

HIGH-ORDER SYMMETRY-PRESERVING DISCRETIZATIONS: APPLICATION TO REPEATED MATRIX-BLOCK STRUCTURES

J. Plana-Riu, D.Santos, F.X. Trias, A. Oliva

*Heat and Mass Transfer Technological Center, Technical University of Catalonia,
C/Colom 11, 08222 Terrassa (Spain)*

josep.plana.riu@upc.edu

Abstract

This work presents a novel methodology for developing high-order symmetry-preserving discretizations of the Navier–Stokes equations. These schemes overcome the classical trade-off between physical fidelity and numerical accuracy inherent in second-order methods by preserving the mathematical symmetries of the continuous equations while achieving fourth-order accuracy. The proposed operators lead to denser sparse matrices, which, when coupled with repeated matrix-block structures, enable the transformation of sparse matrix-vector products (SpMV) into sparse matrix-matrix products (SpMM), substantially increasing arithmetic intensity and overall computational performance. The approach is applied within a projection method framework and tested in turbulent channel flow simulations using the in-house TermoFluids Algebraic code. Results demonstrate significant speed-ups across all projection stages, validating the applicability of the method for modern high-performance computing architectures.

1 Introduction

Users from Computational Fluid Dynamics (CFD) tools usually need to choose between multiple items: robustness, accuracy, physical fidelity, performance, etc. Thus, it is the user who, depending on the objective of the simulation, will decide where to put more efforts in as well as where these efforts are not much required.

In terms of accuracy, without taking anything else in consideration, any user would prefer having high-order discretization schemes that would allow having coarser grids and thus better performance in the runs. Nonetheless, generally higher-order schemes lead to breaking the symmetries present in the continuous formulation of the equations, eventually leading to unstable solutions unless some stabilization technique is employed, and thus the physical fidelity will be reduced notably. Classical second-order symmetry-preserving schemes (the second-order definition in Verstappen and Veldman (2003), Trias et al. (2014)) will indeed preserve these symmetries, at a cost of re-

quiring finer grids to obtain equivalent results, which eventually will lead to reduced performance.

If higher-order symmetry-preserving schemes, i.e. schemes that preserve the inherent continuous symmetries with a higher-than-second order of accuracy, were used, this would tackle the three problems simultaneously and thus no trade-off among accuracy, physical fidelity and performance would not need to be considered. Hence, the simulation would benefit from the higher-order schemes while preserving the mathematical structure raising from the continuous equations.

Moreover, having higher-order discretization schemes lead to sparse matrices with a higher number of non-zeros per row. According to Plana-Riu et al. (2024), having denser matrices provides performance benefits when exploiting repeated matrix block structures. These techniques, firstly introduced by Krasnopolsky (2018), arise with the goal of tackling the problem that comes with the reduced memory bandwidths from current supercomputers, i.e. their peak performance is really high, yet CFD simulations cannot extract their full potential as there is much more data to transfer compared to the amount of computations that need to be performed. According to the roofline theory from Williams et al. (2008), this corresponds to a low arithmetic intensity.

Therefore, the benefit of using these high-order symmetry-preserving schemes is twofold. First of all, it breaks the trade-off from physical fidelity and accuracy that appears with the classical non-preserving high-order schemes and the second-order symmetry-preserving scheme as it allows preserving symmetries with greater accuracy. On the other hand, the performance gains in a parallel-in-time framework notably increases given the denser operators from these schemes, thus giving additional benefits without the use of any additional resources.

2 Towards higher-order symmetry-preserving discretizations

Considering a 1D finite-volume discretization of the diffusive term, this reads as

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{x_i} = \frac{1}{h} \left(\left. \frac{\partial \phi}{\partial x} \right|_{x_{i+1/2}} - \left. \frac{\partial \phi}{\partial x} \right|_{x_{i-1/2}} \right) + \mathcal{O}(h^2), \quad (1)$$

where h is the grid spacing. Considering a box filter with filter length h ,

$$\bar{\phi} = \frac{1}{h} \int_{x-h/2}^{x+h/2} \phi \, dx, \quad (2)$$

the filtered discretization reads

$$\left. \frac{\partial^2 \bar{\phi}}{\partial x^2} \right|_{x_i} = \frac{1}{h} \left(\left. \frac{\partial \bar{\phi}}{\partial x} \right|_{x_{i+1/2}} - \left. \frac{\partial \bar{\phi}}{\partial x} \right|_{x_{i-1/2}} \right). \quad (3)$$

