Dynamic Mesh Adaptation of massive unstructured grids for the simulation of flame fronts and gas/liquid interfaces

V. Moureau, P. Bénard, K. Bioche, J. Carmona, L. Voivenel, Y. Béchane
CORIA, CNRS UMR6614, Normandie Univ, UNIROUEN, INSA Rouen, France

P. Begou, G. Balarac, G. Ghigliotti,– LEGI, Grenoble, France

A. Froehly, C. Dobrzynski – LMB/INRIA Bordeaux, France

R. Mercier, M. Cailler, J. Leparoux, R. Letournel
SAFRAN Tech, Magny-les-H., France

NUMPEX workshop, 6-7 février 2024
Motivation
Motivation

PRECCINSTA burner

BIERA scheme - GTEILS model - non-adiabatic walls
Dynamic mesh adaptation with surface based on flame sensor

P. Berard, G. Lebrun, V. Moureau - COREA

T [K]

2200.0
2000.0
1800.0
1600.0
1400.0
1200.0
1000.0
800.0
600.0
400.0
300.0

Courtesy J. Vernier, J. Leparoux, R. Mercier, SAFRAN TECH
Motivation

Courtesy J. Vernier, J. Leparoux, R. Mercier, SAFRAN TECH
Objectives

• To design a **dynamic mesh adaptation** method under constraints:
  – Based on **unstructured grids** for complex geometries
  – Tailored for **distributed-memory** system (10,000+ cores)
  – **Efficient** enough to be called every 10 fluid iterations
  – Compatible with the modeling approach (**finite-volume + LES**)
  – No remeshing at material interfaces to avoid interpolation errors

• Some choices
  – **Isotropic** mesh adaptation (for now)
  – Parallelism handled by the flow solver (interpolation, data transfer)

• A first check if these objectives are reachable for tets
  – Reduced computation cost of fluid solver: **30 to 500 µs/iter/node**
  – Adaptation of a distributed mesh with **MMG**: order of 100 µs/node

mmgtools.org
A first attempt

- January 2015: first use of MMG directly in YALES2
High-performance dynamic mesh adaptation

- A combination of several kernels

Moving interface method [1, 2]

$M_{\text{init}}$  $M_{\text{target}}$

Parallel edge-cutting  Parallel load-balancing

High-performance dynamic mesh adaptation

- Full adaptation workflow
  - Example for 4 processors

- Skewness is the main measure of the mesh quality

\[
\text{if } S_k > S_{k_{max}} \quad \text{or} \quad \frac{|M - M_{\text{target}}|}{\min(M, M_{\text{target}})} > \epsilon_M
\]

Performance evolution for atomization problems

Flat-spray
8 μm
0.014 ms
24h / 1120 proc

Jet in-cross-flow
10 μm
0.02ms
24h / 3444p
20 μm
0.08ms
24h / 1036p

Multi-point
10 μm
0.05ms
24h / 1456p
20 μm
0.11ms
24h / 1120p

Cône creux
10 μm
0.2ms
24h / 1260p
20 μm
0.4ms
24h / 600p

Flat-spray
8 μm
0.200 ms
24h / 1120 proc


Reduced Efficiency μs.proc/noeuds/ite

Courtesy R. Mercier, SAFRAN TECH
CFD platform
The CFD platform

**YALES2 solvers**
- ICS: Incompressible at constant density
- VDS: Incompressible at variable density
- SPS: Spray with level-set and ghost-fluid method
- ALE: Arbitrary Lagrangian Eulerian
- GFS: Granular flow with Discrete Element Method
- HTS: Heat transfer
- MHD: Magneto-hydro-dyn
- SMS+FS: Structural mech.
- ACS: Acoustics
- BOI: Boiling
- ECS: Compressible flow

