Journal of Computational Physics 549 (2026) 114611

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Sub-filter-scale (SFS) compressible turbulence modeling and shock capturing via the Block-Spectral-Stress (BSS) closure

Matteo Ruggeri [a,*], Victor C. B. Sousa [b], Carlo Scalo [b,c]

[a] School of Aeronautics and Astronautics, Purdue University, 47906, West Lafayette, IN, USA
[b] School of Mechanical Engineering, Purdue University, 47906, West Lafayette, IN, USA
[c] Chief Executive Officer, HySonic Technologies, LLC, 47906-4182, West Lafayette IN, USA

ARTICLE INFO

Keywords:
Turbulence modeling
Shock-capturing
Flux reconstruction
Block spectral method
Wall bounded flow

ABSTRACT

A new Block-Spectral-Stress (BSS) sub-filter-scale (SFS) closure combining turbulence modeling and shock-capturing has been developed for flux-reconstruction numerics improving upon an earlier formulation named Legendre Spectral Viscosity or LSV (Sousa and Scalo, *J. Comput. Phys.* 2022, Vol 460). The BSS method relies on the Legendre spectral representation of the velocity gradients to estimate the SFS kinetic energy, used to compute the SFS momentum stresses, heat-flux, and pressure-work, based on the resolved field. The method is able to capture shocks in one-dimension with numerical order up to 20, with good agreement between *a posteriori* and exact SFS stresses. The same approach is able to adequately preserve the hydrodynamic structure of a vortex impinging on a Mach 1.5 shock in two dimensions and overcome symmetry violations and lack of Galilean invariance of the previous LSV method. Also, unlike LSV, BSS removes the need for explicit spectral modulations of the modeled SFS fluxes. It has been tested in three-dimensional turbulent calculations where it is compared against the Smagorinsky, dynamic Smagorinsky, and Vreman models, all adapted to flux reconstruction numerics. In subsonic and supersonic Taylor-Green Vortex calculations, the BSS model loses accuracy on coarse meshes (i.e. fewer mesh cells), while surpassing other closures on finer ones, making the increasing of polynomial order not advantageous. The same behavior was noticed also in a supersonic Taylor-Green Vortex test case; instead, in fully developed supersonic and hypersonic turbulent channel flow the opposite is observed, where *a posteriori* predictions improve with increasing polynomial order. BSS has proven to be versatile in achieving both shock capturing and turbulence modeling with no case-specific tuning of coefficients, while not necessarily being the best performing closure for anyone particular test case.

1. Introduction

The last years have seen the proliferation of spectral methods, which have the potential to be fully integrated into unstructured solver retaining with high-order numerical convergence [1–10]. However, spectral numerics are not robust in the presence of steep gradients [11–13], commonly present in high-speed flows, which entail both strong shocks and highly compressible turbulence cascade effects [14–16]. Both shock and turbulence formation have a similar energy cascade dynamics [17–19]; differently from shocks, turbulence has a finite spectra therefore standard shock-capturing methods usually tend to remove a dynamically significant part of turbulent spectra and hinder the quality of the simulation. On the other hand, Large Eddy Simulations (LES) techniques–which allow to resolve turbulence on a mesh coarser than Direct Numerical Simulation quality (DNS) by modeling of small scales

[*] Corresponding author.
E-mail address: mruggeri@purdue.edu (M. Ruggeri).

https://doi.org/10.1016/j.jcp.2025.114611
Received 7 July 2025; Received in revised form 9 December 2025; Accepted 19 December 2025
Available online 25 December 2025

turbulence–are not able to numerically sustain shocks. This work aims to address these two problems with a unified model that is able to capture shocks while preserving the broadest possible spectrum of the turbulence in the simulation.

Spectral methods can be implemented with three different approaches: the discontinuous Galerkin (DG) [8,20–23], the staggered grid approach (or spectral differences) [6,7,24], and the flux reconstruction (FR) method [2–4]. The latter is a more recent development in the field of spectral numerics and serves as a basis for our work. It relies on correction (or lifting) functions that are used to propagate information from the cell faces onto the interior solution points, effectively yielding a larger numerical stencil than just the inner-cell numerics alone. Ramirez et al. [25] developed a FR approach specifically for aeroacoustic that relies on the Linearized Euler equations and works on high-order finite volume schemes. Instead, Wang et al. [26] developed an algorithm of implicit time integration of the flux reconstruction method.

The trade-off between stability and accuracy of the FR method relies on the choice of the appropriate correction function. Asthana et al. [27] proved that even when the mesh is refined, the instabilities caused by aliasing errors related to the interpolation of the fluxes from the solution points to the faces are not completely reduced. In the same paper, they mathematically proved that traditional artificial viscosity strategies can stabilize the FR method for all numerical orders for sufficiently fine grids. They also showed how this allows to run 1D shock simulation up to numerical order 120, with the rate of convergence not being affected by the adoption of a shock sensor for localization of the artificial viscosity.

Tonnicello et al. [5] addressed the problem of reproducing the non-monotonic entropy profile across a captured shock [19,28–31] with high-order spectral differences. Most shock-capturing methods are unable to capture this phenomenon, returning a monotonic entropy rise across the shock. They compare Laplacian viscosity with a physical artificial viscosity method, showing that the latter is able to have a better prediction of the entropy even when the Laplacian approach is able to better capture the shock. The main drawback of the proposed approach is that it dissipates acoustic waves, which are critical, for example in hypersonic boundary layer transition in canonical geometries [32]. The authors pointed out that this is likely caused by the shock sensor and suggested that a divergence-based sensor may have better performance. However, previous authors [33,34], who indeed used a divergence-based sensor, still witnessed spurious suppression of acoustic waves. Alternative approaches are DG entropy-stable schemes [35–38]. By their nature, these methods can significantly stabilize the simulation around shocks, but they do not fully address the LES modeling aspect of simulations. Unlike our model, which targets the problem from a physical closure perspective, these approaches target more specifically the solver core numerics.

The current work aims to address this challenge in developing a combined SFS turbulence/shock-capturing model. Section 2 describes the code used for this investigation and the numerics used to solve the compressible Navier-Stokes equations. In the second part of the section the Block-Spectral-Stress (BSS) model is introduced together with a description of the Smagorinsky and dynamic Smagorinsky models implemented with a block-spectral logic, and later used for comparison with BSS. Section 5 presents how the model performs in shock-dominated 1D and 2D flows. Finally, the model's capability to capture SFS turbulence effects (i.e. act as a Large-Eddy Simulation closure) is tested in a subsonic Taylor-Green Vortex (TGV) flow and turbulent wall-bounded flow at supersonic and hypersonic flow speed in Section 7.

2. Models description

The code used to perform the simulations in this manuscript is H^{3}AMR [39–42] (HySonic, High-Order, Hybrid Adaptive Mesh Refinement developed by HySonic Technologies, LLC) an unstructured spectral research code [43] for compressible flows based on flux reconstruction numerics [2]. The current section provides a description of the core numerics (Section 2.1) and of various SFS models such as Smagorinsky [44] (Section 2.2.1), Dynamic Smagorinsky [45] (Section 2.2.2), Vreman's model [46] (Section 2.2.3), and Block-Spectral-Stress (BSS) (Section 3), as implemented in the code, hence adapted to FR numerics.

2.1. Code description (H^{3}AMR)

The vector of conserved quantities reads:

$$\mathbf{Q} = \left[\rho,\ \rho u_1,\ \rho u_2,\ \rho u_3,\ E\right]^T \tag{1}$$

where ρ is the density, u_i the velocity in the i^{th} direction in the physical space defined by the Cartesian coordinates (x_1, x_2, x_3) or, equivalently $(x,\ y,\ z)$, and $E = \rho(e + u_i u_i/2)$ is the specific total energy, and $e = p/\rho(\gamma - 1)$ is the specific internal energy for ideal calorically perfect gases. The flux vectors read:

$$\mathbf{F} = \begin{bmatrix} \rho u_1 \\ \rho u_1 u_1 + p - \mu\sigma_{11} \\ \rho u_1 u_2 - \mu\sigma_{12} \\ \rho u_1 u_3 - \mu\sigma_{13} \\ (E+p)u_1 - k\dfrac{\partial T}{\partial x_1} + \\ \cdots - \mu\sigma_{i1}u_i \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \rho u_2 \\ \rho u_2 u_1 - \mu\sigma_{21} \\ \rho u_2 u_2 + p - \mu\sigma_{22} \\ \rho u_2 u_3 - \mu\sigma_{23} \\ (E+p)u_2 - k\dfrac{\partial T}{\partial x_2} + \\ \cdots - \mu\sigma_{i2}u_i \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} \rho u_3 \\ \rho u_3 u_1 - \mu\sigma_{31} \\ \rho u_3 u_2 - \mu\sigma_{32} \\ \rho u_3 u_3 + p - \mu\sigma_{33} \\ (E+p)u_3 - k\dfrac{\partial T}{\partial x_3} + \\ \cdots - \mu\sigma_{i3}u_i \end{bmatrix} \tag{2}$$

where μ is the dynamic viscosity, p the pressure, T the temperature, and $\sigma_{ij} = \partial u_i/\partial x_j + \partial u_j/\partial x_i - \delta_{ij}(2/3)\partial u_k/\partial x_k$ where δ_{ij} is the Kronecker delta. Given these vectors, the Navier-Stokes equations in physical space can be written as:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x_1} + \frac{\partial \mathbf{G}}{\partial x_2} + \frac{\partial \mathbf{H}}{\partial x_3} = 0 \tag{3}$$

The conservation equations can be recast in the computational space (ξ^1, ξ^2, ξ^3), with mapping defined separately for each mesh element, as:

$$\begin{aligned} \mathcal{Q} &= J\mathbf{Q} \\ \mathcal{F} &= J\left(\frac{\partial \xi^1}{\partial x_1}\mathbf{F} + \frac{\partial \xi^1}{\partial x_2}\mathbf{G} + \frac{\partial \xi^1}{\partial x_3}\mathbf{H}\right) \\ \mathcal{G} &= J\left(\frac{\partial \xi^2}{\partial x_1}\mathbf{F} + \frac{\partial \xi^2}{\partial x_2}\mathbf{G} + \frac{\partial \xi^2}{\partial x_3}\mathbf{H}\right) \\ \mathcal{H} &= J\left(\frac{\partial \xi^3}{\partial x_1}\mathbf{F} + \frac{\partial \xi^3}{\partial x_2}\mathbf{G} + \frac{\partial \xi^3}{\partial x_3}\mathbf{H}\right) \end{aligned} \tag{4}$$

where italics are used for quantities transformed in the computational space; and $\partial \xi^i/\partial x_j$ is the Jacobian matrix for the linear transformation from the physical x_j to computational space ξ^i, and J is the determinant of $\partial \xi^i/\partial x_j$.

Therefore the Navier-Stokes equations can be rewritten as:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{1}{J}\frac{\partial \mathcal{F}}{\partial \xi^1} + \frac{1}{J}\frac{\partial \mathcal{G}}{\partial \xi^2} + \frac{1}{J}\frac{\partial \mathcal{H}}{\partial \xi^3} = 0 \tag{5}$$

which are the equations solved by H^{3}AMR. Considering that J is constant over time we can divide all terms by it and keep $\mathbf{Q}$ in physical space.

The computational domain is defined separately for each mesh element or block; in each element, the conserved quantities and the fluxes are stored in $(N+1)$ Gauss-Legendre quadrature points or solution points per computational direction. This yields a polynomial reconstruction of order N, and an overall solver order of $O{=}N+1$ due to the intra-cell flux exchange. Hereafter, the numerical order (O) refers to the overall solver order, which is equal to the number of solution points per cell.

For the sake of conciseness, the following numerical derivations will be explained with one-dimensional formulations; their 3D extension is simply derived via a tensorial concatenation of one-dimensional operators. Hereafter we will use ξ as a generic computational direction, be it ξ^1, ξ^2, or ξ^3. A subscript will be used to refer to a specific nodal location (e.g. ξ_j) along that direction. Given a set of values of conserved quantities $\mathcal{Q}(\xi)$ and fluxes $\mathrm{F}(\xi)$ on the solution points, their values are interpolated to the left $\xi = -1$ and right $\xi = +1$ interface along a given computational direction, which yields a discontinuity resolved by using the Rusanov [47] method:

$$\mathrm{F}^* = \frac{1}{2}(\mathrm{F}_R + \mathrm{F}_L) - \frac{S^+}{2}(\mathrm{Q}_R - \mathrm{Q}_L) \tag{6}$$

where F^* is the new common value of the flux at the interface, F_R and F_L are the extrapolated values of the fluxes to the right and left of the interface, Q_R and Q_L are the respectively extrapolated conserved quantities, and $S^+ = \max(a_L + |u_L|, a_R + |u_R|)$ is the maximum between the sum of the speed of sound a and velocity u at the left and right of the interface. From the updated values, the derivatives can be computed using the flux reconstruction [2] method along a given computational direction, ξ, evaluated at j-th nodal location, ξ_j:

$$\left.\frac{\partial f}{\partial \xi}\right|_{\xi_i} = D_{ij} f(\xi_j) + \left(f^*_{-\frac{1}{2}} - f(-1)\right) g_l'(\xi_i) + \left(f^*_{+\frac{1}{2}} - f(+1)\right) g_r'(\xi_i) \tag{7}$$

where f is a generic function of ξ, either a conserved quantity or a flux, $f^*_{-\frac{1}{2}}$ and $f^*_{+\frac{1}{2}}$ are the updated values at the left and right interface of the element (using the Rusanov method for the inviscid fluxes and arithmetic average for the conserved quantities the viscous fluxes), D_{ij} is the discrete first derivative operator and $g_l'(\xi_i)$ and $g_r'(\xi_i)$ are the derivatives of the correction functions at the solution points. For the 1D and 2D cases tested in this manuscript, we noticed no appreciable difference between communicating the SFS values at the interface or not. This result is due to the fact that these values are computed from the conserved quantities that already benefit from the interface communication. The additional benefit of not communicating SFS values at the interface is reducing the number of operations. The correction functions have the property of being symmetric $g_r(\xi_j) = g_l(-\xi_j)$ and can be using the Radau polynomials:

$$R_{N+1}(\xi_j) = \frac{(-1)^{N+1}}{2}\left[L_{N+1}(\xi_j) - L_N(\xi_j)\right], \tag{8}$$

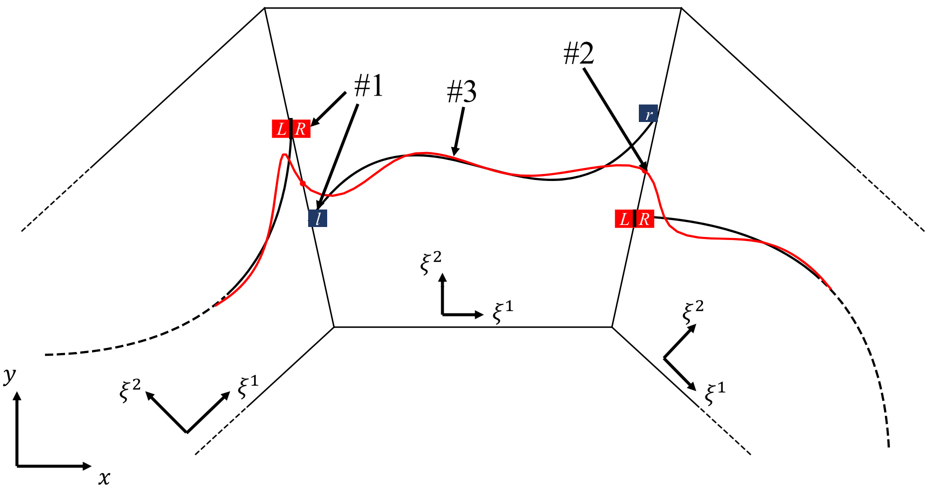


Fig. 1. Qualitative illustration of Flux Reconstruction (FR) with 1D elements which are interpolate to the interface (black lines) and then corrected back to the solution points with their updated values at the interface (red line). Step 1 the internal points are extrapolated to the interfaces. Step 2 the interfaces exchange information with the Rusanov method. Step 3 FR is used to reconstruct the new polynomial. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where $L_{N+1}(\xi_j)$ are Legendre polynomials of order $N+1$ evaluated at the nodal location ξ_j. From the Radau polynomials, the correction functions can be computed as:

$$g_{N+1} = \frac{N+1}{2N+1} R_{R,N+1} + \frac{N}{2N+1} R_{R,N} \tag{9}$$

where g_{N+1} is the function of polynomial order $N+1$ and is applied to the left g_l and from it we can derive g_r.

