



ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 186 (2003) 610–629

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

Spectral element filtering techniques for large eddy simulation with dynamic estimation

H.M. Blackburn^{*}, S. Schmidt

CSIRO Manufacturing and Infrastructure Technology, PO Box 56, Highett, Vic. 3190, Australia

Received 10 October 2001; received in revised form 10 January 2003; accepted 25 January 2003

Abstract

Spectral element methods have previously been successfully applied to direct numerical simulation of turbulent flows with moderate geometrical complexity and low to moderate Reynolds numbers. A natural extension of application is to large eddy simulation of turbulent flows, although there has been little published work in this area. One of the obstacles to such application is the ability to deal successfully with turbulence modelling in the presence of solid walls in arbitrary locations. An appropriate tool with which to tackle the problem is dynamic estimation of turbulence model parameters, but while this has been successfully applied to simulation of turbulent wall-bounded flows, typically in the context of spectral and finite volume methods, there have been no published applications with spectral element methods. Here, we describe approaches based on element-level spectral filtering, couple these with the dynamic procedure, and apply the techniques to large eddy simulation of a prototype wall-bounded turbulent flow, the plane channel, using a mixing length-based eddy viscosity subgrid-scale model. The methods outlined here may be carried over without modification to more complex geometries.

© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Spectral element; Dynamic; Large eddy simulation

1. Introduction

Spectral element methods [20,28,34] are by now well-established for the simulation of turbulent flows in moderately complex geometries. The methods have a number of attractive features in these applications. As high-order finite element techniques, they can deal with arbitrary geometric complexity, and are capable of local mesh adaption [13] by either increasing the number of elements (*h*-refinement) or increasing the polynomial order within elements (*p*-refinement). For smooth solution spaces, the methods provide asymptotically exponential rates of spatial convergence with *p*-refinement, although in the present application it is the low numerical diffusion and dispersion exhibited by the discretisation that is potentially more significant. Compared to low-order finite element methods, the relatively high degree of structure of data

^{*} Corresponding author. Tel.: +61-3-9252-6330; fax: +61-3-9252-6240.

E-mail address: hugh.blackburn@csiro.au (H.M. Blackburn).

within each element promotes efficient numerical implementations, particularly when block methods can be exploited in matrix–matrix multiply operations. The low degree of data connectivity between elements facilitates concurrent implementations which use domain decomposition, and, with efficient element-level operations, promotes the use of matrix-free methods for solution of partial differential equations.

To date however, there have been few published attempts to extend spectral element techniques to the field of large eddy simulation (LES) of turbulent flows. The most broadly successful LES methodology yet developed is the so-called ‘dynamic procedure’ (DP) [9] which uses two-level filtering (with one level of explicit filter application) and assumptions about similarity of turbulent processes at two cut-off length scales in order to compute the coefficients of an underlying sub-grid scale (SGS) turbulence model during runtime. This method has proven robust and quite accurate in a variety of flows, even when allied with comparatively unsophisticated SGS models, such as the Smagorinsky eddy viscosity model [15]. Primarily because it is able to adapt model coefficients to provide the correct behaviour near walls, the DP can be used in complex geometries, and so a marriage of spectral element and DP techniques would seem to be a promising combination for complex geometry LES. What is needed as a precondition are techniques for explicit filtering of scalar fields that sit within the spectral element computational framework.

Here, we describe three alternative and comparatively inexpensive spectral element filtering strategies, employ them for LES using the DP, and investigate their performance for the simulation of turbulent channel flow, a well-documented wall-bounded flow. The filters are all applied locally within each element, which is a key factor in keeping the cost down: global filtering operations for spectral element methods [31] are expensive as they necessitate solving global equations, and are possibly not particularly relevant for application of the DP, where the concentration is on the smallest length scales. In essence, the filters are all spectral filters, but the shape functions to which they are applied have only local support, so the filters can be sharp in spectral space but local in physical space. As no special action is taken in the turbulence modelling to deal with the presence of walls, the methods carry over without alteration to more complex geometries.

2. Spectral element basis functions

In brief, the spectral element method [6,13,19,28,34] is a high-order finite element method that takes advantage of the favourable convergence properties (asymptotically exponential for smooth solutions) of interpolation using families of orthogonal polynomials. These families can be characterised as polynomial eigenfunction solutions of singular Sturm–Liouville differential equations [6]. The most general class of these eigenfunctions are the Jacobi polynomials, while the Chebyshev and Legendre polynomials are among the more commonly used subclasses. In the spectral element method, one typically does not use these expansion functions directly, as they do not provide C^0 continuity across element boundaries; instead, it is the Gauss quadrature nodes and weights associated with the orthogonal polynomials that provide the linkage, and maintain the exponential convergence property. Our discussion is particularised to the Gauss–Lobatto (GL) quadrature points, which include the interval end points.

Defined on the interval $[-1, 1]$ are the $N_p = N + 1$ GL quadrature points x_j and associated weights w_j , related to a set of orthogonal polynomials spanning \mathbb{P}_N . These quadrature points and weights make the following relationship between quadrature and integration exact for all polynomials $p \in \mathbb{P}_{2N-1}$:

$$\sum_{j=0}^N p(x_j)w_j = \int_{-1}^1 p(x)w(x) dx. \quad (1)$$

Here $w(x)$ is the continuous weight function associated with the orthogonal basis underlying the corresponding GL rule, and which appears in the related Sturm–Liouville problem. For Legendre polynomials,

for example, $w(x) = 1$, and we have the Gauss–Lobatto–Legendre (GLL) quadrature points and weights: this is the combination usually employed in the spectral element method. More generally, for Jacobi polynomials $J^{\alpha,\beta}$, the weight function is of the form $w(x) = (1-x)^\alpha(1+x)^\beta$.

Given two functions u, v , continuous on $[-1, 1]$, the continuous and discrete inner products are respectively defined as

$$(u, v)_w = \int_{-1}^1 u(x)v(x)w(x) dx, \quad (2)$$

$$(u, v)_N = \sum_{j=0}^N u(x_j)v(x_j)w_j, \quad (3)$$

and $(u, v)_N = (u, v)_w$ provided $uv \in \mathbb{P}_{2N-1}$. Functions u and v are termed continuously or discretely orthogonal on $[-1, 1]$ if the corresponding inner products are non-zero only when $u = v$.

Fig. 1 illustrates three alternative families of functions associated with the spectral element method on the interval $x = [-1, 1]$, up to fifth-order polynomials. In the first column (Fig. 1(a)) are the Legendre polynomials. In considering a hierarchical basis for filtering, these are a natural candidate family owing to their close association with the quadrature used in the spectral element method (if instead, the Gauss–Lobatto–Chebyshev points and weights were used, we might consider the Chebyshev family as a candidate for filtering). As noted above, the Legendre basis is by definition orthogonal under the weight function $w(x) = 1$. A potential difficulty with these basis functions in the present context is that filtering will break C^0 continuity between elements, which stems from the fact that none of the basis functions is zero at the ends of the interval. This is not necessarily a cause for alarm, as we will be filtering only intermediate fields, not used directly for constructing a solution, which *will* retain C^0 continuity.

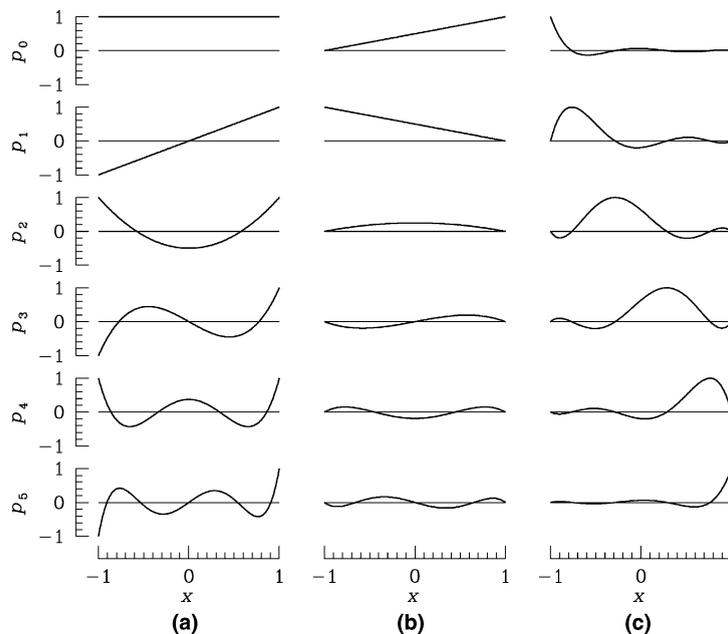


Fig. 1. Families of shape functions associated with the spectral element method, shown for polynomial order $N = 5$ on the interval $[-1, 1]$. (a) The Legendre polynomials; (b) the ‘modal’ polynomials; (c) Lagrange cardinal functions based on the GLL quadrature nodes.

