Centre Tecnològic de Transferència de Calor
UNIVERSITAT POLITÈCNICA DE CATALUNYA

DNS/LES using a minimal set of algebraic kernels

F.Xavier Trias[1], Xavier Álvarez-Farré[1], Àdel Alsalti-Baldellou[1], Andrey Gorobets[2], Assensi Oliva[1]

[1]Heat and Mass Transfer Technological Center, Technical University of Catalonia
[2]Who cares?

CTTC UPC
Centre Tecnològic de Transferència de Calor
UNIVERSITAT POLITÈCNICA DE CATALUNYA

DNS/LES using a minimal set of algebraic kernels: challenges and opportunities

F.Xavier Trias[1], Xavier Álvarez-Farré[1], Àdel Alsalti-Baldellou[1], Andrey Gorobets[2], Assensi Oliva[1]

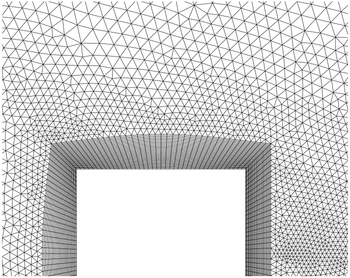
[1]Heat and Mass Transfer Technological Center, Technical University of Catalonia
[2]Who cares?

Contents

Motivation

Research question #1:

- Can we construct numerical discretizations of the Navier-Stokes equations suitable for **complex geometries**, such that the **symmetry properties** are exactly preserved?



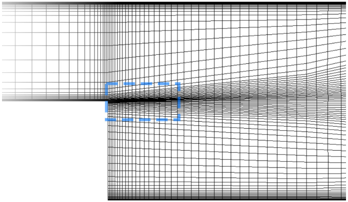
DNS[1] of the turbulent flow around a square cylinder at $Re = 22000$

[1]F.X.Trias, A.Gorobets, A.Oliva. *Turbulent flow around a square cylinder at Reynolds number 22000: a DNS study*, **Computers&Fluids**, 123:87-98, 2015.

Motivation

Research question #1:

- Can we construct numerical discretizations of the Navier-Stokes equations suitable for **complex geometries**, such that the **symmetry properties** are exactly preserved?



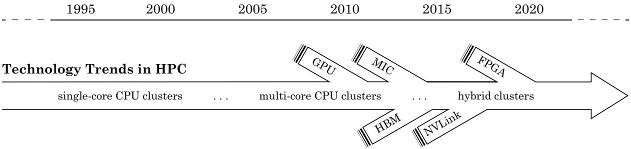
DNS[2] of backward-facing step at $Re_\tau = 395$ and expansion ratio 2

[2] A.Pont-Vílchez, F.X.Trias, A.Gorobets, A.Oliva. *DNS of Backward-Facing Step flow at $Re_\tau = 395$ and expansion ratio 2*. **Journal of Fluid Mechanics**, 863:341-363, 2019.

Motivation

Research question #2:

- How can we develop **portable** and **efficient** CFD codes for large-scale simulations on modern supercomputers?



[3] X.Álvarez, A.Gorobets, F.X.Trias. *A hierarchical parallel implementation for heterogeneous computing. Application to algebra-based CFD simulations on hybrid supercomputers*. **Computers & Fluids**, 214:104768, 2021.

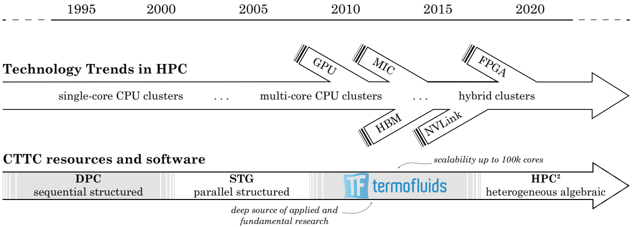
[4] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

[5] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Motivation

Research question #2:

- How can we develop **portable** and **efficient** CFD codes for large-scale simulations on modern supercomputers?



HPC²: portable, algebra-based framework for heterogeneous computing is being developed[3]. Traditional stencil-based data and sweeps are replaced by algebraic structures (sparse matrices and vectors) and kernels. SpMM-based strategies to increase the arithmetic intensity are presented in this conference[4,5].

[3] X.Álvarez, A.Gorobets, F.X.Trias. *A hierarchical parallel implementation for heterogeneous computing. Application to algebra-based CFD simulations on hybrid supercomputers*. **Computers & Fluids**, 214:104768, 2021.

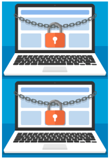
[4] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

[5] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Motivation

Frequently used general purpose CFD codes:

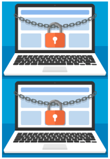
- STAR-CCM+
- ANSYS-FLUENT
- Code-Saturne
- OpenFOAM



Motivation

Frequently used general purpose CFD codes:

- STAR-CCM+
- ANSYS-FLUENT
- Code-Saturne
- OpenFOAM

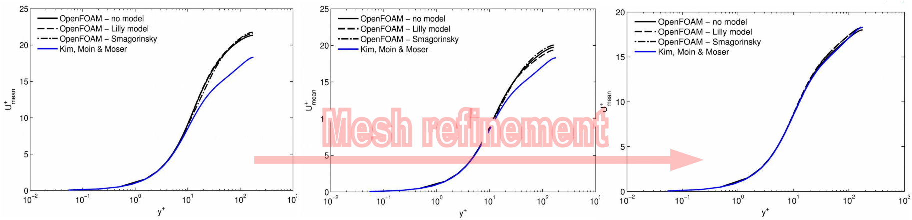


Main common characteristics of LES in such codes:

- **Unstructured finite volume** method, **collocated** grid
- Second-order spatial and temporal discretisation
- Eddy-viscosity type LES models

Motivation

OpenFOAM® LES[6] results of a turbulent channel for at $Re_\tau = 180$

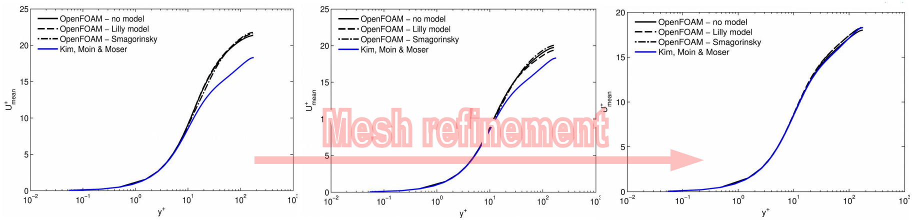


13×76×20	19×78×28	38×78×57
$\Delta x^+=90, \Delta y^+_{wall}=0.5, \Delta z^+=30$	$\Delta x^+=60, \Delta y^+_{wall}=0.5, \Delta z^+=20$	$\Delta x^+=30, \Delta y^+_{wall}=0.5, \Delta z^+=10$

[6] E.M.J.Komen, L.H.Camilo, A.Shams, B.J.Geurts, B.Koren. *A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows*, **Journal of Computational Physics**, 345, 565-595, 2017.

