Swiss Numerical Analysis Day 2015,
Geneva, Switzerland, 17 April 2015.

Efficient BDF time discretization of the Navier–Stokes equations with VMS–LES modeling in a High Performance Computing framework

L. Dede', D. Forti, S. Deparis, and A. Quarteroni

CMCS – Chair of Modelling and Scientific Computing
MATHICSE – Mathematics Institute of Computational Science and Engineering
EPFL – École Polytechnique Fédérale de Lausanne
Switzerland

Introduction

Applications

Turbulent or transitional flows occur in many physical contexts:

- External flows (civil Engineering, Hydrodynamics, or Aeronautical applications);
- Internal flows (e.g. Haemodynamics).

DNS vs. LES

The Direct Numerical Simulation (DNS) is generally computationally challenging or not affordable, for which Large Eddy Simulations (LES) models can instead be used.

Numerical Approximation (LES)

Even with LES modeling, efficient and flexible solvers for the incompressible Navier–Stokes equations, e.g. based on the Finite Elements method, are required to make feasible the numerical simulation, also in a High Performance Computing HPC setting.

Motivations

Goals

- Develop an efficient solver for LES modeling of the incompressible Navier–Stokes equations.
- Use the Finite Elements method for the spatial approximation.
- Use efficient time discretization schemes.
- Use a scalable solver for parallel computing (HPC).
- Solve large scale problems at high Reynolds ($\mathbb{R}e$) numbers.

Methodology

- Variational Multiscale (VMS) approach with LES modeling (**VMS–LES**) [Bazilevs et al., CMAME, 2007], [Codina et al., CMAME, 2007], [Wall et al, CMAME 2010].
- Backward Differentiation Formulas (BDF) for the time discretization to define a semi–implicit scheme (**VMS–LES/BDF**).
- Scalable solver for HPC with Multigrid (ML) preconditioner.

The incompressible Navier–Stokes equations

Navier–Stokes equations for incompressible flows (Newtonian)

$$\begin{aligned} \rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \boldsymbol{f} - \nabla\cdot\boldsymbol{\sigma}(\boldsymbol{u},p) &= 0 \qquad \text{in } \Omega \times (0,T), \\ \nabla\cdot\boldsymbol{u} &= 0 \qquad \text{in } \Omega \times (0,T), \end{aligned}$$

with suitable initial and boundary conditions on $\Gamma_D \subseteq \partial\Omega$; $\boldsymbol{u}$ and p are the velocity and the pressure, ρ the density, $\boldsymbol{f}$ the external forces, $\boldsymbol{\sigma}(\boldsymbol{u},p) = -p\mathbf{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u})$ the stress tensor, μ the dynamic viscosity, and $\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)$ the strain tensor.

Weak formulation

The weak formulation of the Navier–Stokes equations reads:

$$\begin{aligned} &\text{find } \boldsymbol{U} = \{\boldsymbol{u},\, p\} \in \boldsymbol{\mathcal{V}}_{\mathbf{g}} \quad : \\ &(\boldsymbol{w}, \rho\tfrac{\partial \boldsymbol{u}}{\partial t}) + (\boldsymbol{w}, \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}) + (\nabla\boldsymbol{w}, \mu(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)) - (\nabla\cdot\boldsymbol{w}, p) + (q, \nabla\cdot\boldsymbol{u}) = (\boldsymbol{w}, \boldsymbol{f}) \\ &\qquad \text{for all } \boldsymbol{W} = \{\boldsymbol{w}, q\} \in \boldsymbol{\mathcal{V}}_{\mathbf{0}},\ \forall t \in (0,T), \end{aligned} \tag{1}$$

with the function spaces $\mathcal{V}_{\mathbf{g}} = \{\boldsymbol{u} \in \left[H^1(\Omega)\right]^d \ :\ \mathbf{u}|_{\Gamma_D} = \mathbf{g}\}$, $\mathcal{V}_{\mathbf{0}} = \{\boldsymbol{u} \in \left[H^1(\Omega)\right]^d \ :\ \mathbf{u}|_{\Gamma_D} = \mathbf{0}\}$, $\mathcal{Q} = L^2(\Omega)$, $\boldsymbol{\mathcal{V}}_{\mathbf{g}} = \mathcal{V}_{\mathbf{g}} \times \mathcal{Q}$, and $\boldsymbol{\mathcal{V}}_{\mathbf{0}} = \mathcal{V}_{\mathbf{0}} \times \mathcal{Q}$.

