# Computational aspects of DGM as an enabling technology for LES of practical flows

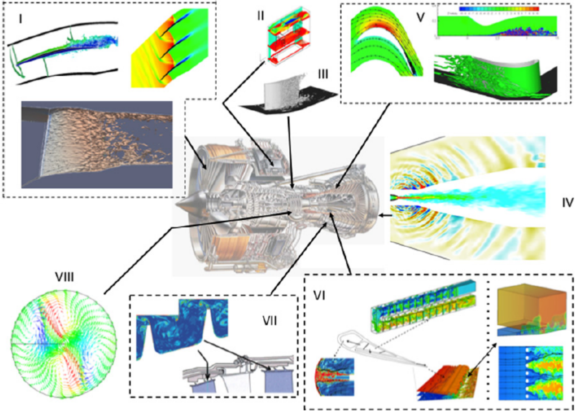
1st Intl. Workshop on HPC-CFD in Energy/Transport Domain

**Koen Hillewaert, Corentin Carton de Wiart**
Argo team leader, research engineer
Contact: koen.hillewaert@cenaero.be

PROD-F-015-01

Doc. ref.: N/A

## Industrial need : propulsion & turbomachinery (Tucker, 2011)



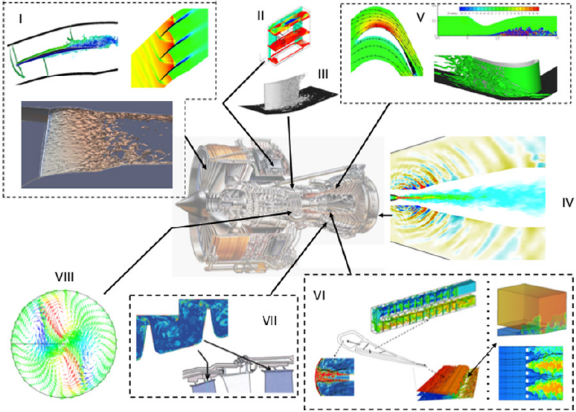

- no clear scale separation for many flow regimes
- (U)RANS is still used due to cost issues, works well for most (design) regimes
- DNS, LES, . . . needed for off-design, transition, cavities, instabilities, noise, combustion

PROD-F-015-01



Cenaero

# Large Eddy Simulation
## Industrial need : propulsion & turbomachinery (Tucker, 2011)



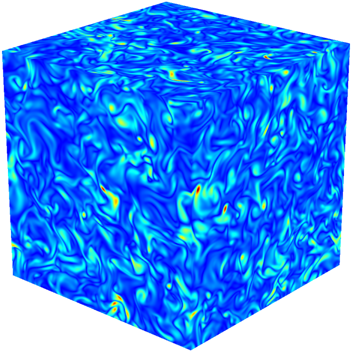

*"Although LES is, obviously, much less model dependent than RANS, grids currently used for more practical simulations are clearly insufficiently fine for the LES model and numerical schemes not to be playing an excessively strong role."*

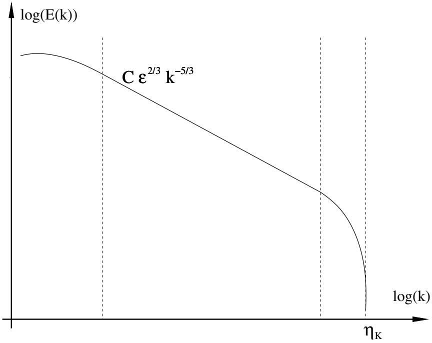




Cenaero

# Large Eddy Simulation
## Principle : DNS



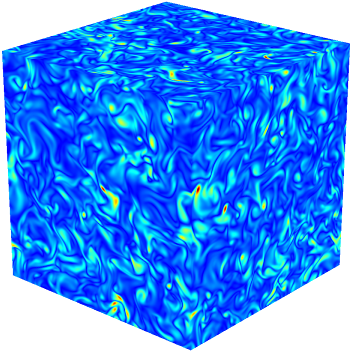


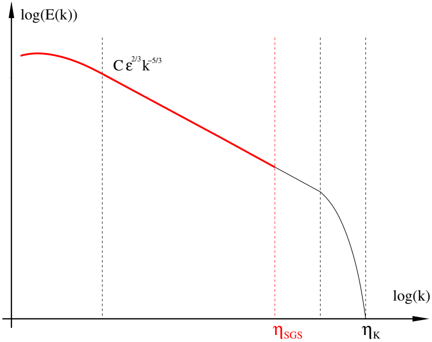

- energetic subrange : creation
- inertial subrange : transfer from larger to smaller scales, no loss
- viscous subrange : destruction $l_K = \left(\frac{\nu^3}{\epsilon}\right)^{1/4} \sim Re^{-3/4}$

