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 Centre
Technical University of Catalonia

15th Ercoftac Symposium on Engineering Turbulence Modelling and Measurements
Dubrovnik, Croatia
Sept. 23rd, 2025

Introduction

$\boxed{\text{Physical fidelity}}$ $\longleftrightarrow$ $\boxed{\text{Order of accuracy}}$

e.g. conservation, symmetry, ...

MUSCL, ENO, WENO, ...

Introduction

High-order symmetry-preserving discretizations

Physical fidelity $\longleftrightarrow$ Order of accuracy

e.g. conservation, symmetry, ...

MUSCL, ENO, WENO, ...

Introduction

- Given a 1D domain discretized with N cells, a second-order FV discretization leads to...

$$\boxed{\begin{array}{c}\textbf{High-order symmetry-preserving}\\ \boxed{\text{Fidelity}} \longleftrightarrow \boxed{\text{Accuracy}}\end{array}}$$

$$\begin{pmatrix} D_{11} & D_{12} & 0 & \cdots & \cdots & 0 & D_{1N} \\ D_{21} & D_{22} & D_{23} & \cdots & \cdots & 0 & 0 \\ 0 & D_{32} & D_{33} & D_{34} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{N1} & 0 & \cdots & \cdots & 0 & D_{N,N-1} & D_{N,N} \end{pmatrix}$$

Introduction

- Given a 1D domain discretized with N cells, a fourth-order FV discretization leads to...

High-order symmetry-preserving

$$\boxed{\text{Fidelity}} \longleftrightarrow \boxed{\text{Accuracy}}$$

$$\begin{pmatrix} D_{11} & D_{12} & D_{13} & \cdots & \cdots & D_{1,N-1} & D_{1N} \\ D_{21} & D_{22} & D_{23} & D_{24} & \cdots & \cdots & D_{2N} \\ D_{31} & D_{32} & D_{33} & D_{34} & D_{35} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{N1} & D_{N2} & \cdots & \cdots & D_{N,N-2} & D_{N,N-1} & D_{N,N} \end{pmatrix}$$

High-order discretizations lead to **denser** matrices

- For a given mesh, more expensive linear algebra operations.
- Is there any way to exploit this?

Towards high-order symmetry-preserving discretizations

Second-order symmetry-preserving discretizations

The second-order symmetry-preserving discretization of the 1D diffusion operator is constructed as:

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

Interpretation

The second-order approximation of the second derivative can be interpreted as applying a box filter to the second derivative.

Towards high-order symmetry-preserving discretizations

Second-order symmetry-preserving discretizations

The second-order symmetry-preserving discretization of the 1D diffusion operator is constructed as:

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

Interpretation

The second-order approximation of the second derivative can be interpreted as applying a box filter to the second derivative.

Box filter with length h

$$\bar{\phi} = \frac{1}{h}\int_{x-h/2}^{x+h/2} \phi \; dx \qquad \phi = \bar{\phi} + \frac{h^2}{24}\frac{\partial^2\phi}{\partial x^2} + \mathcal{O}(h^4)$$

Towards high-order symmetry-preserving discretizations

Fourth-order symmetry-preserving discretizations

According to Trias et al. [1], this box filter can be applied in such a way that

$$\frac{\partial^2 \phi}{\partial x^2} \approx \overline{\frac{\partial^2 \overline{\phi}}{\partial x^2}} + \overline{\frac{\partial^2 \phi'}{\partial x^2}} + \left(\frac{\partial^2 \overline{\phi}}{\partial x^2}\right)' + \left(\frac{\partial^2 \phi'}{\partial x^2}\right)' + \mathcal{O}(h^4)$$

Extension to multiple dimensions

$$\nabla^2 \phi \approx \overline{\nabla^2 \overline{\phi}} + \overline{\nabla^2 \phi'} + \left(\nabla^2 \overline{\phi}\right)' + \left(\nabla^2 \phi'\right)' + \mathcal{O}(h^4)$$

Extension to discrete formulation

$$\tilde{L} = L + LR + RL + RLR = (I + R)L(I + R)$$

[1] Trias et al., 2024, Symmetry-preserving approximate deconvolutions. On: 9th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2024), Lisbon, Portugal, 3-7 June 2024.

