i+1} + \frac{1}{2}(\chi_{4i} + \chi_{4i+2}), \end{aligned} \tag{3.15}$$

where g and r are frozen for the periods of Δt_g and Δt_r, respectively. Δt_g and Δt_r are called as the frozen times for the intermediate and small scales, respectively.

As a second step, (3.11) is approximated at every other grid point at $t = l\Delta t_g$ (l is an integer) using a second-order central difference:

$$-\frac{\Gamma}{2}\chi_{2i+2} + (1+\Gamma)\chi_{2i} - \frac{\Gamma}{2}\chi_{2i-2} = R_{1_{2i}} + \Gamma(r_{2i+1} + r_{2i-1}). \tag{3.16}$$

The χ at every other grid point is obtained by solving (3.16). The $\chi_{2i\pm1}$ are then updated with the frozen r, and $g_{4i\pm2}$ are updated: e.g.,

$$\chi_{2i+1} = r_{2i+1} + \frac{1}{2}(\chi_{2i} + \chi_{2i+2}) \tag{3.17}$$

$$g_{4i+2} = \chi_{4i+2} - \frac{1}{2}(\chi_{4i} + \chi_{4i+4}). \tag{3.18}$$

As a third step, (3.11) is approximated at every point at $t = l\Delta t_r$:

$$-\Gamma\chi_{i+1} + (1+2\Gamma)\chi_i - \Gamma\chi_{i-1} = R_{1_i}. \tag{3.19}$$

The χ at every grid point is obtained by solving (3.19). The $r_{2i\pm1}$ are then updated as

$$r_{2i+1} = \chi_{2i+1} - \frac{1}{2}(\chi_{2i} + \chi_{2i+2}). \tag{3.20}$$

Once χ's are obtained at either $4i$, $2i$ or i points, similar procedures are applied to the other two directions. It is straightforward to extend the procedure described above in the spanwise and wall-normal directions. At the end of these procedures, the streamwise velocity is obtained. Again, the same procedure can be easily applied to the other two velocity components.

Let us write the numerical algorithm of IU:

1. Start with an initial velocity field u^0 or a previous time step $u^{n,k-1} = u^{n-1}$.
2. Solve the discretized momentum equations at $(4i, 4j, 4k)$ grid points (similar to (3.14)) to obtain $\boldsymbol{u}$ at $(4i, 4j, 4k)$ points.
3. Update $\boldsymbol{u}$ at non-$(4i, 4j, 4k)$ points with frozen $\boldsymbol{g}$ and $\boldsymbol{r}$ (see (3.15)).
4. If $t = l\Delta t_g$, go to Step 5. If not, go to Step 2.
5. Solve the discretized momentum equations at $(2i, 2j, 2k)$ grid points (similar to (3.16)) to obtain $\boldsymbol{u}$ at $(2i, 2j, 2k)$ points.
6. Update $\boldsymbol{u}$ at $(2i \pm 1, 2j \pm 1, 2k \pm 1)$ points with frozen $\boldsymbol{r}$ and also update $\boldsymbol{g}$ at $(4i \pm 2, 4j \pm 2, 4k \pm 2)$ points (see (3.17) - (3.18)).
7. If $t = l\Delta t_r$, go to Step 8. If not, go to Step 2.
8. Solve the discretized momentum equations at all the grid points (similar to (3.19)) to obtain $\boldsymbol{u}$ at all points.
9. Update $\boldsymbol{r}$ at $(2i \pm 1, 2j \pm 1, 2k \pm 1)$ points (see (3.20)).
10. Solve the Poisson equation (3.7) at all points, update the velocity (3.8), and go to Step 2.

Note that it is not necessary for us to decompose the velocity into the same levels of scales in all the directions. That is, one may decompose the flow into two scales in the wall-normal direction and three scales in the streamwise and spanwise directions.

The interpolation used in obtaining the neighboring velocity (e.g., (3.15)) deteriorates the momentum conservation property and the mass conservation is easily violated unless the Poisson equation (3.7) is solved at each time step. However, the requirement of the Poisson solution at each time step clearly diminishes the advantage of using the IU method.

The modification and application of the present multilevel scheme to the turbulent channel flow are in progress and will be reported in the future.

REFERENCES

Chen, M. & Temam, R. 1991 Incremental Unknowns for Solving Partial Differential Equations. *Numer. Math.* **59,** 255-271.

Chen, M. & Temam, R. 1993 Nonlinear Galerkin Method with Multilevel Incremental Unknowns. *Contributions in Numerical Mathematics, WSSIAA.* **2,** 151-164.

DEBUSSCHE, A., DUBOIS, T., & TEMAM, R. 1995a The Nonlinear Galerkin Method: A multiscale method applied to the simulation of homogeneous turbulent flows. *Theoretical and Computational Fluid Dynamics.* **7(4),** 279-315.

DUBOIS, T., JAUBERTEAU, F., & TEMAM, R. 1995b Dynamic multilevel methods in turbulence simulations. *Computational Fluid Dynamics Review*, M. Hafez and K. Oshima ed., Wiley Publishers.

DUBOIS, T., JAUBERTEAU, F., & TEMAM, R. 1996 A Comparative Study of Multilevel Schemes in Homogeneous Turbulence. *Proceeding of ICNMFD.* **15.**

GOTTLIEB, D. & ORSZAG, S. A. 1977 Numerical Analysis of Spectral Methods: Theory and Applications. *SIAM-CBMS, Philadelphia.*

KIM, J., MOIN, P., & MOSER, R. 1987 Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177,** 133-166.

SHEN, J. 1994 Efficient spectral-Galerkin method I. direct solvers for second- and fourth-order equations by using Legendre polynomials. *SIAM J. Sci. Comput.* **15,** 1489–1505.

SHEN, J. 1995 Efficient spectral-Galerkin method II. direct solvers for second- and fourth-order equations by using Chebyshev polynomials. *SIAM J. Sci. Comput.* **16,** 74–87.

SHEN, J. 1996 Efficient Chebyshev-Legendre Galerkin methods for elliptic problems. *Proceedings of ICOSAHOM'95. Houston J. Math.*, 233-239.