Spatial approximation: Finite Elements and VMS–LES modelling

Finite Elements discretization

We define $X_r^h := \left\{ v^h \in C^0(\overline{\Omega}) \ : \ v^h|_K \in \mathbb{P}_r, \text{ for all } K \in \mathcal{T}_h \right\}$ as the Finite Elements function space of degree $r \geq 1$ over Ω triangulated with a mesh $\mathcal{T}_h$ of tetrahedrons; h_K is the diameter of the element $K \in \mathcal{T}_h$.

Multiscale direct–sum decomposition

The space $\boldsymbol{\mathcal{V}}$ ($\boldsymbol{\mathcal{V}}_\mathbf{g}$ or $\boldsymbol{\mathcal{V}}_\mathbf{0}$) is decomposed into the coarse and fine scales subspaces:

$$\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{V}}^h \oplus \boldsymbol{\mathcal{V}}',$$

$\boldsymbol{\mathcal{V}}^h$ is the coarse scale function space associated to the Finite Elements discretization X_r^h and $\boldsymbol{\mathcal{V}}'$ is an infinite dimensional function space representing the fine scales. For the weak formulation (1), we apply the decomposition to the trial and test functions:

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{w}^h + \boldsymbol{w}', \\ q &= q^h + q', \\ \boldsymbol{u} &= \boldsymbol{u}^h + \boldsymbol{u}', \\ p &= p^h + p'. \end{aligned}$$

Spatial approximation: Finite Elements and VMS–LES modelling

VMS–LES modeling [Bazilevs et al., CMAME, 2007]

The fine scale velocity and pressure components are chosen as:

$$\begin{aligned} \boldsymbol{u}' &\simeq -\tau_M(\boldsymbol{u}^h)\, \boldsymbol{r}_M(\boldsymbol{u}^h, p^h), \\ p' &\simeq -\tau_C(\boldsymbol{u}^h)\, r_C(\boldsymbol{u}^h); \end{aligned}$$

$\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)$ and $r_C(\boldsymbol{u}^h)$ are the residuals of the momentum and continuity equations:

$$\begin{aligned} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) &= \rho\frac{\partial \boldsymbol{u}^h}{\partial t} + \rho\boldsymbol{u}^h \cdot \nabla\boldsymbol{u}^h + \nabla p^h - \mu\Delta\boldsymbol{u}^h - \boldsymbol{f}, \\ \boldsymbol{r}_C(\boldsymbol{u}^h) &= \nabla\cdot\boldsymbol{u}^h. \end{aligned}$$

The stabilization parameters τ_M and τ_C are set as:

$$\tau_M = \tau_M(\boldsymbol{u}^h) = \left(\frac{\sigma^2\rho^2}{\Delta t^2} + \frac{\rho^2}{h_K^2}|\boldsymbol{u}^h|^2 + \frac{\mu^2}{h_K^4}C_p\right)^{-1/2}, \qquad (2)$$

$$\tau_C = \tau_C(\boldsymbol{u}^h) = \frac{h_K^2}{\tau_M(\boldsymbol{u}^h)}, \qquad (3)$$

with σ a constant equal to the order of the time discretization chosen with time step Δt and $C_p = 60 \cdot 2^{r-2}$ a constant related to an inverse inequality.

Spatial approximation: Finite Elements and VMS–LES modelling

Semi–discrete VMS–LES formulation (weak residual)

$$\text{find } \boldsymbol{U}^h = \{\boldsymbol{u}^h,\, p^h\} \in \boldsymbol{\mathcal{V}}_{\mathbf{g}}^h \quad : \quad A(\boldsymbol{W}^h, \boldsymbol{U}^h) - F(\boldsymbol{W}^h) = 0$$

$$\text{for all } \boldsymbol{W}^h = \{\boldsymbol{w}^h,\, q^h\} \in \boldsymbol{\mathcal{V}}_{\mathbf{0}}^h,\ \forall t \in (0, T),$$

where:

$$A(\boldsymbol{W}^h, \boldsymbol{U}^h) := A^{NS}(\boldsymbol{W}^h, \boldsymbol{U}^h) + A^{VMS}(\boldsymbol{W}^h, \boldsymbol{U}^h)$$

$$F(\boldsymbol{W}^h) := (\boldsymbol{w}^h, \boldsymbol{f}),$$

$$A^{NS}(\boldsymbol{W}^h, \boldsymbol{U}^h) := \left(\boldsymbol{w}^h, \rho\, \frac{\partial \boldsymbol{u}^h}{\partial t}\right) + \left(\boldsymbol{w}^h, \rho(\boldsymbol{u}^h \cdot \nabla)\boldsymbol{u}^h\right) + \left(\nabla \boldsymbol{w}^h, \mu(\nabla \boldsymbol{u}^h + (\nabla \boldsymbol{u}^h)^T)\right) - \left(\nabla\cdot \boldsymbol{w}^h, p^h\right) + \left(q^h, \nabla\cdot \boldsymbol{u}^h\right),$$

$$A^{VMS}(\boldsymbol{W}^h, \boldsymbol{U}^h) := \left(\rho \boldsymbol{u}^h \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \tau_M(\boldsymbol{u}^h)\, \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right) - \left(\nabla\cdot \boldsymbol{w}^h, \tau_C(\boldsymbol{u}^h)\, r_C(\boldsymbol{u}^h)\right) + \left(\rho \boldsymbol{u}^h \cdot (\nabla \boldsymbol{u}^h)^T, \tau_M(\boldsymbol{u}^h)\, \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right) - \left(\nabla \boldsymbol{w}^h, \tau_M(\boldsymbol{u}^h)\, \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \otimes \tau_M(\boldsymbol{u}^h)\, \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right).$$

VMS–SUPG and VMS–LES terms

Time discretization: implicit BDF schemes

Backward Differentiation Formulas (BDF)

We partition $[0, T]$ into N_t subintervals of size $\Delta t = \frac{T}{N_t}$ with $t_n = n\,\Delta t$ for $n = 0, \ldots, N_t$; $\boldsymbol{u}_n^h \approx \boldsymbol{u}^h(t_n)$ and $p_n^h \approx p^h(t_n)$.
Depending the order σ of the BDF scheme, the approximation of the time derivative of the velocity reads:

$$\frac{\partial \boldsymbol{u}^h}{\partial t} \approx \frac{\alpha_\sigma \boldsymbol{u}_{n+1}^h - \boldsymbol{u}_{n,\mathrm{BDF}\sigma}^h}{\Delta t},$$

where for BDF schemes of orders $\sigma = 1, 2, 3$ we have:

$$\boldsymbol{u}_{n,\mathrm{BDF}\sigma}^h = \begin{cases} \boldsymbol{u}_n^h & \text{if } n \geq 1, \quad \text{for } \sigma = 1 \quad (\mathrm{BDF1}), \\ 2\boldsymbol{u}_n^h - \frac{1}{2}\boldsymbol{u}_{n-1}^h & \text{if } n \geq 2, \quad \text{for } \sigma = 2 \quad (\mathrm{BDF2}), \\ 3\boldsymbol{u}_n^h - \frac{3}{2}\boldsymbol{u}_{n-1}^h + \frac{1}{3}\boldsymbol{u}_{n-2}^h & \text{if } n \geq 3, \quad \text{for } \sigma = 3 \quad (\mathrm{BDF3}), \end{cases}$$

$$\alpha_\sigma = \begin{cases} 1, & \text{for } \sigma = 1 \quad (\mathrm{BDF1}), \\ \frac{3}{2}, & \text{for } \sigma = 2 \quad (\mathrm{BDF2}), \\ \frac{11}{6}, & \text{for } \sigma = 3 \quad (\mathrm{BDF3}). \end{cases}$$