Fig. 1(b) illustrates the so-called ‘modal’ polynomial basis, which is the set of functions:

$$\begin{aligned}
 p_0(x) &= \frac{1}{2}(1+x), \\
 p_1(x) &= \frac{1}{2}(1-x), \\
 p_i(x) &= \frac{1}{4}(1+x)(1-x)J_i^{1,1}(x), \quad i \geq 2.
 \end{aligned}
 \tag{4}$$

This is the family associated with the p -version of the finite element method [39], and are the shape functions used in the ‘modal’ version of the spectral element method [19]. Like the Legendre polynomials, it provides a hierarchy of modes, although these are neither continuously or discretely orthogonal. For $i \geq 2$, the modes are zero at the ends of the interval, meaning that filtering higher-order modes will preserve C^0 continuity across element boundaries.

Fig. 1(c) shows the family of Lagrange interpolants (or Lagrange cardinal functions) for the GLL nodes, $N = 5$. These are the shape functions employed in the ‘nodal’ version of the spectral element method. The functions do not form a hierarchical set, so it is not immediately obvious how to perform a filtering operation. On the other hand, it is only the shape functions associated with the end-point nodes that are non-zero at the ends of the interval, leaving open the possibility of preserving C^0 continuity. In fact, since each successive function is only non-zero at its associated GLL point, and zero at the others, the functions are discretely (although not continuously) orthogonal. In the corresponding spectral element method, this property carries over to produce mass matrices that are diagonal when GLL quadrature is used in their construction.

3. Element-level filtering and projection

The current work is directed principally to the tensor-product spectral element basis, which may be applied to two-dimensional distorted-quadrilateral and three-dimensional distorted-hexahedral elements. In the modal version of the spectral element method, the shape functions are tensor products of the one-dimensional modal polynomials and in the nodal version they are tensor products of the GLL Lagrange interpolants. However, the central ideas needed for filtering are most simply developed using a single-element one-dimensional space.

3.1. Low-pass spectral filtering via matrix–vector polynomial transform

If the basis functions p form a hierarchical set in the sense that they have progressively higher spatial frequencies, then low-pass filtering in the spectral domain makes sense. The mechanism for transforming between the nodal representation $u(x_j)$ and the spectral representation u_k^* is the discrete polynomial transform (DPT), as outlined in Appendix A. In one space dimension, the forward and inverse transformations may be implemented as matrix–vector multiplications

$$\mathbf{u}^* = \mathbf{B}^{-1}\mathbf{u}, \quad \mathbf{u} = \mathbf{B}\mathbf{u}^*,
 \tag{5}$$

where matrix entry B_{jk} represents basis function p_k evaluated at quadrature point x_j .

Using the DPT, a spectral filter can be applied in the space of the coefficients u_k^* . This process is illustrated in diagrammatic form in Fig. 2. Defining a filter vector l_k and matrix $\mathbf{L} = \text{diag}(l_0, \dots, l_N)$, the filtered nodal values $\bar{u}(x_j)$ are in practice produced by application of a single matrix operator

$$\bar{\mathbf{u}} = \mathbf{B}\mathbf{L}\mathbf{B}^{-1}\mathbf{u} = \mathbf{F}\mathbf{u}.
 \tag{6}$$

In general, \mathbf{F} has no special structure and the one-dimensional filtering of the $N_p = N + 1$ nodal values of \mathbf{u} is an $\mathcal{O}(N_p^2)$ operation, but this operation count could potentially be reduced by exploiting symmetries of

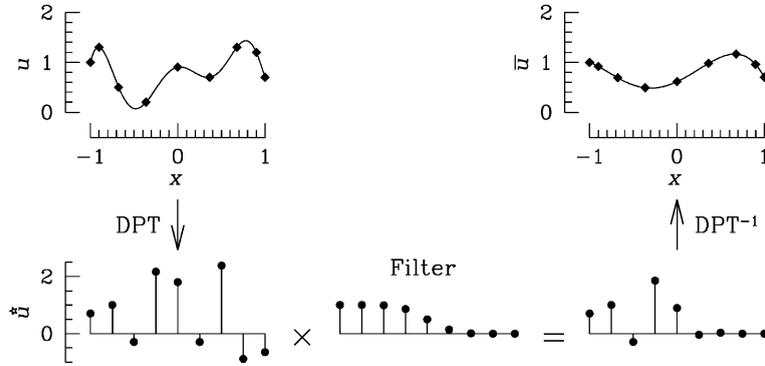


Fig. 2. Illustration of filtering in the space of spectral coefficients u_k^* via the discrete polynomial transform (DPT).

the basis functions [4]. The filter weights l_k can be chosen in any appropriate way; for lowpass filtering, the l_k could be a spectral top-hat filter, or a set that is smooth in spectral space, such as the Boyd–Vandeven coefficients [26]. As a spectral filtering procedure, the method is idempotent (i.e., $\mathbf{FF} = \mathbf{F}$) if the vector of filter coefficients l_k has a sharp cutoff in spectral space.

In work presented here, low-pass spectral filtering has been applied alternatively in either the Legendre or modal polynomial basis space; in Sections 4 and 5, suffixes L and M will be applied to denote the use of these two filtering techniques.

3.2. Thin-out filtering via Lagrange-interpolant projection

If the basis functions *do not* form a hierarchical set, e.g., for the Lagrange interpolants, then filtering can be implemented by projecting to a lower-order set of basis functions in the same family. Defining I_N^M as the operator that interpolates a polynomial of order N with $N_p = N + 1$ nodes onto a set of $M_p = M + 1$ nodal points we produce the Lagrange-interpolant projector $F = I_M^N I_N^M$. This projects the original nodal values $u(x_k)$ to another set at the same nodal locations x_k , but derived from the Lagrange interpolants through the smaller $(M + 1)$ set of GLL nodes. This process is illustrated in diagrammatic form in Fig. 3.

The operators I_N^M and I_M^N can be derived in any appropriate fashion from the unique polynomials passing through the corresponding points. Perhaps the most straightforward way is to directly employ the Lagrange interpolants, from which

$$I_{Njk}^M = \left\{ \prod_{p=0}^N (x_j - x_p) \right\} / \prod_{\substack{q=0 \\ q \neq k}}^N (x_k - x_q), \quad j = 0, \dots, M, \quad k = 0, \dots, N, \tag{7}$$

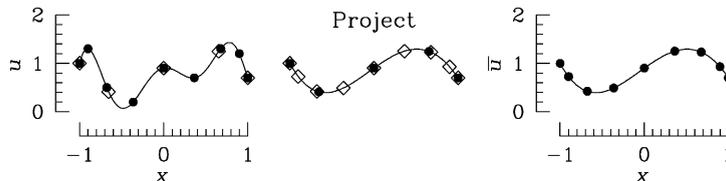


Fig. 3. Illustration of filtering using projection onto a lower-order Lagrange polynomial interpolant, using the GLL family of nodes. At each stage, ● represents the data at the interpolant nodes and ◇ the interpolated data extracted for projection.

where all the x values are GLL nodes. As matrix operators, \mathbf{I}_N^M is $M_p \times N_p$, \mathbf{I}_M^N is $N_p \times M_p$ and $\mathbf{F} = \mathbf{I}_N^M \mathbf{I}_M^N$ is $N_p \times N_p$, the cost of filtering by applying the projector is again $\mathcal{O}(N_p^2)$. The operator \mathbf{F} is idempotent.

The technique of employing a projector of this form has previously been employed for the stabilisation of spectral element direct numerical simulation [8], however in that case the projection was to just one order lower, i.e., $M_p = N_p - 1$, whereas here $M_p = (N_p + 1)/2$ is typically used, where $/$ represents integer division.

In Sections 4 and 5, the suffix P will be employed to denote use of Lagrange-interpolant projection filtering.

3.3. A comparison of filtering strategies in one space dimension

In Fig. 4 is presented a comparison of the filtering techniques introduced above as applied to the (arbitrary) one-dimensional data set shown in Fig. 4(a). For all cases, the solid line represents the polynomial interpolant through the current data set, but for cases (b–d) the original interpolant (case a) is also shown, as a dotted line. In Fig. 4(b) we have the results of applying DPT-based filtering in the Legendre spectral coefficients, using a zero-lag Boyd–Vandeven filter of ninth order. The filtered data are obviously smoother, but the end values are not preserved, as noted in Section 2. In a multi-element case, this will break C^0 continuity, but breakage does not necessarily represent a problem, because the filter will be applied to an intermediate field and the results used in an element-by-element fashion.

Fig. 4(c) shows the equivalent DPT-based filtering, but in the modal basis, rather than the Legendre basis. Here we see that the end values of the original data are preserved, leading to preservation of C^0 continuity in a multi-element case. Fig. 4(d) shows the outcome of Lagrange-interpolant projection filtering; again the end values of the original data are preserved. The points where the new and original interpolants intersect correspond to the intermediate lower-order projected values.