Fig. 1 shows how this process works, where the black line shows the extrapolation to the interfaces of the element (Step #1). Then, the values at the interface are updated to a new common value shown as red dot at the interface (Step #2). The updated quantity is interpolated back to the solution points, making the reconstruction within the element C^0 continuous among the elements (Step #3).

2.2. Filtered navier-stokes formalism

The Navier-Stokes equations in physical space can be filtered with a spatial filter, indicated with an overbar $\bar{(\cdot)}$, which is assumed to commute with the derivative operation. For compressible flows, it is suggested to use a Favre-based filter:

$$\check{f} = \frac{\overline{\rho f}}{\bar{\rho}}, \tag{10}$$

which leads to the Favre-filtered Navier-Stokes equations, where the unclosed SFS quantities are shown on the right hand side:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho}\check{u}_j}{\partial x_j} = 0 \tag{11}$$

$$\frac{\partial \bar{\rho}\check{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{\rho}\check{u}_i\check{u}_j + \bar{p}\delta_{ij} - \mu\check{\sigma}_{ij}\right) = -\bar{\rho}\tau_{ij}, \tag{12}$$

$$\frac{\partial \bar{E}}{\partial t} + \frac{\partial}{\partial x_j}\left((\bar{E}+\bar{p})\check{u}_j - k\frac{\partial \check{T}}{\partial x_j} - \mu\check{\sigma}_{ij}\check{u}_i\right) = \mu\epsilon - \frac{1}{2}\left(\frac{\gamma\pi_j}{\gamma-1} + \bar{\rho}C_p q_j\right) - \frac{1}{2}\bar{\rho}\zeta_j, \tag{13}$$

$$\frac{\bar{p}}{\gamma-1} - \bar{E} + \frac{1}{2}\bar{\rho}\check{u}_i\check{u}_i = -\frac{1}{2}\bar{\rho}\tau_{ii}. \tag{14}$$

The last is the state equation for an ideal gas, k and C_p are the thermal conductivity and heat capacity at a constant pressure of the gas. The filtering operation yields the unclosed SFS terms: the SFS stress tensor τ_{ij}, the SFS heat flux q_j, the SFS pressure-work π_j, the SFS kinetic energy advection ζ^j, and the SFS heat dissipation ϵ. The last two terms are small and can be considered negligible; for this reason, they are not considered in most of the existing LES models. The Smagorinsky and Dynamic Smagorinsky models assume negligible also the SFS pressure-work π_j, which is instead modeled by the BSS method because it leads to better results when the model is used for shock-capturing. The quantities can be analytically computed as follows:

$$\tau_{ij} = \widecheck{u_i u_j} - \check{u}_i \check{u}_j \tag{15}$$

$$q_j = \widecheck{T u_j} - \check{T} \check{u}_j \tag{16}$$

$$\pi_j = \overline{p u_j} - \overline{p} \check{u}_j \tag{17}$$

$$\zeta_j = \widecheck{u_k u_k u_j} - \check{u}_k \check{u}_k \check{u}_j \tag{18}$$

$$\epsilon = \frac{\partial \overline{\sigma_{ij} u_i}}{\partial x_j} - \frac{\partial \check{\sigma}_{ij} \check{u}_i}{\partial x_j} \tag{19}$$

In the following, three alternative SFS models are formulated in the computational space, consistent with the implementation of BSS in H^{3}AMR. The SFS quantities are then added to the physical flux vectors (vectors in (2)):

$$\mathbf{F}^{\mathrm{SFS}} = \begin{bmatrix} 0 \\ \bar{\rho}\tau_{11} \\ \bar{\rho}\tau_{12} \\ \bar{\rho}\tau_{13} \\ \frac{1}{2}\left(\frac{\gamma\pi_1}{\gamma-1} + \bar{\rho}C_p q_1\right) \end{bmatrix} \mathbf{G}^{\mathrm{SFS}} = \begin{bmatrix} 0 \\ \bar{\rho}\tau_{21} \\ \bar{\rho}\tau_{22} \\ \bar{\rho}\tau_{23} \\ \frac{1}{2}\left(\frac{\gamma\pi_2}{\gamma-1} + \bar{\rho}C_p q_2\right) \end{bmatrix} \mathbf{H}^{\mathrm{SFS}} = \begin{bmatrix} 0 \\ \bar{\rho}\tau_{31} \\ \bar{\rho}\tau_{32} \\ \bar{\rho}\tau_{33} \\ \frac{1}{2}\left(\frac{\gamma\pi_3}{\gamma-1} + \bar{\rho}C_p q_3\right) \end{bmatrix} \tag{20}$$

and then the code continues to work as described in Section 2.1.

2.2.1. Smagorinsky

Smagorinsky [44] proposed a model to close the filtered Navier-Stokes equations based on the resolved strain-rate tensor. The model was developed for incompressible flows and hence closing only τ_{ij}, but can be extended to compressible flows with at least a closure for q_j. The model relies on the eddy viscosity, which is computed as:

$$\nu_t = 2C\Delta^2 |\check{S}| \tag{21}$$

where $C = 0.0256$ is the Smagorinsky constant, Δ is (in our case) the characteristic computational length scale is

$$|\check{S}| = \sqrt{2\check{S}_{ij}\check{S}_{ij}} \tag{22}$$

the norm of the strain-rate tensor, and $\check{S}_{ij}$ is the strain-rate tensor, which is computed using the computational-space derivatives as

$$\check{S}^{ij} = \frac{1}{2}\left(\frac{\partial \check{u}_i}{\partial \xi^j} + \frac{\partial \check{u}_j}{\partial \xi^i}\right). \tag{23}$$

The computational length scale Δ can be estimated as $\Delta = \sqrt[3]{w^i w^j w^k}$ where w^i, w^j, and w^k are, respectively, the Gauss-Legendre quadrature weights in the i-th, j-th, and k-th computational directions. For an orthogonal mesh, this formulation matches the one in physical space, provided the spatial length scale is the physical one.

The SFS stress tensor is computed with an eddy viscosity assumption:

$$\tau^{ij} = -\nu_t(\check{S}_{ij} - \frac{1}{3}\check{S}_{kk}\delta_{ij}) \tag{24}$$

where δ^{ij} is the Kronecker delta. The SFS heat flux is modeled as:

$$q_j = -\frac{\nu_t}{Pr_t}\frac{\partial \check{T}}{\partial \xi^j} \tag{25}$$

where $Pr_t = 0.9$ is the turbulent Prandtl number, and it is assumed to be constant in this formulation.

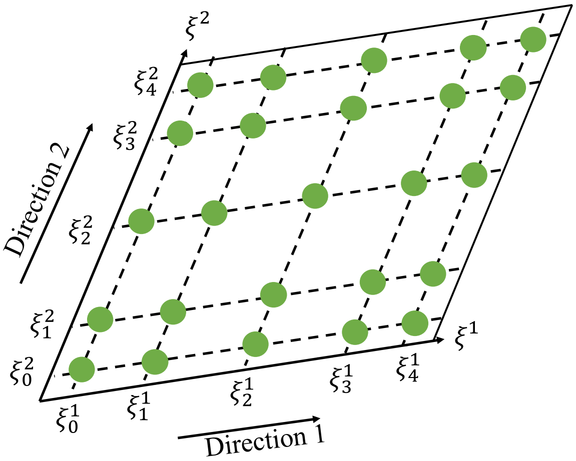


Fig. 2. Qualitative illustration of a two-dimensional element with O internal points located at the zeros of the Gauss-Legendre polynomial order N=$O-1$. In the entire manuscript ξ^i_k denotes the k-th node along the computational direction i.

2.2.2. Dynamic smagorinsky

To overcome the overly dissipative nature of the Smagorinsky model, Germano et al. [48] developed a dynamic model, capable of modulating its intensity based on the local and instantaneous levels of turbulence. The original model was proposed for incompressible flow by Germano et al. [48] and then extended to compressible flow by Moin et al. [45]. The Dynamic Smagorinsky model is based on the standard Smagorinsky model, but instead of assuming C and Pr_t constant, they are computed dynamically according to Eqs. 15 and 19 in [45] derived from algebraic identities, which entail numerous auxiliary test-filtering operations. The model was developed to work with finite-difference or finite-volume codes, but it was never applied to block-spectral numerical codes. We chose to limit the averaging operator within each element, resulting in block-uniform values of C and Pr_t. Because of the small domain covered by the elements, it can happen that the constants are negative, in which case we impose a lower limit of 0 for C and 0.01 for Pr_t.

2.2.3. Vreman'S model

Based on the Smagorinsky model, Vreman [46] developed an eddy viscosity model specifically for wall-bounded turbulent flows. The model estimates the kinematic eddy viscosity ν_t to close the filtered Navier-Stokes equation with a reduction of the dissipation in transitional and/or near-wall regions, making it particularly suitable for wall-bounded flows. The eddy viscosity is computed as:

$$\nu_t = C_{Vr}\sqrt{\frac{\beta_{11}\beta_{22}-\beta_{12}^2+\beta_{11}\beta_{33}-\beta_{13}^2+\beta_{22}\beta_{33}-\beta_{23}^2}{\alpha_{ij}\alpha_{ij}}} \tag{26}$$

where

$$\beta_{ij} = (\Delta^m)^2\alpha_{mi}\alpha_{mj}, \qquad \alpha_{ij} = \partial\check{u}_j/\partial\xi^i \tag{27}$$

where C_{Vr} is a constant that in this case is set to 0.07 and Δ^m is the grid spacing along the m^{th} computational direction.

2.2.4. Legendre spectral viscosity (LSV)

The Legendre Spectral Viscosity (LSV) [49] was specifically developed for flux-reconstruction numerics as an extension of the Quasi-Spectral Viscosity (QSV) method [17,32] developed and tested on compact finite difference numerics. Unlike previously presented models, LSV has a tensorial dissipation matrix $\mathcal{D}_{ij}$ instead of a scalar eddy viscosity ν_t. The dissipation matrix is computed from the estimated tensorial cutoff energy inside each element:

$$E^i_N(\check{u}_j) = \left|\frac{1}{\gamma_N}\sum_{k=0}^{N}\frac{1}{2}\left(\check{u}_j(\xi^i_k)\right)^2 L_N(\xi^i_k)w_k\right| \tag{28}$$

where $\check{u}_j(\xi^i_k)$ is the physical j-th velocity component along the i-th computational direction evaluated at the k-th node along that direction, which leads to non Galileain invariance in the kinetic energy estimation, $L_N(\xi^i_k)$ is the value of the highest Legendre mode at ξ^i_k: here i is the direction along which the value is filtered and k the nodal location (see Fig. 2), w_k is the Gauss Legendre quadrature weight, $\gamma_N = 2/(2N+1)$, and N is the polynomial order used to reconstruct the solution inside the element. From the cutoff energy the model computes the SFS scale $v^i(\check{u}_j) = \sqrt{\frac{N}{2}E^i_N(\check{u}_j)}$, which allows to compute the dissipation matrix as $\mathcal{D}_{ij} = v^i(\check{u}_j)\ell^j$.

The dissipation matrix is used to compute the SFS stress, heat flux, and pressure-work as:

$$\tau_{ij} = -\mathcal{M}\left[\sigma_e, \frac{1}{2}\left(\mathcal{D}_{ij}\frac{\partial \check{u}_i}{\partial \xi^j} + \mathcal{D}_{ji}\frac{\partial \check{u}_j}{\partial \xi^i}\right)\right] \tag{29}$$

$$q_j = -\mathcal{M}\left[\sigma_e, \mathrm{diag}(\mathcal{D})_j \frac{\partial \check{T}}{\partial \xi^j}\right] \tag{30}$$

$$\pi_j = -\mathrm{diag}(\mathcal{D})_j \frac{\partial \bar{p}}{\partial \xi^j} \tag{31}$$

where $\mathrm{diag}(\mathcal{D})$ is a vector comprising the diagonal of $\mathcal{D}_{ij}$. The summation over repeated indices (j and i here) is not implied. The $\mathcal{M}$ is the modulation operation defined as:

$$\mathcal{M}\left[\sigma_e, \mathbf{u}\right] = \mathcal{L}^{-1}[\sigma_e, \mathcal{L}[\mathbf{u}]] \tag{32}$$

where $\mathcal{L}$ is the Legendre transform:

$$\tilde{u}_n = \mathcal{L}[\mathbf{u}] = \frac{1}{\gamma_n}\sum_{k=0}^{N} u(\xi_k) L_n(\xi_k) w_k \tag{33}$$

and $\mathcal{L}^{-1}$ the inverse operation, with spectral modulation $\sigma_e(\eta)$:

$$\mathcal{L}^{-1}[\sigma_e, \tilde{\mathbf{u}}] = \sum_{n=0}^{N} \sigma_e(\eta)\tilde{u}_n L_n(\xi_k) \tag{34}$$

where $\gamma_n = 2/(2n+1)$, $\eta = n/N$, N the polynomial order, and ξ_j the point. The modulation transfer function σ_e used in equations (29)-(30) is defined as:

$$\sigma_e(\eta) = 0.2 + 0.8\frac{e^{4\eta} - 1}{e^4 - 1} \tag{35}$$

and aimed at reducing the dissipation at low wavenumbers. The SFS pressure-work (equation (31)) is not modulated because it is shown to preserve pressure positivity in strong shocks in the current form.