Similarly, the partial first derivatives are discretized using a second-order approximation,

$$\left. \frac{\partial \bar{\phi}}{\partial x} \right|_{x_{i+1/2}} \approx \frac{\phi_{i+1} - \phi_i}{h} + \mathcal{O}(h^2), \quad (4)$$

which filtered counterparts follow similarly to the second-order derivative,

$$\left. \frac{\partial \bar{\phi}}{\partial x} \right|_{x_{i+1/2}} = \frac{\partial \bar{\phi}}{\partial x} \Big|_{x_{i+1/2}} = \frac{\phi_{i+1} - \phi_i}{h}. \quad (5)$$

Therefore, the second-order finite-volume discretization corresponds to

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{x_i} \approx \left. \frac{\partial^2 \bar{\phi}}{\partial x^2} \right|_{x_i}, \quad (6)$$

which can be extended to multidimensional problems as

$$\nabla^2 \phi = \nabla^2 \bar{\phi} + \mathcal{O}(h^2). \quad (7)$$

By definition, a box filter can be defined as $\phi = \bar{\phi} + \phi' = \bar{\phi} - \frac{h^2}{24} \nabla^2 \bar{\phi} + \mathcal{O}(h^4)$. This can be applied to Eq. (7), leading to

$$\nabla^2 \phi = \nabla^2 \bar{\phi} + \nabla^2 \phi' + (\nabla^2 \bar{\phi})' + (\nabla^2 \phi')'. \quad (8)$$

Therefore, the order of accuracy of Eq. (7) can be improved by introducing additional terms. This reads as

$$\nabla^2 \phi \approx \nabla^2 \bar{\phi} + \nabla^2 \phi' + (\nabla^2 \bar{\phi})' + \mathcal{O}(h^4). \quad (9)$$

If this procedure is applied to the 1D case, it leads to the 5-point fourth-order approximation of the second derivative. Extending it to a discrete level, this methodology can be applied to extend the classical second-order approximation of the discrete Laplacian L to a fourth-order approximation by

$$\tilde{L} = (I + R)L(I + R), \quad (10)$$

where I is the identity matrix of size n , being n the number of nodes of the discrete domain, and R is the discrete filter residual, which is defined as

$$R = -\frac{1}{24} T_{cs}^T T_{cs}, \quad (11)$$

where T_{cs} is the cell-to-face identity matrix. According to Verstappen and Veldman (2003), the diffusive operator should correspond to a semi-negative defined symmetric matrix, which is actually fulfilled by $\tilde{L}$. Namely by definition in a symmetry-preserving framework, $L = MG$. Therefore, if this is applied to Eq. (10),

$$\tilde{L} = \tilde{M} \tilde{G} \underbrace{=}_{\tilde{M} = -\tilde{G}^T \Omega_s \tilde{G}} -\tilde{G}^T \Omega_s \tilde{G}, \quad (12)$$

where $\tilde{M} = (I + R)M$, $\tilde{G} = G(I + R)$, and Ω_s is the diagonal matrix containing the staggered volumes i associated to each face. Hence, the definition of the high-order Laplacian actually preserves the duality of the gradient and the divergence.

With regards to the convective term, its order of accuracy can be improved from second- to fourth-order in the same manner, i.e.

$$\tilde{C} = (I + R)C(I + R), \quad (13)$$

where C is the second-order symmetry-preserving discretization of the convective term defined as $MU_s \Pi$, where $U_s = \text{diag}(\mathbf{u}_s)$ is a diagonal matrix containing the velocities normal to the faces, and Π is the cell-to-face interpolator, which in a symmetry-preserving framework corresponds to $1/2|T_{cs}|$. Introducing it onto the high-order approximation, it reads

$$\tilde{C} = \tilde{M} U_s \tilde{\Pi}, \quad (14)$$

where $\tilde{\Pi} = \Pi(I + R)$. Therefore, these new operators will indeed have a greater number of entries. These sparsity patterns and number of non-zeros will depend on the operator, the dimensionality of the problem as well as the kind of mesh that is being used, i.e. hexahedral or tetrahedral, for 3D implementations; squares or triangles, for the 2D counterparts. Table 1 depicts the number of non-zeros per row depending on all these aforementioned parameters.