PROD-F-015-01



Cenaero





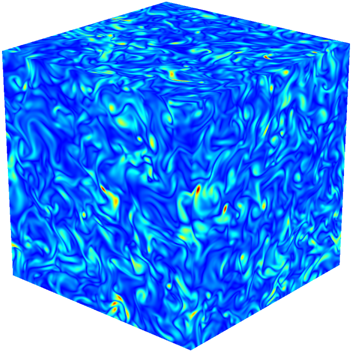

- resolved / captured scales
- subgrid scales

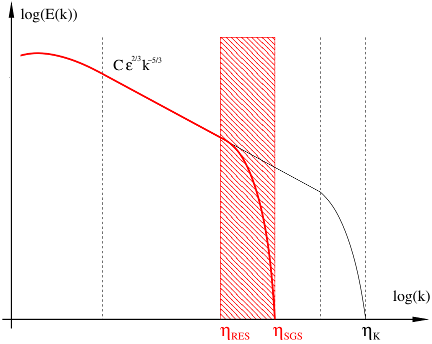
PROD-F-015-01



Cenaero

# Large Eddy Simulation
## Principle : multiscale LES






- captured scales
  - fully resolved scales
  - model scales
- subgrid scales

PROD-F-015-01



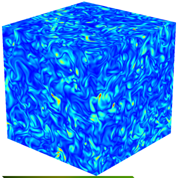
Cenaero

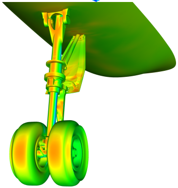
Academic codes (Spectral, FD, ...)

- + high order of accuracy
- + low computational cost
- + low / optimized dissipation and dispersion
- − no / limited geometric flexibility
- − some models are tuned to case

Industrial codes (FVM, stabilised FEM)

- + geometric flexibility
- + robustness
- ∼ moderate scalability and efficiency
- − (formally) $2^{nd}$ order of accuracy
  - − high dispersion and dissipation (RANS/shocks)
  - − degradation near boundaries and complex geometry
- − kinetic energy preserving methods degrade stability





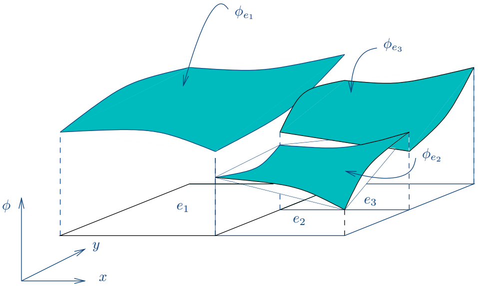
Discretisation is as important as model → DGM

- + error relegated to very high frequencies
- + unstructured meshes for complex geometry
- + high resolution → HPC, computational efficiency
- + adaptive resolution





Cenaero




- **Discontinuous Finite element approximation :** $\tilde{u}_m \approx u_m = \sum_i \phi_i \mathbf{u}_{im} \in \mathcal{V}$
  - regular functions in each element
  - potentially *discontinuous* at interfaces
- **Galerkin variational formulation** : $\mathbf{u}_{im}$ formally defined by

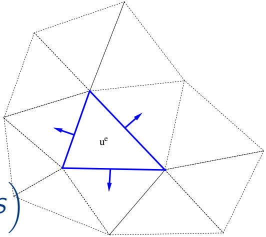
$$\int_\Omega v_m \left( \frac{\partial u_m}{\partial t} + \nabla \cdot \mathbf{g}_m(u) \right) dV = 0 \ , \ \forall v \in \mathcal{V}$$

PROD-F-015-01



Cenaero

Straightforward development




$$\int_\Omega v_m \frac{\partial u_m}{\partial t} dV + \int_\Omega v_m \nabla \cdot \mathbf{g}_m dV = 0 \ , \ \forall v \in \mathcal{V}$$

$$= \sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV + \sum_e \int_e v_m \nabla \cdot \mathbf{g}_m dV$$

$$= \sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV + \sum_e \left( - \int_e \nabla v_m \cdot \mathbf{g}_m dV + \oint_{\partial e} v_m \mathbf{g}_m \cdot \mathbf{n} dS \right)$$

$$= \sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV - \sum_e \int_e \nabla v_m \cdot \mathbf{g}_m dV + \sum_f \int_f [\![ v_m \mathbf{g}_m ]\!] dS$$

DGM replaces $[\![ v_m \mathbf{g}_m ]\!]$ by $\gamma$ to ensure stability

$$\sum_e v_m \frac{\partial u_m}{\partial t} - \sum_e \int_e \nabla v_m \cdot \mathbf{g}_m dV + \sum_f \int_f \gamma_m(u^+, u^-, v^+, v^-, \mathbf{n}) dS = 0 \ , \ \forall v \in \mathcal{V}$$

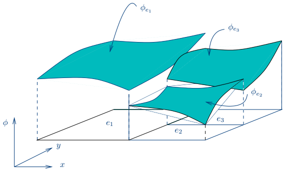
PROD-F-015-01



Cenaero

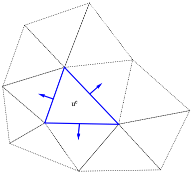
Global formulation : $\forall v \in \mathcal{V} = \cup_e \mathcal{V}(e)$

$$\sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV - \sum_e \int_e \nabla v_m \cdot \mathbf{g}_m dV + \sum_f \int_f \gamma_m(u^+, u^-, v^+, v^-, \mathbf{n}) dS = 0$$