Higher-order symmetry-preserving discretizations

Obtention of M, G, Π, C

Second-order diffusive operator

$$L = MG$$

With this definition, $\tilde{M}$ and $\tilde{G}$ can be obtained from $\tilde{L}$ as:

Fourth-order diffusive operator

$$\tilde{L} = \tilde{M}\tilde{G}$$

- $\tilde{M} = (I + R)M$
- $\tilde{G} = G(I + R)$

Higher-order symmetry-preserving discretizations

Obtention of M, G, Π, C

Second-order diffusive operator

$$L = MG$$

With this definition, $\tilde{M}$ and $\tilde{G}$ can be obtained from $\tilde{L}$ as:

Fourth-order diffusive operator

$$\tilde{L} = \tilde{M}\tilde{G}$$

- $\tilde{M} = (I + R)M$
- $\tilde{G} = G(I + R)$

Second-order convective operator

$$C = MU_s\Pi$$

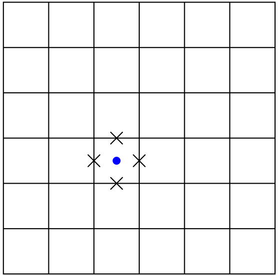
Similarly, $\tilde{C} = (I + R)C(I + R)$. With this definition, $\tilde{M}, \tilde{\Pi}$ can be obtained from $\tilde{C}$ as:

Fourth-order convective operator

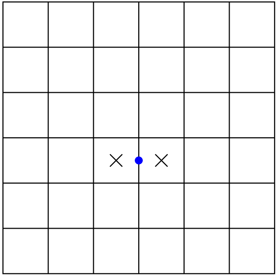
$$\tilde{C} = \tilde{M}U_s\tilde{\Pi}$$

- $\tilde{M} = (I + R)M$
- $\tilde{\Pi} = \Pi(I + R)$

Higher-order symmetry-preserving discretizations

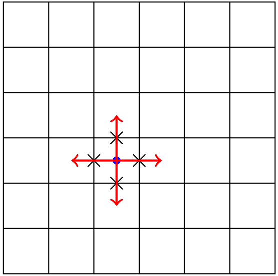


M: 6 entries

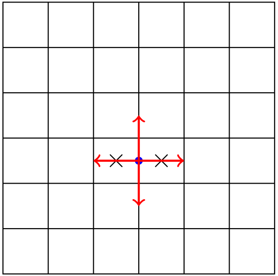


G, Π: 2 entries

Higher-order symmetry-preserving discretizations

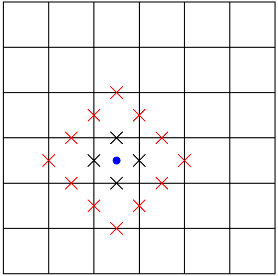


$\tilde{M}$: ? entries

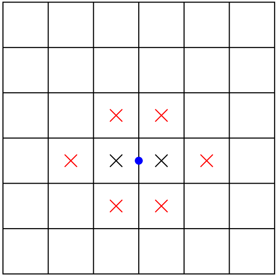


$\tilde{G}, \tilde{\Pi}$: ? entries

Higher-order symmetry-preserving discretizations



$\tilde{M}$: 16 entries



$\tilde{G}, \tilde{\Pi}$: 8 entries

Higher-order symmetry-preserving discretizations

	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	19	3	18
$2D^{quad}$	4	16	2	8	5	25	4	24
$3D^{hex}$	6	36	2	12	7	63	6	62

Application to a projection method

Incompressibility constraint in projection methods

$$L\mathbf{p}_c = M\mathbf{u}_c^p$$

Application to a projection method

Incompressibility constraint in projection methods

$$L\mathbf{p}_c = M\mathbf{u}_c^p$$

Extending to fourth-order symmetry-preserving discretizations

$$\tilde{L}\tilde{\mathbf{p}}_c = \tilde{M}\tilde{\mathbf{u}}_c^p$$

where $\tilde{\mathbf{x}}$ approximates $\mathbf{x}$ with fourth-order accuracy: $\mathbf{x} = (I + R)\tilde{\mathbf{x}}$.