Application to a projection method

Projection methods, such as the fractional step method or the PISO algorithm, apply their incompressibility constraint making use of the Helmholtz-Hodge theorem. Hence, it yields the Poisson equation for pressure, $\mathbf{p}_c$,

$$L\mathbf{p}_c = M\mathbf{u}_s^p, \quad (15)$$

Table 1: Number of non-zeros per row for the divergence, gradient, cell-to-face interpolator, Laplacian and convective operators.

	M	$\tilde{M}$	G, Π	$\tilde{G}, \tilde{\Pi}$	L	$\tilde{L}$	C	$\tilde{C}$
$1D$	2	4	2	4	3	5	2	4
$2D^{tri}$	3	9	2	6	4	10	3	9
$2D^{quad}$	4	16	2	8	5	13	4	12
$3D^{hex}$	6	36	2	12	7	51	6	50

which arises from the projection step $\mathbf{u}_s = \mathbf{u}_s^p - G\mathbf{p}_c$. If the high-order operators are considered, Eq. (15) reads as

$$\tilde{L}\tilde{\mathbf{p}}_c = \tilde{M}\mathbf{u}_s^p. \quad (16)$$

This equation will arise from the projection step $\tilde{\mathbf{u}}_s = \mathbf{u}_s^p - \tilde{G}\tilde{\mathbf{p}}_c$. Introducing the definitions for the high-order operators and simplifying $I + R$, Eq. (16) recovers

$$L(I + R)\tilde{\mathbf{p}}_c = M\mathbf{u}_s^p, \quad (17)$$

from which is straightforward to obtain that the outcome from the Poisson equation for both second- and high-order schemes is going to be the same. Therefore, the Laplacian used for the Poisson solution is going to be the second-order given it is lighter.

3 Exploiting repeated block structures in the Navier-Stokes equations

Current supercomputers have the greatest peak performance in history. However, CFD simulations cannot exploit the full potential as their memory bandwidth limits the performance that can be extracted out of the computer. According to the roofline model from Williams et al. (2008), the maximum performance that can be extracted in these situations corresponds to

$$\pi = \beta I, \quad (18)$$

where π is the computer performance, β is the memory bandwidth, and I corresponds to the arithmetic intensity, defined as the ratio of computations and data transferred. As β is fixed by the machine used, this study aims at giving methods to increment I .

In order to increment this arithmetic intensity, previous studies (Krasnopolsky (2018), Alsalti-Baldellou et al. (2023), Plana-Riu et al. (2024)) focused their efforts in, under circumstances where the matrices show repeated block structures, playing with these blocks to reduce the amount of data to transfer by transforming the sparse matrix-vector products (SpMV) into sparse matrix-matrix products (SpMM), and eventually improving I .

Let us consider a sparse matrix $A = I_m \otimes \tilde{A} \in \mathbb{R}^{sn \times sm}$, where $\tilde{A} \in \mathbb{R}^{n \times m}$ is a repeated block from A . Consider s vectors $\tilde{\mathbf{x}} \in \mathbb{R}^m$ such that $\mathbf{x} =$

$(\tilde{\mathbf{x}}_1 \tilde{\mathbf{x}}_2 \dots \tilde{\mathbf{x}}_s)^T$. Therefore, the outcome of the SpMV $A\mathbf{x}$ is going to be the same as the outcome of the equivalent SpMM,

$$\tilde{A}(\tilde{\mathbf{x}}_1 \tilde{\mathbf{x}}_2 \dots \tilde{\mathbf{x}}_s). \quad (19)$$

Namely by definition, the maximum possible speed-up, $P_{m, \text{SpMM}}$ will be related to the ratio of I , which for a SpMM of a matrix A with n_c columns, n_r rows, $\text{nnz}(A)$ non-zeros and s right-hand sides is defined as follows,

$$\begin{aligned} I_{\text{SpMM}}(m) &= \\ &= \frac{(2\text{nnz}(A) + 1)s}{12\text{nnz}(A) + 4(n_r/s + 1) + 8(n_r + n_c + s)}. \end{aligned} \quad (20)$$