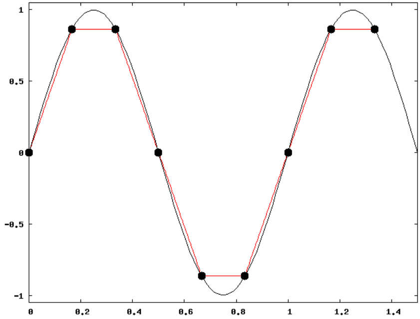

Elementwise Galerkin : $\forall \phi_i \in \mathcal{V}(e)$ , $\forall e \in \mathcal{E}$

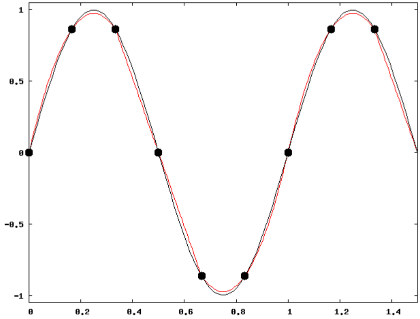
$$\int_e \phi_i \frac{\partial u^e_m}{\partial t} dV - \int_e \nabla \phi_i \cdot \mathbf{g}_m dV + \int_{\partial e} \gamma_m(u^e, u^*, \phi_i, 0, \mathbf{n}) dS = 0$$




PROD-F-015-01

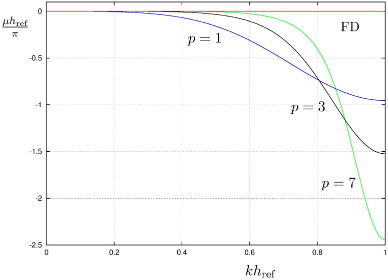
# Discontinuous Galerkin Method
## Resolution vs accuracy : impact of order

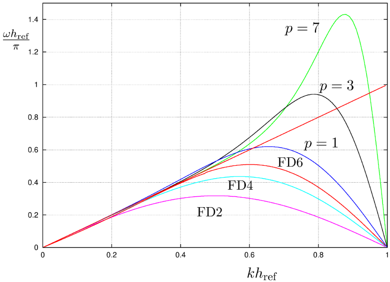



PROD-F-015-01

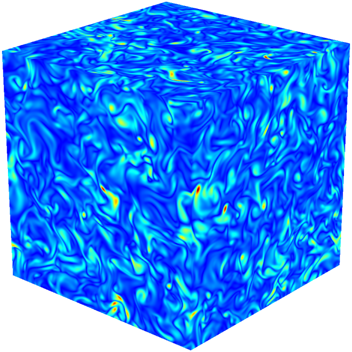
# Discontinuous Galerkin Method
## Resolution vs accuracy : error properties of DGM

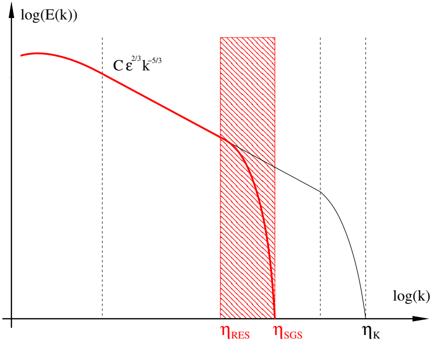



PROD-F-015-01

# Discontinuous Galerkin Method
## Resolution vs accuracy : LES and numerical error





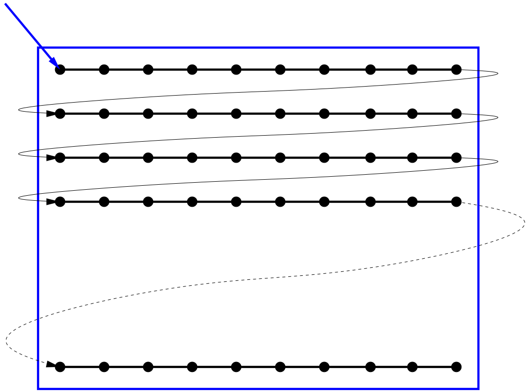

- captured scales
  - fully resolved scales
  - model scales
- subgrid scales : modeled

Implicit LES ?

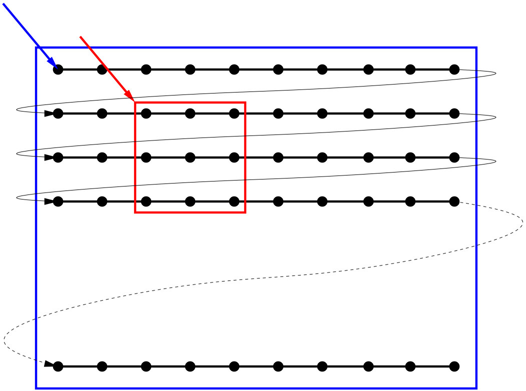
PROD-F-015-01



Cenaero



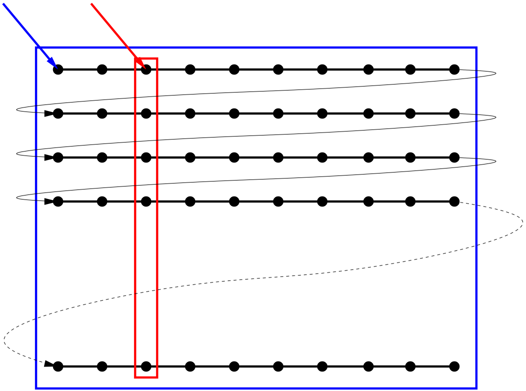
$$\mathbf{A} \in \mathbb{R}^{m \times n} = (\mathbf{a}, n, m, lda)$$
$$\mathbf{A}_{ij} = *(\mathbf{a} + i * n + j)$$
$$\mathbf{A}_{ij} = *(\mathbf{a} + i * lda + j)$$



