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

1 Introduction

Within the framework of Computational Fluid Dynamics (CFD) simulations, the user usually needs to make a trade-off between three different items: accuracy, physical fidelity and performance, without forgetting the presence of stability, which will also be a concern in turbulent simulations, as generally its explicit nature makes stable time-steps rather small to preserve stability.

In terms of accuracy, any user would prefer using a high-order discretization as it allows using a coarser grid to obtain equivalent results, which indeed improves the performance of the run. Nonetheless, the presence of a general high-order discretization will not preserve the symmetries present in the continuous nature of the equations and thus the physical fidelity will be lost. On the other hand, using a classical second-order symmetry-preserving discretization will preserve this physical fidelity that was previously missing, yet to achieve good enough accuracy, a finer grid will be needed, losing eventually performance.

Hence, the use of higher-order symmetry-preserving discretizations would tackle the three problems simultaneously, as it would eliminate (or at least limit) the trade-off previously mentioned: the simulation would have high-order accuracy while maintaining the physical fidelity as the continuous operators' properties will be preserved. Moreover, a coarser grid could be used maintaining the errors, thus leading to improved performance.

Nonetheless, in some situations the grid size is determined not by accuracy but by physical lengthscales, such as in direct numerical solution (DNS) runs, in which all scales should be resolved in order to solve the flow properly. In these cases, a higher-order will be more expensive than the classical second-order, as matrices (or stencil-like operations) will be denser, thus having a greater number of operations to be performed, leading to increased wall-clock times.

In order to deal with this issue, this abstract presents a general method to obtain a high-order symmetry-preserving discretization together with a method that, given the presence of repeateb matrix block structures in the simulation, i.e. symmetries, geometrical repetitions, parallel-in-time simulations, etc., to improve the performance of the method which will make the use of these high-order discretizations lighter and faster.

2 High-order symmetry-preserving discretizations

Consider the 1D finite-volume discretization of the diffusive term:

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

where h is the grid spacing. It can be proved that this second-order derivative is equivalent to box-filtering the second-order derivative of the filtered ϕ,

$$\left.\frac{\partial^2\phi}{\partial x^2}\right|_{x_i} = \left.\overline{\frac{\partial^2\bar{\phi}}{\partial x^2}}\right|_{x_i}, \tag{2}$$

which can be easily extended to a multidimensional problem such that,

$$\nabla^2\phi = \overline{\nabla^2\bar{\phi}} + \mathcal{O}(h^2). \tag{3}$$

By definition, any symmetric filter, such as the box filter, is $\phi = \bar{\phi} - \frac{h^2}{24}\nabla^2\phi + \mathcal{O}(h^4)$, which applied to Eq. (3), it allows improving the order of accuracy of the second-order derivative as follows

$$\nabla^2\phi = \overline{\nabla^2\bar{\phi}} + \overline{\nabla^2\phi'} + (\nabla^2\bar{\phi})' + \mathcal{O}(h^4), \tag{4}$$

which corresponds to a second-order approximate deconvolution and in the 1D case leads to the classical 5-point fourth-order approximation of the second derivative.

From a discrete standpoint, the standard second-order approximation to the Laplacian operator, $L = MG$ should be replaced by

$$\tilde{L} = (I + R)L(I + R), \tag{5}$$

where R is the discrete filter residual defined as

$$R = -\frac{1}{24} T_{cs}^T T_{cs}, \tag{6}$$

being T_{cs} the cell-to-face incidence matrix. Eq. (5) will eventually lead to a negative semi-definite symmetric matrix that preserves the continuous property of the Laplacian operator. Hence, the high-order Laplacian operator may be built as

$$\tilde{L} = \tilde{M}\tilde{G} = -\tilde{G}^T \Omega_s \tilde{G}, \tag{7}$$

being $\tilde{G} = G(I + R)$, $\tilde{M} = (I + R)M$, and recalling that $G = -\Omega_s^{-1} M^T$, and Ω_s a diagonal matrix containing the staggered volumes. This interpolation, given a 1D case, will lead to a 7-point fourth-order Laplacian discretization.