Therefore, the speed-up will have its upper bound as

$$P_m = \frac{I_{\text{SpMM}}(m)}{I_{\text{SpMV}}} = \frac{I_{\text{SpMM}}(m)}{I_{\text{SpMM}}(1)}. \quad (21)$$

A lower bound can be computed assuming zero temporal locality when accessing the input vector coefficients. This is introduced in Eq. (20) by replacing $8n_c s$ by $8\text{nnz}(A)s$.

For $m \rightarrow \infty$, the upper-bound ends up depending on the number of non-zeros per row. For square matrices, i.e. $n_r = n_c$,

$$\lim_{s \rightarrow \infty} P_m \approx_{n_r \gg 1} \frac{12\text{nnz}(A)/n_r + 20}{16}. \quad (22)$$

Thus, high-order schemes such as those presented in Section 2 will take a greater benefit given they are denser matrices, i.e. they have a greater number of non-zeros per row. Figure 1 gives a graphical example on how this speed-up is obtained.

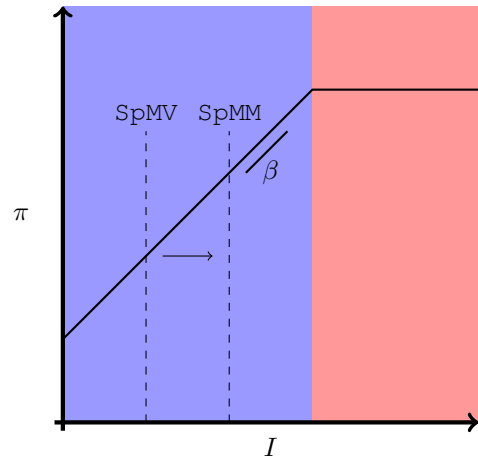


Figure 1: Simplified version of a roofline model in which the memory-bound (blue) and compute-bound (red) regions are depicted. The goal of the present paper is represented in pushing the arithmetic intensity I towards the compute-bound zone.

A simple way to make these block structures appear is considering multiple parameter simulations. Let a flow simulation with an arbitrary parameter of which s values are of interest, with exact geometry and mesh. Therefore, these cases will have identical divergence, gradient and Laplacian operators, which makes it possible to run all s simulations simultaneously. Following the notation from Trias et al. (2014), the semi-discrete equations can be rewritten as

$$M\mathbf{U}_s = \mathbf{0}_s, \quad (23a)$$

$$\Omega \frac{d\mathbf{U}_c}{dt} + M\mathbf{U}_s \Pi \mathbf{U}_c = D\mathbf{U}_c - \Omega G_c \mathbf{P}_c, \quad (23b)$$

where $\mathbf{U}_c = (\mathbf{u}_{c,1} \ \mathbf{u}_{c,2} \ \dots \ \mathbf{u}_{c,s})$ is the dense matrix containing the s rhs of the velocity field, $\mathbf{P}_c = (\mathbf{p}_{c,1} \ \mathbf{p}_{c,2} \ \dots \ \mathbf{p}_{c,s})^T$ is the equivalent for the pressure field and $\mathbf{U}_s$ is the velocity at the faces for the s rhs. Note that in these circumstances, all SpMV are converted to SpMM and therefore all relevant parameters are being simulated simultaneously.

This way of expressing the Navier-Stokes equations, i.e. rewriting all the SpMV as SpMM, allows exploiting this potential performance benefit not only in the solution of the Poisson equation as in Krasnopol'sky (2018) but in the whole projection method itself. Moreover, considering the bigger number of entries that the higher-order schemes provide, as shown in Table 1, the bigger the potential performance gains as shown by Eq. (22).