$$\mathbf{B} \in \mathbb{R}^{p \times q} = (\mathbf{b}, p, q, lda)$$

$$\mathbf{b} = \mathbf{a} + i_b * lda + j_b$$

$$\mathbf{B}_{ij} = *(\mathbf{b} + i * lda + j) = *(\mathbf{a} + (i + i_b) * n + (j + j_b))$$



$$\mathbf{b} \in \mathbb{R}^p = (\mathbf{b}, p, lda)$$

$$\mathbf{b} = \mathbf{a} + i_b * lda$$

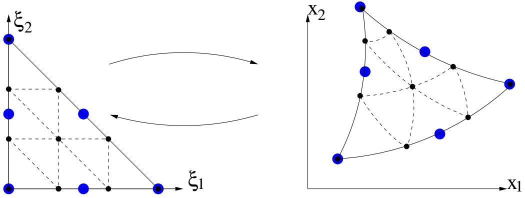
$$\mathbf{b}_i = *(\mathbf{b} + i * lda) = *(\mathbf{a} + (i + i_b) * n$$

PROD-F-015-01



Cenaero

Solutions expanded in parametric coordinates $u_m = \sum_{i=1}^{N_\phi} \mathbf{u}_{im} \phi_i(\boldsymbol{\xi})$




Classical Gauss-Legendre quadrature in parametric space

$$\int_V \nabla \phi_i \cdot (\mathbf{f}_m(u) + \mathbf{d}_m(u, \nabla u)) dV \approx \sum_{q=1}^{N_q} w_q \left( \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}_{kl}^{-1} \left( f_m^l(u) + d_m^l(u, \nabla u) \right) |\mathbf{J}| \right)_{\xi_q}$$

Jacobian $\mathbf{J}$ of the mapping $\boldsymbol{\xi} \rightarrow \mathbf{x}$

$$\mathbf{J}_{kl} = \frac{\partial x^k}{\partial \xi^l} = \sum_{i=1}^{N_c} \frac{\partial \psi_i}{\partial \xi^l} \qquad \left(\mathbf{J}^{-1}\right)_{kl} = \frac{\partial \xi^k}{\partial x^l}$$

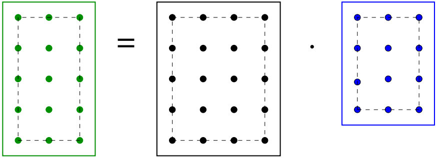
PROD-F-015-01



Cenaero

# Efficient assembly

## Residual assembly : split and reinterpretation of data structures

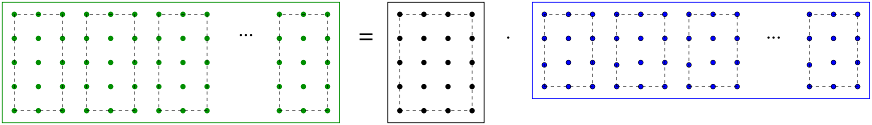


1. Collocation

$$\mathfrak{u}_{qm} = u_m(\xi_q) = \sum_{i=1}^{N_\phi} \phi_i(\xi_q)\mathbf{u}_{im} = \sum_{i=1}^{N_\phi} \mathfrak{C}_{qi}\mathbf{u}_{im}$$

$$\mathfrak{g}^l_{qm} = \left(\frac{\partial u_m}{\partial \xi^l}\right)_{\xi_q} = \sum_{i=1}^{N_\phi} \left(\frac{\partial \phi_i}{\partial \xi^l}\right)_{\xi_q} \mathbf{u}_{im} = \sum_{i=1}^{N_\phi} \mathfrak{G}^l_{qi}\mathbf{u}_{im}$$

PROD-F-015-01




1. Collocation over all elements

$$\mathfrak{u}_{qm} = u_m(\xi_q) = \sum_{i=1}^{N_\phi} \phi_i(\xi_q)\mathbf{u}_{im} = \sum_{i=1}^{N_\phi} \mathfrak{C}_{qi}\mathbf{u}_{im}$$

$$\mathfrak{g}^l_{qm} = \left(\frac{\partial u_m}{\partial \xi^l}\right)_{\xi_q} = \sum_{i=1}^{N_\phi} \left(\frac{\partial \phi_i}{\partial \xi^l}\right)_{\xi_q} \mathbf{u}_{im} = \sum_{i=1}^{N_\phi} \mathfrak{G}^l_{qi}\mathbf{u}_{im}$$