Application to a projection method

Incompressibility constraint in projection methods

$$L\mathbf{p}_c = M\mathbf{u}_c^p$$

Extending to fourth-order symmetry-preserving discretizations

$$\tilde{L}\tilde{\mathbf{p}}_c = \tilde{M}\tilde{\mathbf{u}}_c^p$$

where $\tilde{\mathbf{x}}$ approximates $\mathbf{x}$ with fourth-order accuracy: $\mathbf{x} = (I + R)\tilde{\mathbf{x}}$.

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

Application to a projection method

Incompressibility constraint in projection methods

$$L\mathbf{p}_c = M\mathbf{u}_c^p$$

Extending to fourth-order symmetry-preserving discretizations

$$\tilde{L}\tilde{\mathbf{p}}_c = \tilde{M}\tilde{\mathbf{u}}_c^p$$

where $\tilde{\mathbf{x}}$ approximates $\mathbf{x}$ with fourth-order accuracy: $\mathbf{x} = (I + R)\tilde{\mathbf{x}}$.

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

Application to a projection method

Incompressibility constraint in projection methods

$$L\mathbf{p}_c = M\mathbf{u}_c^p$$

Extending to fourth-order symmetry-preserving discretizations

$$\tilde{L}\tilde{\mathbf{p}}_c = \tilde{M}\tilde{\mathbf{u}}_c^p$$

where $\tilde{\mathbf{x}}$ approximates $\mathbf{x}$ with fourth-order accuracy: $\mathbf{x} = (I + R)\tilde{\mathbf{x}}$.

$$L\mathbf{p}_c = M\tilde{\mathbf{u}}_c^p$$

Application to a projection method

Incompressibility constraint in projection methods

$$L\mathbf{p}_c = M\mathbf{u}_c^p$$

Extending to fourth-order symmetry-preserving discretizations

$$\tilde{L}\tilde{\mathbf{p}}_c = \tilde{M}\tilde{\mathbf{u}}_c^p$$

where $\tilde{\mathbf{x}}$ approximates $\mathbf{x}$ with fourth-order accuracy: $\mathbf{x} = (I + R)\tilde{\mathbf{x}}$.

$$L\mathbf{p}_c = M\tilde{\mathbf{u}}_c^p$$

Outcome:

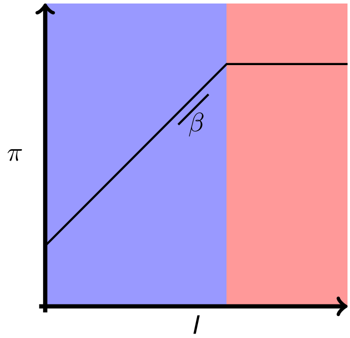
The Poisson equation does not change when using fourth-order symmetry-preserving discretizations.

Exploiting repeated matrix-block structures

Roofline model[2]

$$\pi = \beta I$$

- π: performance
- β: memory bandwidth
- I: arithmetic intensity, flops per byte



$$\underbrace{A\mathbf{x} \underbrace{=}_{A=I_2\otimes\tilde{A}} \begin{pmatrix} \tilde{A} & 0 \\ 0 & \tilde{A} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}}_{\texttt{SpMV}}$$

[2]Williams et al., 2009, Roofline: an insightful visual performance model for multicore architectures, Communications of the ACM 52

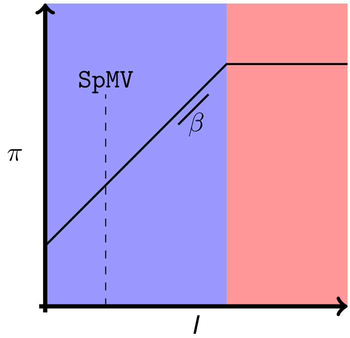
Exploiting repeated matrix-block structures

Roofline model[2]

$$\pi = \beta I$$

- π: performance
- β: memory bandwidth
- I: arithmetic intensity, flops per byte

$$\underbrace{A\mathbf{x} \underbrace{=}_{A=I_2\otimes\tilde{A}} \begin{pmatrix} \tilde{A} & 0 \\ 0 & \tilde{A} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}}_{\text{SpMV}}$$