3.4. Filtering in multiple space dimensions

The filtering techniques described above can be carried over with little modification to multiple space dimensions when tensor-product basis functions are used, for example, in isoparametrically mapped quadrilateral or hexahedral elements. Filters are applied in similar tensor-product fashion; the advantage obtained by this approach is that the filtering operations are efficient, since by taking advantage of

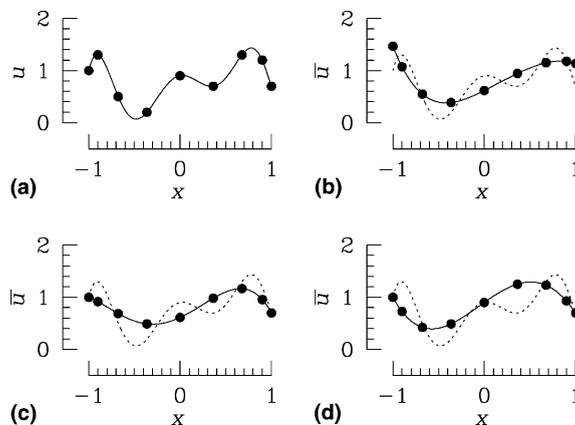


Fig. 4. A comparison of filtering methods for $N_p = 9$, for an arbitrary set of values with GLL abscissae: (a) the original data; (b) DPT filtered values using the Legendre polynomial basis; (c) DPT filtered values using the modal polynomial basis; (d) filtering via Lagrange-interpolant projection. In (b–d), the dotted line represents the original interpolant. Note that cases (c,d) preserve end values.

sum-factorisation methods [32], they have an $\mathcal{O}(N_p^{D+1})$ cost in each element, where N_p is the number of points along the edge of an element and D is the number of space dimensions. This is the same order of cost as applying a gradient operator.

For example, in two space dimensions, \mathbf{u} can now be taken as a two-dimensional matrix of coefficients in each element. If \mathbf{u} has row-major ordering, then a two-dimensional tensor-product filtering operation is performed by

$$\bar{\mathbf{u}} = \mathbf{F}\mathbf{u}\mathbf{F}^T, \quad (8)$$

where \mathbf{F} is a one-dimensional filter/projection matrix. This convolution is applied to the data of each element, with the same matrix \mathbf{F} in each case (for equal-order elements): the cost of applying the filter is $\mathcal{O}(N_p^3 N_{el})$, where N_{el} is the number of elements. The extension to three space dimensions follows naturally. The associated matrix–matrix multiply operations are typically implemented through vendor-optimised

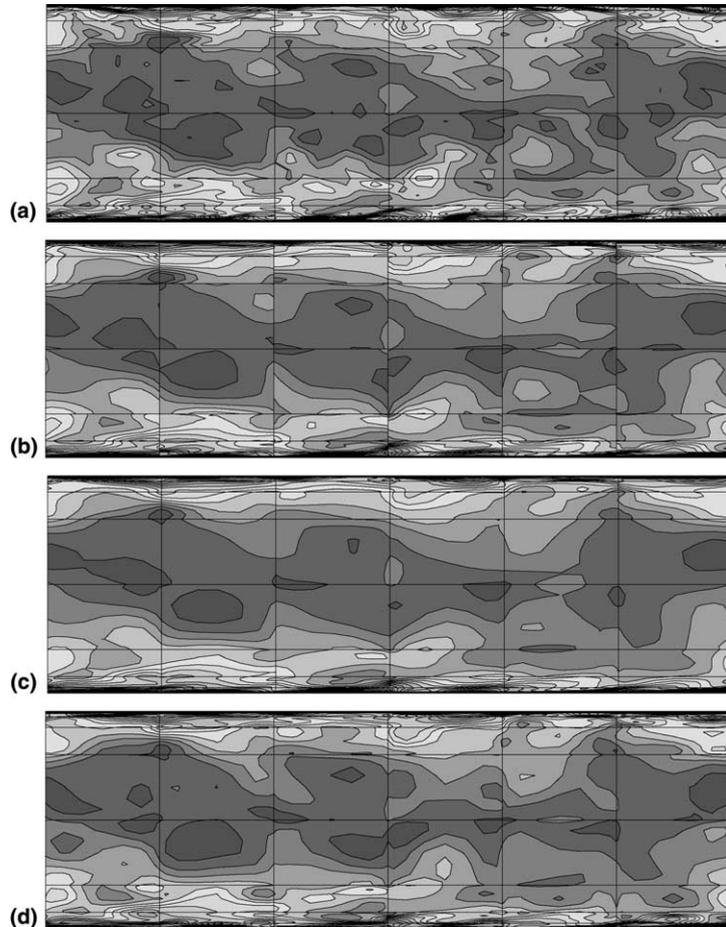


Fig. 5. An illustration of filtering techniques in two space dimensions, showing contours of streamwise velocity on planes normal to the spanwise direction from a spectral element LES of turbulent channel flow. In each planar slice, rectangular outlines show the boundaries of the 60 spectral elements used in the mesh (6 streamwise \times 10 cross flow), and the contour levels are the same in each case. (a) Unfiltered field; (b) filtered with the DPT and Boyd–Vandeven filter coefficients in the Legendre basis space; (c) the same filter, but applied in the modal basis; (d) filtered by projection from 8×8 tensor-product GLL Lagrange basis to the 4×4 basis.

library subroutines, however it is also straightforward to vectorise the operations of (8) over elements, if desired.

In Fig. 5, we present a comparison of two-dimensional filtering operations as applied to streamwise velocity data from a spectral element LES of turbulent channel flow, analogous to the one-dimensional comparisons shown in Fig. 4. It can be seen that all three types of filtering operation (b–d) produce a similar degree of smoothing to the original data. A detailed examination reveals breakage of C^0 continuity across element boundaries for Fig. 5(b), where filtering is applied in the Legendre basis.

As a final point, it should be noted that filtering and differentiation in the spectral element basis do not commute for any of the techniques outlined here. Where both filtering and differentiation operations are required (e.g., in forming the filtered rate-of-strain tensor, see Appendix B), we have carried out filtering after differentiation.

4. Implementation of dynamic LES

The filtering methods outlined above, in conjunction with the dynamic Smagorinsky model (DSM, see also [7,37,38]), form the basis for our LES method. The Germano–Lilly dynamic procedure ([9,27], reviewed in Appendix B) returns an estimate of a spatially varying mixing length l_s , which in turn is used to compute a turbulent eddy viscosity ν_t . The associated SGS stress is

$$\tau = -2l_s^2 |\overline{\mathbf{S}}| \overline{\mathbf{S}} = -2\nu_t \overline{\mathbf{S}}, \tag{9}$$

where $\overline{\mathbf{S}} = [\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T]/2$ is the resolved-scale rate-of-strain tensor and $|\overline{\mathbf{S}}| = (2\overline{\mathbf{S}} : \overline{\mathbf{S}})^{1/2}$.

The momentum and continuity equations for the resolved-scale velocity field $\overline{\mathbf{u}}$ are written as

$$\partial_t \overline{\mathbf{u}} = -\mathcal{N}(\overline{\mathbf{u}}) - \nabla \Pi + \nu \nabla^2 \overline{\mathbf{u}} + 2\nabla \cdot \nu_t \overline{\mathbf{S}}, \tag{10}$$

$$\nabla \cdot \overline{\mathbf{u}} = 0, \tag{11}$$

where $\mathcal{N}(\overline{\mathbf{u}}) = (\overline{\mathbf{u}} \cdot \nabla \overline{\mathbf{u}} + \nabla \cdot \overline{\mathbf{u}} \overline{\mathbf{u}})/2$ is the skew-symmetric form of the nonlinear advection terms, and Π is a modified pressure (see Appendix B). The skew-symmetric form of the advection terms is found to reduce aliasing errors in spectral calculations, and although comparatively computationally expensive, is found to give good results especially when resolution is at a premium [2,21,41]. The notations DSM-L, DSM-M and DSM-P will be used to denote the use of the dynamic Smagorinsky model with, respectively, filtering in the Legendre or modal polynomial bases via DPT, and Lagrange-interpolant projection filtering.