Motivation

Open∇FOAM® LES[6]results of a turbulent channel for at $Re_{\tau} = 180$



$13 \times 76 \times 20$ — $\Delta x^+ = 90, \Delta y^+_{wall} = 0.5, \Delta z^+ = 30$

$19 \times 78 \times 28$ — $\Delta x^+ = 60, \Delta y^+_{wall} = 0.5, \Delta z^+ = 20$

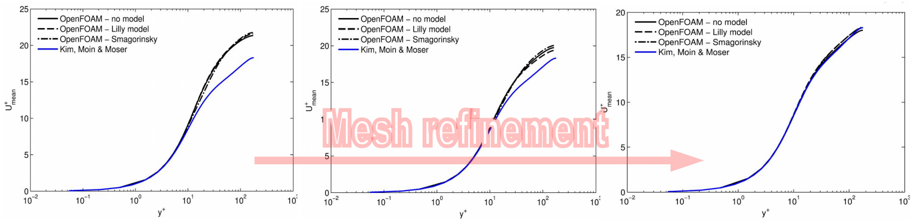
$38 \times 78 \times 57$ — $\Delta x^+ = 30, \Delta y^+_{wall} = 0.5, \Delta z^+ = 10$

- Are LES results are merit of the SGS model?

[6] E.M.J.Komen, L.H.Camilo, A.Shams, B.J.Geurts, B.Koren. *A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows*, **Journal of Computational Physics**, 345, 565-595, 2017.

Motivation

Open∇FOAM® LES[6] results of a turbulent channel for at $Re_\tau = 180$



$13 \times 76 \times 20$ — $\Delta x^+ = 90, \Delta y^+_{wall} = 0.5, \Delta z^+ = 30$

$19 \times 78 \times 28$ — $\Delta x^+ = 60, \Delta y^+_{wall} = 0.5, \Delta z^+ = 20$

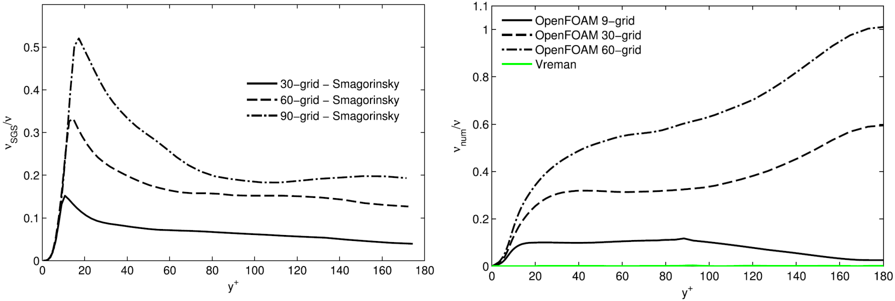
$38 \times 78 \times 57$ — $\Delta x^+ = 30, \Delta y^+_{wall} = 0.5, \Delta z^+ = 10$

- Are LES results are merit of the SGS model? Apparently **NOT**!!! ✗

[6] E.M.J.Komen, L.H.Camilo, A.Shams, B.J.Geurts, B.Koren. *A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows*, **Journal of Computational Physics**, 345, 565-595, 2017.

Motivation

OpenFOAM® LES[7] results of a turbulent channel for at $Re_\tau = 180$

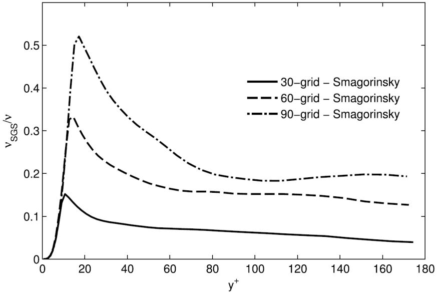


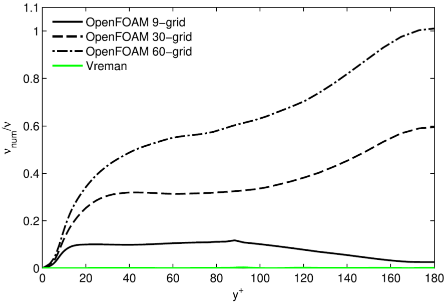
$$\nu_{num} \neq 0$$

[7] E.M.J.Komen, L.H.Camilo, A.Shams, B.J.Geurts, B.Koren. *A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows*, **Journal of Computational Physics**, 345, 565-595, 2017.

Motivation

Open∇FOAM® LES[7] results of a turbulent channel for at $Re_\tau = 180$





$$\nu_{SGS} < \nu_{num} \neq 0$$

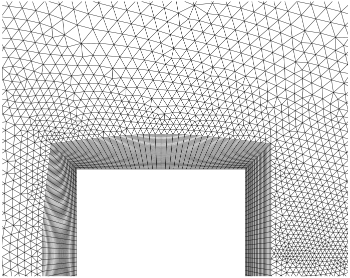
---

[7] E.M.J.Komen, L.H.Camilo, A.Shams, B.J.Geurts, B.Koren. *A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows*, **Journal of Computational Physics**, 345, 565-595, 2017.

Motivation

Research question #1:

- Can we construct numerical discretizations of the Navier-Stokes equations suitable for **complex geometries**, such that the **symmetry properties** are exactly preserved?



DNS[1] of the turbulent flow around a square cylinder at $Re = 22000$

[1] F.X.Trias, A.Gorobets, A.Oliva. *Turbulent flow around a square cylinder at Reynolds number 22000: a DNS study*, **Computers&Fluids**, 123:87-98, 2015.