Time discretization: implicit BDF schemes

The fully implicit BDF scheme for VMS–LES modeling

For a BDF scheme of order σ:

find $\boldsymbol{u}_{n+1}^h \in \mathcal{V}_{\mathbf{g}}^h$ and $p_{n+1}^h \in \mathcal{Q}^h$:

$$\left(\boldsymbol{w}^h, \rho\, \frac{\alpha_\sigma \boldsymbol{u}_{n+1}^h - \boldsymbol{u}_{n,\mathrm{BDF}\sigma}^h}{\Delta t}\right) + \left(\boldsymbol{w}^h, \rho\, (\boldsymbol{u}_{n+1}^h \cdot \nabla)\boldsymbol{u}_{n+1}^h\right)$$
$$+ \left(\nabla \boldsymbol{w}^h, \mu\, \left(\nabla \boldsymbol{u}_{n+1}^h + (\nabla \boldsymbol{u}_{n+1}^h)^T\right)\right) - \left(\nabla\cdot\boldsymbol{w}^h, p_{n+1}^h\right) + \left(q^h, \nabla\cdot\boldsymbol{u}_{n+1}^h\right)$$
$$+ \left(\rho\, \boldsymbol{u}_{n+1}^h \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \tau_M(\boldsymbol{u}_{n+1}^h)\, \boldsymbol{r}_M(\boldsymbol{u}_{n+1}^h, p_{n+1}^h)\right) - \left(\nabla\cdot\boldsymbol{w}^h, \tau_C(\boldsymbol{u}_{n+1}^h)\, r_C(\boldsymbol{u}_{n+1}^h)\right)$$
$$+ \left(\rho\, \boldsymbol{u}_{n+1}^h \cdot (\nabla \boldsymbol{w}^h)^T, \tau_M(\boldsymbol{u}_{n+1}^h)\, \boldsymbol{r}_M(\boldsymbol{u}_{n+1}^h, p_{n+1}^h)\right)$$
$$- \left(\nabla \boldsymbol{w}^h, \tau_M(\boldsymbol{u}_{n+1}^h)\, \boldsymbol{r}_M(\boldsymbol{u}_{n+1}^h, p_{n+1}^h) \otimes \tau_M(\boldsymbol{u}_{n+1}^h)\, \boldsymbol{r}_M(\boldsymbol{u}_{n+1}^h, p_{n+1}^h)\right)$$
$$= \left(\boldsymbol{w}^h, \boldsymbol{f}_{n+1}\right),$$

for all $\boldsymbol{w}^h \in \mathcal{V}_{\mathbf{0}}^h$ and $q^h \in \mathcal{Q}^h$, $\forall n \geq \sigma - 1$,

given $\boldsymbol{u}_n^h, \ldots, \boldsymbol{u}_{n+1-\sigma}^h$, with $\boldsymbol{f}_{n+1} = \boldsymbol{f}(t_{n+1})$.

The problem is nonlinear both in $\boldsymbol{u}_{n+1}^h$ and p_{n+1}^h.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Time discretization: the semi–implicit BDF scheme (VMS–LES/BDF)

Extrapolation with Newton–Gregory backward polynomials

The variables $\boldsymbol{u}_{n+1}^h$ and p_{n+1}^h are linearized by means of extrapolation with Newton–Gregory backward polynomials.

We consider the following extrapolations of orders $\sigma = 1, 2, 3$ for the velocity and pressure variables at the discrete time t_{n+1}:

$$\boldsymbol{u}_{n+1,\sigma}^h = \begin{cases} \boldsymbol{u}_n^h & \text{if } n \geq 0, & \text{for } \sigma = 1 \ \text{(BDF1)}, \\ 2\,\boldsymbol{u}_n^h - \boldsymbol{u}_{n-1}^h & \text{if } n \geq 1, & \text{for } \sigma = 2 \ \text{(BDF2)}, \\ 3\,\boldsymbol{u}_n^h - 3\,\boldsymbol{u}_{n-1}^h + \boldsymbol{u}_{n-2}^h & \text{if } n \geq 2, & \text{for } \sigma = 3 \ \text{(BDF3)} \end{cases}$$

$$p_{n+1,\sigma}^h = \begin{cases} p_n^h & \text{if } n \geq 0, & \text{for } \sigma = 1 \ \text{(BDF1)}, \\ 2\,p_n^h - p_{n-1}^h & \text{if } n \geq 1, & \text{for } \sigma = 2 \ \text{(BDF2)}, \\ 3\,p_n^h - 3\,p_{n-1}^h + p_{n-2}^h & \text{if } n \geq 2, & \text{for } \sigma = 3 \ \text{(BDF3)}. \end{cases}$$

The extrapolations of $\boldsymbol{u}_{n+1}^h$ and p_{n+1}^h induce similar extrapolations on the residuals and stabilization parameters.