The momentum equations are integrated in time using a mixed explicit–implicit time-splitting scheme [17]; all the results presented here were computed using a second-order time variant of the method. In order to minimise the impact of the diffusive contribution of the divergence of SGS stress on the stability of the method, we take the approach outlined in [18], where the sum of the molecular and turbulent eddy viscosities $\nu_{\text{eff}} = \nu + \nu_t$ is decomposed into a spatially constant component ν_{ref} and a spatially varying component $(\nu_{\text{eff}} - \nu_{\text{ref}})$. The momentum equations (10) thus become

$$\partial_t \overline{\mathbf{u}} = -\mathcal{N}(\overline{\mathbf{u}}) - \nabla \Pi + \nu_{\text{ref}} \nabla^2 \overline{\mathbf{u}} + 2\nabla \cdot (\nu_{\text{eff}} - \nu_{\text{ref}}) \overline{\mathbf{S}}. \tag{12}$$

When integrating (12), the term $2\nabla \cdot (\nu_{\text{eff}} - \nu_{\text{ref}}) \overline{\mathbf{S}}$ is treated explicitly, along with the nonlinear terms, while the term $\nu_{\text{ref}} \nabla^2 \overline{\mathbf{u}}$ is treated implicitly, enhancing the overall numerical stability of the scheme. In addition, the global Helmholtz problem solved for the implicit viscous terms enforces the correct velocity boundary conditions and ensures C^0 continuity of the solution on element boundaries. The value of ν_{ref} is chosen to be approximately equal to the maximum value of ν_{eff} , which is unknown a priori, but ν_{ref} can be adjusted during the computation without adverse effects.

We have used a spectral element–Fourier method [1,16] for spatial discretisation, with two-dimensional spectral elements covering a planar domain of arbitrary geometric complexity, and Fourier expansions in a homogeneous direction – consult [13,19] for further detail. This approach allows us to test the element-level filtering ideas advanced in Section 3, while maintaining efficient parallel solution techniques for the global linear Helmholtz problems inherent in the time-splitting. Filtering in the homogeneous direction is carried out in the Fourier-transformed space using smooth spectral filters of the Boyd–Vandeven type. Solutions are not explicitly dealiased in the homogeneous direction (e.g., by the 3/2 rule), although the use of skew-symmetric nonlinear terms is helpful in minimising aliasing errors.

As noted in Appendix B, stability problems associated with negative values of the dynamic estimate usually force adoption of some constraint, and various means have been proposed. We have used a combination of homogeneous averaging of the numerator and denominator of (B.13), in conjunction with lowpass filtering of the estimate in time [5]

$$l_S^{2(n+1)} = \varepsilon l_S^{2(*)} + (1 - \varepsilon) l_S^{2(n)}, \quad (13)$$

where $l_S^{2(*)}$ is the initial estimate computed with homogeneous averaging, and $l_S^{2(n)}$ is the value from the previous timestep. This first-order recursive filtering has an associated time constant $\kappa = -\ln(1 - \varepsilon)/\Delta t$. In addition, clipping of the total viscosity is used, although this typically becomes inactive as κ is decreased. The initial estimate of l_S^2 can be supplied by the Smagorinsky model or any other appropriate method.

5. Application to turbulent channel flow

Turbulent channel flow provides a convenient and well-documented setting in which to test the performance of the method. The computational domain is doubly periodic (streamwise and spanwise), allowing a choice of homogeneous direction for Fourier expansions: we have taken this to be spanwise. For all simulations, the domain extents are 2δ in the wall-normal (y) direction, $2\pi\delta$ in the streamwise (x) direction and $\pi\delta$ in the spanwise (z) direction, as illustrated in Fig. 6. Previous investigations have established that for this flow, these domain extents are sufficient to overcome most contamination effects resulting from periodic correlations. In order to drive the flow in the streamwise direction, a body force per unit mass of magnitude τ_w/ρ was applied in the appropriate component of the filtered momentum equations, where τ_w is the time-average wall shear stress. In presenting results, the Reynolds decomposition $\bar{\mathbf{u}} = \mathbf{U} + \mathbf{u}$ for the

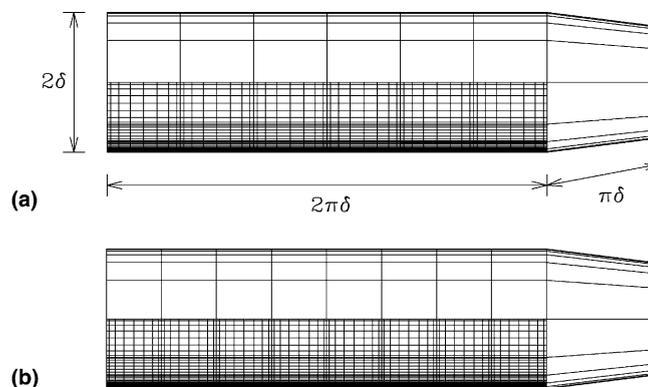


Fig. 6. Perspective views of the spectral element–Fourier meshes for plane channel flows: (a) 60-element (10 cross flow \times 6 streamwise) mesh for $Re_\tau = 650$; (b) 96-element mesh for $Re_\tau = 1015$. In the lower half of each mesh, element GLL nodes are shown for eighth-order tensor-product shape functions.

resolved velocity field is employed, where \mathbf{U} is the ensemble average of $\bar{\mathbf{u}}$. All velocities are normalised by the friction velocity, e.g., $U^+ = U/u_\tau$, where $u_\tau = (\tau_w/\rho)^{1/2}$.

In designing meshes for simulations, we employ rules-of-thumb for wall-resolving LES as outlined in [35]. In addition we have found that using one element to cover the viscous sublayer ($y^+ \lesssim 10$, where $y^+ = yu_\tau/\nu$) and another to cover the buffer layer ($10 \lesssim y^+ \lesssim 35$), where turbulent energy production is greatest, provides good results. A geometric progression of wall-normal element sizes is then used to reach the centre of the channel.

5.1. $Re_\tau = 650$ LES

The first test case is for $Re_\tau = u_\tau\delta/\nu = 650$ (or approximately 14,000 based on δ and the bulk velocity). This enables comparison to $Re_\tau = 590$ direct numerical simulation (DNS) results [30], and to experimental results for $Re_\tau = 640$ and 708 [14,40]. The computational mesh is as illustrated in Fig. 6(a), with 10 elements in the wall-normal direction and 6 in the streamwise direction. 64 planes of data were employed in the spanwise direction. With eighth-order tensor-product shape functions employed in each element ($N = 8$), the total number of independent mesh nodes is 311,040. The associated mesh sizes in wall units are $\Delta z^+ = 32$, $\Delta x^+ = 85$, and near the wall, a wall-normal lengthscale $\Delta y^+ = 1.2$; here, Δx^+ and Δy^+ are calculated using the element side lengths divided by N .

Fig. 7 shows the DNS and experimental data in comparison with three spectral element calculations that *do not* employ the DP. The experimental and DNS comparison data are seen in Fig. 7(a) to be in quite good agreement for the mean velocity profile (of the two sets of experimental results, those of [14] are in best agreement with the DNS far from the wall), but the agreement is less satisfactory for the streamwise rms velocity fluctuation, as can be seen from an examination of Fig. 7(b). Of the two sets of experimental data, those of [40] would appear, by comparison to the DNS, to provide a more reliable estimate of the peak value of u_{rms}^+ , but closer to the wall ($5 \lesssim y^+ \lesssim 10$), the results of [14] are better.

The spectral element simulation results also shown in Fig. 7 are revealing. The results for the ‘no-model’ case (i.e., under-resolved DNS) provide a good match to the DNS data [30] for the mean and wall-normal fluctuating velocities in the viscous sublayer and into the start of the buffer layer, although u_{rms}^+ is high, indicating that the near-wall region is, as expected, slightly under-resolved. However the predicted mean velocity in the outer region ($y^+ \gtrsim 35$) is significantly low, of order 5% low at the centre plane. Results for a non-dynamic Smagorinsky-based LES without wall damping are bad throughout the flow, and the mean velocity near the centre plane is too low by approximately 10%. With wall damping included, Smagorinsky-based LES performs well in reproducing the mean velocity profile and the profile of fluctuating streamwise velocity, although the wall-normal fluctuating velocities are too low in the inner region. The addition of the SGS model improves the near-wall predictions of u_{rms}^+ , but makes those for v_{rms}^+ deteriorate.

For both non-dynamic Smagorinsky-based LES, the model constant was adjusted to $c_s = 0.12$, mesh length scales were computed within each element as $\bar{\Delta} = (\Delta x \Delta y \Delta z)^{1/3}$, and for the case in which wall damping was used, the near-wall mesh length scale was multiplied by $[1 - \exp(-y^{+3}/A^{+3})]^{1/2}$ [36], where the van Driest length scale, $A^+ = 26$. This form of the near-wall damping enforces the correct asymptotic variation of mixing length with distance from the wall, i.e., $l_s^2 \sim y^3$ [37], although there are step changes in l_s^2 at element boundaries, produced by corresponding changes in $\bar{\Delta}$. It should be emphasised that in order to apply the near-wall damping the wall shear stress must be known (as it appears in y^+), and that the Smagorinsky constant $c_s = 0.12$ was tuned by trial and error in order to provide a good match between computed and measured mean velocity profiles. For the DP, neither of these values are needed.