Symmetry-preserving discretization

Continuous

$$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$$

$$\nabla \cdot \boldsymbol{u} = 0$$

Symmetry-preserving discretization

Continuous

$$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$$

$$\nabla \cdot \boldsymbol{u} = 0$$

Discrete

$$\Omega \frac{d\boldsymbol{u}_h}{dt} + \mathsf{C}(\boldsymbol{u}_h)\, \boldsymbol{u}_h = \mathsf{D}\boldsymbol{u}_h - \mathsf{G}\boldsymbol{p}_h$$

$$\mathsf{M}\boldsymbol{u}_h = \boldsymbol{0}_h$$

Symmetry-preserving discretization

Continuous	Discrete
$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$	$\Omega \frac{d\boldsymbol{u}_h}{dt} + \mathsf{C}(\boldsymbol{u}_h)\,\boldsymbol{u}_h = \mathsf{D}\boldsymbol{u}_h - \mathsf{G}\boldsymbol{p}_h$
$\nabla \cdot \boldsymbol{u} = 0$	$\mathsf{M}\boldsymbol{u}_h = \boldsymbol{0}_h$
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \int_\Omega \boldsymbol{a}\boldsymbol{b}\, d\Omega$	$\langle \boldsymbol{a}_h, \boldsymbol{b}_h \rangle_h = \boldsymbol{a}_h^T \Omega \boldsymbol{b}_h$

Symmetry-preserving discretization

Continuous	Discrete
$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$	$\Omega \frac{d \boldsymbol{u}_h}{dt} + \mathsf{C}(\boldsymbol{u}_h)\, \boldsymbol{u}_h = \mathsf{D} \boldsymbol{u}_h - \mathsf{G} \boldsymbol{p}_h$
$\nabla \cdot \boldsymbol{u} = 0$	$\mathsf{M} \boldsymbol{u}_h = \boldsymbol{0}_h$
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \int_\Omega \boldsymbol{a} \boldsymbol{b} d\Omega$	$\langle \boldsymbol{a}_h, \boldsymbol{b}_h \rangle_h = \boldsymbol{a}_h^T \Omega \boldsymbol{b}_h$
$\langle C(\boldsymbol{u}, \varphi_1), \varphi_2 \rangle = - \langle C(\boldsymbol{u}, \varphi_2), \varphi_1 \rangle$	$\mathsf{C}(\boldsymbol{u}_h) = -\mathsf{C}^T(\boldsymbol{u}_h)$

Symmetry-preserving discretization

Continuous	Discrete
$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$	$\Omega \frac{d\boldsymbol{u}_h}{dt} + \mathsf{C}(\boldsymbol{u}_h)\boldsymbol{u}_h = \mathsf{D}\boldsymbol{u}_h - \mathsf{G}\boldsymbol{p}_h$
$\nabla \cdot \boldsymbol{u} = 0$	$\mathsf{M}\boldsymbol{u}_h = \boldsymbol{0}_h$
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \int_\Omega \boldsymbol{a}\boldsymbol{b} d\Omega$	$\langle \boldsymbol{a}_h, \boldsymbol{b}_h \rangle_h = \boldsymbol{a}_h^T \Omega \boldsymbol{b}_h$
$\langle C(\boldsymbol{u}, \varphi_1), \varphi_2 \rangle = -\langle C(\boldsymbol{u}, \varphi_2), \varphi_1 \rangle$	$\mathsf{C}(\boldsymbol{u}_h) = -\mathsf{C}^T(\boldsymbol{u}_h)$
$\langle \nabla \cdot \boldsymbol{a}, \varphi \rangle = -\langle \boldsymbol{a}, \nabla \varphi \rangle$	$\Omega \mathsf{G} = -\mathsf{M}^T$

Symmetry-preserving discretization

Continuous	Discrete
$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$	$\Omega \frac{d\boldsymbol{u}_h}{dt} + \mathsf{C}(\boldsymbol{u}_h)\boldsymbol{u}_h = \mathsf{D}\boldsymbol{u}_h - \mathsf{G}\boldsymbol{p}_h$
$\nabla \cdot \boldsymbol{u} = 0$	$\mathsf{M}\boldsymbol{u}_h = \boldsymbol{0}_h$
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \int_\Omega \boldsymbol{a}\boldsymbol{b} d\Omega$	$\langle \boldsymbol{a}_h, \boldsymbol{b}_h \rangle_h = \boldsymbol{a}_h^T \Omega \boldsymbol{b}_h$
$\langle C(\boldsymbol{u}, \varphi_1), \varphi_2 \rangle = - \langle C(\boldsymbol{u}, \varphi_2), \varphi_1 \rangle$	$\mathsf{C}(\boldsymbol{u}_h) = -\mathsf{C}^T(\boldsymbol{u}_h)$
$\langle \nabla \cdot \boldsymbol{a}, \varphi \rangle = - \langle \boldsymbol{a}, \nabla \varphi \rangle$	$\Omega \mathsf{G} = -\mathsf{M}^T$
$\left\langle \nabla^2 \boldsymbol{a}, \boldsymbol{b} \right\rangle = \left\langle \boldsymbol{a}, \nabla^2 \boldsymbol{b} \right\rangle$	$\mathsf{D} = \mathsf{D}^T \quad def -$

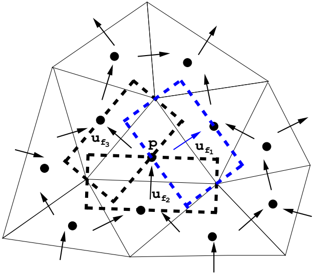
Why collocated arrangements are so popular?

- STAR-CCM+
- ANSYS-FLUENT
- Code-Saturne
- OpenFOAM

$$\boxed{\Omega_s \frac{d\boldsymbol{u}_s}{dt} + \mathsf{C}\left(\boldsymbol{u}_s\right)\boldsymbol{u}_s = \mathsf{D}\boldsymbol{u}_s - \mathsf{G}\boldsymbol{p}_c; \quad \mathsf{M}\boldsymbol{u}_s = \boldsymbol{0}_c}$$

In staggered meshes

- p-$\boldsymbol{u}_s$ coupling is naturally solved ✓
- $\mathsf{C}\left(\boldsymbol{u}_s\right)$ and D difficult to discretize ✗



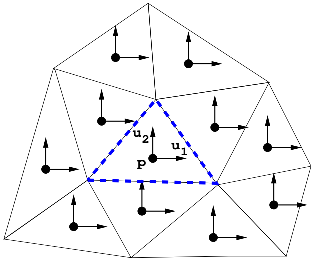
Why collocated arrangements are so popular?

- STAR-CCM+
- ANSYS-FLUENT
- Code-Saturne
- OpenFOAM

$$\Omega_c \frac{d\boldsymbol{u}_c}{dt} + \mathsf{C}\left(\boldsymbol{u}_s\right)\boldsymbol{u}_c = \mathsf{D}\boldsymbol{u}_c - \mathsf{G}_c\boldsymbol{p}_c; \quad \mathsf{M}_c\boldsymbol{u}_c = \boldsymbol{0}_c$$

In <u>collocated meshes</u>

- p-$\boldsymbol{u}_c$ coupling is cumbersome ✗
- $\mathsf{C}\left(\boldsymbol{u}_s\right)$ and D easy to discretize ✓
- Cheaper, less memory,... ✓

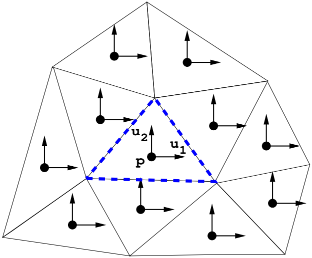


Why collocated arrangements are so popular?

Everything is easy except the pressure-velocity coupling...