Ensemble averaging of multiple flow states

An application where these repeated block structures appears naturally is in the ensemble averaging of multiple flow states. Under certain conditions where the statistically steady state fraction of a turbulent simulation can be considered ergodic, i.e., an ensemble average of m different simulations is equivalent to the time average of a single simulation (Tosi et al. (2022)). Assuming that the time average of the single simulation has been performed for T_A time units after a transition time of T_T , the ensemble averaged relevant quantities can be computed as

$$\langle \phi \rangle = \frac{1}{m} \sum_{i=1}^m \frac{1}{T_A/m} \int_{T_T}^{T_T+T_A/m} \phi \, dt, \quad (24)$$

where ϕ and $\langle \phi \rangle$ are the relevant quantity and its ensemble averaged counterpart, respectively. By running these flow states simultaneously in the same device, the repeated block structures will appear.

4 Numerical experiments

This section presents the numerical experiments performed to assess the performance of the high-order symmetry-preserving discretizations presented in Section 2 and their application to the ensemble averaging of multiple flow states. In order to test this performance, a turbulent planar channel flow at a Reynolds

number of $Re_\tau = 180$ is considered, with a domain size of $4\pi\delta \times 2\delta \times 4\pi\delta/3$. The used mesh corresponds to a $128 \times 128 \times 128$ Cartesian mesh, which corresponds to approximately 2.1 million nodes. These simulations have been run in a single node of the JFF2 cluster, corresponding to 2x Intel Xeon 6230 processors, which sums up to 40 CPUs. Therefore, the granularity of the runs corresponds to 52428 cells per core, which is equivalent to the strong scaling limit according to Mosqueda-Otero et al. (2024). The simulations have been run in an OpenMP environment, with 40 threads per run.

The high-order operators have been implemented in the in-house CFD code, TermoFluids Algebraic (tfa), in which the calculations are carried out using the HPC² framework, which allows the portability of the code to different HPC architectures, and allows converting SpMVs into SpMMs. Therefore, the performance gains from the high-order symmetry-preserving operators in a repeated matrix-block structure can be assessed.

The relevant metric in this performance comparison will be the speed-up P_m , which is defined as

$$P_m = \frac{mT_{\text{SpMV}}}{T_{\text{SpMM},m}}, \quad (25)$$

where T_{SpMV} is the time spent in the SpMV and $T_{\text{SpMM},m}$ is the time spent in the SpMM with m right-hand sides. Therefore, the speed-ups for the SpMV operations originating from the divergence, gradient, and interpolator operators are going to be computed, as their sparsity patterns differ from one another. Moreover, the speed-ups for the whole diffusive and convective operators are going to be computed similarly to Eq. (25), as it is in the whole of these operations that the overall speed-up is going to translate to the whole iteration.

Figure 2 (top) shows the obtained speed-ups for the used operators in the simulations according to the application of the method in a projection method. It shows that the speed-ups for the high-order methods are generally greater than the second-order gradient, which is present as it is required as well for the projection step. Note that the high-order gradient appears multiplying the velocity field, as the implementation of the diffusive term is performed as $\tilde{M}\tilde{G}$, given that it makes the implementation flexible for LES simulations. As shown in Eq. (22), a matrix with a greater number of non-zeros per row will eventually lead to a greater potential speed-up. Therefore, as the high-order divergence, $\tilde{M}$ has a greater number of non-zeros per row (in the 3D hexahedral case, 36 non-zeros per row) compared to the high-order gradient and interpolator, which in the hexahedral case have 12 non-zeros per row, the shown speed-up is greater for the divergence operator than for the gradient and interpolator, which behave equally as their sparsity pattern is identical.