Fig. 8 shows results computed using the dynamic model with Smagorinsky-based SGS treatment and the techniques outlined in Sections 3 and 4. These plots serve to compare the influence of the three different filtering techniques; it can be seen that while all three types of filter provide broadly comparable results, filtering in the Legendre basis (DSM-L) provides the best agreement to the DNS data for the mean velocity

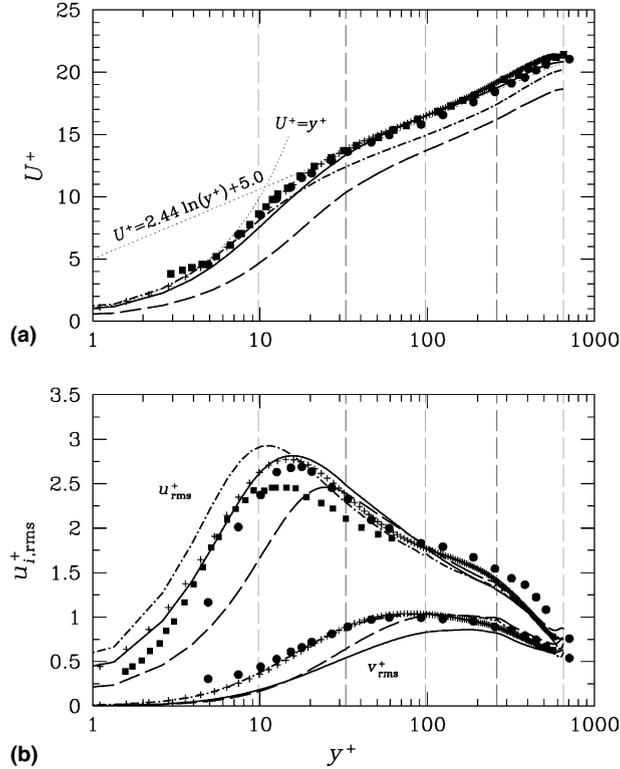


Fig. 7. Channel flow comparison data for $Re_\tau = 650$, (a) mean and (b) rms (u^+ , streamwise; v^+ , wall-normal) velocity profiles: \blacksquare , \bullet , experimental measurements [14,40]; $+$, DNS data [30]. *Non-dynamic* spectral element simulations: dot-dashed line: no SGS model; dashed line: Smagorinsky model, $c_s = 0.12$, without wall damping; solid line: Smagorinsky model, $c_s = 0.12$, van Driest wall damping, $A^+ = 26$. Vertical dashed lines indicate element boundaries.

profile, although projection-filtering (DSM-P) is nearly as satisfactory. Near the wall, DSM-P provides the best agreement to the DNS data for u_{rms}^+ and v_{rms}^+ , but in the outer region, DSM-L and DSM-M are better. The corresponding plots of resolved-scale Reynolds stress $\langle u^+v^+ \rangle$ are shown in Fig. 9, where the resolved-scale contribution is highest for DSM-P.

For all results shown in Figs. 8 and 9, the mixing length update weight $\varepsilon = 0.1$; with timestep $u_\tau \Delta t / \delta = 236 \times 10^{-6}$ the corresponding recursive filter characteristic time constant $\kappa \delta / u_\tau = 446$. Fig. 10 illustrates the temporal variation and ensemble-average value of the dynamic estimate of mixing length for DSM-L and $\kappa \delta / u_\tau = 446$; the profiles shown are averages in both homogeneous directions, streamwise and spanwise. The locations where sharp dips in l_S^2 occur correspond to element boundaries. This feature is common to all of the filtering techniques described here. The value of κ influences the degree of observed fluctuation of l_S^2 (including negative values) about its ensemble-average value, but this value does not vary significantly with κ . Likewise, profiles of mean and fluctuating velocity and resolved-scale Reynolds stress are not significantly influenced by large changes in κ (these data are not presented); what does vary however is the relative amount of clipping that must be applied to maintain the total viscosity ν_{eff} non-negative for each timestep, as shown in Table 1 for DSM-P.

Fig. 11 shows the ensemble-averaged profiles of l_S^2 in wall units, demonstrating that the DSM produces, near the wall, the expected scaling $l_S^2 \sim y^3$ [9,37]. In this region, the large reductions of l_S^2 at element boundaries, prominent in Fig. 10, do not occur.

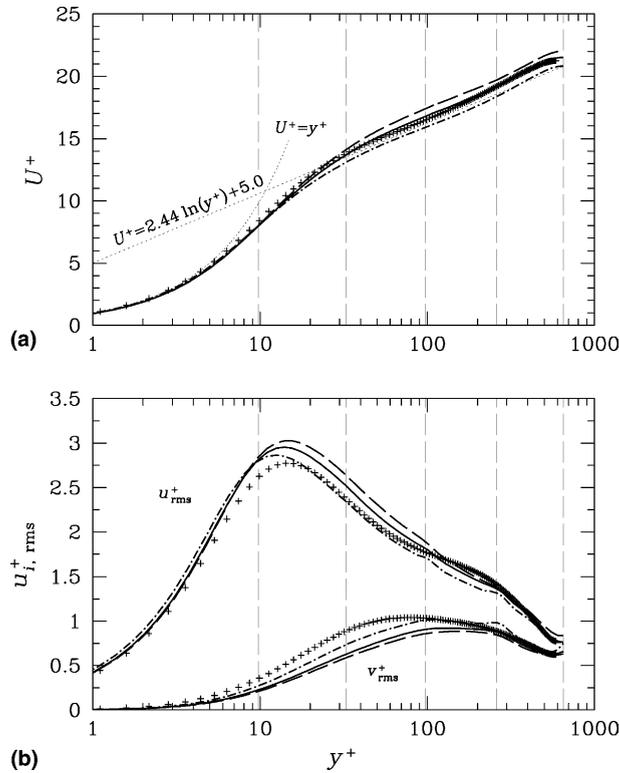


Fig. 8. Channel flow dynamic model simulation results for $Re_\tau = 650$, showing influence of filtering technique: (a) mean and (b) rms (u^+ , streamwise; v^+ , wall-normal) velocity profiles: +, DNS data [30]; dot-dashed line, DSM-P; solid line, DSM-L; dashed line, DSM-M.

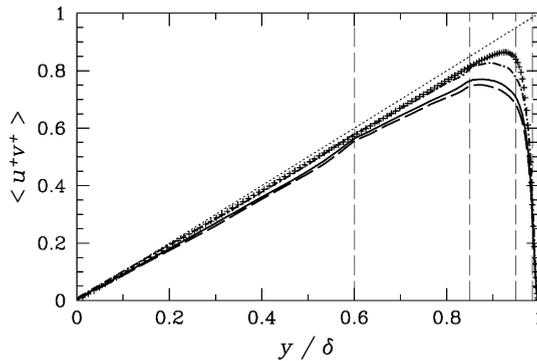


Fig. 9. Channel flow dynamic model simulation results for $Re_\tau = 650$, showing influence of filtering technique on resolved-scale Reynolds stress profiles: +, DNS data [30]; dot-dashed line, DSM-P; solid line, DSM-L; dashed line, DSM-M; dotted line, total of resolved, subgrid and viscous contributions.

5.2. $Re_\tau = 1015$ LES

The second test case is for $Re_\tau = 1015$ (bulk flow Reynolds number approximately 23,000). This enables comparison to experimental results for $Re_\tau = 1014$ and 1017 [14,40]. The computational mesh is shown in

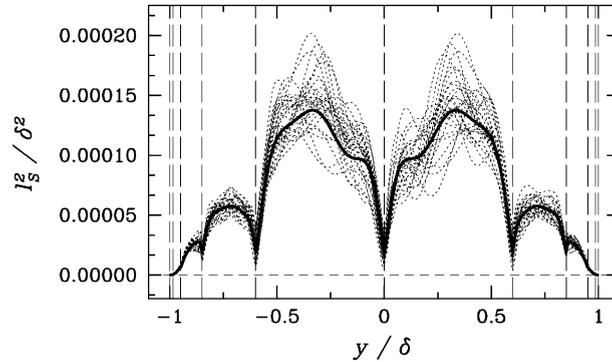


Fig. 10. Temporal variation of the profile of spanwise and streamwise averaged values of I_S^2/δ^2 for DSM-L with $\varepsilon = 0.1$ at $Re_\tau = 650$. Dotted lines represent instantaneous values, full line the time-averaged value, and dashed vertical lines the element boundaries.