With regards to the convective term, the same exact procedure is applied, where the high-order convective operator is defined as follows,

$$\tilde{C} = (I + R)C(I + R), \tag{8}$$

where C is the standard second-order symmetry-preserving convective operator defined as $C = MU_s\Pi$, being U_s the diagonal matrix containing the velocities at the faces, and $\Pi = \frac{1}{2}|T_{cs}|$ is the standard cell-to-face mid-point interpolation. The reader is referred to Trias et al. (2024) for details in the construction of each operator. Introducing then the definition for the convective term to Eq. (8), the high-order discrete form of the convective term can be obtained,

$$\tilde{C} = \tilde{M} U_s \tilde{\Pi}, \tag{9}$$

where $\tilde{\Pi} = \Pi(I + R)$. This interpolation, eventually, will lead to 4-point, zero-diagonal, fourth-order convective operator given a 1D case.

3 Matrix block structures: from `SpMV` to `SpMM`

As shown in the previous section, the matrices $\tilde{L}$ and $\tilde{C}$ will be denser than L and C, as the presence of more points will end up generating sparse matrices with more non-zeros per row. From a general standpoint, this will make sparse matrix-vector products (`SpMV`) more expensive as the amount of data to transfer will increase, plus the number of operations to perform will increase as well. Hence, given a fixed grid size, the time spent in a high-order `SpMV` will be greater than in a second-order `SpMV`.

Nonetheless, under proper circumstances in which repeated matrix block structures are present, those can be used to reduce the amount of data to transfer while preserving the obtention of results, transforming `SpMV` to sparse matrix-matrix operations (`SpMM`). These circumstances may be generated by the user in cases in which multiple runs may be executed simultaneously with the same geometry, so that the operators can be reused. This is the case of a parallel-in-time setup (Krasnopolsky, 2018) or a multiple parameter simulation (Tosi et al., 2022). Moreover, these block structures appear naturally in the matrices, if arranged correctly, in simulations in which symmetries or repeated geometries are present (Alsalti-Baldellou et al., 2024).

Let us consider a parallel-in-time framework, in which multiple simulations are run at the same time in the same device. In terms of the incompressible Navier-Stokes equations, these can be written as

$$\frac{d}{dt}\begin{pmatrix} \Omega\mathbf{u}_1 & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \Omega\mathbf{u}_m \end{pmatrix} + \begin{pmatrix} C & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \end{pmatrix} =$$

$$\begin{pmatrix} D & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & D \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \end{pmatrix} - \begin{pmatrix} G & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & G \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{pmatrix}, \tag{10}$$

where $D = \nu L$. Note that in Eq. (10), the operators are denoted without $\tilde{\cdot}$. Nonetheless, the definition is general as it may be used for both second- or high-order operators.

As it can be seen all simulations share the same exact operators and thus, from a computational point-of-view, it will be equivalent in terms of results to perform a `SpMM` in such a way that the matrix A is transferred only once instead of m-times.

$$\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \equiv A \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{pmatrix} \tag{11}$$

By reducing the number of data to be transferred while maintaining the amount of operations to perform, the operational intensity of the kernel is increased, which eventually leads to speed-up in the operation. This speed-up will be bound by the ratio of operational intensities between the `SpMM` and the equivalent `SpMV`.

One of the parameters relevant in the computation of the operational intensities is the number of non-zeros per row that the corresponding sparse matrices have. Table 1 shows the comparison between the classical second-order implementation and the current high-order. This will indeed benefit the higher-order schemes compared to the second-order, thus making them more suitable for this transformation form `SpMV` to `SpMM`.

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
$3D^{tet}$	4	16	2	8	5	17	4	16
$3D^{hex}$	6	36	2	12	7	25	6	24

4 Preliminary results

Some runs have been performed in order to test the performance of the SpMM compared to running simulations with a single right-hand side. These runs were performed under a turbulent planar channel flow simulation at $\mathrm{Re}_\tau = 180$ and a grid of 160^3, with hyperbolic tangent refinement in the y direction with $\gamma = 1.5$. These have been performed using the in-house code TermoFluids Algebraic with hpc^2 as the mathematical engine and Chronos for the solution of the Poisson equation, in 64 cores of a MareNostrum 5 General Purpose Partition node, which led to a load per CPU of 64k cells. All simulations have been run for 1 time unit, running 1, 2, 4, and 8 flow states simultaneously.