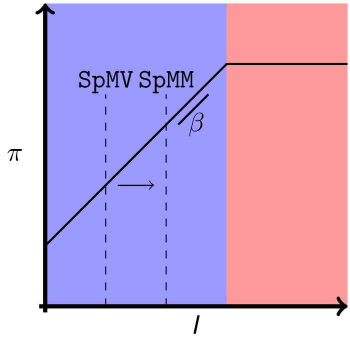
[2] Williams et al., 2009, Roofline: an insightful visual performance model for multicore architectures, Communications of the ACM 52

Exploiting repeated matrix-block structures

Roofline model[2]

$$\pi = \beta I$$

- π: performance
- β: memory bandwidth
- I: arithmetic intensity, flops per byte



$$\underbrace{A\mathbf{x} \underbrace{=}_{A = I_2 \otimes \tilde{A}} \begin{pmatrix} \tilde{A} & 0 \\ 0 & \tilde{A} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}}_{\text{SpMV}} \equiv \underbrace{\tilde{A} \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{pmatrix}}_{\text{SpMM}}$$

[2] Williams et al., 2009, Roofline: an insightful visual performance model for multicore architectures, Communications of the ACM 52

Exploiting repeated matrix-block structures

Defining the optimal (upper-bound) speed-up as:

$$P_m = \frac{\pi_m}{\pi_1} = \frac{I_{\mathtt{SpMM}}(m)}{I_{\mathtt{SpMV}}}$$

By definition,

$$I_{\mathtt{SpMM}}(m) = \frac{(2\mathrm{nnz}(A)+1)m}{12\mathrm{nnz}(A)+4(n_r/m+1)+8(n_r+n_c+m)},$$

Then,

$$P_m = \frac{m[3(\mathrm{nnz}(A)+n_r+1)+2n_c]}{3\mathrm{nnz}(A)+(1+2m)(n_r+1)+2mn_c}.$$

Exploiting repeated matrix-block structures

Defining the optimal (upper-bound) speed-up as:

$$P_m = \frac{\pi_m}{\pi_1} = \frac{I_{\mathtt{SpMM}}(m)}{I_{\mathtt{SpMV}}}$$

By definition,

$$I_{\mathtt{SpMM}}(m) = \frac{(2\mathrm{nnz}(A)+1)m}{12\mathrm{nnz}(A)+4(n_r/m+1)+8(n_r+n_c+m)},$$

Then,

$$P_m = \frac{m[3(\mathrm{nnz}(A)+n_r+1)+2n_c]}{3\mathrm{nnz}(A)+(1+2m)(n_r+1)+2mn_c}.$$

What happens when $m \to \infty$?

Assuming $n_r \gg 1$ and $n_c = n_r$,

$$\lim_{m\to\infty} P_m \approx \frac{12\mathrm{nnz}(A)/n_r + 20}{16}$$

Exploiting repeated matrix-block structures

Defining the optimal (upper-bound) speed-up as:

$$P_m = \frac{\pi_m}{\pi_1} = \frac{I_{\mathtt{SpMM}}(m)}{I_{\mathtt{SpMV}}}$$

By definition,

$$I_{\mathtt{SpMM}}(m) = \frac{(2\mathrm{nnz}(A)+1)m}{12\mathrm{nnz}(A) + 4(n_r/m+1) + 8(n_r+n_c+m)},$$

Then,

$$P_m = \frac{m[3(\mathrm{nnz}(A)+n_r+1)+2n_c]}{3\mathrm{nnz}(A) + (1+2m)(n_r+1) + 2mn_c}.$$

What happens when $m \to \infty$?

Assuming $n_r \gg 1$ and $n_c = n_r$,

$$\lim_{m\to\infty} P_m \approx \frac{12\mathrm{nnz}(A)/n_r + 20}{16}$$

- Upper-bound speed-up depends on density of the matrix.