Table 1

Relative number of grid points at which clipping was performed as a function of recursive smoothing time constant κ for the $Re_\tau = 650$ DSM-P

ε	0.99	0.1	0.01	0.001
$\kappa\delta/u_\tau$	19.5×10^3	446	42.5	4.24
Clipping fraction	0.02	0.0075	0.0025	0

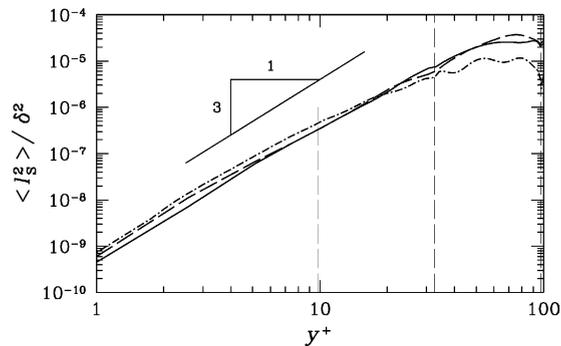


Fig. 11. Channel flow dynamic model simulation results for $Re_\tau = 650$, near-wall variation of the profile of $\langle I_S^2 \rangle / \delta^2$, showing that $I_S^2 \sim y^{+3}$ in the viscous sublayer. Dot-dashed line, DSM-P; solid line, DSM-L; dashed line, DSM-M.

Fig. 6(b), now with 12 elements in the wall-normal direction and 8 in the streamwise direction. 96 planes of data were employed in the spanwise direction. With eighth-order tensor-product shape functions again employed in each element, the total number of independent mesh nodes is 595,968, slightly less than double the number used for $Re_\tau = 650$.

Mean and rms velocity profiles for the $Re_\tau = 1015$ simulations are shown in Fig. 12, while Reynolds stress profiles are shown in Fig. 13: in all cases the experimental results of [40] are presented for comparison (those of [14] are similar). The outcomes for the different filtering methods are similar to those found at $Re_\tau = 650$ (cf. Figs. 8 and 13); filtering in the Legendre basis space (DSM-L) produces the best agreement

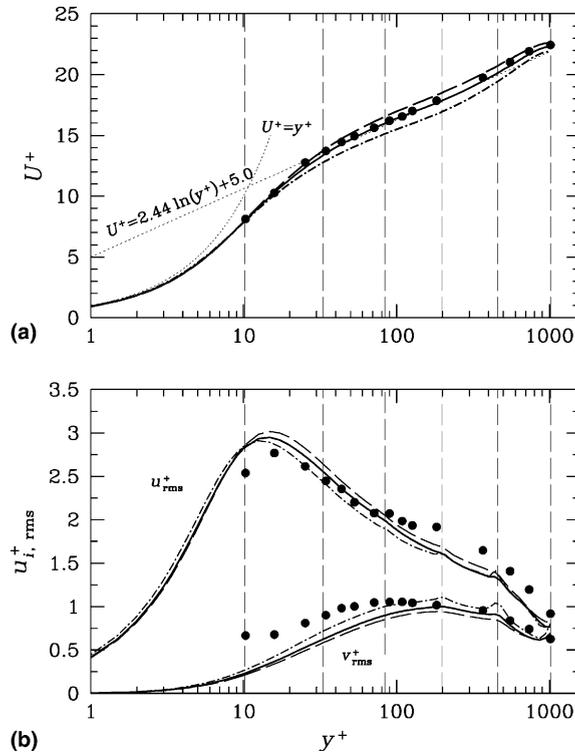


Fig. 12. Channel flow dynamic model simulation results for $Re_\tau = 1015$, showing influence of filtering technique: (a) mean and (b) rms (u^+ , streamwise; v^+ , wall-normal) velocity profiles: \bullet , experimental measurements [40]; dot-dashed line, DSM-P; solid line, DSM-L; dashed line, DSM-M.

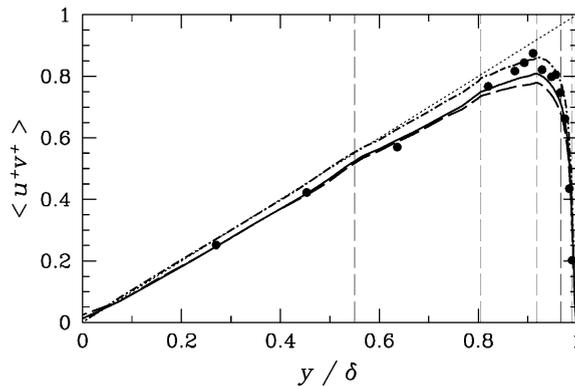


Fig. 13. Channel flow dynamic model simulation results for $Re_\tau = 1015$, showing influence of filtering technique on resolved-scale Reynolds stress profiles: \bullet , experimental measurements [40]; dot-dashed line, DSM-P; solid line, DSM-L; dashed line, DSM-M; dotted line, total of resolved, subgrid and viscous contributions.

with the validation data for the mean velocity profile; all cases slightly overpredict the peak values of fluctuating streamwise velocity, and underpredict the wall-normal values (as is the case for the plain Smagorinsky model with van Driest damping).

6. Discussion and conclusions

The results from the spectral element dynamic Smagorinsky model simulations of turbulent channel flow are, taken overall, very encouraging, with filtering in the Legendre polynomial basis delivering the best predictions of mean velocity profiles, in very good agreement with available validation data. The near-wall predicted values of fluctuating velocities do not compare quite as favourably with the validation data, however the near-wall results are similar to those of other DSM simulations with comparable resolution [9,21,22], suggesting that prediction errors (e.g., u_{rms}^+ high and v_{rms}^+ low, compared to experimental results) here are mostly associated either with marginal resolution, or the SGS model. Further from the wall, the peaks in fluctuation velocity components and Reynolds stresses near element boundary locations suggests that the SGS contribution is lower near element boundaries (where grid points are concentrated). This is also reflected in the estimates of l_s^2 having local minima at element boundaries near the centre of the channel. Near the wall, the DSM produces estimates of mixing lengths that have the expected variation with distance from the wall, and which do not show element-boundary effects, which is as would be expected for wall-resolving LES.

Filtering which breaks C^0 continuity (e.g., filtering in the Legendre basis) does not appear to cause problems, as the filter is applied only to intermediate fields, used to construct eddy viscosity estimates. Similar considerations and reasoning were supplied for application of filtering in the Legendre basis in a spectral element (non-dynamic) large eddy ocean simulation [26]. In that case, filters were applied to vorticity and divergence fields, used in substeps prior to computing the velocity field. The model had no explicit SGS scheme, and filtering was applied in part to stabilise and smooth the solutions. In [3], methods of constructing expansion functions in which filtering could be directly applied on an element-by-element basis without breaking inter-element continuity or boundary conditions were discussed. It would seem that the modal basis functions have similar properties to the functions described in [3], also that filtering in either the modal basis or by using spectral projection could provide similar outcomes to filtering in the bases described there.

Differentiation and filtering do not commute for any of the spectral element filters presented. This does not appear to be important; while it is common in channel flow DSM LES to employ Fourier expansions in the two homogeneous directions and to use sharp spectral filters *only in those directions*, thus avoiding filter commutation errors [12,37], the predictions [9,21,22] are very similar to those presented here.

The cost of the dynamic model is significant, with total simulation times approximately 1.5 times greater than for the equivalent Smagorinsky model, which again is approximately twice as expensive as simulation without a SGS stress model (these figures will vary with the design of the remainder of the simulation code, and computer architecture). There are 24 filter applications per timestep. In comparison to the non-dynamic Smagorinsky model, the return is that the model constant does not need to be known in advance, or tuned to an appropriate value, and that special action to adjust the constant near walls (through van Driest damping) is not required.

The low-pass time filtering of the dynamic estimate appears to perform well, although one expects that for small values of the time constant κ (i.e., large equilibration times), its performance will degrade in highly unsteady flows such as wake flows. It is preferable to other schemes such as Lagrangian averaging [29] which are also applicable to complex geometries principally in that it is computationally cheap to apply; the streamwise interpolation required for Lagrangian averaging is expensive in applications with unstructured grids. The fact that flow statistics seem to be relatively insensitive to values of κ suggests it may be possible in some flows to deploy the dynamic model essentially in an adaption phase, compute a suitable spatial distribution of the model parameter (here, distribution of l_s^2), and then use non-dynamic simulation for further work.

The adoption of high-order spatial discretisations appears to be an important element of successful LES with explicit turbulence models [10,21]. As high-order discretisations, spectral element methods would appear to be very suitable for such application. For complex geometries, other appropriate simulation

technologies include B-spline embedded grids [22,23], and block-structured finite difference methods used in conjunction with high-order schemes [21,24,33]. Part of the motivation for the present work has been to demonstrate that spectral element methods can provide an attractive alternative to other high-order methods for application to large eddy simulation.

Acknowledgements

This work was supported via a Merit Allocation Grant from the Australian Partnership for Advanced Computing (APAC). We would like to thank the staff of this institution for their assistance.

Appendix A. Discrete polynomial transform via matrix–vector operations

The unique polynomial interpolant $I_N u$ of the nodal values of u at the set of quadrature points x_j

$$I_N u(x_j) = u(x_j) \tag{A.1}$$

can be represented by the spectral expansion

$$I_N u = \sum_{k=0}^N u_k^* p_k, \tag{A.2}$$

where u_k^* are spectral coefficients and p_k are polynomial basis functions, not necessarily discretely orthogonal, but spanning \mathbb{P}_N . From (A.1) and (A.2)

$$u(x_j) = \sum_{k=0}^N u_k^* p_k(x_j), \tag{A.3}$$

or in matrix–vector form

$$\mathbf{u} = \mathbf{B}\mathbf{u}^*, \tag{A.4}$$

where the matrix entry B_{jk} represents basis function p_k evaluated at quadrature point x_j .