Time discretization: the semi–implicit BDF scheme

The semi–implicit BDF scheme for VMS–LES modeling (**VMS–LES/BDF**)

For a BDF scheme of order σ:

find $\boldsymbol{u}^h_{n+1} \in \mathcal{V}^h_{\mathbf{g}}$ and $p^h_{n+1} \in \mathcal{Q}^h$:

$$\left(\boldsymbol{w}^h, \rho\, \frac{\alpha_\sigma \boldsymbol{u}^h_{n+1} - \boldsymbol{u}^h_{n,\mathrm{BDF}\sigma}}{\Delta t}\right) + \left(\boldsymbol{w}^h, \rho\, (\boldsymbol{u}^h_{n+1,\sigma} \cdot \nabla)\boldsymbol{u}^h_{n+1}\right) + \left(\nabla \boldsymbol{w}^h, \mu \left(\nabla \boldsymbol{u}^h_{n+1} + (\nabla \boldsymbol{u}^h_{n+1})^T\right)\right)$$

$$- \left(\nabla\cdot \boldsymbol{w}^h, p^h_{n+1}\right) + \left(q^h, \nabla\cdot \boldsymbol{u}^h_{n+1}\right) + \left(\rho\, \boldsymbol{u}^h_{n+1,\sigma} \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \tau_M^{n+1,\sigma}\, \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right)$$

$$- \left(\nabla\cdot \boldsymbol{w}^h, \tau_C^{n+1,\sigma}\, r_C(\boldsymbol{u}^h_{n+1})\right) + \left(\rho\, \boldsymbol{u}^h_{n+1,\sigma} \cdot (\nabla \boldsymbol{w}^h)^T, \tau_M^{n+1,\sigma}\, \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right)$$

$$- \left(\nabla \boldsymbol{w}^h, \tau_M^{n+1,\sigma} \widehat{\boldsymbol{r}}_M^{n+1,\sigma} \otimes \tau_M^{n+1,\sigma} \widetilde{\boldsymbol{r}}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right)$$

$$- \left(\nabla \boldsymbol{w}^h, \tau_M^{n+1,\sigma} \widehat{\boldsymbol{r}}_M^{n+1,\sigma} \otimes \tau_M^{n+1,\sigma}\, \rho\, \alpha_\sigma\, \frac{\boldsymbol{u}^h_{n+1}}{\Delta t}\right)$$

$$+ \left(\nabla \boldsymbol{w}^h, \tau_M^{n+1,\sigma}\, \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) \otimes \tau_M^{n+1,\sigma}\, \rho\, \frac{\boldsymbol{u}^h_{n,BDF}}{\Delta t}\right) = \left(\boldsymbol{w}^h, \boldsymbol{f}_{n+1}\right),$$

for all $\boldsymbol{w}^h \in \mathcal{V}^h_{\mathbf{0}}$ and $q^h \in \mathcal{Q}^h$, $\forall n \geq \sigma - 1$,

given $\boldsymbol{u}^h_n, \ldots, \boldsymbol{u}^h_{n+1-\sigma}$.

The problem is linear both in $\boldsymbol{u}^h_{n+1}$ and p^h_{n+1}.

Time discretization: the semi–implicit BDF scheme

Extrapolated stabilization parameters

$$\tau_M^{n+1,\sigma} := \left(\frac{\sigma^2 \rho^2}{\Delta t^2} + \frac{\rho^2}{h_K^2} \left| \boldsymbol{u}^h_{n+1,\sigma} \right|^2 + \frac{\mu^2}{h_K^4} C_p \right)^{-1/2},$$

$$\tau_C^{n+1,\sigma} = \frac{h_K^2}{\tau_M^{n+1,\sigma}}.$$

Extrapolated residuals

$$\boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) := \rho \left(\frac{\alpha_\sigma \boldsymbol{u}^h_{n+1} - \boldsymbol{u}^h_{n,\mathrm{BDF}\sigma}}{\Delta t} \right) + \rho\, \boldsymbol{u}^h_{n+1,\sigma} \cdot \nabla \boldsymbol{u}^h_{n+1} + \nabla p^h_{n+1} - \mu\, \Delta \boldsymbol{u}^h_{n+1} - \boldsymbol{f}_{n+1},$$