Results are shown in Fig. 1, where three different set-ups have been tested: 7 non-zeros, 13 non-zeros and 27 non-zeros per row. It can be seen in that case that the performance obtained improves the bigger the number of non-zeros per row, as the increase in arithmetic intensity is greater, which leads to bigger speed-ups compared to having a fewer number of non-zeros per row.

5 Concluding remarks

In this extended abstract a methodology to extend the order of accuracy for a symmetry-preserving framework in a general geometry is presented and the possibility to use high operational intensity algorithms in order to improve the efficiency of the implementation is depicted, with application in geometries with symmetries, repetitions; or in parallel-in-time set-ups. The use of higher-order schemes in the framework of exploiting repeated matrix structures is benefitial as higher-order discrete operators generate denser sparse matrices, e.g. there is a greater number of non-zeros per row. This will generate a bigger growth of operational intensity when incrementing the number of right-hand sides, thus leading to greater speed-ups in the SpMM kernel and, overall, in the whole iteration and simulation. Moreover, the implementation of this method together with testing in a repeated matrix block structure framework is expected to be presented in the conference.

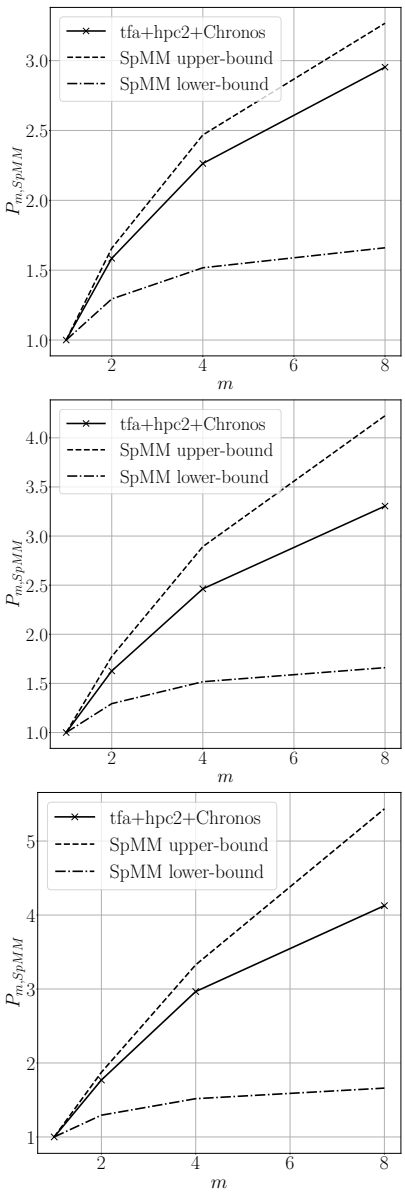

Figure 1: Speed-ups obtained in the SpMM kernel for 7 non-zeros (top), 13 non-zeros (center), and 27 non-zeros (bottom) per row.

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. and D.S. are also supported by the Catalan Agency for Management of University and Research Grants (AGAUR). Calculations were performed on the MareNostrum 5 GCC supercomputer at the BSC. The authors thankfully acknowledge these institutions.

References

Alsalti-Baldellou, À., Álvarez-Farré, X., Colomer, G., Gorobets, A., Pérez-Segarra, C.D., Oliva, A., Trias, F.X. (2024), Lighter and faster simulations on domains with symmetries. *Comput. Fluids*, Vol. 275, 106247

Krasnopolsky, B.I. (2018), An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles, *Comput. Phys. Commun.*, Vol. 229, pp. 8-19

Tosi, R., Núñez, M., Pons-Prats, J., Principe, J., Rossi, R. (2022), On the use of ensemble average techniques to accelerate the uncertainty quantification of CFD predictions in wind egineering, *J. Wind Eng. Ind. Aerodyn.*, Vol. 228, 105105

Trias, F.X., Álvarez-Farré, X., Alsalti-Baldellou, À., Gorobets, A., Oliva, A. (2024), An efficient eigenvalue bounding method: CFL condition revisited, *Comput. Phys. Commun*, Vol. 305, 109351