3. Block-Spectral-Stress (BSS)

The Block-Spectral-Stress (BSS) method builds upon the Legendre-Spectral Viscosity (LSV) closure of Sousa and Scalo [49], and specifically redesigned to overcome issues presented in Section 6.

The BSS model estimates the high-wavenumber resolved kinetic energy element-by-element using the gradient of the velocity in each solution point:

$$E^i_N(\check{u}_j) = \left| \frac{1}{\gamma_N}\sum_{k=0}^{N} \frac{1}{2}\left(\ell^j \left.\frac{\partial \check{u}_j}{\partial \xi^i}\right|_{\xi^i_k}\right)^2 L_N(\xi^i_k) w_k \right| \tag{36}$$

based on the square of the computational derivative of $\check{u}_j$ along the i direction computed with the correction polynomials, $L_N(\xi^i_k)$ is the value of the highest Legendre mode at the nodal location ξ^i_k (i is the direction along which the value is filtered), w_k is the Gauss Legendre quadrature weight, ℓ^j is the computational length scale of the solution point, which in the computational space is equal to w_j, $\gamma_N = 2/(2N+1)$, and N is the polynomial order of the function used to reconstruct the solution inside the element.

From the estimated cutoff energy, it is possible to compute the SFS scale $v^i(\check{u}_j) = \sqrt{\frac{N}{2}E^i_N(\check{u}_j)}$, where the factor $N/2$ is the average grid spacing of the element in the computational space. From that it is possible to estimate the dissipation needed by the model with a mixing-length argument,

$$\mathcal{D}_{ij} = v^i(\check{u}_j)\ell^j. \tag{37}$$

The SFS quantities can then be computed as:

$$\tau_{ij} = -C_{ij}\frac{1}{2}\left(\mathcal{D}_{ij}\frac{\partial \check{u}_i}{\partial \xi^j} + \mathcal{D}_{ji}\frac{\partial \check{u}_j}{\partial \xi^i}\right) \tag{38}$$

Table 1
Coefficients of Eqs. (38, 39, 40). Where the first two are the coefficients of a constant matrix respectively at the diagonal and off-diagonal. The last two are single constant coefficients. In the following work we will show that there is no need to tune these coefficients.

C_{ii}	$C_{i\neq j}$	C_q	C_p
1.0	0.2	1.0	1.0

$$q_j = -C_q \mathrm{diag}(\mathcal{D})_j \frac{\partial \check{T}}{\partial \xi^j} \tag{39}$$

$$\pi_j = -C_p \mathrm{diag}(\mathcal{D})_j \frac{\partial \bar{p}}{\partial \xi^j} \tag{40}$$

where $\mathrm{diag}(\mathcal{D})$ is the diagonal of (37), note that the summation over repeated indices is not implied in Eqs. (38)–(40). C_q and C_p are constants, and C_{ij} is a matrix of constants, with values reported in Table 1. Unlike the Smagorinsky model, the diagonal of the strain-rate tensor is not removed, as it provides the normal SFS stresses that support the shock-capturing capabilities of the model. The BSS closure does not modulate the SFS stresses in the spectral space as LSV does.

4. Baseline stability considerations of flux reconstruction numerics

In this section, we discuss important details regarding the stability of the underline FR scheme using the Sod-shock tube test case [50], a standard case widely used to test shock-capturing methods [17,49,51,52] in 1D. The case simulates a tube with a gas with higher pressure and density on the left and lower values on the right separated by a membrane at $x = 0$, which is removed at time zero. A shock and an expansion wave form, propagating, respectively, to the right and left of the membrane. The problem is described by the Euler equations with (normalized) initial conditions:

$$[u_1, p, \rho] = \begin{cases} [0, 1, 1] & \text{if } x \leq 0 \\ [0, 0.1, 0.125] & \text{if } \quad x > 0 \end{cases} \tag{41}$$

where u_1 is the velocity, p the pressure, and ρ the density. The most significant numerical challenge for the high-order scheme is given by the shock and the contact discontinuity.

After implementing the LSV model [49] in H^{3}AMR, we found that the model performed as reported in the original paper for the 2D shock-vortex interaction case. However, when comparing the two base codes without additional models, we found that the code used for the LSV experiments on the Sod shock tube was more stable than H^{3}AMR. The former was implemented in Python following the procedure on the right of Fig. 3 (workflow B). Instead, H^{3}AMR is implemented in C+ + following the steps on the left of Fig. 3 (workflow A). Both workflows were independently reproduced and verified with an independent 1D C+ + toy code.

Fig. 3 highlights in red the main difference where H^{3}AMR computes the fluxes at the interface from the conserved quantities extrapolated at the interface, instead of the Python code used in [49] extrapolations of the flux values from the solution points to the interfaces. Workflow B leads to more stable results in 1D, as shown in Fig. 4. All codes were tested with the same time integration scheme (Runge-Kutta 4 [53]), same time-step $dt = 0.001$ (which corresponds to a CFL lower than 0.1), same mesh (20 elements), and the same numerical order ($O = 4$). In Fig. 4a it is possible to see that workflow A tends to be less stable. Indeed, it is possible to see an undershoot of the pressure at the initial discontinuity. Differently from the other workflow, that remains more stable and close to the shock. Workflow (A) is able to simulate only up to $t = 0.032$ because of the occurance of negative pressure value. In Fig. 4b it is possible to see that workflow (B) is able to continue integrating past $t = 0.25$. This problem could be further delayed by using positivity-preserving limiters at the interfaces. The interfaces are only used for flux exchange and, when the values extrapolated are not physical, such as negative pressure or temperature, they are set to 0 and then inputted to a Rusanov approximate Riemann solver. As long as the outputted flux value remains physical, the simulation remains stable. However, because the difference in stability of workflow (A) and (B) only exists in 1D (and disappears in multi-dimensional cases), we chose to retain the original workflow (A) without additional stabilization methods.

Fig. 4b also shows small differences between the two coding languages for the same workflow, exacerbated by the higher order. Among the main ones, there is (1) the way in which the Gauss-Legendre points are computed, which then affects all the numerical operators such as the derivative operator (D_{ij}) and (2) built-in functions in Python that are not in C+ +. For the first problem, we found that using the default functions in the Lapack and Eigen libraries lead to slightly different values of the Legendre points that may lead to non-zero derivatives even though the values at the solution points are constant, specifically when using Eigen. Another small difference is due to the usage of the numpy library in Python, which leads to slightly different results in the vector multiplication operator.

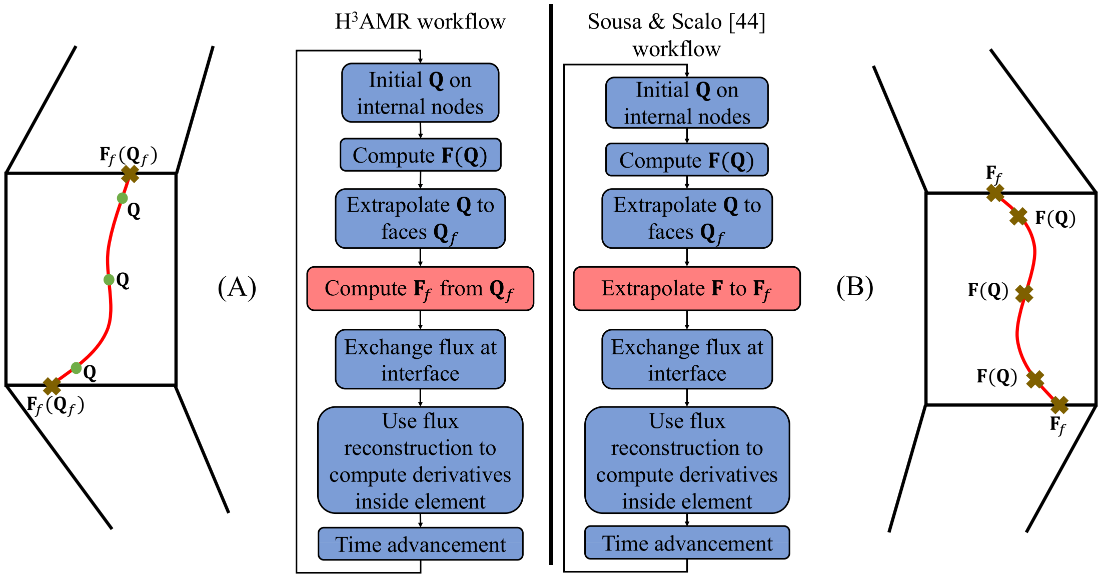

$\mathbf{F}_f(\mathbf{Q}_f)$
$\mathbf{Q}$
$\mathbf{Q}$
$\mathbf{Q}$
$\mathbf{F}_f(\mathbf{Q}_f)$
(A)
H^{3}AMR workflow
Initial $\mathbf{Q}$ on internal nodes
Compute $\mathbf{F}(\mathbf{Q})$
Extrapolate $\mathbf{Q}$ to faces $\mathbf{Q}_f$
Compute $\mathbf{F}_f$ from $\mathbf{Q}_f$
Exchange flux at interface
Use flux reconstruction to compute derivatives inside element
Time advancement
Sousa & Scalo [44] workflow
Initial $\mathbf{Q}$ on internal nodes
Compute $\mathbf{F}(\mathbf{Q})$
Extrapolate $\mathbf{Q}$ to faces $\mathbf{Q}_f$
Extrapolate $\mathbf{F}$ to $\mathbf{F}_f$
Exchange flux at interface
Use flux reconstruction to compute derivatives inside element
Time advancement
(B)
$\mathbf{F}_f$
$\mathbf{F}(\mathbf{Q})$
$\mathbf{F}(\mathbf{Q})$
$\mathbf{F}(\mathbf{Q})$
$\mathbf{F}_f$

Fig. 3. Comparison of two flux reconstruction procedures, one computing the fluxes at the interface from the extrapolated conserved quantities (workflow A), on the left, the other extrapolating flux values from the solution points, on the right (workflow B). We developed a separated 1D C+ + toy code that reproduces the stability characteristics of these two workflows.

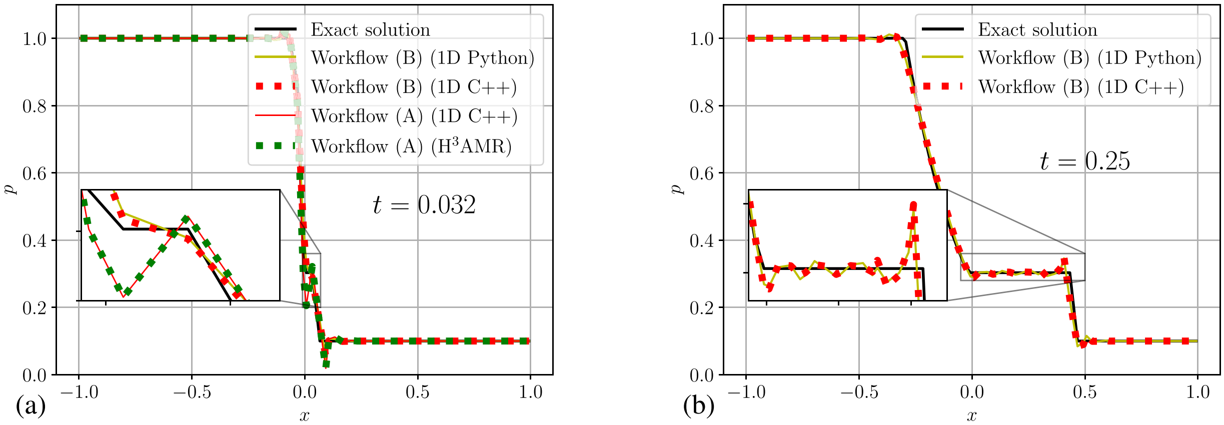


Fig. 4. Comparison of Sod shock tube pressure p evolution with workflow (B) (1D Python code) used by Sousa and Scalo [49], workflow (B) (1D C+ + code), workflow (A) (1D C+ + code) and workflow (A) (H^{3}AMR). All codes run with Runge-Kutta 4 on 20 elements numerical order 4 ($O = 4$) and no shock-capturing. Results are shown for times $t = 0.032$ (a) and $t = 0.25$ (b).

It is important to point out that even though all the aforementioned differences play a role in the numerical stability of 1D test case, they do not play a significant role in the 2D and 3D cases, where both workflows and, interchangeably, either programming languages, exhibit similar stability margins. Workflow (A) is inherently more accurate than (B) because the fluxes are computed directly at the interface and the one extrapolated cannot be as accurate given the non-linear operation used to compute the fluxes. Moreover, workflow (B) cannot be implemented in complex codes in a straightforward way, as it is for a 1D code. Specifically, in H^{3}AMR the fluxes are defined in physical or computational space depending on the operation, and at the interface between elements the physical fluxes need to be decomposed into tangential and perpendicular components updated by the flux solver and then recomposed and recast in a flux in computational space that is used to update the solution point. All these operations are not needed in a 1D code because the **G** and **H** terms in equation (3) do not exist. In addition, more complex codes may require different flux solvers at the interface, (such as Rusanov, HLL, or HLLC) H- p-refinement [54] and therefore having to further adjust the fluxes, or they may also be able to work with different fluids and therefore use generic equations of state. Hereafter, we will adopt workflow (A) and fully unstructured code numerics to generate results.