- STAR-CCM+ CD-adapco SIEMENS
- ANSYS-FLUENT ANSYS FLUENT
- Code-Saturne CODE SATURNE eDF GPL
- OpenFOAM OpenFOAM® GPL v3 Free Software

$$\Omega_c \frac{d\boldsymbol{u}_c}{dt} + \mathsf{C}\left(\boldsymbol{u}_s\right)\boldsymbol{u}_c = \mathsf{D}\boldsymbol{u}_c - \mathsf{G}_c\boldsymbol{p}_c; \quad \mathsf{M}_c\boldsymbol{u}_c = \boldsymbol{0}_c$$



In collocated meshes

- p-$\boldsymbol{u}_c$ coupling is cumbersome ✗
- $\mathsf{C}\left(\boldsymbol{u}_s\right)$ and D easy to discretize ✓
- Cheaper, less memory,... ✓

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

- Mass: $M\Gamma_{c\to s}\boldsymbol{u}_c = M\Gamma_{c\to s}\boldsymbol{u}_c - \mathsf{L}_c\mathsf{L}^{-1} M\Gamma_{c\to s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c \left(\mathsf{L} - \mathsf{L}_c\right) \boldsymbol{p}_c \neq 0$ ✗

[8] F.X.Trias, O.Lehmkuhl, A.Oliva, C.D.Pérez-Segarra, R.W.C.P.Verstappen. *Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids*, **Journal of Computational Physics**, 258 (1): 246-267, 2014.

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

- Mass: $M\Gamma_{c\to s}\boldsymbol{u}_c = M\Gamma_{c\to s}\boldsymbol{u}_c - \boxed{L_c L^{-1}} M\Gamma_{c\to s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c \boxed{(L - L_c)} \boldsymbol{p}_c \neq 0$ ✗

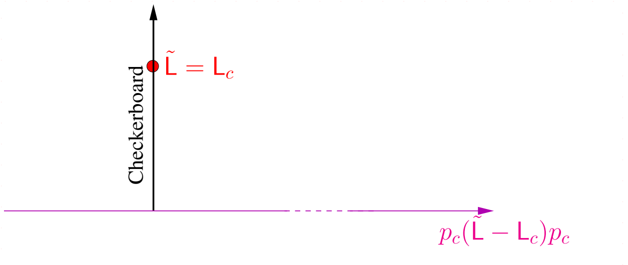
[8] F.X.Trias, O.Lehmkuhl, A.Oliva, C.D.Pérez-Segarra, R.W.C.P.Verstappen. *Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids*, **Journal of Computational Physics**, 258 (1): 246-267, 2014.

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

- Mass: $\mathsf{M}\Gamma_{c\to s}\boldsymbol{u}_c = \mathsf{M}\Gamma_{c\to s}\boldsymbol{u}_c - \boxed{\mathsf{L}_c\mathsf{L}^{-1}}\mathsf{M}\Gamma_{c\to s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c\boxed{(\mathsf{L}-\mathsf{L}_c)}\boldsymbol{p}_c \neq 0$ ✗



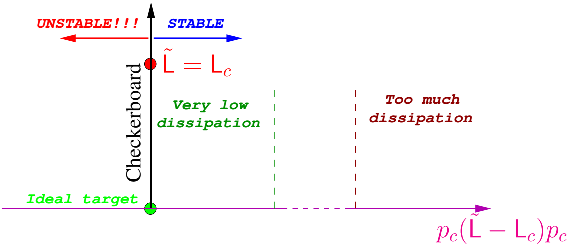
[8] F.X.Trias, O.Lehmkuhl, A.Oliva, C.D.Pérez-Segarra, R.W.C.P.Verstappen. *Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids*, **Journal of Computational Physics**, 258 (1): 246-267, 2014.

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

- Mass: $\mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c = \mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c - \boxed{\mathsf{L}_c\mathsf{L}^{-1}}\mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c\boxed{(\mathsf{L}-\mathsf{L}_c)}\boldsymbol{p}_c \neq 0$ ✗



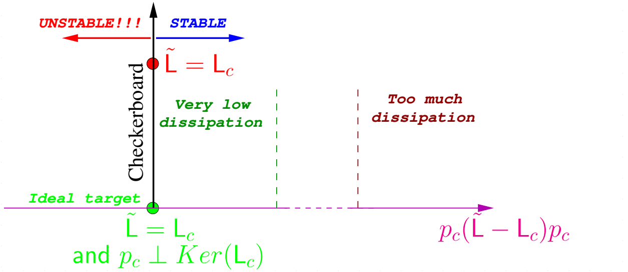
[8] F.X.Trias, O.Lehmkuhl, A.Oliva, C.D.Pérez-Segarra, R.W.C.P.Verstappen. *Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids*, **Journal of Computational Physics**, 258 (1): 246-267, 2014.

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

- Mass: $\mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c = \mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c - \boxed{\mathsf{L}_c\mathsf{L}^{-1}}\mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c\boxed{(\mathsf{L}-\mathsf{L}_c)}\boldsymbol{p}_c \neq 0$ ✗



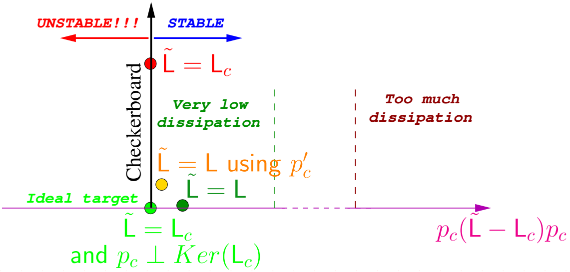
[8]Shashank, J.Larsson, G.Iaccarino. *A co-located incompressible Navier-Stokes solver with exact mass, momentum and kinetic energy conservation in the inviscid limit*, **Journal of Computational Physics**, 229: 4425-4430,2010.

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

- Mass: $\mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c = \mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c - \boxed{\mathsf{L}_c\mathsf{L}^{-1}}\mathsf{M}\Gamma_{c\rightarrow s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c\boxed{(\mathsf{L}-\mathsf{L}_c)}\boldsymbol{p}_c \neq 0$ ✗



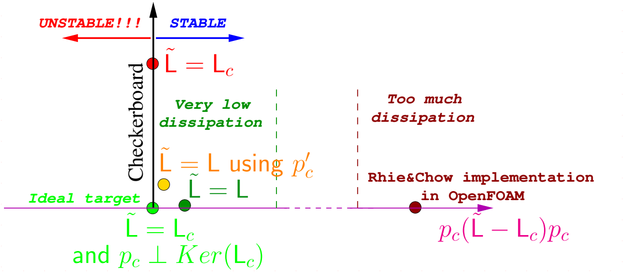
[8] F.X.Trias, O.Lehmkuhl, A.Oliva, C.D.Pérez-Segarra, R.W.C.P.Verstappen. *Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids*, **Journal of Computational Physics**, 258 (1): 246-267, 2014.