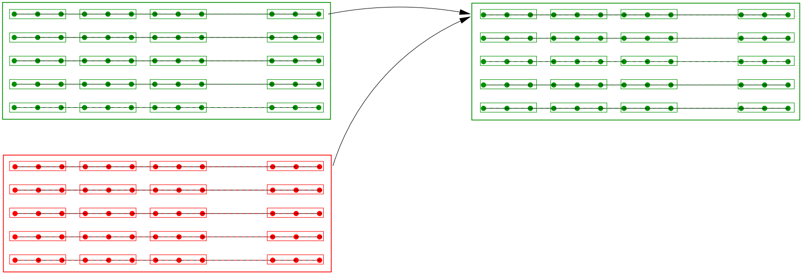




Cenaero

# Efficient assembly

## Residual assembly : split and reinterpretation of data structures



1. Collocation over all elements
2. Flux evaluation per quadrature point

$$\mathfrak{f}^k_{qm} = |\mathbf{J}|\mathbf{J}^{-1}_{kl}\left(f^k_m(u) + d^k_m(u,\nabla u)\right)_{\xi_q}$$

$$= \left(\frac{\partial u_m}{\partial x^k}\right)_{\xi_q} = \sum_{l=1}^{d}\left(\mathbf{J}^{-1}_{lk}\frac{\partial u_m}{\partial \xi^l}\right)_{\xi_q}$$

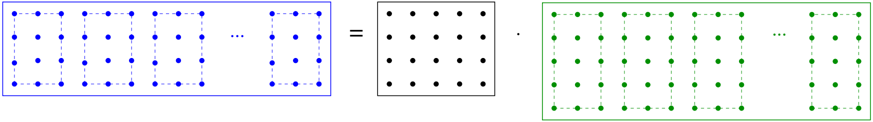
PROD-F-015-01



Cenaero

# Efficient assembly

## Residual assembly : split and reinterpretation of data structures



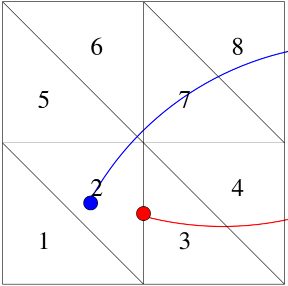

1. Collocation over all elements
2. Flux evaluation per quadrature point
3. Flux redistribution over all elements

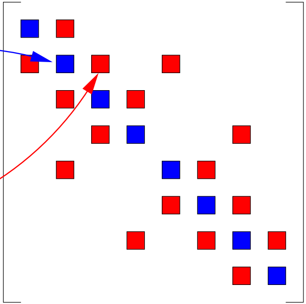
$$\mathbf{r}_{im} += \sum_{q=1}^{N_q} w_q \left( \frac{\partial \phi_i}{\partial \xi^k} \right)_{\xi_q} \mathfrak{f}^k_{qm} = \sum_{q=1}^{N_q} \mathfrak{R}^l_{iq} \mathfrak{f}^l_{qm}$$

PROD-F-015-01



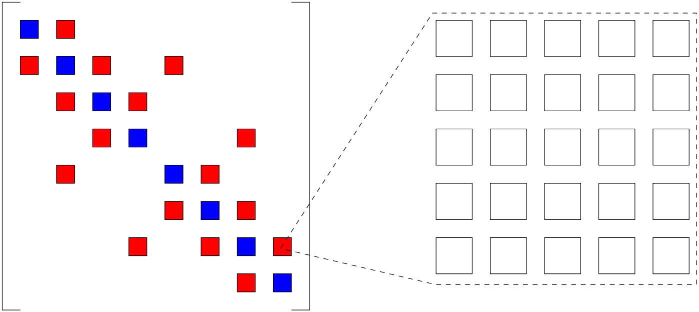
Cenaero



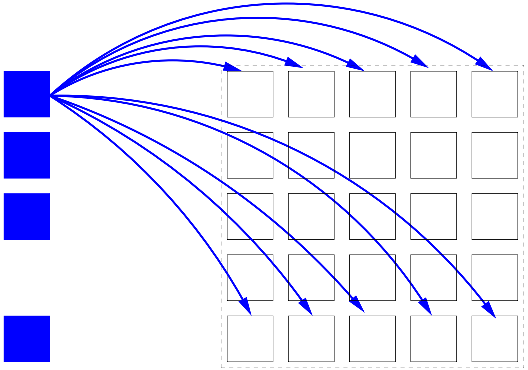



block CSR structure

- large block size $N_\phi N_v \sim p^3 N_v$ (eg. DGM(4) hex, Navier-Stokes : 625)
- computations recastable in dense gemm, inversion and gemv
  - datastructure can be deallocated/allocated on the fly
  - internal renumbering independent of the mesh
  - mixed precision
- very low indexing overhead allowing simple and flexible datastructure
- typically single precision (bJacobi-, bILU-GMRES for damped inexact Newton)

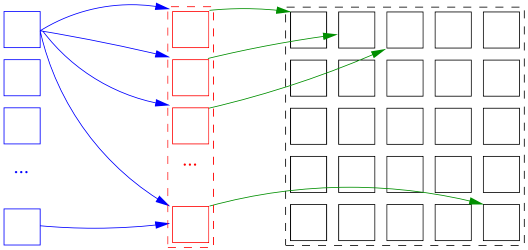


$$\mathbf{r}_{im} = \sum_q w_q \left( |\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left( w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left( |\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$