In order to determine the expansion coefficients u_k^* from the nodal values $u(x_j)$ we take the discrete inner product of (A.1) with the basis functions. Each basis function p_j generates an equation of the form

$$(I_N u, p_j)_N = (u, p_j)_N. \tag{A.5}$$

Introducing the spectral expansion (A.3)

$$\left(\sum_{k=0}^N u_k^* p_k, p_j \right)_N = (u, p_j)_N, \tag{A.6}$$

or

$$\sum_{i=0}^N \sum_{k=0}^N u_k^* p_k(x_i) p_j(x_i) w_i = \sum_{i=0}^N u(x_i) p_j(x_i) w_i. \tag{A.7}$$

In matrix–vector form

$$\mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{u}^* = \mathbf{B}^T \mathbf{W} \mathbf{u}, \tag{A.8}$$

where $\mathbf{W} = \text{diag}(w_0, \dots, w_N)$. Thus the spectral coefficients can be computed as

$$\mathbf{u}^* = [\mathbf{B}^T \mathbf{W} \mathbf{B}]^{-1} \mathbf{B}^T \mathbf{W} \mathbf{u} = \mathbf{M} \mathbf{u}. \quad (\text{A.9})$$

This is the discrete polynomial transform (DPT), also referred to as the ‘matrix multiplication transformation’ [4].

Alternatively, one can derive directly from the corresponding inverse relationship (A.4) that $\mathbf{M} = \mathbf{B}^{-1}$. However, if the basis functions are orthogonal with respect to the weight function w , the symmetric matrix $\mathbf{B}^T \mathbf{W} \mathbf{B}$ is diagonal. This simplification allows the expense of inverting a system of equations in order to produce the forward DPT to be avoided. For example, if the weight function $w(x) = 1$, as it is for the orthogonal Legendre basis,

$$[\mathbf{B}^T \mathbf{W} \mathbf{B}]^{-1} = \text{diag}(c_0, \dots, c_N), \quad (\text{A.10})$$

where

$$c_k = \begin{cases} k + 1/2 & \text{for } k < N, \\ N/2 & \text{for } k = N. \end{cases} \quad (\text{A.11})$$

In the case of the basis functions being the GLL Lagrange interpolants, the nodal and spectral coefficients are identical, as the operator \mathbf{B} is the identity matrix.

It should be noted that in practical applications of spectral element methods, N is typically 15 or less, so that the cost of computing the DPT coefficients is insignificant, particularly as this can be performed in a pre-processing stage, and the cost amortised over the subsequent time-integration period. Rather, it is the cost of *applying* the DPT at each step that is significant.

Appendix B. Dynamic Smagorinsky model

The conceptual basis of LES as proposed by Leonard [25] is a convolution of the exact turbulent velocity field \mathbf{u} with a filter kernel K that gives the resolved scale velocity field $\bar{\mathbf{u}}$,

$$\bar{\mathbf{u}}(\mathbf{x}) = \int K(\bar{\Delta}, |\mathbf{x} - \mathbf{x}'|) \mathbf{u}(\mathbf{x}') d^3 \mathbf{x}'. \quad (\text{B.1})$$

where $\bar{\Delta}$ is the filter length, which is usually be taken to be the grid size.

An evolution equation for $\bar{\mathbf{u}}$ is obtained by convolving the incompressible Navier–Stokes equations with the spatial filter kernel on the grid scale,

$$\partial_t \bar{\mathbf{u}} + \nabla \cdot \overline{\mathbf{u} \mathbf{u}} = -\nabla \bar{P} + \nu \nabla^2 \bar{\mathbf{u}}, \quad (\text{B.2})$$

where $\bar{P} = \bar{p}/\rho$. This first filtering operation is taken as implicit in the formulation (i.e., not explicitly carried out). Here, by assumption, filtering and differentiation commute, i.e., $\overline{\partial \mathbf{u} / \partial x} = \partial \bar{\mathbf{u}} / \partial x$, although in general, they do not, and an additional commutation error arises, in addition to other modelling errors [12]. As in conventional turbulence modelling, the nonlinear terms are have to be modelled, because the filtered dyad $\overline{\mathbf{u} \mathbf{u}}$ cannot be expressed in terms of the known resolved components $\bar{\mathbf{u}}$, and an extra stress τ is introduced to close the equations, such that $\tau = \overline{\mathbf{u} \mathbf{u}} - \bar{\mathbf{u} \mathbf{u}}$. The momentum equations then become

$$\partial_t \bar{\mathbf{u}} + \nabla \cdot \bar{\mathbf{u} \mathbf{u}} = -\nabla \bar{P} + \nu \nabla^2 \bar{\mathbf{u}} - \nabla \cdot \tau. \quad (\text{B.3})$$

The turbulence modelling task is to estimate the subgrid-scale stress τ from the resolved velocity field $\bar{\mathbf{u}}$.

The dynamic method [9] seeks to exploit the expectation that in the inertial range of the turbulence energy spectrum, the turbulence physics are statistically self-similar when viewed at different length scales, and specifically at the grid length scale $\bar{\Delta}$, representative of the computational mesh, and at a larger test-filter length scale $\tilde{\Delta}$. If the same turbulence model can be applied to the portions of the energy spectrum that reside at sizes larger than these two length scales, then the model coefficients should be the same in each case. The DP provides the framework for working back from the grid- and test-filtered velocity fields to obtain the model coefficient(s), which are then applied in estimation of the SGS stress τ .

Applying a second filter with an associated size of $\tilde{\Delta}$ to the filtered Navier–Stokes equation (B.3) leads to a similar stress tensor on the test-filter level \mathbf{T}

$$\partial_t \tilde{\mathbf{u}} + \nabla \cdot \tilde{\mathbf{u}}\tilde{\mathbf{u}} = -\nabla \tilde{P} + \nu \nabla^2 \tilde{\mathbf{u}} - \nabla \cdot \mathbf{T}, \tag{B.4}$$

with

$$\mathbf{T} = \widetilde{\mathbf{u}\mathbf{u}} - \tilde{\mathbf{u}}\tilde{\mathbf{u}}. \tag{B.5}$$

Assuming similar physics underly both stresses τ and \mathbf{T} , they can be estimated with exactly the same model. Filtering of τ gives $\tilde{\tau} = \widetilde{\mathbf{u}\mathbf{u}} - \tilde{\mathbf{u}}\tilde{\mathbf{u}}$ in which the first term on the right matches the one in (B.5) and can therefore be eliminated, leading to the relation known as ‘Germano’s identity’

$$\mathcal{L} = \mathbf{T} - \tilde{\tau} = \widetilde{\mathbf{u}\mathbf{u}} - \tilde{\mathbf{u}}\tilde{\mathbf{u}}, \tag{B.6}$$

which can be used with any stand-alone SGS model – all terms, except any model parameters, can be evaluated.

For simplicity and robustness, we employ the Smagorinsky (mixing length) eddy viscosity model for the deviatoric components of the SGS stress τ , so that

$$\tau - \frac{1}{3}\text{tr}(\tau)\mathbf{1} = -2\nu_t \bar{\mathbf{S}} = -2(c_s \bar{\Delta})^2 |\bar{\mathbf{S}}| \bar{\mathbf{S}}, \tag{B.7}$$

where $\bar{\mathbf{S}}$ is the traceless, symmetric, resolved-scale rate-of-strain tensor

$$\bar{\mathbf{S}} = \frac{1}{2} [\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T], \tag{B.8}$$

c_s is the Smagorinsky constant and $|\bar{\mathbf{S}}| = (2\bar{\mathbf{S}} : \bar{\mathbf{S}})^{1/2}$. The isotropic component of SGS stress is notionally combined with the filtered pressure, to obtain a modified pressure $\Pi = \bar{P} + \frac{1}{3}\text{tr}(\tau)$, which is then used in place of \bar{P} in the evolution equation (B.3).