Pressure-velocity coupling on collocated grids

A vicious circle that cannot be broken...

In summary[8]:

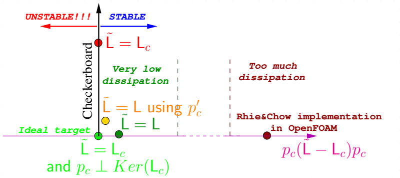
- Mass: $\mathsf{M}\Gamma_{c\to s}\boldsymbol{u}_c = \mathsf{M}\Gamma_{c\to s}\boldsymbol{u}_c - \boxed{\mathsf{L}_c\mathsf{L}^{-1}}\mathsf{M}\Gamma_{c\to s}\boldsymbol{u}_c \approx \mathbf{0}_c$ ✗
- Energy: $\boldsymbol{p}_c\boxed{(\mathsf{L}-\mathsf{L}_c)}\boldsymbol{p}_c \neq 0$ ✗

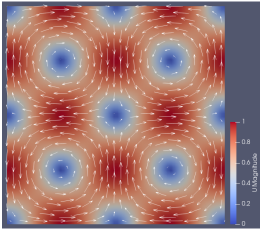


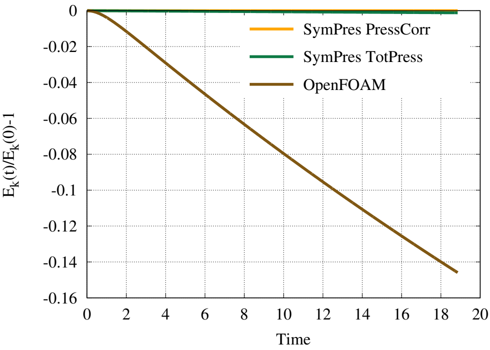
[8] E.Komen, J.A.Hopman, E.M.A.Frederix, F.X.Trias, R.W.C.P.Verstappen. "A symmetry-preserving second-order time-accurate PISO-based method". **Computers & Fluids**, 225:104979, 2021.

Pressure-velocity coupling on collocated grids

A vicious circle that ~~cannot be broken~~ can almost be broken...





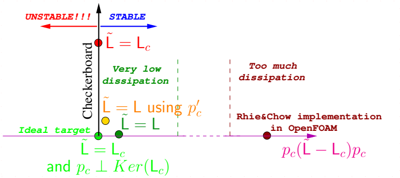


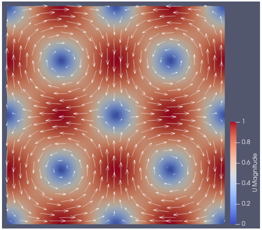
Results for an inviscid Taylor-Green vortex[9]

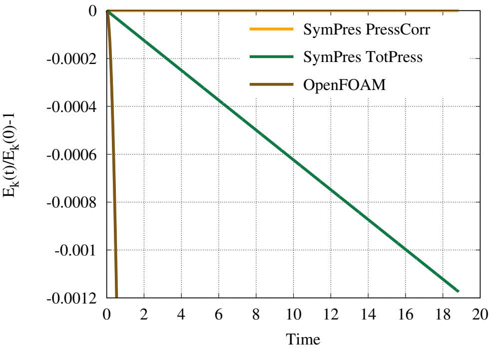
[9] E.Komen, J.A.Hopman, E.M.A.Frederix, F.X.Trias, R.W.C.P.Verstappen. "A symmetry-preserving second-order time-accurate PISO-based method". **Computers & Fluids**, 225:104979, 2021.

Pressure-velocity coupling on collocated grids

A vicious circle that ~~cannot be broken~~ can almost be broken...







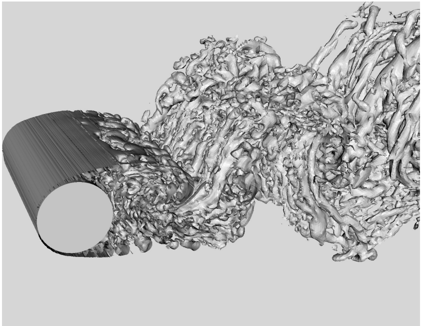
Results for an inviscid Taylor-Green vortex[9]

[9] E.Komen, J.A.Hopman, E.M.A.Frederix, F.X.Trias, R.W.C.P.Verstappen. "A symmetry-preserving second-order time-accurate PISO-based method". **Computers & Fluids**, 225:104979, 2021.

Pressure-velocity coupling on collocated grids

Examples of simulations

Despite these inherent limitations, symmetry-preserving collocated formulation has been successfully used for DNS/LES simulations[10]:

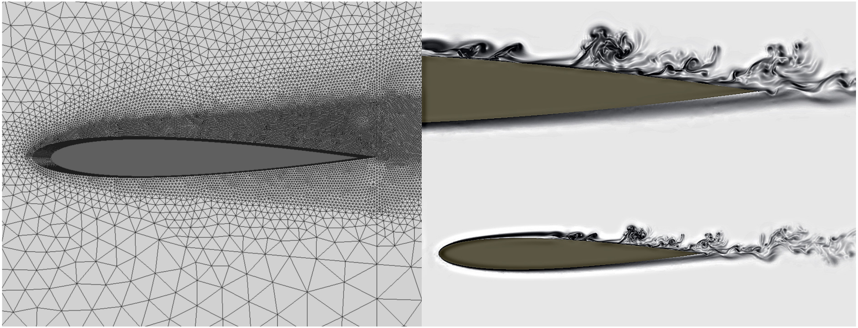


[10] R.Borrell, O.Lehmkuhl, F.X.Trias, A.Oliva. *Parallel Direct Poisson solver for discretizations with one Fourier diagonalizable direction*. **Journal of Computational Physics**, 230:4723-4741, 2011.

Pressure-velocity coupling on collocated grids

Examples of simulations

Despite these inherent limitations, symmetry-preserving collocated formulation has been successfully used for DNS/LES simulations[10]:

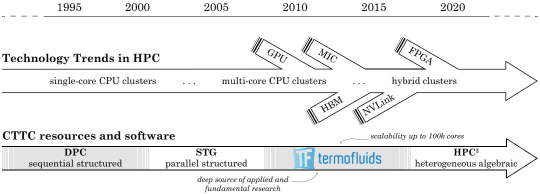


[10] F.X.Trias and O.Lehmkuhl. *A self-adaptive strategy for the time-integration of Navier-Stokes equations*. **Numerical Heat Transfer, part B**, 60(2):116-134, 2011.

Algebra-based approach naturally leads to portability

Research question #2:

- How can we develop **portable** and **efficient** CFD codes for large-scale simulations on modern supercomputers?