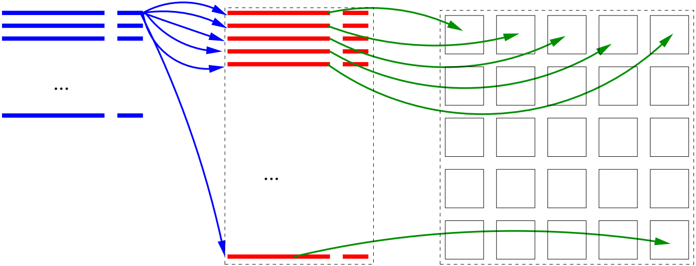
$$\mathbf{r}_{im} = \sum_q w_q \left( |\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left( w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left( |\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

- subblock (right) is proxy → not contiguous in memory → addition is done row per row




$$\mathbf{r}_{im} = \sum_q w_q \left( |\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left( w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left( |\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

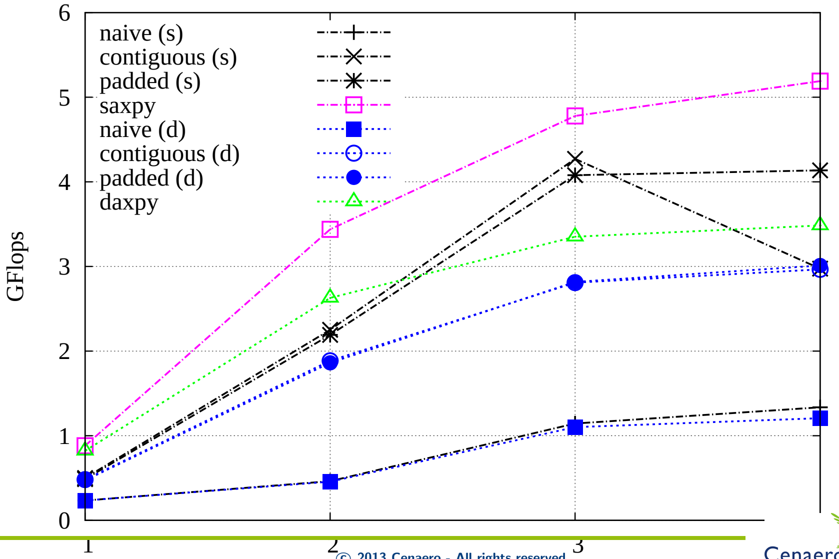
- subblock (right) is proxy → not contiguous in memory
- intermediate blocks contiguous → $N_q$ contiguous sums (blue) + single copy (green)



$$\mathbf{r}_{im} = \sum_q w_q \left( |\mathbf{J}| \frac{\partial \phi_i}{\partial \xi^k} \mathbf{J}^{-1}_{kl} f^l_m(u) \right)_{\xi_q} \Rightarrow \frac{\partial \mathbf{r}_{im}}{\partial \mathbf{u}_{jn}} = \sum_q \left( w_q \frac{\partial \phi_i}{\partial \xi^k} \phi_j \right)_{\xi_q} \left( |\mathbf{J}| \mathbf{J}^{-1}_{kl} \frac{\partial f^l_m}{\partial u_n}(u) \right)_{\xi_q}$$

- subblock (right) is proxy → not contiguous in memory
- intermediate blocks contiguous → $N_q$ contiguous sums (blue) + single copy (green)
- padding increases flop efficiency in the assembly sums (blue)

PROD-F-015-01


GFlops
0
1
2
3
4
5
6
1
2
3
naive (s)
contiguous (s)
padded (s)
saxpy
naive (d)
contiguous (d)
padded (d)
daxpy





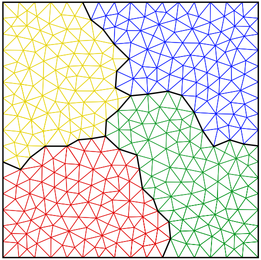
Cenaero

# Assembly

$$\sum_e v_m \frac{\partial u_m}{\partial t} - \sum_e \int_e \nabla v_m \cdot \mathbf{g}_m dV + \sum_f \int_f \gamma_m(u^+, u^-, v^+, v^-, \mathbf{n}) dS = 0 \ , \ \forall v \in \mathcal{V}$$

Shared memory

- volume terms - embarassingly parallel
- computation of interface terms
- redistribution $\rightarrow$ coloring



PROD-F-015-01



Cenaero

# Assembly

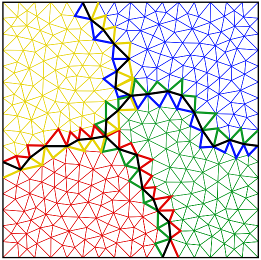
$$\sum_e v_m \frac{\partial u_m}{\partial t} - \sum_e \int_e \nabla v_m \cdot \mathbf{g}_m dV + \sum_f \int_f \gamma_m(u^+, u^-, v^+, v^-, \mathbf{n}) dS = 0 \ , \ \forall v \in \mathcal{V}$$

Shared memory

- volume terms - embarassingly parallel
- computation of interface terms
- redistribution $\rightarrow$ coloring