$$\widehat{\boldsymbol{r}}_M^{n+1,\sigma} := \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1,\sigma}, p^h_{n+1,\sigma}),$$

$$\widetilde{\boldsymbol{r}}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) := \rho \boldsymbol{u}^h_{n+1,\sigma} \cdot \nabla \boldsymbol{u}^h_{n+1} + \nabla p^h_{n+1} - \mu \Delta \boldsymbol{u}^h_{n+1} - \boldsymbol{f}_{n+1}.$$

The extrapolation of the stabilization parameters and residuals is based on the extrapolated variables $\boldsymbol{u}^h_{n+1,\sigma}$ and $p^h_{n+1,\sigma}$.

Time discretization: the semi–implicit BDF scheme

Remarks

- The fully discrete semi–implicit formulation (VMS–LES/BDF) yields a linear problem in the variables $\boldsymbol{u}_{n+1}^h$ and p_{n+1}^h to be solved only once at each time step t_n.
- At the discrete level, assembling and solving the linear system is done only once for each time step t_n (fewer terms than with Newton linearization).
- BDF schemes with extrapolation yields stable time discretizations of the Navier–Stokes equations at the continuous level (under suitable boundary conditions).
- The stability in respect of the time discretization for the VMS–LES/BDF scheme is not guaranteed a priori (for all Δt).
- In practice, stability in respect of the time discretization is also obtained for relatively large time steps Δt.

Finite Elements library *LifeV*

Numerical implementation

The implementation of the VMS–LES/BDF mehod for the incompressible Navier–Stokes equations is carried out in the Finite Elements Library *LifeV*.

LifeV

- Finite Elements Library for the numerical approximation of PDEs
- Research code oriented to the development and test of new numerical methods and algorithms
- C++, object oriented, parallel
- *LifeV* relies on *Trilinos* Library/packages
- Modular design; use of the Expression Template Assembly (ETA) package
- Developers: CMCS – EPFL, E(CM)2 – Emory, MOX – Polimi, REO, ESTIME – INRIA
- Distributed under LGPL, `http://www.lifev.org`

Multigrid (ML) preconditioner

Solver

We use the GMRES method through the *Belos* package of *Trilinos* to solve the linear system. The stopping criterion is based on the relative residual, with tolerance $tol = 10^{-6}$.
The right–preconditioning strategy is used.

ML Preconditioner

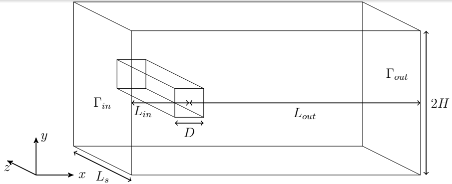
We use the Multigrid preconditioners available from the *ML* package of *Trilinos*. The preconditioner is built on the system matrix [Gee et. al, Sandia National Labs., 2006].

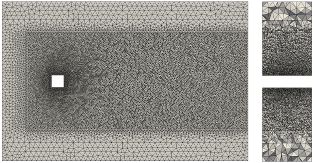
- Default parameters setting: `NSSA`, nonsymmetric smoothed aggregation variant for highly nonsymmetric operators;
- `max_levels` = 3, `cycle_applications` = 3, `pde_equations` = 4;
- smoother: Gauss-Seidel, 3 sweeps;
- aggregation: `type = Uncoupled-MIS`.

Flow past a squared cylinder

Benchmark problem

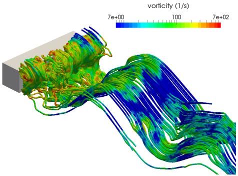
Flow past a squared cylinder at $\mathbb{R}e = 22{,}000$ [Koobus and Farhat, 2004] (experimental data available).