HPC²: portable, algebra-based framework for heterogeneous computing is being developed. Traditional stencil-based data and sweeps are replaced by algebraic structures (sparse matrices and vectors) and kernels. `SpMM`-based strategies to increase the arithmetic intensity are/were presented in this conference[11,12].

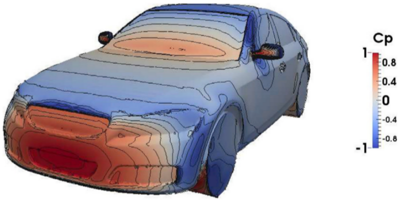
[11] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

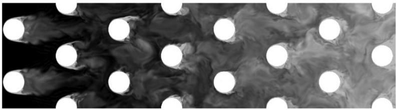
[12] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

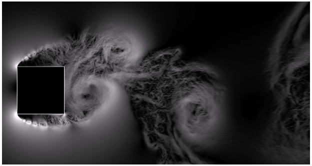
Algebra-based approach naturally leads to portability, to simple and analyzable formulations

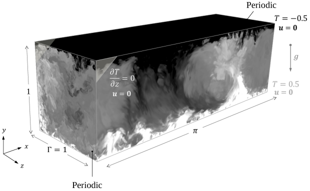
Continuous	Discrete
$\frac{\partial \boldsymbol{u}}{\partial t} + C(\boldsymbol{u}, \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{u} - \nabla p$	$\Omega \frac{d\boldsymbol{u}_h}{dt} + \mathsf{C}(\boldsymbol{u}_h)\boldsymbol{u}_h = \mathsf{D}\boldsymbol{u}_h - \mathsf{G}\boldsymbol{p}_h$
$\nabla \cdot \boldsymbol{u} = 0$	$\mathsf{M}\boldsymbol{u}_h = \boldsymbol{0}_h$
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \int_\Omega \boldsymbol{a}\boldsymbol{b}\, d\Omega$	$\langle \boldsymbol{a}_h, \boldsymbol{b}_h \rangle_h = \boldsymbol{a}_h^T \Omega \boldsymbol{b}_h$
$\langle C(\boldsymbol{u}, \varphi_1), \varphi_2 \rangle = -\langle C(\boldsymbol{u}, \varphi_2), \varphi_1 \rangle$	$\mathsf{C}(\boldsymbol{u}_h) = -\mathsf{C}^T(\boldsymbol{u}_h)$
$\langle \nabla \cdot \boldsymbol{a}, \varphi \rangle = -\langle \boldsymbol{a}, \nabla \varphi \rangle$	$\Omega \mathsf{G} = -\mathsf{M}^T$
$\left\langle \nabla^2 \boldsymbol{a}, \boldsymbol{b} \right\rangle = \left\langle \boldsymbol{a}, \nabla^2 \boldsymbol{b} \right\rangle$	$\mathsf{D} = \mathsf{D}^T \quad def -$

Algebra-based approach naturally leads to portability, to simple and analyzable formulations and opens the door to new strategies[13,14] to improve its perfomance...





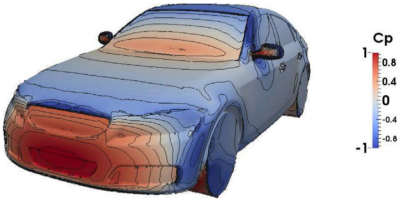


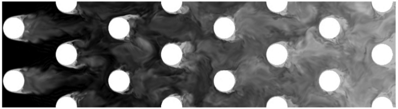


[13] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

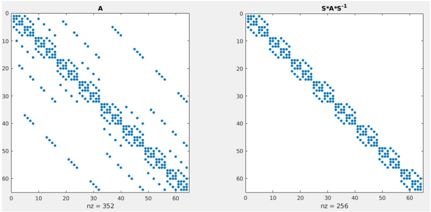
[14] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Algebra-based approach naturally leads to portability, to simple and analyzable formulations and opens the door to new strategies[13,14] to improve its perfomance...





$$\hat{L} = SLS^{-1} = I \otimes L_{inn} + diag(\boldsymbol{d})$$



SpMM can be used $\implies$ **higher AI**

[13] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

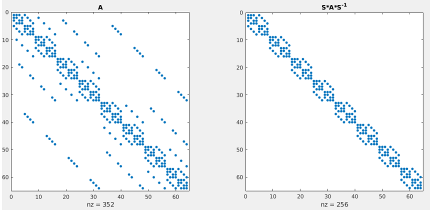
[14] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Algebra-based approach naturally leads to portability, to simple and analyzable formulations and opens the door to new strategies[13,14] to improve its perfomance...

Benefits for Poisson solver are 3-fold:

$$\hat{\mathsf{L}} = \mathsf{S}\mathsf{L}\mathsf{S}^{-1} = \mathsf{I} \otimes \mathsf{L}_{inn} + diag(\boldsymbol{d})$$

- Higher arithmetic intensity (AI)
- Reduction of memory footprint
- Reduction in the number of iterations



`SpMM` can be used $\Longrightarrow$ **higher AI**

[13] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

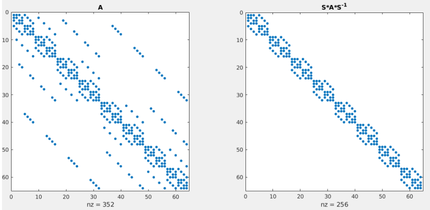
[14] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Algebra-based approach naturally leads to portability, to simple and analyzable formulations and opens the door to new strategies[13,14] to improve its perfomance...

Benefits for Poisson solver are 3-fold:

$$\hat{\mathsf{L}} = \mathsf{S}\mathsf{L}\mathsf{S}^{-1} = \mathsf{I} \otimes \mathsf{L}_{inn} + diag(\boldsymbol{d})$$

- Higher arithmetic intensity (AI)
- Reduction of memory footprint
- Reduction in the number of iterations



→ Overall speed-up up to **x2-x3** ✓
→ Memory reduction of ≈**2** ✓

SpMM can be used ⟹ **higher AI**

[13] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

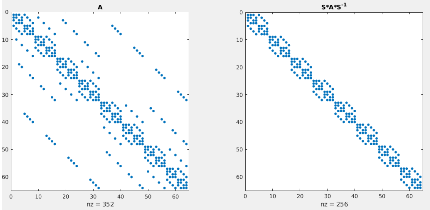
[14] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Algebra-based approach naturally leads to portability, to simple and analyzable formulations and opens the door to new strategies[13,14] to improve its perfomance...