Distributed memory

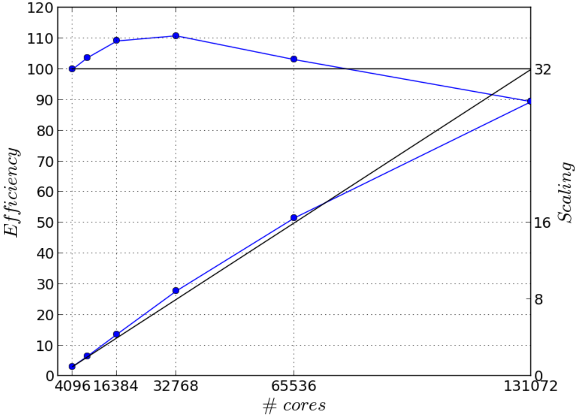
- limited exchange vs computation
- communication hiding
  1. non-blocking send of ghost cells
  2. evaluation of volume and boundary terms
  3. receiving ghost cells
  4. evaluation of interface terms



PROD-F-015-01



Cenaero




- strong scaling on BlueGene/Q at Jülich SC
- hybrid MPI(2)/OpenMP(32)
- Newton-GMRES-Jacobi single precision

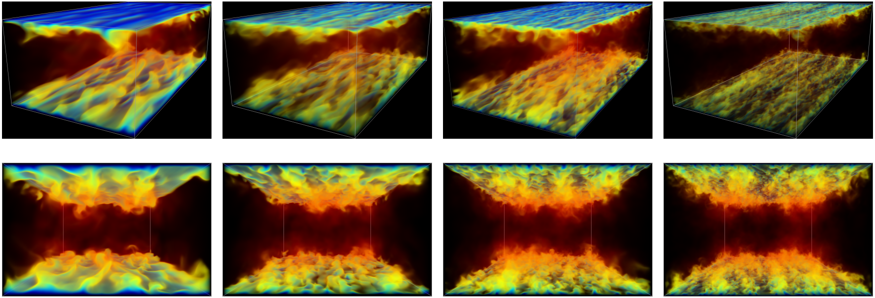




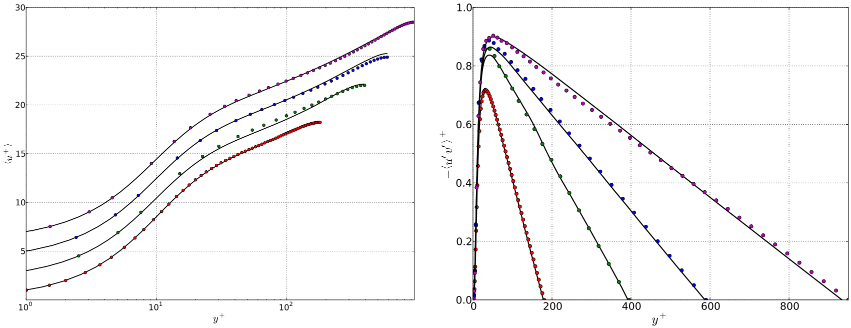
PRACE Cenaero

# Validation and applications

## Channel flow : Setup

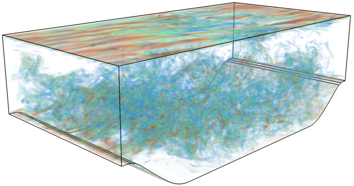


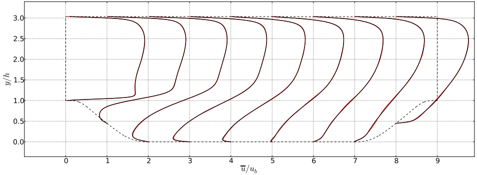
| $Re_\tau$ | *dof* ($nx \times ny \times nz$) | $\Delta x^+ \times \Delta y^+ \times \Delta z^+$ | $\Delta t^+$ | Ref. |
|---|---|---|---|---|
| 395 | $64 \times 48 \times 48$ | $38.78 \times 2.45 \times 25.85$ | $2.\ 10^{-3}$ | [?] |
| 590 | $96 \times 64 \times 96$ | $38.61 \times 2.36 \times 19.3$ | $1.\ 10^{-3}$ | [?] |
| 950 | $192 \times 96 \times 192$ | $31.01 \times 1.52 \times 15.5$ | $5.\ 10^{-4}$ | [?] |

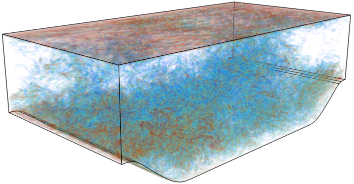


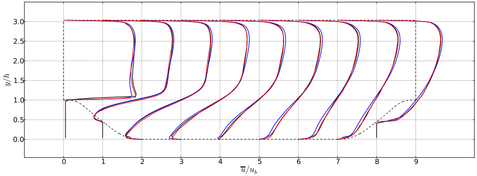
30
25
20
15
10
5
$\langle u^+ \rangle$
$10^0$
$10^1$
$10^2$
$y^+$
1.0
0.8
0.6
0.4
0.2
0.0
$-\langle u'v' \rangle^+$
0
200
400
600
800
$y^+$