	M	$\tilde{M}$	G, Π	$\tilde{G}, \tilde{\Pi}$
$2D^{quad}$	4	16	2	8
$3D^{hex}$	6	36	2	12

Exploiting repeated matrix-block structures

Reformulating the Navier-Stokes equations

Semi-discrete incompressible Navier-Stokes equations

$$M\mathbf{u}_s = \mathbf{0}_s, \qquad \Omega \frac{d\mathbf{u}_c}{dt} + MU_s\Pi\mathbf{u}_c = D\mathbf{u}_c - \Omega G_c\mathbf{p}_c$$

[3] Alsalti-Baldellou et al., 2024, A multigrid reduction framework for domains with symmetries, SIAM Journal on Scientific Computing 46

[4] Plana-Riu et al., 2025, Exploting repeated matrix-block strucutres for more efficient CFD on modern supercomputers, arXiv:2508.06710 [preprint]

Exploiting repeated matrix-block structures

Reformulating the Navier-Stokes equations

Semi-discrete incompressible Navier-Stokes equations

$$M\mathbf{u}_s = \mathbf{0}_s, \qquad \Omega\frac{d\mathbf{u}_c}{dt} + MU_s\Pi\mathbf{u}_c = D\mathbf{u}_c - \Omega G_c\mathbf{p}_c$$

Let $\mathbf{U}_c = [\mathbf{u}_{c,1}, \mathbf{u}_{c,2}, \dots, \mathbf{u}_{c,m}]$ and $\mathbf{P}_c = [\mathbf{p}_{c,1}, \mathbf{p}_{c,2}, \dots, \mathbf{p}_{c,m}]$

(Block-structure) semi-discrete incompressible Navier-Stokes equations

$$M\mathbf{U}_s = \mathbf{0}_s, \qquad \Omega\frac{d\mathbf{U}_c}{dt} + MU_s\Pi\mathbf{U}_c = D\mathbf{U}_c - \Omega G_c\mathbf{P}_c$$

e.g. domains with symmetries or repeated geometries[3], parallel-in-time[4] simulations, etc.

[3] Alsalti-Baldellou et al., 2024, A multigrid reduction framework for domains with symmetries, SIAM Journal on Scientific Computing 46

[4] Plana-Riu et al., 2025, Exploting repeated matrix-block strucutres for more efficient CFD on modern supercomputers, arXiv:2508.06710 [preprint]

Exploiting repeated matrix-block structures

Ensemble averaging of multiple flow states

Required condition[5]

Ergodicity: time average $=$ Ensemble average

[5] Tosi et al., 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

Exploiting repeated matrix-block structures

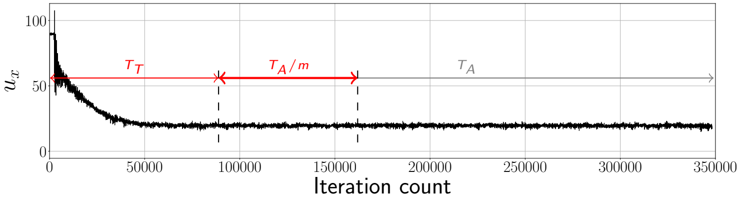
Ensemble averaging of multiple flow states

Required condition[5]

Ergodicity: time average = Ensemble average

Let ϕ be the variable of interest, then:

$$\langle\phi\rangle = \frac{1}{T_A}\int_{T_T}^{T_T+T_A} \phi \; dt \equiv \frac{1}{m}\sum_{i=1}^{m} \frac{1}{T_A/m}\int_{T_T}^{T_T+T_A/m} \phi_i \; dt$$



[5] Tosi et al., 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

Numerical experiments

Turbulent planar channel flow: numerical setup

- Domain: $L_x \times L_y \times L_z = 4\pi\delta \times 2\delta \times 4\pi\delta/3$
- Grid: $N_x \times N_y \times N_z = 128 \times 128 \times 128$ (2.1 million nodes)
- $Re_\tau = u_\tau \delta/\nu = 180$
- Time integration: RK3 + `AlgEigCD`
- Initial condition: parabolic profile + random perturbation
- Boundary conditions: periodic in x and z, no-slip in y
- Code: `tfa+hpc2`
- Discretization strategies:
    - Second-order symmetry-preserving (`SP2`)
    - Fourth-order symmetry-preserving (`SP4`)