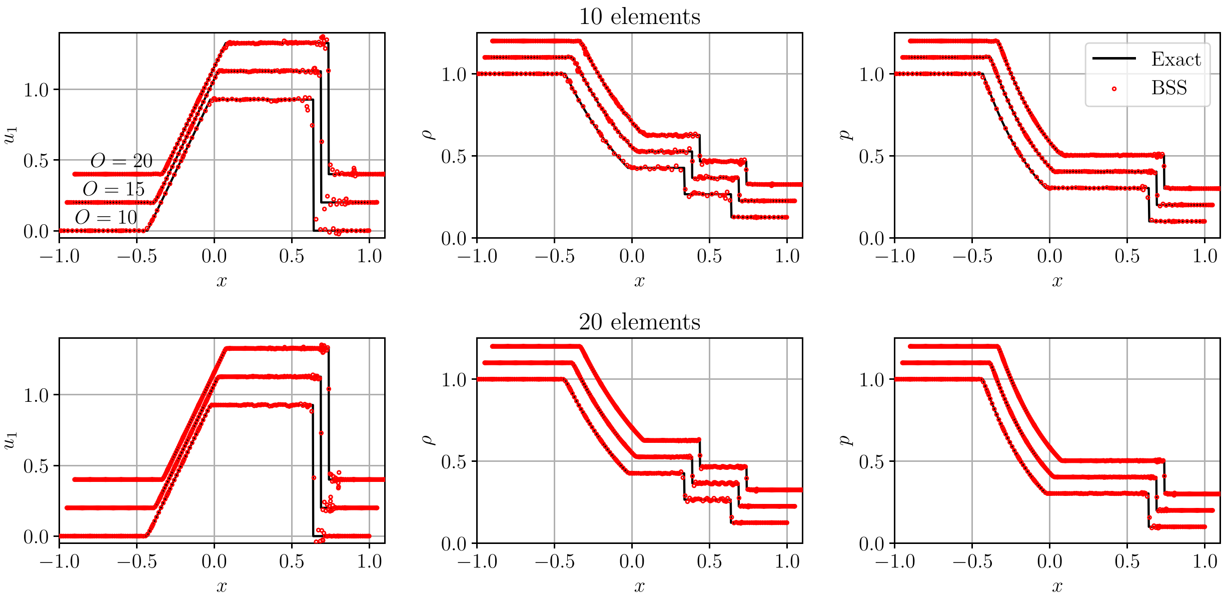


Fig. 5. Sod shock tube velocity u_1, density ρ and pressure p at $t = 0.364$. The first row has a domain of 10 elements and the second of 20 elements. Inside each element the solution is reconstructed with solver accuracy (O) of 10, 15, and 20, which corresponds to 100, 150, and 200 DOFs on the first row and 200, 300, and 400 DOF on the second row.

5. Performance assessment of BSS as shock-capturing scheme

In this section, we are going to assess BSS's capability to act as a shock-capturing closure, in a one-dimensional Sod-shock tube problem (Section 5.1) and a two-dimensional shock vortex interaction case (Section 5.2). Results are obtained via workflow (A) (see Fig. 4) and from the H^{3}AMR code.

5.1. Sod shock tube

Fig. 5 compares the velocity, pressure, and density profiles at $t = 0.364$ of the simulated case with BSS and the analytical solution in different meshes and numerical order O resolution. On the top row, the results are obtained using 10 mesh elements and in each plot the numerical order is O = 10, 15, and 20 from the bottom to the top. Most of the oscillations around the shock are visible in the velocity profile, whereas the pressure surprisingly looks mostly smooth. Increasing the numerical order sharpens the shock and reduces the post-shock oscillations while increasing the pre-shock ones. However, this phenomenon is not dependent on the shock itself, but on its proximity to the inter-element boundary. The bottom row of Fig. 5 presents the results obtained on a mesh with 20 elements. The most visible outcome is the fact that the oscillations are reduced when compared to the same numerical order, on a finer domain, which also allows for a better capturing of the shock. Therefore, it is possible to conclude that increasing the number of degrees of freedom (DOF), i.e. the total number of solution points, reduces the oscillation cased by the shock, but to have a better resolution of the shock, it is better to increase the number of elements and not only the numerical order.

Fig. 6 shows the velocity, pressure, density, SFS stresses, SFS heat flux, and SFS pressure-work for 10 mesh elements and numerical order 20. The exact SFS quantities are computed using equations (38)-(40) on the analytical solution on the same mesh with numerical order 72 filtered down to 20 and the BSS one is obtained *a posteriori*. As expected, considering that the shock has an infinite spectrum, the exact SFS quantities exhibit point-to-point oscillations in the Legendre spectral space η. The modeled SFS quantities follow very closely the exact ones. The last row shows the Legendre transform of the *a posteriori* versus the exact SFS stresses, SFS heat flux, and SFS pressure-work in the element with the shock. For all cases, the spectrum of the *a posteriori* SFS quantities matches the behavior of the exact solution spectra, in sign and magnitude. There are some differences in the spectra of the SFS stresses, mostly due to the discrete numerical nature of the investigation; indeed, if the results were continuous, the exact solution spectra would oscillate further.

5.2. Shock vortex interaction

This section investigates the interaction between a vortex with zero-net circulation and a steady shock. For this investigation, the domain is $[0, 2L] \times [0, L]$, the freestream velocity before the shock is $V_0 = 1.5\sqrt{\gamma p_0/\rho_0}$, the initial shock location is $x_s/L = 1/2$, and the vortex center at $x_{cv}/L = 1/4$ and $y_{cv}/L = 1/2$.

The vortex-induced tangential velocity is given by:

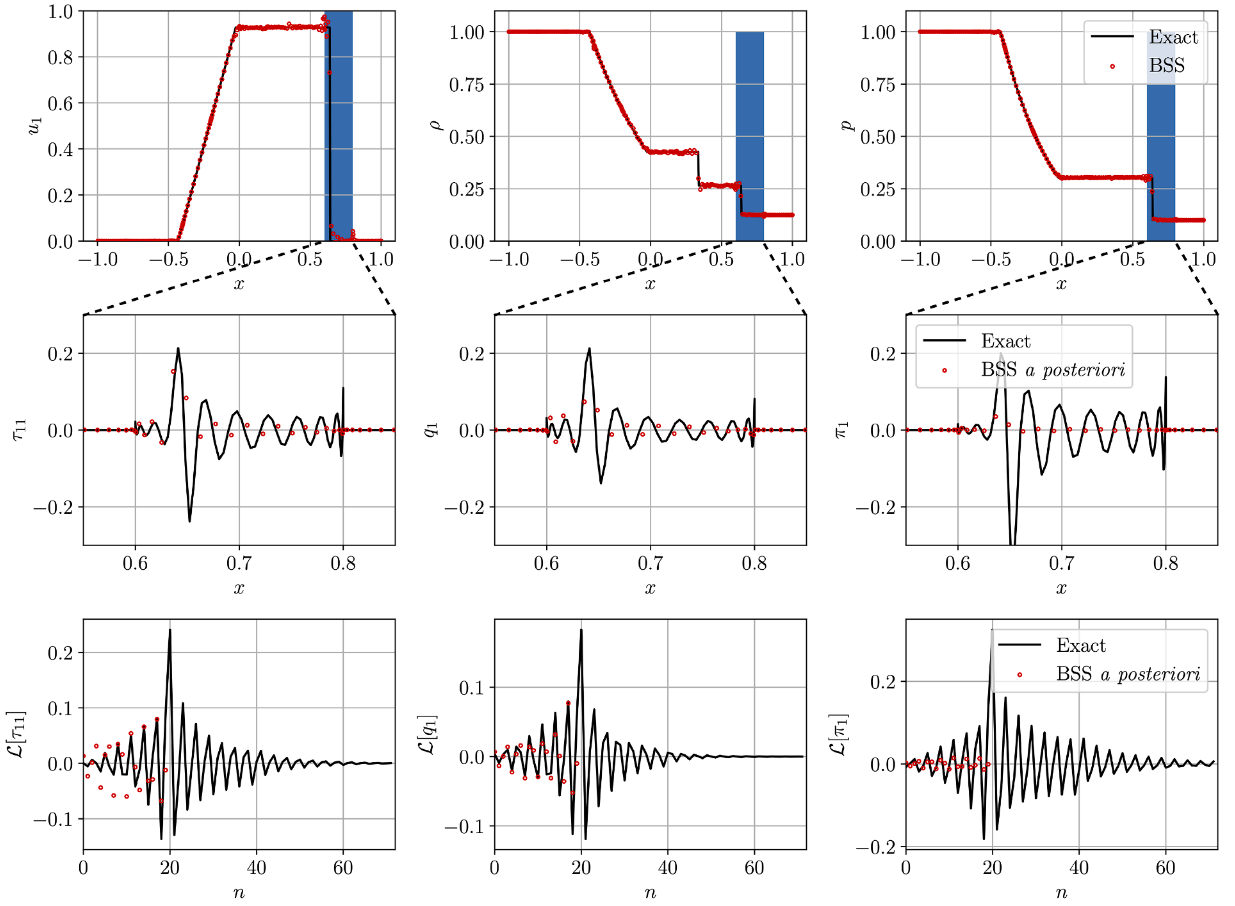


Fig. 6. Top row: Sod-shock tube velocity u_1, density ρ, pressure p, SFS stresses τ_{11}, SFS heat flux q_1, and SFS pressure-work π_1 at $t = 0.364$, for a 10 elements mesh with numerical order $O = 20$. Middle row: SFS stresses in physical space. Bottom row: Legendre transform of the SFS quantities. Solid lines are exact values, while red circles are the *a posteriori* quantities. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\frac{u_\theta(r)}{u_\theta(a)} = \begin{cases} \frac{r}{a}, & \text{if } r \leq a \\ \frac{\eta}{2}\left(\frac{r}{b} - \frac{b}{r}\right), & \text{if } a < r \leq b \\ 0, & \text{otherwise,} \end{cases} \tag{42}$$

which differentiates between an inner circle $r \leq a$ and an outer ring $a < r \leq b$, where $a/L = 0.075$, $b/L = 0.175$ and $\eta = 2(b/a)/[1 - (b/a)^2]$. The total velocity is $\mathbf{u} = u_\theta(r)\hat{\mathbf{e}}_\theta + V_0\hat{\mathbf{e}}_x$ where the maximum tangential velocity is $u_\theta(a) = 0.9V_0$. The pressure field is derived from ideal gas and isentropic relations, and its gradient is taken to balance the centrifugal forces:

$$\frac{\partial P}{\partial r} = \rho\frac{u_\theta^2(r)}{r}, \quad P = \rho RT, \quad \frac{P}{P_0} = \left(\frac{\rho}{\rho_0}\right)^\gamma. \tag{43}$$

Other works [5,55,56] that have performed simulations on this problem reported that the shock-vortex interaction leads to the creation of two smaller vortices with the upper one in a forward position with respect to the lower one. Fig. 7 shows the evolution of the vortex passing through the shock, deforming the shock into an S-shape.

Fig. 8 shows the effects of increasing numerical order (rows-wise) and increasing mesh refinement (columns-wise). The total number of degrees of freedom DOF is kept constant along each row increasing their number of a factor of 4. As previously seen in the Sod-shock tube investigation, when the total number of degrees of freedom is the same, the simulation with the lowest numerical order (i.e. highest mesh-element count) has the best result. This is most evident in the middle row results, where increasing the numerical order to $O = 16$, while keeping the DOFs the same at 131072, reduces the quality of the numerical solution. At 524,288 DOFs, there is little variation in the numerical solution when trading mesh refinement with polynomial refinement.

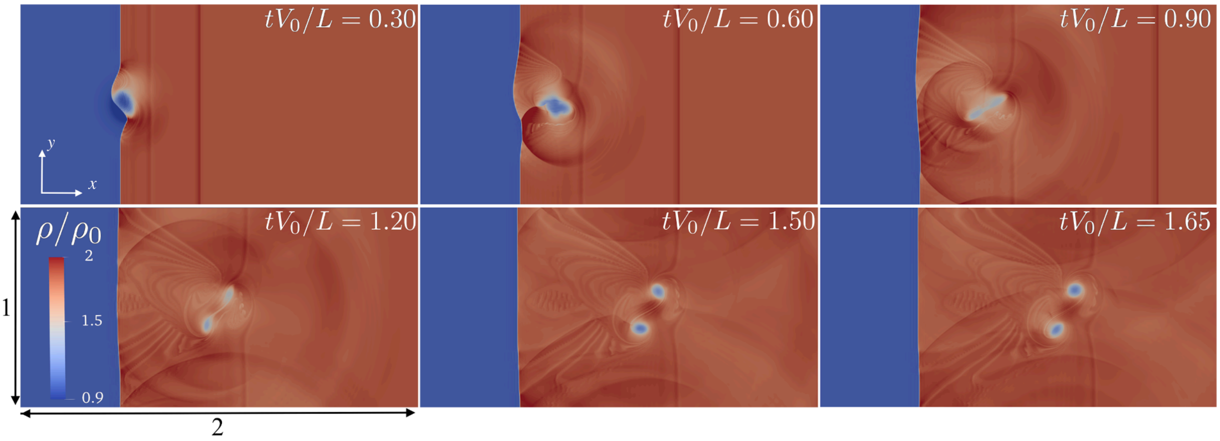


Fig. 7. Shock-vortex interaction evolution using numerical order $O = 4$ and 256x128 elements (524,768 DOF).

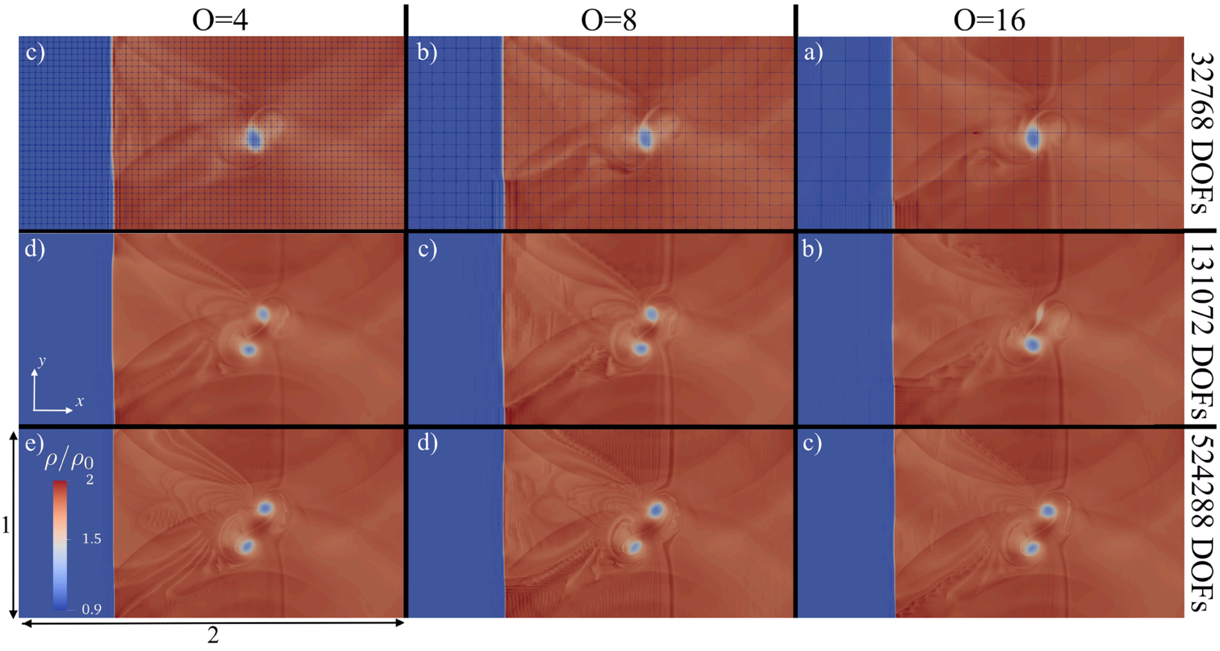


Fig. 8. Shock-vortex interaction at $tV_0/L = 1.65$ with different numerical order and mesh resolution. Top row (for all columns): 32768 DOFs ; Middle row: 131072 DOFs; Bottom row: 524288 DOFs. The meshes have in x and y: a) 16x8, b) 32x16, c) 64x32, d) 128x64, and e) 256x128.