Other SpMM-based strategies to **increase AI** and **reduce memory** footprint:

- Multiple transport equations
- Parametric studies
- Parallel-in-time simulations
- Go to higher-order?

$$\hat{L} = SLS^{-1} = I \otimes L_{inn} + diag(\boldsymbol{d})$$



SpMM can be used $\Longrightarrow$ **higher AI**

[13] À.Alsalti-Baldellou, X.Álvarez-Farré, A.Gorobets, A.Oliva, F.X.Trias. *Strategies to increase the arithmetic intensity of the linear solvers*. **ParCFD'22** Don't miss it!

[14] X.Álvarez-Farré, À.Alsalti-Baldellou, A.Gorobets, A.Oliva, F.X.Trias. *On the benefits and applications of sparse matrix-matrix product on various parallel architectures*. **ParCFD'22** Don't miss it!

Rethinking standard CFD operations

In summary...

<u>Leitmotiv</u>: relying on a **minimal set of (algebraic) kernels** is crucial for code **portability** and **maintenance**!!!

[15] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Rethinking standard CFD operations

In summary...

Leitmotiv: relying on a **minimal set of (algebraic) kernels** is crucial for code **portability** and **maintenance**!!!

- Computational challenge: `SpMV` has a low AI
  - Solution: make use of `SpMM` whenever possible (multiple transport equations, spatial symmetries, parallel-in-time simulations, parametric studies,...) to increase AI and, therefore, perfomance.
  - Positive side-effects: reduction of memory footprint (crucial for GPUs), improvement of the convergence for the Poisson solver...

[15] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Rethinking standard CFD operations

In summary...

Leitmotiv: relying on a **minimal set of (algebraic) kernels** is crucial for code **portability** and **maintenance**!!!

- Computational challenge: `SpMV` has a low AI
  - Solution: make use of `SpMM` whenever possible (multiple transport equations, spatial symmetries, parallel-in-time simulations, parametric studies,...) to increase AI and, therefore, perfomance.
  - Positive side-effects: reduction of memory footprint (crucial for GPUs), improvement of the convergence for the Poisson solver...
- Implementation challenge: there still exists a list of standard CFD methods that do not seem to fit well on an algebraic framework (*e.g.* flux limiters[15], boundary conditions, CFL condition,...).
  - Dilemma: "add more and more specific kernels" vs "rethink them"

[15] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Rethinking standard CFD operations

In summary...

Leitmotiv: relying on a **minimal set of (algebraic) kernels** is crucial for code **portability** and **maintenance**!!!

- Computational challenge: `SpMV` has a low AI
  - Solution: make use of `SpMM` whenever possible (multiple transport equations, spatial symmetries, parallel-in-time simulations, parametric studies,...) to increase AI and, therefore, perfomance.
  - Positive side-effects: reduction of memory footprint (crucial for GPUs), improvement of the convergence for the Poisson solver...
- Implementation challenge: there still exists a list of standard CFD methods that do not seem to fit well on an algebraic framework (*e.g.* flux limiters[15], boundary conditions, **CFL condition**,...).
  - Dilemma: "add more and more specific kernels" vs **"rethink them"**

[15] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Rethinking standard CFD operations

CFL-like condition

Step #1: forget about classical formulae from textbooks...

$$\Delta t \leqslant C_{conv} \left(\frac{\Delta x}{U} \right)_{\min} \quad \text{and} \quad \Delta t \leqslant C_{diff} \left(\frac{\Delta x^2}{\nu} \right)_{\min}$$

Rethinking standard CFD operations

CFL-like condition

Step #1: forget about classical formulae from textbooks...

$$\Delta t \leqslant C_{conv} \left(\frac{\Delta x}{U} \right)_{\min} \quad \text{and} \quad \Delta t \leqslant C_{diff} \left(\frac{\Delta x^2}{\nu} \right)_{\min}$$

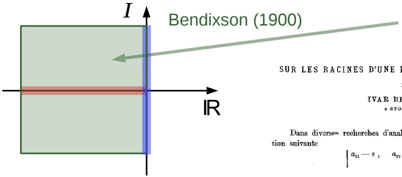
...and replace it by an eigenbounding problem of $\mathsf{C}(\boldsymbol{u}_s)$ and D matrices

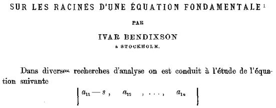
$$\boxed{\Omega_s \frac{d\boldsymbol{u}_s}{dt} + \mathsf{C}(\boldsymbol{u}_s)\,\boldsymbol{u}_s = \mathsf{D}\boldsymbol{u}_s - \mathsf{G}\boldsymbol{p}_c; \quad \mathsf{M}\boldsymbol{u}_s = \mathbf{0}_c}$$

$$(-\mathsf{C}(\boldsymbol{u}_s) + \mathsf{D})\,\boldsymbol{v} = \lambda \boldsymbol{v}$$

$$\mathsf{C}(\boldsymbol{u}_s)\,\boldsymbol{v} = \lambda \boldsymbol{v}$$

$$\mathsf{D}\boldsymbol{v} = \lambda \boldsymbol{v}$$




SUR LES RACINES D'UNE ÉQUATION FONDAMENTALE[1]

PAR

IVAR BENDIXSON

à STOCKHOLM.

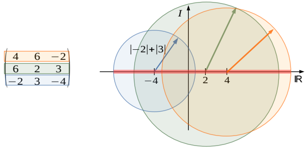
Dans diverses recherches d'analyse on est conduit à l'étude de l'équation suivante

$$\begin{vmatrix} a_{11} - s, & a_{12} & , \dots , & a_{1n} \end{vmatrix}$$

Rethinking standard CFD operations

CFL-like condition

Step #2: compute eigenbounds of $\mathsf{C}\left(\boldsymbol{u_s}\right)$ and D in an inexpensive way[16]



$$\rho(\mathsf{D}) \leqslant \rho^{Gersh}(\mathsf{D})$$

$$\rho(\mathsf{C}) \leqslant \rho^{Gersh}(\mathsf{C})$$

[16] F.X.Trias and O.Lehmkuhl. *A self-adaptive strategy for the time-integration of Navier-Stokes equations*. **Numerical Heat Transfer, part B**, 60(2):116-134, 2011.