Numerical experiments

Turbulent planar channel flow: numerical setup

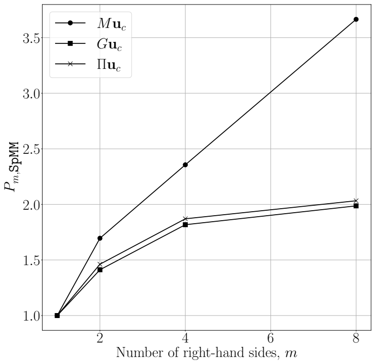
- Domain: $L_x \times L_y \times L_z = 4\pi\delta \times 2\delta \times 4\pi\delta/3$
- Grid: $N_x \times N_y \times N_z = 128 \times 128 \times 128$ (2.1 million nodes)
- $Re_\tau = u_\tau \delta/\nu = 180$
- Time integration: RK3 + `AlgEigCD`
- Initial condition: parabolic profile + random perturbation
- Boundary conditions: periodic in x and z, no-slip in y
- Code: `tfa+hpc2`
- Discretization strategies:
  - Second-order symmetry-preserving (`SP2`)
  - Fourth-order symmetry-preserving (`SP4`)

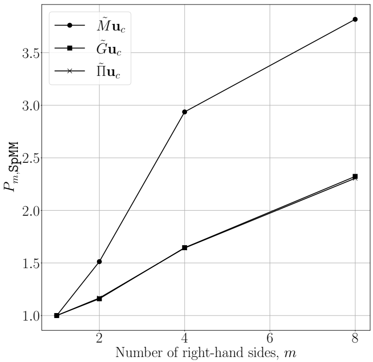
Objective of the test

Assess the speed-up improvements for the `SP4` discretization compared to `SP2` when exploiting repeated matrix-block structures.

Numerical experiments

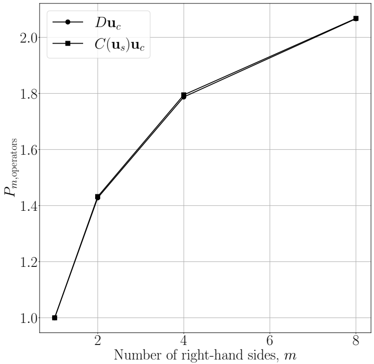
Turbulent planar channel flow: M, G, Π results

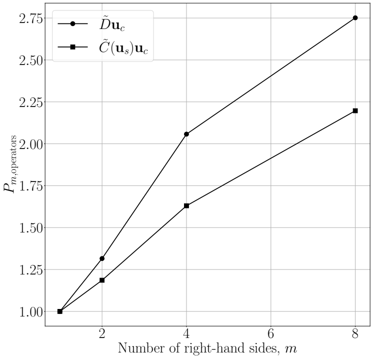




Numerical experiments

Turbulent planar channel flow: C, D results



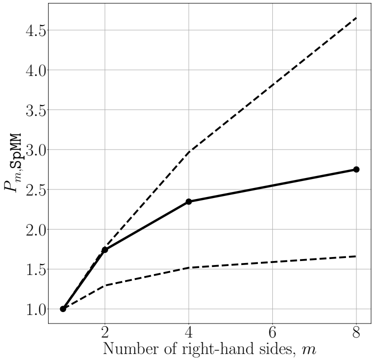


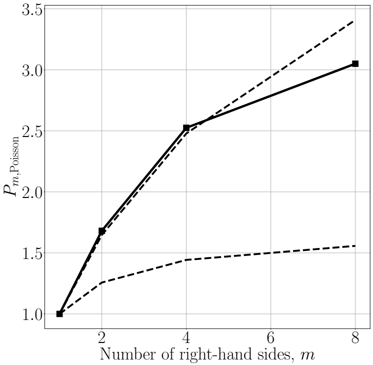
Numerical experiments

Turbulent planar channel flow: speed-up in the Poisson equation

Reminder

Using SP4 does not affect the definition of the Poisson equation





Conclusions

- High-order symmetry-preserving discretizations for collocated grids are presented.
- Its construction does not require addition operators to those of a second-order discretization.
- Higher-order discretizations lead to denser matrices.
- Additional costs of higher-order discretizations can be mitigated by exploiting repeated matrix-block structures.
- The diffusive operator obtains a 30% improvement in performance for $m = 8$ (compared to $\mathtt{SP2}$).
- The convective operator obtains a 5% improvement in performance for $m = 8$ (compared to $\mathtt{SP2}$).
- Repeated matrix-block structures can be exploited for the Poisson equation, with great speed-ups.

Acknowledgements

The work presented in this conference 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*.