6. BSS Comparison against LSV by sousa and scalo [49]

After implementing the LSV model in the block spectral code H^{3}AMR we realized that there were some differences in the results and found some conceptual issues in LSV. The first is presented in Section 4 and is caused by the background numerical workflow, which significantly helped the model. An additional one is cased by the modulation (Section 6.1) and the estimation of the cutoff kinetic energy (Section 6.2). Table 2 summarizes the differences between LSV and BSS in support of the following discussion.

6.1. Sod shock tube test case comparison

When implementing LSV in an unstructured spectral solver like H^{3}AMR, with the numerics as in the left of Fig. 4 we found that the model significantly under performs on Sod shock tube test case in [49]. This is mainly caused by the modulation operation $\mathcal{M}[\sigma_e, \cdot]$, which tends to reduce the strength of the stresses. Fig. 9 shows the profiles obtained using the LSV and BSS models when solving a Sod shock tube problem. It is visible that LSV have weaker SFS stress and heat flux than BSS. Instead, the pressure-work has a similar strength because is not modulated. Even though the physical reason behind the idea of modulation is consistent with theory because it aims at reducing the SFS stresses imposed on the large scale dynamics, it does not work practically because it is applied on

Table 2
Main differences between the Legendre Spectral Viscosity (LSV) and Block-Spectral-Stress (BSS) models.

	LSV	BSS
$E_N^i(\check{u}_j)$	$\left\lvert\frac{1}{\gamma_N}\sum_{k=0}^{N}\frac{1}{2}\left(\check{u}_j(\xi_k^i)\right)^2 L_N(\xi_k^i)w_k\right\rvert$	$\left\lvert\frac{1}{\gamma_N}\sum_{k=0}^{N}\frac{1}{2}\left(\ell^j\frac{\partial\check{u}_j}{\partial\xi^i}\Big\rvert_{\xi_k^i}\right)^2 L_N(\xi_i^k)w_k\right\rvert$
τ_{ij}	$-\mathcal{M}\left[\sigma_e,\frac{1}{2}\left(\mathcal{D}_{ij}\frac{\partial\check{u}_i}{\partial\xi^j}+\mathcal{D}_{ji}\frac{\partial\check{u}_j}{\partial\xi^i}\right)\right]$	$-C_{ij}\frac{1}{2}\left(\mathcal{D}_{ij}\frac{\partial\check{u}_i}{\partial\xi^j}+\mathcal{D}_{ji}\frac{\partial\check{u}_j}{\partial\xi^i}\right)$
q_j	$-\mathcal{M}\left[\sigma_e,\mathrm{diag}(\mathcal{D})_j\frac{\partial\check{T}}{\partial\xi^j}\right]$	$-C_q\mathrm{diag}(\mathcal{D})_j\frac{\partial\check{T}}{\partial\xi^j}$

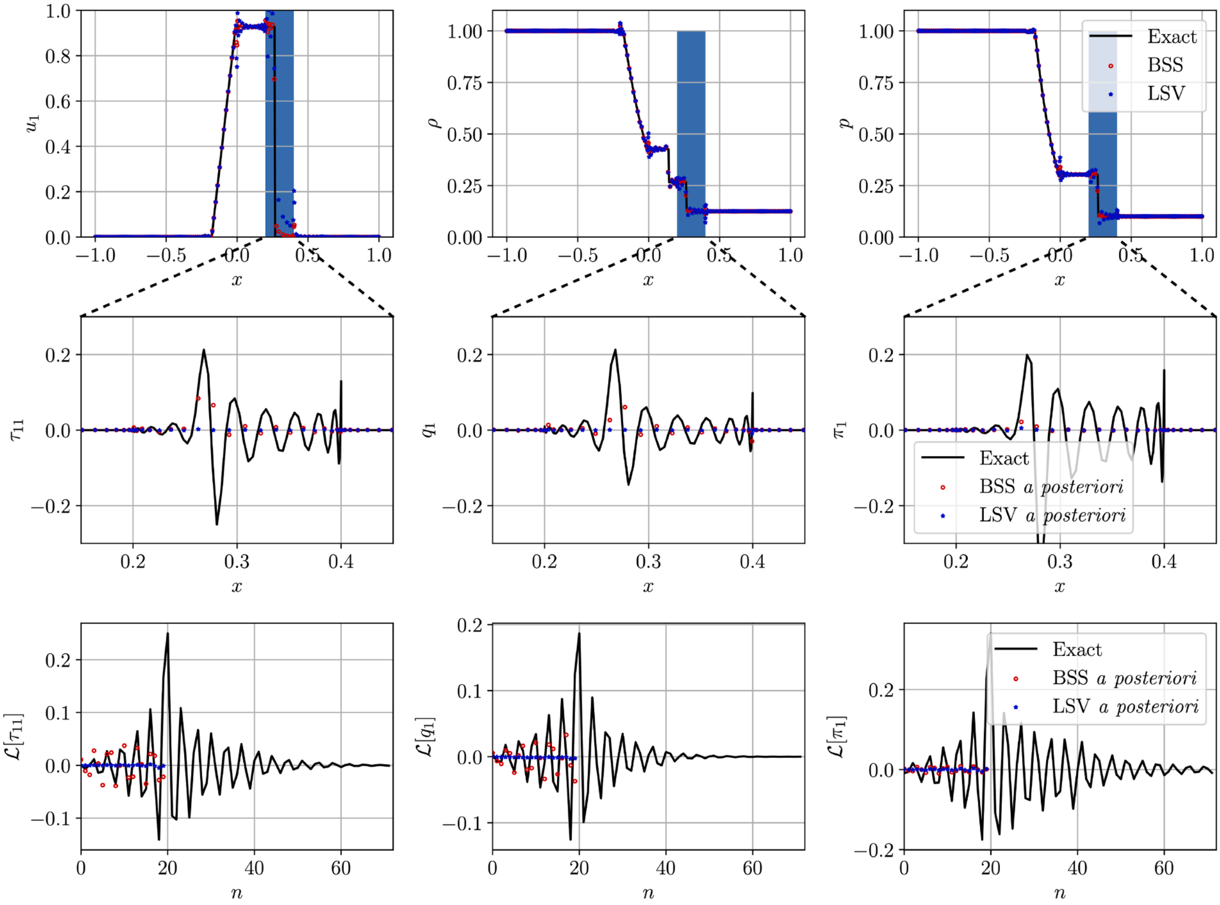


Fig. 9. Comparison of the LSV (blue star) and BSS (red circle) model on the Sod shock tube case using H^{3}AMR numerics (presented on the left of Fig. 4) with order 20 ($O = 20$) and 10 elements. Top row: Sod-shock tube [50] velocity u_1, density ρ, pressure p, SFS stresses τ_{11}, SFS heat flux q_1, and SFS pressure-work π_1. Middle row: SFS stresses computed using the LSV and BSS models. Bottom row: Legendre transform of the three SFS quantities computed in the element with the shock. Solid lines are exact values, while red circles are the *a posteriori* BSS quantities, and the blue stars are the *a posteriori* LSV quantities. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

single mesh elements that do not capture the large scale of dynamics. Hence the modulation is applied on the large scale dynamics inside the element that is small when compared to the entire domain. Therefore, the modulation strategy of LSV is more suitable for domains with larger elements compared to the characteristic length-scales of the flow and hence high numerical order. For complex geometries this may be unfeasible and, moreover, with additional dimensions, high numerical orders numerics lose stability.

6.2. Freely convecting vortex test case comparison

After implementing LSV in H^{3}AMR we were able to replicate the performance reported in [49] but noticed that the estimation of the kinetic energy violates symmetry for a vortex in a uniform flow, resulting in asymmetric SFS stress, heat flux, and pressure work, that would not only be different from the exact one but also deform the vortex in a non-physical manner. Fig. 10 shows the estimated kinetic energy on the left and the SFS stress on the right computed with BSS on top and LSV on the bottom row. The estimated kinetic

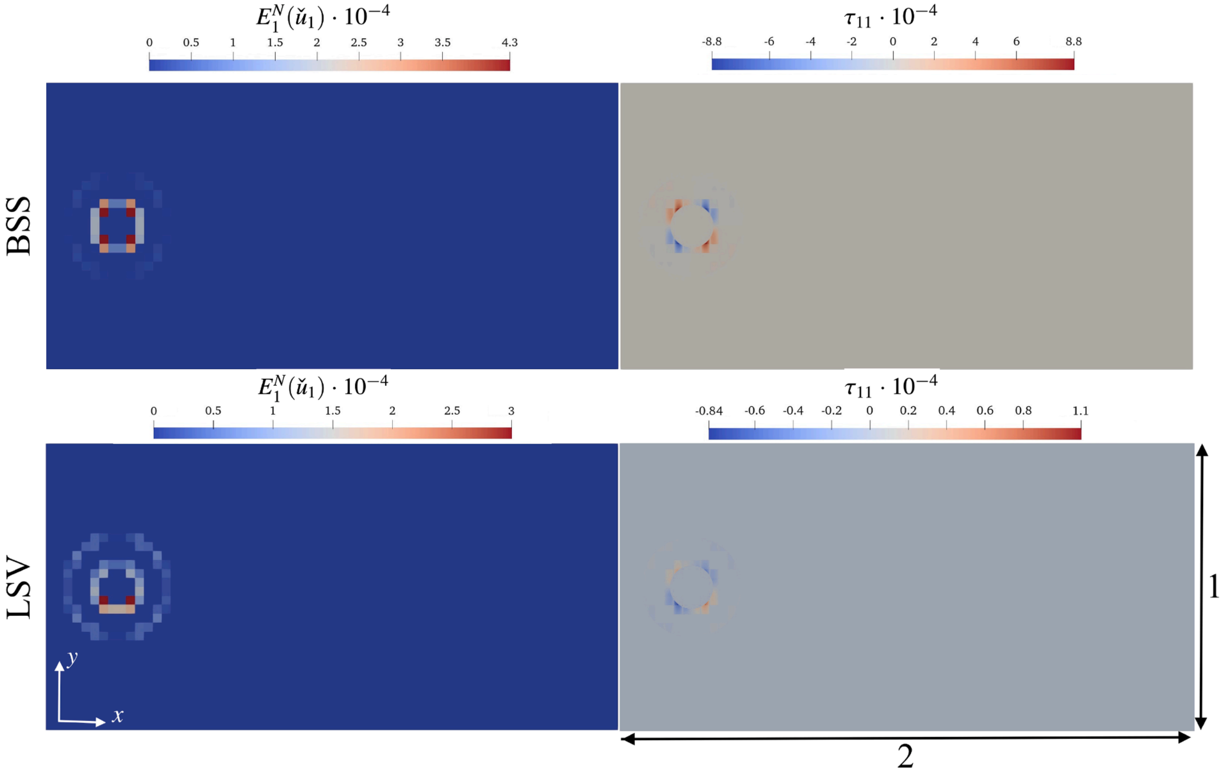


Fig. 10. Comparison of cutoff kinetic energy (on the left) and the SFS stress (on the right) of BSS (top row) and LSV (bottom row) on a vortex. It is visible how the LSV estimated kinetic energy is not symmetric.

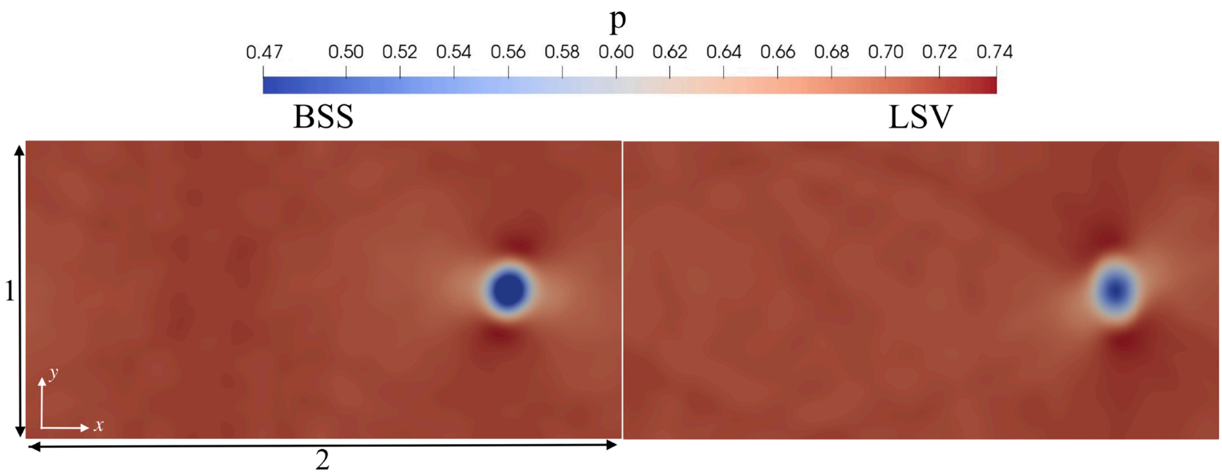


Fig. 11. Pressure distribution of a vortex circulating in a $M = 1.5$ flow with BSS (on the left) and LSV (on the right) active. The snapshot is taken at $tV_0/L = 6$ and is visible how LSV deforms the vortex.

energy of LSV is slightly smaller than the BSS one, but more importantly it is asymmetric, with the lower part of the vortex having a stronger kinetic energy. This is because the vortex is rotating counter-clockwise and the flow underneath is faster because it is added to the uniform flow in the background. Therefore, we changed the estimation of the kinetic energy to be based on the velocity gradient rather than the velocity, which makes the model independent of the background flow and also more aware of the strength of the shock. As noticed for the Sod shock tube case, the SFS stress is more than an order of magnitude smaller for LSV when compared to the one computed by BSS. This is caused by the modulation, given that cutoff kinetic energy is not that different.