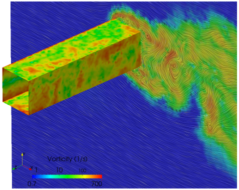


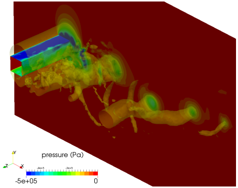


Space and time discretizations

- FEM: $\mathbb{P}1$–$\mathbb{P}1$ 1,323,056 DOFs; $\mathbb{P}2$–$\mathbb{P}2$ 9,209,040 DOFs.
- Different choices of BDF order ($\sigma = 1$ or 2) and Δt.



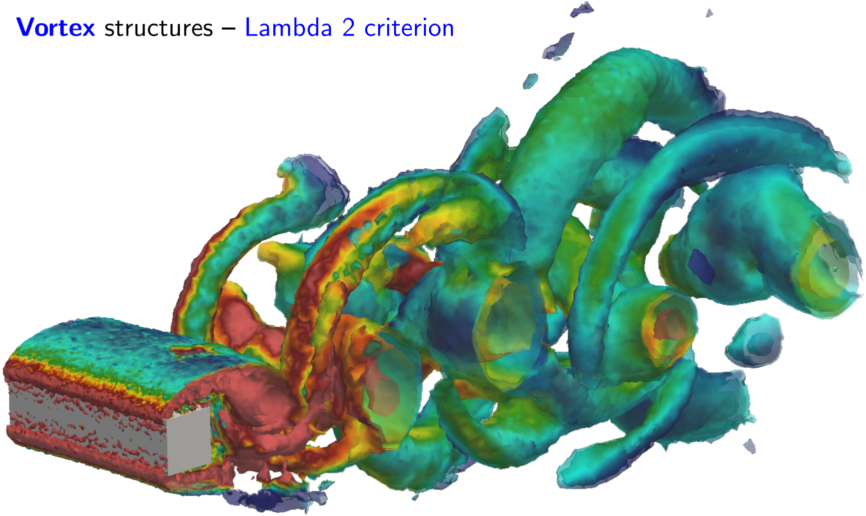




ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Flow past a squared cylinder

Vortex structures – Lambda 2 criterion

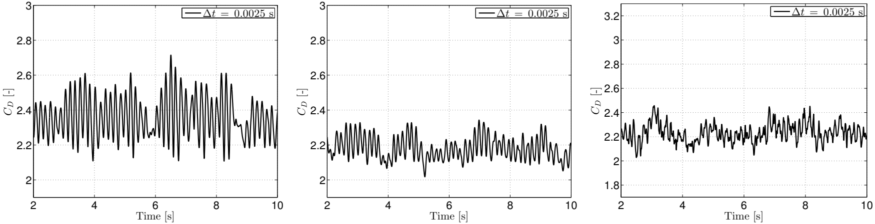


Flow past a squared cylinder

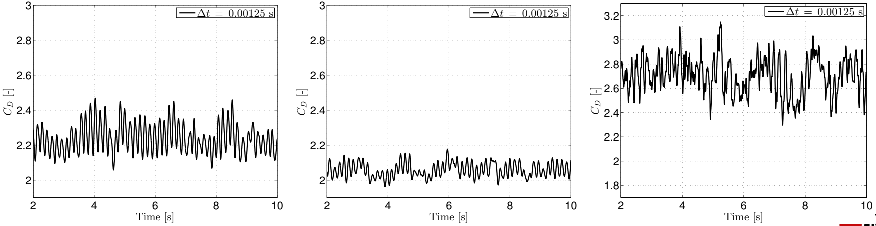
Comparison of drag coefficient C_D vs. time t for different discretizations

FEM $\mathbb{P}1$–$\mathbb{P}1$ and $\mathbb{P}2$–$\mathbb{P}2$; BDF orders $\sigma = 1$ and 2.

$\Delta t = 0.0025\ s$



$\Delta t = 0.00125\ s$



$\mathbb{P}1$–$\mathbb{P}1$, *BDF1* | $\mathbb{P}1$–$\mathbb{P}1$, *BDF2* | $\mathbb{P}2$–$\mathbb{P}2$, *BDF2*

Flow past a squared cylinder

Comparison of results for different discretizations and literature

FEM $\mathbb{P}1$–$\mathbb{P}1$ and $\mathbb{P}2$–$\mathbb{P}2$; BDF orders $\sigma = 1$ and 2.