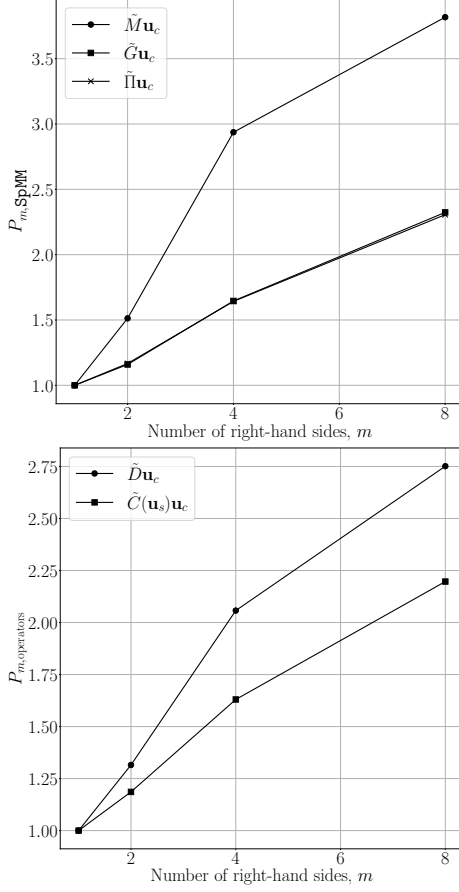


Figure 2: Speed-up P_m for the SpMM of the used operators in the high-order framework: the high-order divergence, gradient, interpolator (top). Speed-up P_m for the whole diffusive and convective operators (bottom).

With regards to the whole diffusive and convective operators, Figure 2 (bottom) shows that the computation of these terms, which are not only based on SpMV's but also other operations such as elementwise vector products ($\text{axt}y$), have speed-ups which eventually translate to the whole iteration. Note that the translation from SpMV to SpMM is also applied in the numerical solution of the Poisson equation, which highly relies on sparse matrix-vector products and thus a big speed-up is also obtained there. The performance benefits from this SpMM-based Poisson solver are shown in Plana-Riu et al. (2024) for the classical second-order discretization, which is the one used in the present methodology.

In comparison, Figure 3 shows the speed-ups obtained for the second-order symmetry-preserving discretizations. While the benefit obtained in the individual operators is around a 4% and 16% with the largest number of right-hand sides for the divergence and gradient operators, respectively, the benefit obtained in the whole diffusive and convective operators corresponds to a 33% and 6% for the diffusive and convective terms, respectively. These speed-up

figures thus take into consideration also the other operations required in the computation of these terms. Note, moreover, that the trends for the speed-ups in the second-order operators show that the speed-up tends to saturate, while for the higher-order counterparts, the speed-up is still increasing. This highlights the fact that for denser matrices, a bigger number of right-hand sides can be exploited, leading to greater speed-ups.

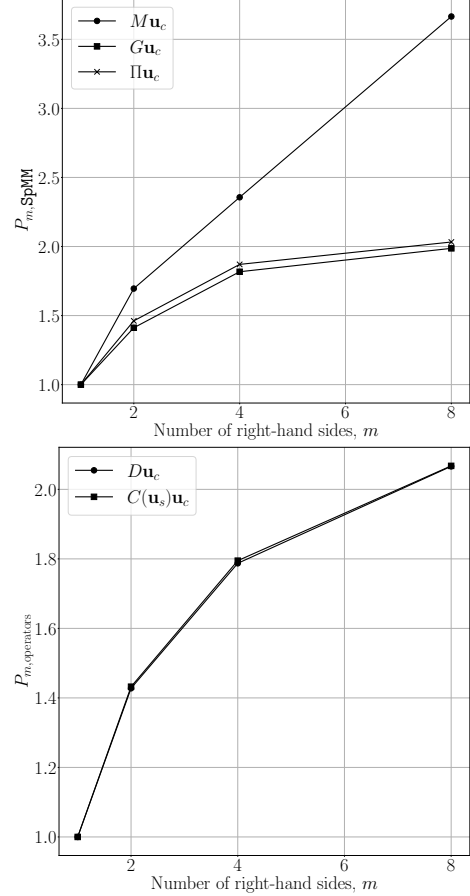


Figure 3: Speed-up P_m for the SpMM of the used operators in the second-order framework: the second-order divergence, gradient, interpolator (top). Speed-up P_m for the whole diffusive and convective operators (bottom).

Therefore, using these high-order symmetry-preserving discretizations leads to a performance benefit throughout the whole projection method stages: in the computation of the predictor velocity with the computation of the diffusive and convective terms, in the numerical solution of the Poisson equation, and in the projection of the predictor velocity using the pressure field. Thus, bigger performance gains compared to those from Krasnopolsky (2018) and Plana-Riu et al. (2024) are expected, as the SpMM is applied throughout the whole simulation, and the gains are increased in the computation of the predictor velocity given the denser operators arising from the high-order discretizations.