As mentioned before, for the shock vortex interaction, it is difficult to notice significant differences between the two models up to $tV_0/L = 1.65$ and the velocity of the bulk flow after the shock is significantly smaller than the one before; therefore the asymmetry

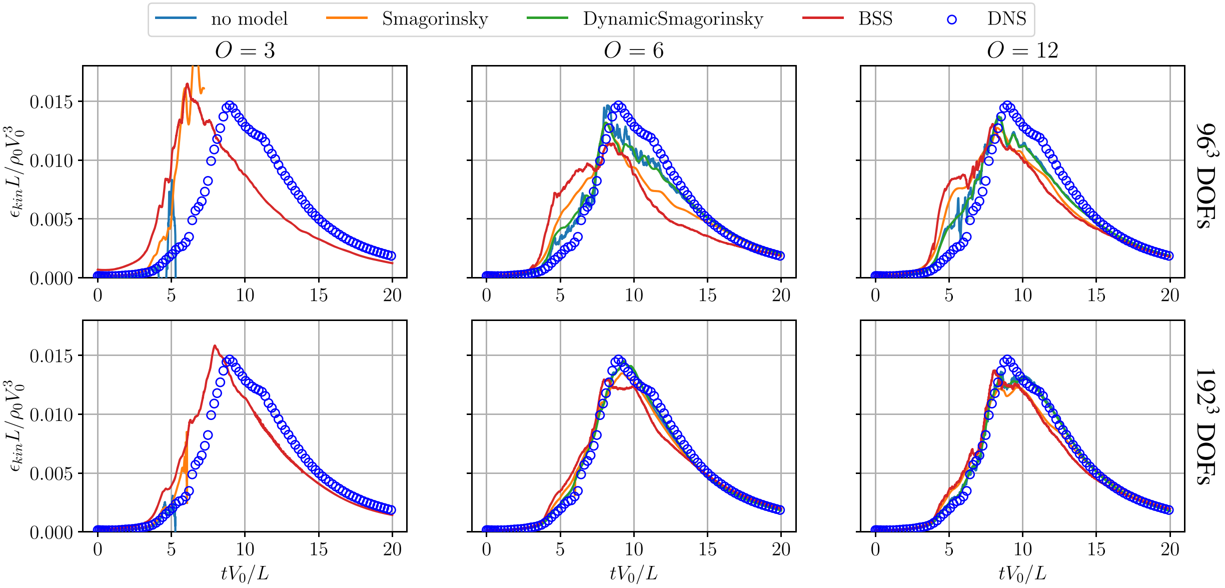


Fig. 12. Normalized dissipation ϵ_{kin} with different degrees of freedom (DOF) and numerical order. From left to right the meshes use for the 96^3 DOF case have respectively 32^3, 16^3, and 8^3 elements. For the 192^3 DOF respectively 64^3, 32^3, and 16^3 elements. Solid lines are the simulations performed with LES models, while blue circles are the DNS results for 288^3 DOFs simulation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

is smaller. In Fig. 11 we show the pressure contour plot of a vortex in a flow at $M = 1.5$ as described in equation (42). The snapshot shows the vortex shape after $tV_0/L = 6$, hence after flowing 4.5 times in the domain. From the pressure contour it is possible to see that the left vortex simulated with LSV does not have a circular shape and it is more diffused than the one simulated with BSS.

7. Performance assessment of BSS as SFS turbulence model

7.1. Taylor-Green Vortex (TGV)

The Taylor-Green Vortex (TGV) is a three-dimensional flow in a triply-period box, exhibiting turbulent breakdown and relaxation towards decaying homogeneous isotropic turbulence. The simulation starts at $t = 0$ s and ends at $t = 200$ s in a $[-\pi L, \pi L]^3$ domain with the following initial conditions:

$$\rho(\mathbf{x},0)/\rho_0 = 1, \tag{44}$$

$$\frac{u_1(\mathbf{x},0)}{V_0} = \sin\left(\frac{x_1}{L}\right)\cos\left(\frac{x_2}{L}\right)\cos\left(\frac{x_3}{L}\right), \tag{45}$$

$$\frac{u_2(\mathbf{x},0)}{V_0} = -\cos\left(\frac{x_1}{L}\right)\sin\left(\frac{x_2}{L}\right)\cos\left(\frac{x_3}{L}\right), \tag{46}$$

$$\frac{u_3(\mathbf{x},0)}{V_0} = 0, \tag{47}$$

$$\frac{p(\mathbf{x},0)}{\rho_0 V_0^2} = \frac{p_0}{\rho_0 V_0^2} + \frac{1}{16}\left[\cos\left(\frac{2x_1}{L}\right) + \cos\left(\frac{2x_2}{L}\right)\right]\left[\cos\left(\frac{2x_3}{L}\right) + 2\right] \tag{48}$$

where $\rho_0 = 1$, $V_0 = 0.1$, $p_0 = 1/\gamma$, and $L = 1$ are the non-dimensionalized density, velocity, pressure, and length scale. The Mach number of the flow is $M_0 = V_0/\sqrt{\gamma p_0/\rho_0} = 0.1$. The Reynolds number is $Re = \rho_0 V_0 L/\mu_0 = 5000$ where the viscosity is considered to be constant at $\mu_0 = 2 \cdot 10^{-5}$. All simulations use a third-order Runge-Kutta time advancement method with $CFL = 0.1$. Grid convergence is achieved at numerical order O = 9 and 32^3 elements mesh.

Similarly to the shock-vortex interaction case, Fig. 12 reports how the dissipation predicted by different methods behaves by varying the degrees of freedom (DOF) and numerical orders. When using O = 3 only BSS is able to yield stable results, whereas the other methods blow up. Unlike in a global spectral method, where the blow-up is caused by the dissipation becoming negative, in

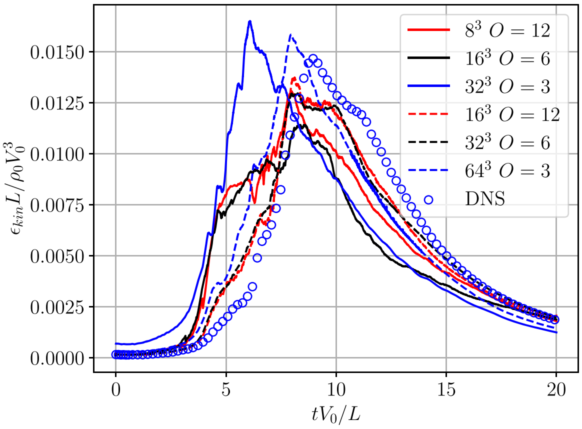


Fig. 13. Kinetic energy dissipation for different combinations of mesh size and numerical orders using the BSS method. Solid lines indicate cases with 96^3 total DOFs and dashed lines 192^3 DOF. Red, black, and blue colors indicate simulation respectively with numerical orders of 12, 6, and 3. Solid lines are the simulations performed with LES models, while blue circles are the DNS results for 288^3 DOFs simulation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Time, number of iterations, and the second per iteration all normalized respect to the no model case for TGV simulation with no model, Smagorinsky, Dynamic Smagorinsky, and Legendre Spectral Viscosity (BSS) methods. The results are obtained on a 16^3 elements mesh, numerical order 6, and up to the same physical time in the simulation.

	Wall Time	# iterations	Time per iteration
No model	1.000	1.000	1.000
Smagorinsky	1.009	0.987	1.024
Dynamic Smagorinsky	0.940	0.801	1.173
BSS	1.067	0.990	1.078

block-spectral numerics the blow-up is caused by oscillations in a single block, which in some cases can affect global statistics as the dissipation. When running with 96^3 DOFs, the BSS is more dissipative than the other methods, especially when turbulent break down starts. In the 192^3 DOFs case, BSS is the only method able to capture the plateau after the dissipation peak. The profile of the dynamic Smagorinsky methods is similar to the profile without any model but without the oscillation present in the latter part of the evolution. This result is caused by the fact that the model is applied in a element-by-element nature; therefore, it has a small effect on the entire simulation domain, but is strong enough to filter down the solution.

Fig. 13 plots the dissipation of the different meshes and numerical orders to highlight how the method performs with different setups. All cases with 96^3 DOFs have very different dissipation, indeed in the 32^3 $\mathrm{O}=3$ the peak dissipation is predicted to occur sooner and with higher intensity. The $\mathrm{O}=12$ case is the one with the best performance, offering more numerical bandwidth to the BSS closure, resulting in a better estimation of turbulent SFS activity. The cases with 192^3 DOFs are closer to the reference DNS resolution and for this reason, the mesh, or the order, have an equally large impact on the solution with better results achieved refining the mesh, as seen in the shock-vortex interaction case. The largest difference is the dissipation peak that is stronger for the $\mathrm{O}=12$ simulation due to a larger Legendre spectral bandwidth available to the model.

Table 3 reports the computational time, the number of iterations, and the time per iteration by all SFS models and a no-model case for a 16^3 mesh size with $\mathrm{O}=6$. As expected, the Dynamic Smagorinsky model is the slowest due to the large number of auxiliary filtered quantities needed to estimate the dynamic coefficients. The standard Smagorinsky is the one with the lowest overhead. BSS has an intermediate overhead, being 6% slower than no model baseline. From the number of iterations it is possible to conclude that all models lead to a lower time step constraint induced by the SFS stresses because all of them require fewer iterations than without a model. This also lead to an higher wall time per iteration also induced by the additional operations.

7.2. Supersonic channel flow

A compressible turbulent channel flow has been used to asses BSS performance in wall bounded flow, specifically in a supersonic Mach 1.5 and hypersonic Mach 6.0 cases. The flow conditions match those used by Chen and Scalo [57], all flow parameters are normalized using the channel's half-width (δ), the speed of sound at the wall (a_w), the wall temperature (T_w), and the bulk density ρ_w. The results are plotted using the Trettel-Larsson transformation [58], where the mean velocity is transformed as:

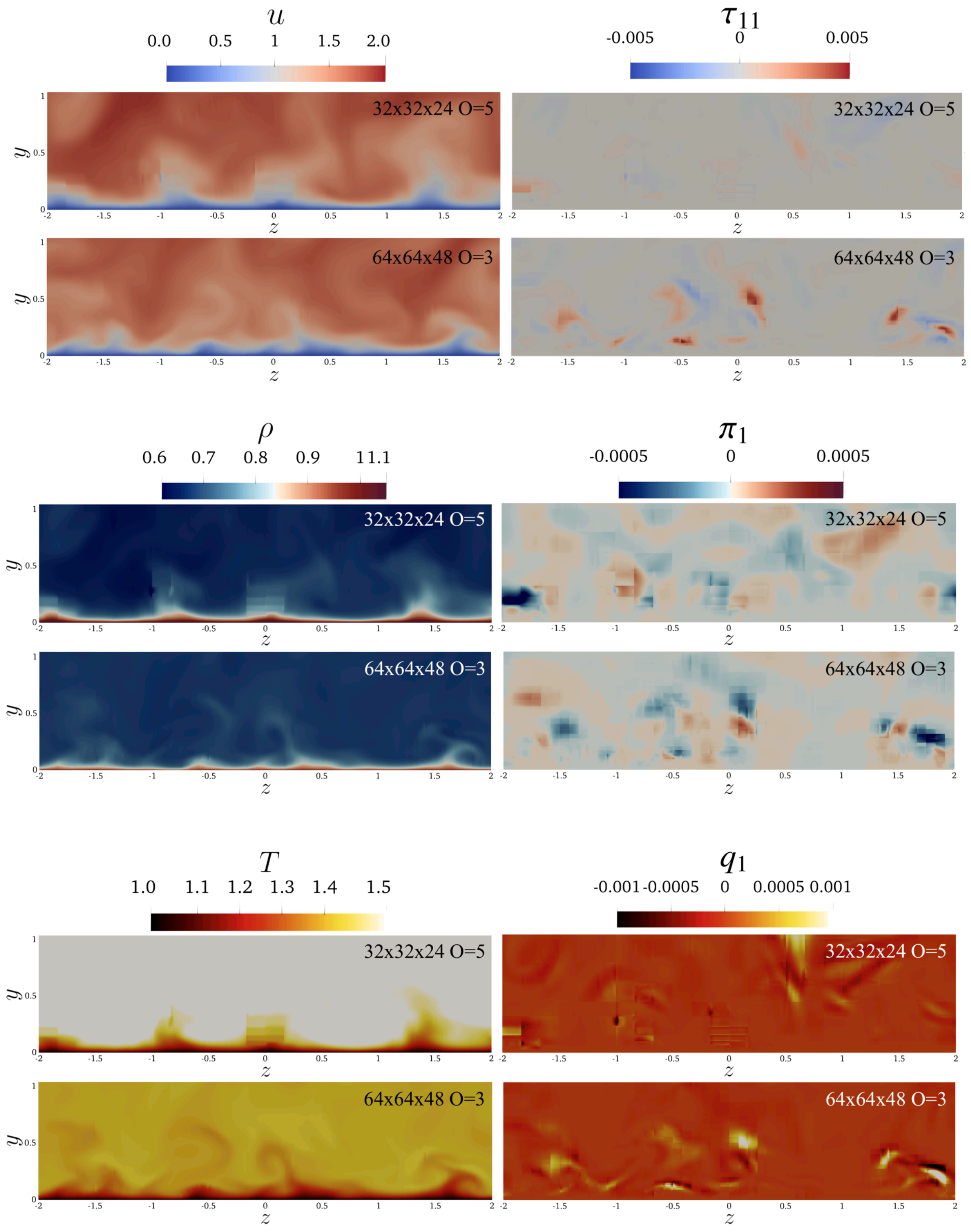


Fig. 14. y-z section of channel flow at Mach 1.5 with no-slip isothermal boundary condition at the wall. On the left column the velocity, density, and temperature contours and on the right one the SFS stresses, pressure-strain, and heat flux. Results shown for various mesh sizes and numerical orders.