Numerical Results

FEM	Δt	BDF σ	$\overline{C}_D$	rms(C_D)	rms(C_L)	Strouhal
$\mathbb{P}1$–$\mathbb{P}1$	0.005 s	1	2.49	0.23	1.49	0.133
	0.0025 s	1	2.35	0.11	1.18	0.138
	0.00125 s	1	2.24	0.08	0.89	0.142
	0.005 s	2	2.27	0.09	0.87	0.144
	0.0025 s	2	2.16	0.07	0.66	0.146
	0.00125 s	2	2.05	0.04	0.58	0.146
$\mathbb{P}2$–$\mathbb{P}2$	0.005 s	2	2.00	0.10	0.58	0.142
	0.0025 s	2	2.24	0.12	0.98	0.141
	0.00125 s	2	2.71	0.15	1.5	0.129

Literature (LES)

LES method	$\overline{C}_D$	rms(C_D)	rms(C_L)	Strouhal
VMS–F.V. [Koobus., 2004]	2.10	0.18	1.08	0.136
Smagorinsky [Rodi, 1997]	1.66–2.77	0.1–0.27	0.38–1.79	0.07–0.15
Dynamic LES [Sohankar, 2000]	2.00–2.32	0.16–0.20	1.23–1.54	0.127–0.135

CHNIQUE AUSANNE

Flow past a squared cylinder

Scalability results of the solver

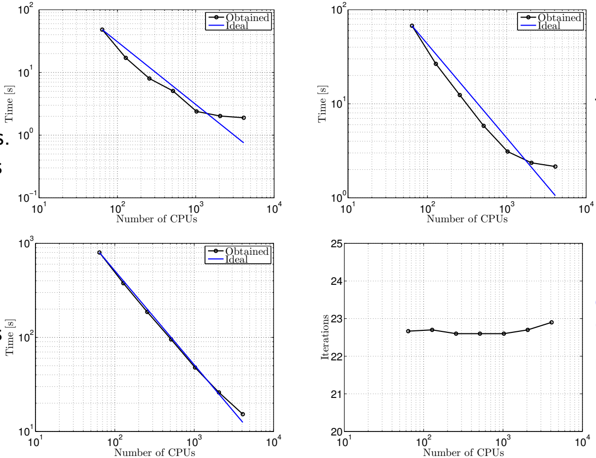
Linear solver based on GMRES and ML preconditioner; simulations performed using FEM $\mathbb{P}2$–$\mathbb{P}2$, *BDF2*, $\Delta t = 0.0025\,s$.

Preconditioner assembly time vs. number of CPUs

Time to solve the linear system vs. number of CPUs

Time to perform a time step vs. number of CPUs

Number of GMRES iterations vs. number of CPUs



Computations carried out with Piz Dora, a Cray XC40 supercomputer at the Swiss National Supercomputing Center (CSCS).

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Conclusions

- We considered a semi–implicit scheme based on BDF formulas and extrapolation for the time discretization of the Navier–Stokes equation with VMS–LES modeling (**VMS–LES/BDF**); spatial discretization was performed with low order Finite Elements method.
- We solved problems in a parallel computing framework (HPC), for which we showed the scalability of the solver with Multigrid preconditioning.
- We applied the method to internal and external flow problems at high Reynolds numbers.
- The numerical tests showed that the discretization based on the VMS–LES/BDF method is efficient, versatile, accurate, and stable for wide ranges of the time steps.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

References

- Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual–based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods Applied Mechanics and Engineering*, **197**:173–201, 2007.
- O. Colomés, S. Badia, R. Codina, and J. Principe. Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, **285**:32–63, 2015.
- D. Forti and L. Dedè. Semi–implicit BDF time discretization of the Navier–Stokes equations with VMS–LES modeling in a High Performance Computing framework. *MATHICSE report*, **09.2015**, 2015.
- B. Koobus and C. Farhat. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes. Application to vortex shedding. *Computer Methods Applied Mechanics and Engineering*, **193**:1367–1383, 2004.
- W. Rodi, J. Ferziger, M. Breuer, and M. Pourquié. Status of large eddy simulations: results of a workshop. *ASME Journal of Fluids Engineering*, **119**:248–262, 1997.