5 Conclusions

This work presents a methodology to obtain high-order symmetry-preserving discretizations of the Navier-Stokes equations which tackle the trade-off between physical fidelity and accuracy that appears with the classical second-order symmetry-preserving schemes, where the preservation of the continuous symmetries leads to a reduced order of accuracy. The high-order discretizations presented are based on the classical second-order symmetry-preserving discretizations, which are extended to a fourth-order accuracy by introducing the higher-order residuals from the box filter.

As an application example, these high-order discretizations are applied to a projection method, where the methodology leads to having this increased order of accuracy while preserving the second-order operators for the numerical solution of the Poisson equation. Therefore, with the ultimate goal of higher-order schemes, which corresponds to having coarser grids for the same accuracy, the solution of the Poisson equation is going to be lighter, which is a benefit for the performance of the whole projection method, as it is well-known that the Poisson equation is the most expensive part of any incompressible flow simulation.

These high-order schemes lead therefore to denser operators, which has been shown to lead to performance benefits if repeated block structures in the operators are present and exploited. This is exemplified in the present work by considering multiple flow states run simultaneously, which implicitly leads to these repeated block structures and thus it allows converting the sparse matrix-vector products (SpMV) into sparse matrix-matrix products (SpMM), leading to incremented arithmetic intensities and thus performance benefits in these products. Figure 2 shows the obtained speed-ups for the SpMM of the high-order divergence, gradient and interpolator operators, as well as the second-order gradient operator.

This results in an incremented performance in the computation of both diffusive and convective terms as shown in Figure 2 (bottom), which will have a greater weight in the overall profiling of the projection method, as the high-order discretization will allow having a coarser grid for the Poisson equation. Therefore, increasing the performance in these terms will lead to a greater performance benefit in the overall simulation.

The obtained results align with the theoretical expectations from Eq. (22), which states that the denser the sparse matrix, the greater the potential speed-up that can be obtained from the conversion of the SpMV into SpMM .

Acknowledgments

The work presented in this abstract is supported by the *Ministerio de Economía y Competitividad*, Spain, SIMEX project (PID2022-142174OB-

I00). J.P-R. is supported by a *FI AGAUR-Generalitat de Catalunya* fellowship (2022 FI.B 00810) from the Catalan Agency for Management of University and Research Grants (AGAUR). D.S. is supported by a *FI AGAUR-Generalitat de Catalunya* fellowship (2022 FI.B 00173), extended and financed by *Universitat Politècnica de Catalunya and Banco Santander*.

References

- Alsalti-Baldellou, A., Álvarez-Farré, X., Trias, F. X., and Oliva, A. (2023). Exploiting spatial symmetries for solving Poisson's equation. *Journal of Computational Physics*, 486. 112133
- Krasnopol'sky, B. I. (2018). An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles. *Computer Physics Communications*, 229, 8–19.
- Mosqueda-Otero, M., Alsalti-Baldellou, A., Álvarez-Farré, X., Plana-Riu, J., Colomer, G., Trias, F.X., and Oliva, A. (2024). A portable algebraic implementation for reliable overnight industrial LES. 35th Parallel CFD International Conference 2024.
- Plana-Riu, J., Trias, F.X., Alsalti-Baldellou, À., Colomer, G., and Oliva, A. (2024). Performance analysis of parallel-in-time techniques in modern supercomputers. 9th European Congress on Computational Methods in Applied Sciences and Engineering.
- Tosi, R., Núñez, M., Pons-Prats, J., Principe, J., Rossi, R. (2022). On the use of ensemble averaging techniques to accelerate the Uncertainty Quantification of CFD predictions in wind engineering. *Journal of Wind Engineering and Industrial Aerodynamics*, 228. 105105
- Trias, F. X., Álvarez-Farré, X., Alsalti-Baldellou, A., Gorbets, A., and Oliva, A. (2024). An efficient eigenvalue bounding method: CFL condition revisited. *Computer Physics Communications*, 305. 109351
- Trias, F. X., Lehmkuhl, O., Oliva, A., Pérez-Segarra, C. D., and Verstappen, R. W. C. P. (2014). Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids. *Journal of Computational Physics*, 258, 246–267.
- Verstappen, R. W. C. P., and Veldman, A. E. P. (2003). Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1), 343–368.
- Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: An insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4), 65–76.