# Validation and applications
## 2D periodichill (ERCOFTAC)





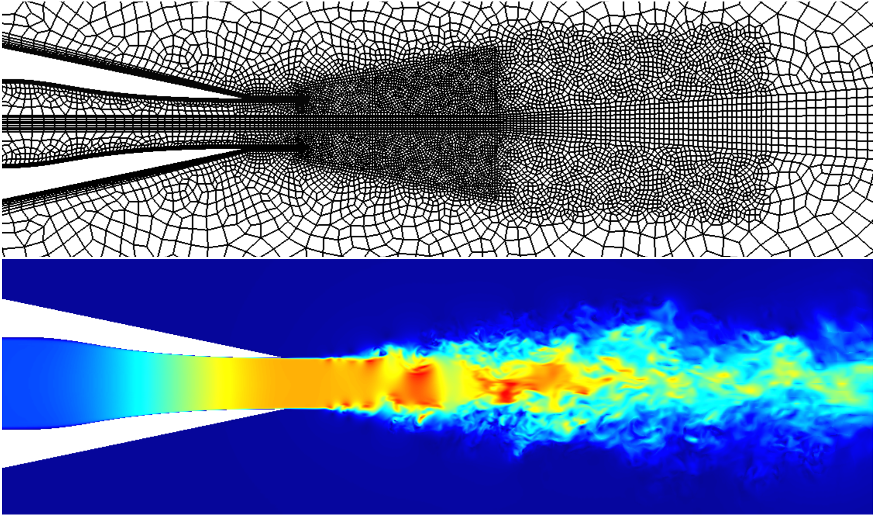


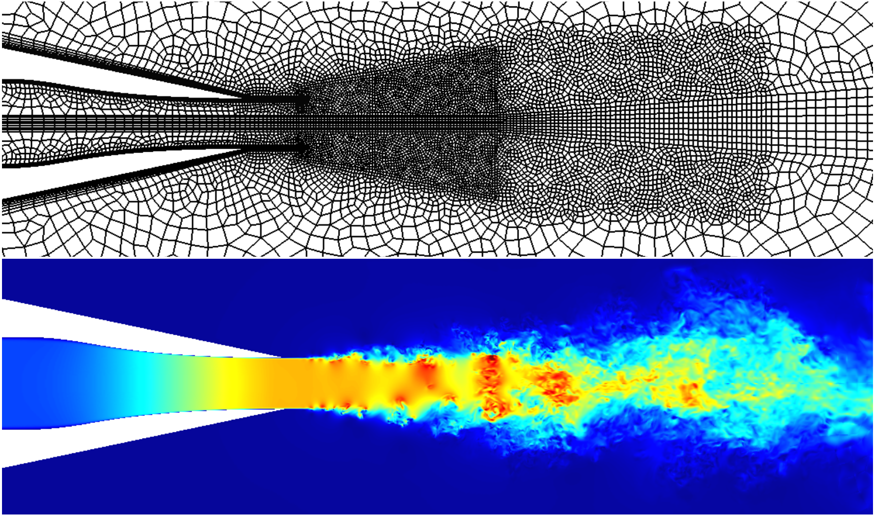

PROD-F-015-01

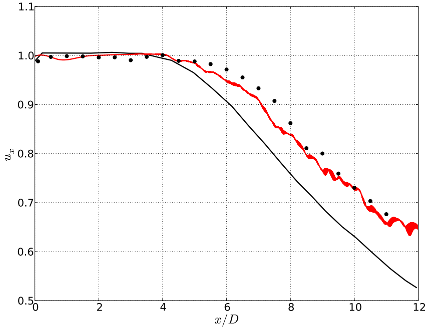


Cenaero

# Validation and applications
## JEAN Nozzle








PROD-F-015-01



Cenaero

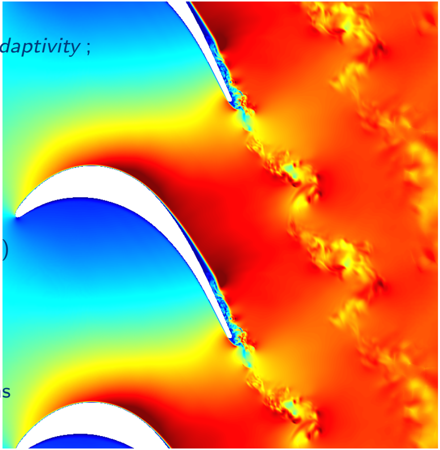
# Conclusions

DGM enabling industrial DNS and LES

- accuracy on unstructured meshes and *adaptivity* ;
- ILES approach ;
- highly efficient assembly ;
- excellent strong scaling.

Current work

- turbulence modeling strategy (ILES)
  - cross-validation experiments (VKI)
  - wall-modeled LES
  - LES vs shock capturing
  - synthetic turbulence
- grid generation & post/co-processing
- non-matching & rotor-stator connections
- multiphysics coupling : CHT, FSI, CAA
- *hp-adaptivity*
- *code optimisation and acceleration strategies*



PROD-F-015-01



Cenaero

Academic collaborations

- Jean-François Remacle (UCLouvain)
- Grégoire Winckelmans (UCLouvain)
- Laurent Bricteux (UMons)
- Matthieu Duponcheel (UCLouvain)

Funding

- European Regional Development Fund (contract N° EP1A122030000102)
- FP6 *Adigma*
- FP7 *IDIHOM*
- PRACE computational grant *PAdDLES* and *noFUDGe*