Using the same model for \mathbf{T} produces

$$\mathbf{T} - \frac{1}{3}\text{tr}(\mathbf{T})\mathbf{1} = -2(c_s \tilde{\Delta})^2 |\tilde{\mathbf{S}}| \tilde{\mathbf{S}} \tag{B.9}$$

and introducing both modelled quantities, (B.7) and (B.9), into the deviatoric components of (B.6) gives

$$\mathcal{L} - \frac{1}{3}\text{tr}(\mathcal{L})\mathbf{1} = -2(c_s \bar{\Delta})^2 \mathcal{M} \tag{B.10}$$

with

$$\mathcal{M} = \left(\frac{\tilde{\Delta}}{\bar{\Delta}}\right)^2 |\tilde{\mathbf{S}}| \tilde{\mathbf{S}} - |\bar{\mathbf{S}}| \bar{\mathbf{S}}, \tag{B.11}$$

where typically $\tilde{\Delta}/\bar{\Delta} = 2$ is assumed. In order to obtain a scalar dynamic estimate, the tensor equation (B.10) is reduced by double contraction [27], in which process the isotropic component of \mathcal{L} is eliminated, since \mathcal{M} is deviatoric

$$\mathcal{L} : \mathcal{M} = -2(c_S \bar{\Delta})^2 \mathcal{M} : \mathcal{M}, \quad (\text{B.12})$$

from which the dynamic estimate can be extracted

$$(c_S(\mathbf{x}, t) \bar{\Delta})^2 = -\frac{1}{2} \frac{\mathcal{L} : \mathcal{M}}{\mathcal{M} : \mathcal{M}}. \quad (\text{B.13})$$

This procedure evaluates a local and time-dependent value of $c_S \bar{\Delta}$ and is updated every timestep. Note that the product $(c_S \bar{\Delta})$ can be treated as a mixing length l_S , without explicitly specifying the length scale on the grid level – this is advantageous in the current context as it enables us to bypass the need to define $\bar{\Delta}$.

The dynamic estimate (B.13) can generate locally negative values of the eddy viscosity ν_t , which, if persistent, can destabilise the time-integration procedure. Hence it is common to employ some kind of ad hoc averaging or limiting of the dynamic estimate to ensure stability. These procedures include: averaging in homogeneous directions [9,21], temporal smoothing [5], integral constraint [11], Lagrangian averaging [29] and clipping [42].

References

- [1] C.H. Amon, A.T. Patera, Numerical calculation of stable three-dimensional tertiary states in grooved-channel flow, *Phys. Fluids A* 1 (2) (1989) 2005–2009.
- [2] G.A. Blaisdell, E.T. Spyropoulos, J.H. Qin, The effect of formulation of nonlinear terms on aliasing errors in spectral methods, *Appl. Numer. Math.* 21 (1996) 207–219.
- [3] J.P. Boyd, Two comments on filtering (artificial viscosity) for Chebyshev and Legendre spectral and spectral element methods: preserving boundary conditions and interpretation of the filter as diffusion, *J. Comput. Phys.* 143 (1998) 283–288.
- [4] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, second ed., Dover, New York, 2001.
- [5] M. Breuer, Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modeling aspects, *Int. J. Numer. Meth. Fluids* 28 (1998) 1281–1302.
- [6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods in Fluid Dynamics*, Springer, Berlin, 1988, 2nd printing.
- [7] J.H. Ferziger, Large eddy simulation, in: Gatski, Hussaini, Lumley (Eds.), *Simulation and Modeling of Turbulent Flows*, Oxford University Press, New York, 1996, pp. 109–154 (Chapter 3).
- [8] P.F. Fischer, J.S. Mullen, Filter-based stabilization of spectral element methods. *Comptes Rendus à l'Académie des Sciences Paris, Ser. 1, Anal. Numer.* 332 (2001) 265–270.
- [9] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A* 3 (7) (1991) 1760–1765.
- [10] S. Ghosal, An analysis of numerical errors in large-eddy simulations of turbulence, *J. Comput. Phys.* 125 (1996) 187–206.
- [11] S. Ghosal, T.S. Lund, P. Moin, K. Akselvoll, A dynamic localization model for large-eddy simulation of turbulent flows, *J. Fluid Mech.* 286 (1995) 229–255.
- [12] S. Ghosal, P. Moin, The basic equations for the large eddy simulation of turbulent flows in complex geometry, *J. Comput. Phys.* 118 (1995) 24–37.
- [13] R.D. Henderson, Adaptive spectral element methods for turbulence and transition, in: T.J. Barth, H. Deconinck (Eds.), *High-order Methods for Computational Physics*, Springer, Berlin, 1999, pp. 225–324 (Chapter 3).
- [14] A.K.M.F. Hussain, W.C. Reynolds, Measurements in fully-developed turbulent channel flow, *ASME J. Fluid Engng.* 97 (1975) 568–578.
- [15] J. Jiménez, R.D. Moser, Large-eddy simulations: where are we and what can we expect?, *AIAA J.* 38 (4) (2000) 605–612.
- [16] G.E. Karniadakis, Spectral element–Fourier methods for incompressible turbulent flows, *Comput. Methods Appl. Mech. Engng.* 80 (1990) 367–380.
- [17] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 97 (2) (1991) 414–443.
- [18] G.E. Karniadakis, S.A. Orszag, V. Yakhot, Renormalization group theory simulation of transitional and turbulent flow over a backward-facing step, in: B. Galperin, S.A. Orszag (Eds.), *Large Eddy Simulation of Complex Engineering and Geophysical Flows*, Cambridge University Press, Cambridge, 1993, pp. 159–177 (Chapter 8).
- [19] G.E. Karniadakis, S.J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, Oxford, 1999.
- [20] K.Z. Korczak, A.T. Patera, An isoparametric spectral element method for solution of the Navier–Stokes equations in complex geometry, *J. Comput. Phys.* 62 (1986) 361–382.

- [21] A.G. Kravchenko, P. Moin, On the effect of numerical errors in large eddy simulations of turbulent flows, *J. Comput. Phys.* 131 (1997) 310–322.
- [22] A.G. Kravchenko, P. Moin, R. Moser, Zonal embedded grids for numerical simulations of wall-bounded turbulent flows, *J. Comput. Phys.* 127 (1996) 412–423.
- [23] A.G. Kravchenko, P. Moin, K. Shariff, B-spline method and zonal grids for simulations of complex turbulent flows, *J. Comput. Phys.* 151 (1999) 757–789.
- [24] S.K. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.* 103 (1992) 16–42.
- [25] A. Leonard, Energy cascade in large-eddy simulations of turbulent fluid flows, *Adv. Geophys.* 18A (1974) 237–248.
- [26] J.G. Levin, M. Iskandarani, D.B. Haidvogel, A spectral filtering procedure for eddy-resolving simulations with a spectral element ocean model, *J. Comput. Phys.* 137 (1997) 130–154.
- [27] D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method, *Phys. Fluids A* 4 (3) (1992) 633–634.
- [28] Y. Maday, A.T. Patera, Spectral element methods for the incompressible Navier–Stokes equations, in: A.K. Noor, J.T. Oden (Eds.), *State-of-the-Art Surveys on Computational Mechanics*, ASME, New York, 1989, pp. 71–143 (Chapter 3).
- [29] C. Meneveau, T.S. Lund, W.H. Cabot, A Lagrangian dynamic subgrid-scale model of turbulence, *J. Fluid Mech.* 319 (1996) 353–385.
- [30] R.D. Moser, J. Kim, N.N. Mansour, Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$, *Phys. Fluids*. 11 (4) (1999) 943–945.
- [31] J.S. Mullen, P.F. Fischer, Filtering techniques for complex geometry fluid flows, *Comm. Numer. Meth. Eng.* 15 (1999) 9–18.
- [32] S.A. Orszag, Spectral methods for problems in complex geometries, *J. Comput. Phys.* 33 (1980) 70–92.
- [33] A. Pascarelli, U. Piomelli, G.V. Candler, Multi-block large-eddy simulations of turbulent boundary layers, *J. Comput. Phys.* 157 (2000) 256–279.
- [34] A.T. Patera, A spectral element method for fluid dynamics: laminar flow in a channel expansion, *J. Comput. Phys.* 54 (1984) 468–488.
- [35] U. Piomelli, Large-eddy simulations: where we stand, in: C. Liu, Z. Liu (Eds.), *Advances in DNS/LES*, Louisiana, August, AFOSR, 1997, pp. 93–104.
- [36] U. Piomelli, J.H. Ferziger, P. Moin, Models for large eddy simulations of turbulent channel flows including transpiration, Technical Report TF-32, Department of Mechanical Engineering, Stanford University, 1987.
- [37] S.B. Pope, *Turbulent Flows*, Cambridge University Press, Cambridge, 2000.
- [38] P. Sagaut, *Large Eddy Simulation for Incompressible Flows*, Springer, Berlin, 2001.
- [39] B.A. Szabó, I. Babuška, *Finite Element Analysis*, Wiley, New York, 1991.
- [40] T. Wei, W.W. Willmarth, Reynolds-number effects on the structure of a turbulent channel flow, *J. Fluid Mech.* 204 (57–95) (1989).
- [41] T.A. Zang, On the rotation and skew-symmetric forms for incompressible flow simulations, *Appl. Numer. Math.* 7 (1991) 27–40.
- [42] Y. Zang, R.L. Street, J.R. Koseff, Large eddy simulation of turbulent cavity flow using a dynamic subgrid-scale model, *Phys. Fluids A* 5 (12) (1993) 3186–3196.