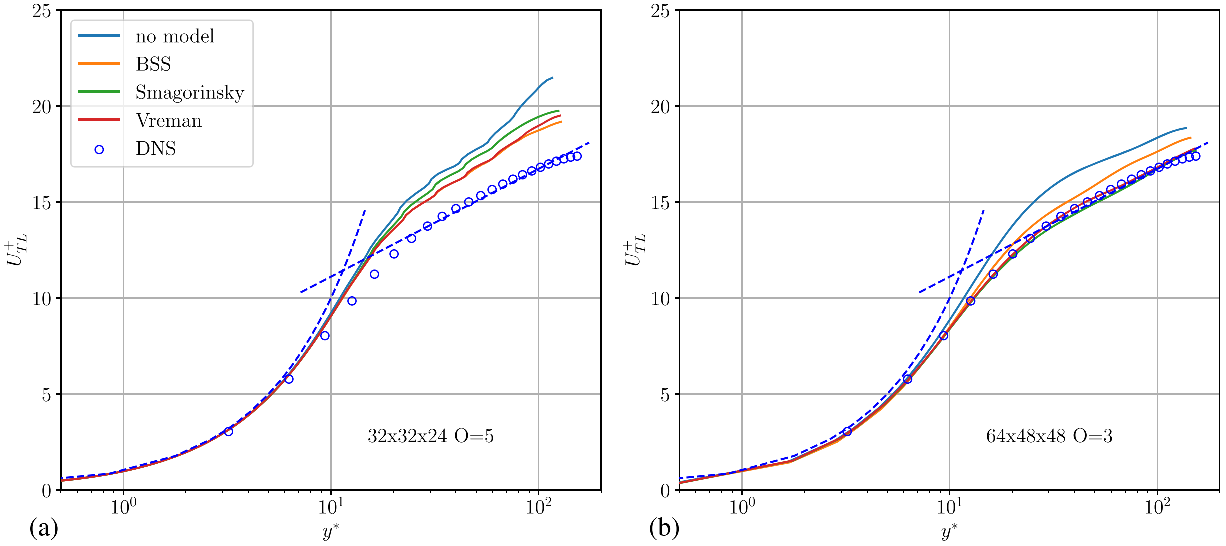


Fig. 15. TL-transformed mean velocity profile for Mach 1.5 compressible channel flow. Results obtained for a 32x32x24 elements mesh and solver accuracy of order $O = 5$ (a) and a 64x48x48 elements mesh with solver accuracy of order $O = 3$ (b). Solid lines are the simulations performed with LES models, while blue circles are the DNS results obtained on a 320x240x240 DOFs mesh O=5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$U^+_{TL} = \int_0^{\check{u}/u_\tau} \sqrt{\frac{\overline{\rho}}{\rho_w}} \left(1 + \frac{1}{2}\frac{1}{\overline{\rho}}\frac{\partial \overline{\rho}}{\partial y} y - \frac{1}{\overline{\mu}}\frac{\partial \overline{\mu}}{\partial y} y\right) d\left(\frac{\check{u}}{u_\tau}\right) \tag{49}$$

where $\check{u}$ is the Favre-filtered streamwise velocity and $u_\tau = \sqrt{\overline{\tau}_w/\overline{\rho}_w}$, plotted versus:

$$y^* = \frac{\overline{\rho}(y) u^*_\tau}{\overline{\mu}(y)} y \tag{50}$$

where $u_\tau = \sqrt{\overline{\tau}_w/\overline{\rho}(y)}$. The reference log-law profile is computed as:

$$U^+_{TL} = \frac{1}{\kappa}\ln(y^*) + C \tag{51}$$

where $\kappa = 0.41$ and $C = 5.5$.

Fig. 14 shows on the left column the streamwise velocity, density, and temperature contours and on the right one the SFS stresses, pressure-strain, and heat flux for a 64x48x48 elements mesh order 3 and 32x32x24 elements mesh order $O = 5$. It is possible to see that the finer mesh is able to better resolve the viscous sublayer and the resolved flow looks more continuous; instead on the coarse mesh the layer is thicker and sometimes the flow looks discontinuous cell by cell, but this problem is mainly due to post-processing software interpolation and partially to the block spectral nature of the solver. The SFS quantities are not as continuous as the flow; also for the finer mesh case, in this instance, the effect is because BSS computes these quantities element-by-element and does not enforce continuity. It is interesting to notice that the SFS quantities have a stronger magnitude in the finer mesh case, where the numerical order is lower, and they are weaker with high numerical order. This result agrees with the fact that a high numerical order is able to better resolve the dynamic properties of the flow.

Fig. 15 presents more quantitative results for the same case with the plots of the TL-transformed mean velocity. The BSS model has the best performance together with Vreman's model on the coarser mesh (Fig. 15a); indeed the Smagorinsky model is not able to predict the intercept as well as BSS. On the finer mesh the Smagorinsky and Vreman models following more closely the DNS results than BSS. This is consistent with BSS' SFS quantities being stronger when using lower numerical order (see Fig. 15b). Indeed, BSS is more dependent on the numerical order than the other two methods because to compute the SFS energy dissipation it needs an accurate description of the energy spectrum. Instead, Smagorinsky and Vreman models benefit from mesh refinement.

Fig. 16 plots the results for the Mach 6.0 hypersonic case. The mesh used for the simulation has 96x75x48 elements and uses numerical order $O = 3$. For this case, the Smagorinsky and BSS model lead to a similar solution following closely the DNS results; instead, Vreman's model predicts a lower log-law intercept. In both Mach 1.5 and 6.0 cases the Dynamic Smagorinsky did not work, probably because the averaging is performed element by element and not over wall parallel planes as suggested by the authors for this kind of simulation. Considering that the averaging strategy would have to be customized for the specific type of flow we can conclude that Smagorinsky, Vreman, and BSS may be a more robust choice for different flow conditions in an unstructured spectral code.

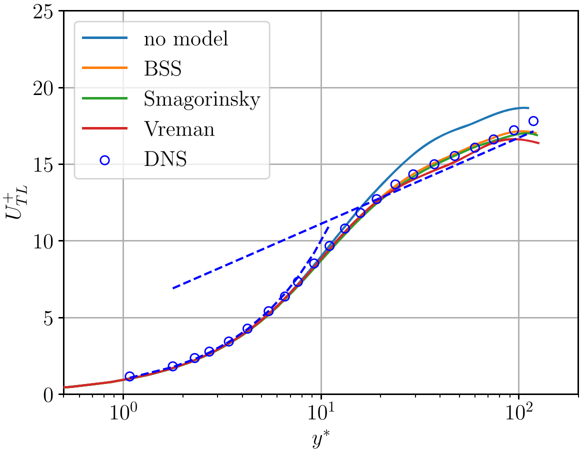


Fig. 16. TL-transformed mean velocity profile for Mach 6.0 compressible channel flow. On the left using a 96x75x48 elements mesh and solver accuracy order $O = 3$. Solid lines are the simulations performed with LES models, while blue circles are the DNS results by Chen and Scalo [57]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

8. Combining shock-capturing and SFS turbulence modeling: The supersonic taylor-Green vortex test case

We discussed before the performance of BSS as a shock capturing method in Section 5 and in Section 7 its behavior as an LES model compared to existing approaches for the subsonic Taylor Green Vortex and supersonic and hypersonic channel flow case. Here we are going to assess BSS performance on a supersonic TGV problem that involves subfilter-scale and shock capturing modeling. The initial condition of the flow uses the equations in Section 7.1 with $M_0 = V_0/\sqrt{\gamma p_0/\rho_0} = 1.25$ and $Re = \rho_0 V_0 L/\mu_0 = 1,600$. In this investigation, the time is normalized as for the subsonic TGV case and the quantities of interest are:

$$E_{kin} = \frac{1}{2\rho_0 V_0^2 \mid \Omega \mid} \int_\Omega \rho \boldsymbol{u} \cdot \boldsymbol{u} d\Omega \tag{52a}$$

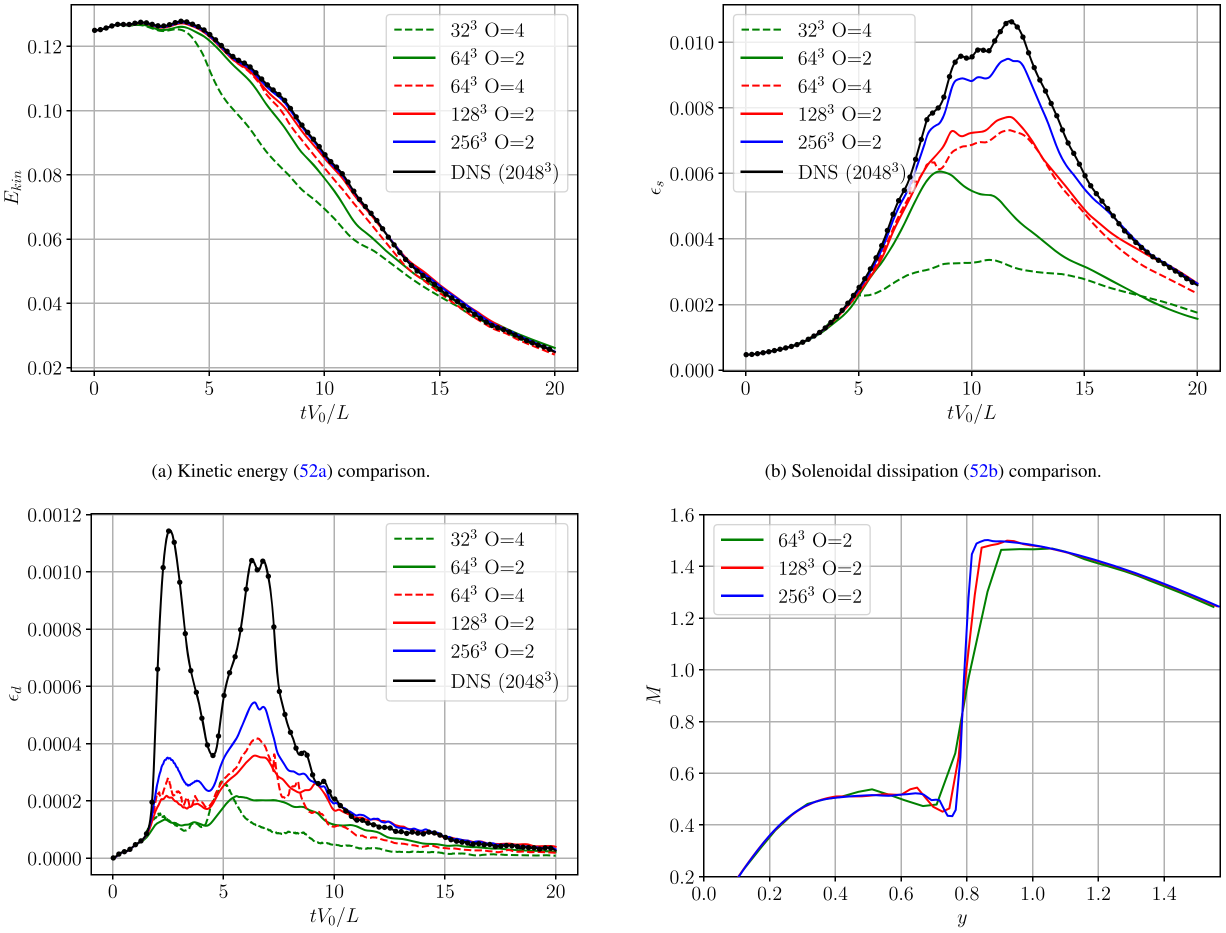
$$\epsilon_s = \frac{L^2}{Re V_0^2 \mid \Omega \mid} \int_\Omega \frac{\mu(\mathrm{T})}{\mu_0} \boldsymbol{\omega} \cdot \boldsymbol{\omega} d\Omega \tag{52b}$$

$$\epsilon_d = \frac{4L^2}{3 Re V_0^2 \mid \Omega \mid} \int_\Omega \frac{\mu(\mathrm{T})}{\mu_0} (\nabla \cdot \boldsymbol{u})^2 d\Omega \tag{52c}$$

where Ω is the domain and $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ the vorticity of the flow. In this work, the BSS model with different numerical orders and resolutions is compared with the DNS results obtained by Lusher and Sandham [59]. In [60] BSS is compared for the same case with different models.

The simulations were performed on 4 different mesh resolutions (32^3, 64^3, 128^3, and 256^3) and two numerical orders (O= 2 and O= 4), so that some cases have the same degrees of freedom but different numerical orders (DOF 128^3, 256^3, and 512^3). The 512^3 DOF was investigated only on the O= 2 due to computational cost restrictions. As shown in Fig. 17a, the 128^3 DOFs case significantly deviates from the DNS results, particularly when using a coarser mesh. However, as the mesh is refined, the results begin to align more closely with the DNS and the 256^3 DOFs case. Among the tested configurations, the 128^3 DOFs case exhibits the largest variation between numerical orders: the $O = 4$ scheme closely matches the DNS up to $tV_0/L = 4.5$, after which it becomes more dissipative.

The plots in Fig. 17b and in Fig. 17c show the solenoidal and dilatational dissipation, and it is possible to see how in the low DOF cases the dissipation is significantly underestimated, especially when comparing the 128^3 DOF cases. All these results show behavior similar to the results obtained for the Sod shock tube and subsonic TGV cases, where the BSS model achieved better performance when using a finer mesh. From the solenoidal dissipation plot in Fig. 17b it is clearly visible how finer meshes lead to better results when using the same DOFs. Instead in the dilatational plot in Fig. 17c the coarser mesh is able to better predict the peaks because it depends on the gradients which are better approximated with higher numerical order. Fig. 17d shows the Mach profile for all O= 2 cases at $x = 0$, $z = 0$, and $tV_0/L = 2.5$, which correspond to the peak of the dilatational dissipation. It is visible how the shock is better approximated with finer meshes as for the Sod shock tube case.



(a) Kinetic energy (52a) comparison.

(b) Solenoidal dissipation (52b) comparison.

(c) Dilatational dissipation (52c) comparison.

(d) Mach profile at $x = 0$, $z = 0$, and $tV_0/L = 2.5$.

Fig. 17. BSS results compare to DNS results of Lusher and Sandham [59] in black with circles, while BSS results was represent with solid lines without circles for 128^3, 256^3, and 512^3 DOFs.

9. Conclusion

In this paper, we presented a new shock-capturing and LES model for high-order finite volume methods called the Block-Spectral-Stress (BSS) closure. The method relies on the last Legendre mode of the velocity gradient inside each element to estimate the subfilter scale stresses, heat-flux, and pressure-work. The main idea is that both turbulence and shock formation have a cascade of energy from large to small scales, and hence a closure for a turbulence model should work also as shock-capturing. Specifically, we noticed that in our model, the subfilter scale pressure-work, which is rarely modeled in LES closures, is important in capturing the shock.

BSS has been implemented in a spectral unstructured code called H^{3}AMR and validated on simulations with shocks and turbulence. In the Sod shock tube case, the model is stable up to numerical order 20. From this investigation, we also notice that the shock is better predicted when the mesh is refined and not as much when the numerical order is increased. Another shock investigation was conducted on the shock-vortex interaction case, where the model was demonstrated to be able to capture the shock without altering the shape of the vortex. Also in this case, mesh refinement leads to faster convergence than increasing the numerical order.