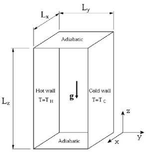
Rethinking standard CFD operations

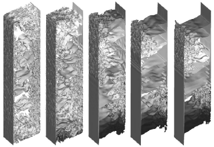
CFL-like condition

Step #2: compute eigenbounds of $\mathsf{C}\,(\boldsymbol{u}_s)$ and D in an inexpensive way[16]

$$\rho(\mathsf{D}) \leqslant \rho^{Gersh}(\mathsf{D})$$

$$\rho(\mathsf{C}) \leqslant \rho^{Gersh}(\mathsf{C})$$





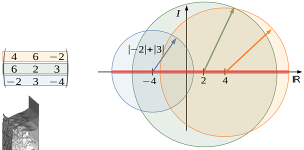


Table 1. Tests for the air-filled (Pr = 0.71) differentially heated cavity at Rayleigh number Ra = 3 × 10^{10} and height aspect ratio 4; averaged results correspond to the statistically steady state

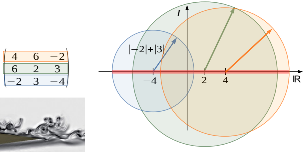
	N_x	N_y	N_z	$\varphi/(\pi/2)$	$\overline{\delta t}_{CFL+AB2}$	$\overline{\delta t}_{EigenCD+\kappa 1L2}$	$\overline{\delta t}_{EigenCD+\kappa 1L2}/\overline{\delta t}_{CFL+AB2}$
MeshA	128	338	778	0.072	1.04×10^{-4}	3.02×10^{-4}	2.90
MeshB	64	168	338	0.158	4.31×10^{-4}	1.21×10^{-3}	2.80
MeshC	32	84	168	0.252	1.80×10^{-3}	4.69×10^{-3}	2.59
MeshD	32	56	112	0.408	4.21×10^{-3}	8.75×10^{-3}	2.08
MeshE	16	42	84	0.504	6.88×10^{-3}	1.35×10^{-3}	1.96

[16] F.X.Trias and O.Lehmkuhl. *A self-adaptive strategy for the time-integration of Navier-Stokes equations*. **Numerical Heat Transfer, part B**, 60(2):116-134, 2011.

Rethinking standard CFD operations

CFL-like condition

Step #2: compute eigenbounds of $\mathsf{C}(\boldsymbol{u}_s)$ and D in an inexpensive way[16]



$$\rho(\mathsf{D}) \leqslant \rho^{Gersh}(\mathsf{D})$$

$$\rho(\mathsf{C}) \leqslant \rho^{Gersh}(\mathsf{C})$$

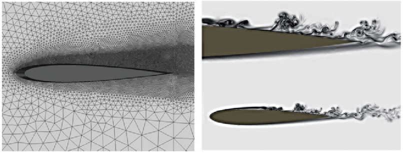


Table 2. Tests for the flow around a NACA 0012 airfoil at Reynolds number 5×10^4 and an angle of attack of 5°; averaged results correspond to the statistically steady state

	N_x	Mesh2D	$\varphi/(\pi/2)$	$\overline{\delta t}_{CFL+AB2}$	$\overline{\delta t}_{EigenCD+\kappa 1L2}$	$\overline{\delta t}_{EigenCD+\kappa 1L2}/\overline{\delta t}_{CFL+AB2}$
UMeshA	64	$\approx 2.65 \times 10^5$	0.593	4.69×10^{-5}	1.30×10^{-4}	2.77
UMeshB	32	$\approx 4.69 \times 10^4$	0.956	1.61×10^{-4}	6.86×10^{-4}	4.27

[16] F.X.Trias and O.Lehmkuhl. *A self-adaptive strategy for the time-integration of Navier-Stokes equations*. **Numerical Heat Transfer, part B**, 60(2):116-134, 2011.

Rethinking standard CFD operations

CFL-like condition

Step #3:
reformulate the problem in a way that we avoid constructing $\mathsf{C}\left(\boldsymbol{u}_s\right)$ and D

$$\rho(\mathsf{D}) \leqslant \rho^{Gersh}(\mathsf{D}) \ldots$$

$$\rho(\mathsf{C}) \leqslant \rho^{Gersh}(\mathsf{C}) \ldots$$

where T_{cs} is the face-to-cell oriented incidence matrix; $\tilde{\Delta}_s \equiv A_s \Lambda_s \Delta_s^{-1}$ (diffusivity-like fluxes) and $\tilde{F}_s \equiv A_s U_s$ (mass fluxes) are diagonal matrices.

Rethinking standard CFD operations

CFL-like condition

Step #3 ... #4 (some maths that would take too long to explain): reformulate the problem in a way that we avoid constructing $\mathsf{C}(\boldsymbol{u}_s)$ and D

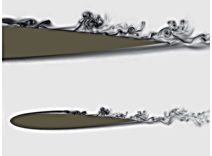
$$\rho(\mathsf{D}) \leqslant \rho^{Gersh} \ldots (\mathsf{D}) \leqslant \rho^{Gersh}(T_{cs} T_{cs}^T \tilde{\Delta}_s) = \max(\underbrace{\overbrace{|T_{cs} T_{cs}^T|}^{\text{Constant matrix}} diag(\tilde{\Delta}_s)}_{\texttt{SpMV}})$$

$$\rho(\mathsf{C}) \leqslant \rho^{Gersh} \ldots (\mathsf{D}) \leqslant \rho^{Gersh}(T_{cs} T_{cs}^T |\tilde{F}_s|) = \max(\underbrace{\overbrace{|T_{cs} T_{cs}^T|}^{\text{Constant matrix}} diag(\tilde{F}_s)}_{\texttt{SpMV}})$$

where T_{cs} is the face-to-cell oriented incidence matrix; $\tilde{\Delta}_s \equiv A_s \Lambda_s \Delta_s^{-1}$ (diffusivity-like fluxes) and $\tilde{F}_s \equiv A_s U_s$ (mass fluxes) are diagonal matrices.

Concluding remarks

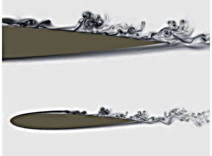
- **Preserving symmetries** either using staggered or collocated formulations is the key point for **reliable LES/DNS** simulations.

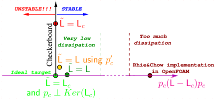


[17] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Concluding remarks

- **Preserving symmetries** either using staggered or collocated formulations is the key point for **reliable LES/DNS** simulations.
- Algebra-based approach naturally leads to **portability**, to simple and **analyzable** formulations and opens the door to **new strategies to improve its perfomance**.

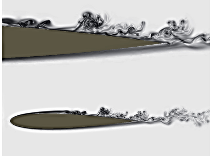


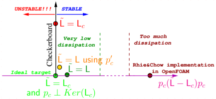


[17] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Concluding remarks

- **Preserving symmetries** either using staggered or collocated formulations is the key point for **reliable LES/DNS** simulations.
- Algebra-based approach naturally leads to **portability**, to simple and **analyzable** formulations and opens the door to **new strategies to improve its perfomance**.





On-going research:

- **Rethinking** standard CFD operations (*e.g.* flux limiters[17], boundary conditions, **CFL**,...) to adapt them into an algebraic framework (Motivation: maintaining a minimal number of basic kernels is crucial for portability!!!)

[17] N.Valle, X.Álvarez, A.Gorobets, J.Castro, A.Oliva, F.X.Trias. *On the implementation of flux limiters in algebraic frameworks*. **Computer Physics Communications**, 271:108230, 2022.

Thank you for your attendance