BSS has been compared to the Smagorinsky, dynamic Smagorinsky, and Vreman models, with the second one adapted to work in a block spectral code. The Taylor-Green vortex was used to test the model in a homogeneous isotropic turbulence simulation. In the case of numerical order 3, BSS is the only model able to prevent simulation blow-up. For numerical order 6 and 12, and coarse meshes, BSS is more dissipative than the other methods but on finer meshes, it is able to reach similar results to the other models. Another turbulence simulation was performed for supersonic and hypersonic channel flow; in this case, the dynamic Smagorinsky model underperforms because of the averaging chosen for the block spectral implementation, instead BSS, the standard Smagorinsky, and Vreman were able to better predict the log-law intercept. Differently from TGV, in the supersonic channel case, BSS achieved better performance in predicting the velocity profile when using a coarser mesh with higher numerical order because of a better prediction of the SFS terms. For the hypersonic case, Smagorinsky is slightly closer to the DNS result than BSS but the difference is minimal, instead Vreman's model estimates a lower velocity closer to the log-law profile instead of the DNS. Finally, we tested the BSS model on a supersonic TGV case that involves both subfilter-scale turbulence and shock capturing modeling. BSS shows similar results to the Sod shock tube and subsonic TGV cases where finer meshes lead to better results when working with similar degrees of freedom.

Therefore, we can conclude that even though BSS is not the best model in every kind of flow is able to work with every flow without significant changes in the code, as in the case of dynamic Smagorinsky which requires different averaging for different kinds of flow. Currently, the model has been used with flux reconstruction numerics on a Legendre base, but we expect that the model can also work with different numerics and is based on high-order finite volume. This last point requires further investigation.

CRediT authorship contribution statement

Matteo Ruggeri: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Victor C. B. Sousa:** Conceptualization; **Carlo Scalo:** Writing – review & editing, Resources, Project administration, Funding acquisition, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Carlo Scalo reports financial support was provided by ONR. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge support through N00014-21-1-2475 (ONR-MURI, PI: Venkat Raman) and ONR grant nos. N00014-23-1-2560 (Core ONR program, PI: Carlo Scalo) with Dr. Eric Marineau as the program manager.

References

[1] S.K. Godunov, I. Bohachevsky, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, Matematičeskij Sbornik 47 (3) (1959) 271–306.

[2] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous galerkin methods, in: 18Th AIAA Computational Fluid Dynamics Conference, 2007, p. 4079.

[3] H.T. Huynh, A reconstruction approach to high-order schemnes including discontinuous galerkin for diffusion, in: 47Th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2009, p. 403.

[4] H.T. Huynh, Z.J. Wang, P.E. Vincent, High-order methods for computational fluid dynamics: a brief review of compact differential formulations on unstructured grids, Comput. Fluids 98 (2014) 209–220.

[5] N. Tonicello, G. Lodato, L. Vervisch, Entropy preserving low dissipative shock capturing with wave-characteristic based sensor for high-order methods, Comput. Fluids 197 (2020) 104357. https://doi.org/10.1016/j.compfluid.2019.104357

[6] G. Lodato, P. Castonguay, A. Jameson, Structural wall-modeled LES using a high-order spectral difference scheme for unstructured meshes, Flow. Turbul. Combust. 92 (2014) 579–606.

[7] G. Lodato, P. Castonguay, A. Jameson, Discrete filter operators for large-eddy simulation using high-order spectral difference methods, Int. J. Numer. Methods Fluids 72 (2) (2013) 231–258.

[8] J.-B. Chapelier, M. De La Llave Plata, E. Lamballais, Development of a multiscale LES model in the context of a modal discontinuous galerkin method, Comput. Methods Appl. Mech. Eng. 307 (2016) 275–299.

[9] J.-B. Chapelier, G. Lodato, A. Jameson, A study on the numerical dissipation of the spectral difference method for freely decaying and wall-bounded turbulence, Comput. Fluids 139 (2016) 261–280.

[10] J.-B. Chapelier, G. Lodato, A spectral-element dynamic model for the large-Eddy simulation of turbulent flows, J. Comput. Phys. 321 (2016) 279–302.

[11] D. Gottlieb, C.-W. Shu, A. Solomonoff, H. Vandeven, On the gibbs phenomenon i: recovering exponential accuracy from the fourier partial sum of a nonperiodic analytic function, J. Comput. Appl. Math. 43 (1–2) (1992) 81–98.

[12] C. Lanczos, J. Boyd, Discourse on Fourier series, SIAM, 2016.

[13] D. Gottlieb, S.A. Orszag, Numerical analysis of spectral methods: theory and applications, SIAM, 1977.

[14] J. Schumacher, J.D. Scheel, D. Krasnov, D.A. Donzis, V. Yakhot, K.R. Sreenivasan, Small-scale universality in fluid turbulence, Proceed. National Acad. Sci. 111 (30) (2014) 10961–10965.

[15] S. Khurshid, D.A. Donzis, K.R. Sreenivasan, Energy spectrum in the dissipation range, Phys. Rev. Fluids 3 (8) (2018) 082601.

[16] D.A. Donzis, J.P. John, Universality and scaling in homogeneous compressible turbulence, Phys. Rev. Fluids 5 (8) (2020) 084609.

[17] V. C. B. Sousa, C. Scalo, A unified quasi-Spectral viscosity (QSV) approach to shock capturing and large-eddy simulation, J. Comput. Phys. 459 (2022) 111139. https://doi.org/10.1016/j.jcp.2022.111139

[18] P. Gupta, G. Lodato, C. Scalo, Spectral energy cascade in thermoacoustic shock waves, J. Fluid. Mech. 831 (2017) 358-393. https://doi.org/10.1017/jfm.2017.635

[19] P. Gupta, C. Scalo, Spectral energy cascade and decay in nonlinear acoustic waves, Phys. Rev. E 98 (3) (2018) 033117.

[20] F. Bassi, S. Rebay, A high-Order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations, J. Comput. Phys. 131 (2) (1997) 267–279. https://doi.org/10.1006/jcph.1996.5572

[21] B. Cockburn, C.W. Shu, TVB Runge-Kutta local projection discontinuous galerkin finite element method for conservation laws. II. general framework, Math. Comput. 52 (1989) 411–435. https://doi.org/10.1090/S0025-5718-1989-0983311-4

[22] Y. Lv, Development of a nonconservative discontinuous galerkin formulation for simulations of unsteady and turbulent flows, Int. J. Numer. Methods Fluids. 92 (5) (2020) 325–346.

[23] E.J. Ching, R. Johnson, Fully conservative discontinuous galerkin method for supercritical, real-fluid flows, in: AIAA SCITECH 2024 Forum, 2024, p. 0913.
[24] D.A. Kopriva, J.H. Kolias, A conservative staggered-Grid chebyshev multidomain method for compressible flows, J. Comput. Phys. 125 (1) (1996) 244–261. https://doi.org/10.1006/jcph.1996.0091
[25] L. Ramírez, J. Fernández-Fidalgo, J. París, M. Deligant, S. Khelladi, X. Nogueira, A very fast high-order flux reconstruction for finite volume schemes for computational aeroacoustics, Eng. Comput. 41 (1) (2025) 667–680.
[26] L. Wang, W. Trojak, F. Witherden, A. Jameson, Nonlinear p-multigrid preconditioner for implicit time integration of compressible navier–Stokes equations, arXiv preprint arXiv:2202.09733 (2022).
[27] K. Asthana, M.R. López-Morales, A. Jameson, Non-linear stabilization of high-order flux reconstruction schemes via fourier-spectral filtering, J. Comput. Phys. 303 (2015) 269–294. https://doi.org/10.1016/j.jcp.2015.09.041
[28] M. Morduchow, P. Libby, On a complete solution of the one-Dimensional flow equations of a viscous, heat-Conducting, compressible gas, J. Aeronaut. Sci. 16 (11) (1949) 674-84.
[29] J. Smoller, Shock waves and reaction-diffusion equations, 258, Springer Science & Business Media, 2012.
[30] M.D. Salas, A. Iollo, Entropy jump across an inviscid shock wave, Theor. Comput. Fluid Dyn. 8 (5) (1996) 365–375.
[31] J.F. Colombeau, A.Y. Le Roux, Multiplications of distributions in elasticity and hydrodynamics, J. Math. Phys. 29 (2) (1988) 315–319.
[32] T. Toki, V.C.B. Sousa, Y. Chen, G.P. Camillo, A. Wagner, C. Scalo, Large-eddy simulations of conical hypersonic turbulent boundary layers over cooled walls via volumetric rescaling method, J. Fluid Mech. 1003 (2025) A28.
[33] P. Fernandez, C. Nguyen, J. Peraire, A physics-based shock capturing method for unsteady laminar and turbulent flows, in: 2018 AIAA Aerospace Sciences Meeting, 2018, p. 0062.
[34] F. Ducros, P. Comte, M. Lesieur, Large-eddy simulation of a spatially growing boundary layer over an adiabatic flat plate at low mach number, Int. J. Heat Fluid Flow 16 (5) (1995) 341–348.
[35] E. Tadmor, The numerical viscosity of entropy stable schemes for systems of conservation laws. i, Math. Comput. 49 (179) (1987) 91–103.
[36] G.J. Gassner, A skew-symmetric discontinuous galerkin spectral element discretization and its relation to SBP-SAT finite difference methods, SIAM J. Scientif. Comput. 35 (3) (2013) A1233–A1253.
[37] J.-L. Guermond, B. Popov, Invariant domains and first-order continuous finite element approximation for hyperbolic systems, SIAM J. Numer. Anal. 54 (4) (2016) 2466–2489.
[38] X. Zhang, C.-W. Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, J. Comput. Phys. 229 (9) (2010) 3091–3120.
[39] C.L. Running, B.L. Bemis, J.L. Hill, M.P. Borg, J.J. Redmond, K. Jantze, C. Scalo, Attenuation of hypersonic second-mode boundary-layer instability with an ultrasonically absorptive silicon-carbide foam, Exp. Fluids 64 (4) (2023). https://doi.org/10.1007/s00348-023-03615-w
[40] B.L. Bemis, J.L. Brun, C.T. Wanstall, J.L. Hill, M.P. Borg, J.J. Redmond, M. Ruggeri, K. Jantze, C. Scalo, C.L. Running, Ultrasonically absorptive silicon-Carbide foam for boundary-Layer control, in: AIAA SCITECH 2023 Forum, 2023. https://doi.org/10.2514/6.2023-0096
[41] B.L. Bemis, M.C. Sieve, J.L. Hill, M.P. Borg, J.J. Redmond, M. Ruggeri, K. Jantze, C. Scalo, C.L. Running, Effect of porosity on the ability of silicon-Carbide foams to attenuate the second-Mode boundary-Layer instability, in: AIAA Aviation 2023 Forum, 2023.
[42] C.L. Running, B.L. Bemis, J.L. Hill, M.P. Borg, J.J. Redmond, K. Jantze, C. Scalo, Attenuation of hypersonic second-mode boundary-layer instability with an ultrasonically absorptive silicon-carbide foam, Exp. Fluids 64 (4) (2023) 79. https://doi.org/10.1007/s00348-023-03615-w
[43] D.A. Kopriva, J.H. Kolias, A conservative staggered-Grid chebyshev multidomain method for compressible flows, J. Comput. Phys. 125 (1) (1996) 244–261. https://doi.org/10.1006/jcph.1996.0091
[44] J. Smagorinsky, General circulation experiments with the primitive equation i the basic experiment, Mon. Weather Rev. 91 (1963) 99–164. https://doi.org/10.1175/1520-0493
[45] P. Moin, K. Squires, W. Cabot, S. Lee, A dynamic subgrid-scale model for compressible turbulence and scalar transport, Phys. Fluids A: Fluid Dynam. 3 (11) (1991) 2746–2757.
[46] A.W. Vremana, An eddy-viscosity subgrid-scale model for turbulent shear flow : algebraic theory and applications, 2004. https://api.semanticscholar.org/CorpusID:162177196.
[47] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, USSR Comput. Math. Mathemat. Phys. 1 (1961) 267–279.
[48] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A: Fluid Dynam. 3 (7) (1991) 1760–1765.
[49] V.C.B. Sousa, C. Scalo, A legendre spectral viscosity (LSV) method applied to shock capturing for high-order flux reconstruction schemes, J. Comput. Phys. 460 (2022) 111157. https://doi.org/10.1016/j.jcp.2022.111157
[50] A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1) (1978) 1–31. https://doi.org/10.1016/0021-9991(78)90023-2
[51] T. Haga, S. Kawai, On a robust and accurate localized artificial diffusivity scheme for the high-order flux-reconstruction method, J. Comput. Phys. 376 (2019) 534–563. https://doi.org/10.1016/j.jcp.2018.09.052
[52] H. Terashima, S. Kawai, M. Koshi, Consistent numerical diffusion terms for simulating compressible multicomponent flows, Comput. Fluids 88 (2013) 484–495. https://doi.org/10.1016/j.compfluid.2013.10.007
[53] J. Butcher, Runge-kutta methods, Scholarpedia 2 (9) (2007) 3147.
[54] V. Rajora, T. Aslam, W.W. Chapman, C. Meyer, S.F. Son, C. Scalo, A block-spectral adaptive H-/p-refinement strategy for shock-dominated problems, J. Comput. Phys. 514 (2024) 113255.
[55] J.L. Ellzey, M.R. Henneke, J.M. Picone, E.S. Oran, The interaction of a shock with a vortex: shock distortion and the production of acoustic waves, Phys. Fluids 7 (1995) 172–184. https://doi.org/10.1063/1.868738
[56] A. Rault, G. Chiavassa, R. Donat, Shock-Vortex interactions at high mach numbers, J. Sci. Comput. 19 (2003) 347–371. https://doi.org/10.1023/A:1025316311633
[57] Y. Chen, C. Scalo, Trapped waves in supersonic and hypersonic turbulent channel flow over porous walls, J. Fluid Mech. 920 (2021) A24. https://doi.org/10.1017/jfm.2021.428
[58] A. Trettel, J. Larsson, Mean velocity scaling for compressible wall turbulence with heat transfer, Phys. Fluids 28 (2) (2016).
[59] D.J. Lusher, N.D. Sandham, Assessment of low-dissipative shock-capturing schemes for the compressible taylor–Green vortex, AIAA J. 59 (2) (2021) 533–545.
[60] J.-B. Chapelier, D.J. Lusher, W. Van Noordt, C. Wenzel, T. Gibis, P. Mossier, A. Beck, G. Lodato, C. Brehm, M. Ruggeri, C. Scalo, N. Sandham, Comparison of high-order numerical methodologies for the simulation of the supersonic taylor-green vortex flow, Phys. Fluids 36 (5) (2024) 055146. https://doi.org/10.1063/5.0206359