**YALES2 library**

- Differential operators
- Dynamic load balancing
- Parallel I/Os
- Linear solvers
- Dynamic mesh refinement
- Stiff integrators
- Turbulence models
- Transported chemistry
- Level-set interface
- Heat & mass transfer / level-set
- Lagrangian transport
- Discrete Element Method
- Discrete ordinates method
- Look-up table management
- Hybrid parallel communications
- Object-oriented & parallel data structures
- Dynamic mesh adaptation
- DoE / workflow management
- Structural mechanics
- Acoustics
- Boiling
- Compressible flow
- High fidelity
- Multiphysics
- High performance

**Key figures**
- 17 major releases
- 930 000 object-oriented F2008 lines
- 20 000+ commits
- 10 to 20 commits/day
- 300+ active branches
- 1 200+ merge requests
- 120+ contributors
CFD platform: YALES2

- **Features**
  - Unstructured meshes and adaptive grid refinement
  - Low-Mach number Navier-Stokes equations (incompressible and variable density)
  - 4-level domain decomposition [3] and hybrid OpenMP/MPI communications
  - Highly efficient solvers for linear system inversion (PCG, DPCG) [4]
  - 4th-order central **finite-volume method** and 4th-order time integration
  - Two-phase flows (Lagrangian particles), **spray and atomization** (Levelset)
  - Combustion modeling (Tabulated or **finite-rate chemistry**, NOx model, …)
  - Suited for massively parallel computing (>32 000 procs)

The **YALES2 network**

- Developed by CORIA, the French Combustion Community and others
  - 600+ researchers/engineers trained at CORIA since 2009
  - 200+ articles (Google Scholar)
- A unique network to ease collaboration and disseminate numerics, algorithms and models to the community

**Academic partners**
- CORIA, IMAG, LEGI, EM2C, IMFT
- CERFACS, ONERA, IFP-EN, LMA, LJK
- LOMC, PPRIME, LMB/INRIA, ICARE
- ULB, UMONS, UCL, VUB, TU DELFT
- CORNELL U., VERMONT U., SHERBROOKE U.

**HPC centers**
- CRIANN, IDRIS, CINES, TGCC
- ROMEO, GENCI, PRACE

**Industrial partners**
- SAFRAN
- ARIANE GROUP
- GE HYDRO
- SIEMENS/GAMESA
- AIR LIQUIDE
- TOTAL ENERGIES
- ... 

**HPC experts**
- INTEL/CEA/GENCI/UVSQ, HPE
- ATOS, IBM, AMD, NVIDIA

**SMEs**
- GDTech
- YPSO FACTO

www.coria-cfd.fr
The YALES2 network

EXTREME CFD 7TH WORKSHOP & HACKATHON

January 22nd – February 2nd 2024
Merville-Franceville, FRANCE
A few applications
DMA of spray combustion

- LES simulations with finite-rate chemistry using dynamic adaptive mesh refinement.
  - Chemistry: ARC for n-heptane (CERFACS), 25 species, 210 reactions
  - Spray: Lagrangian Particle with dynamic load balancing (Stock et al., IJNMF 2023)

\( \Delta x_{\text{min}} = 200 \, \mu m \),
(300M cells)
(10,000 cores)
Jean-Zay @ IDRIS

A. Stock, V. Moureau, Feature-based adaptive mesh refinement for multi-regime reactive flows, Sub. PROCI, 2024
\[ \Delta x_{\text{min}} = 10 \mu m \]

\[ We_g = \frac{\rho_g u_g^2 d_{\text{inj}}}{\sigma} = 1470 \]

\[ q = \frac{\rho_l u_l^2}{\rho_g u_g^2} = 6 \]

---

R. Janodet\textsuperscript{a},b, C. Guillamón\textsuperscript{b}, V. Moureau\textsuperscript{a}, R. Mercier\textsuperscript{b}, G. Lartigue\textsuperscript{a}, P. Bénard\textsuperscript{a}, T. Ménard\textsuperscript{a}, A. Berlemont\textsuperscript{a}

\textsuperscript{a}CORIA, CNRS UMR6614, IRSA and University of Rouen, France

\textsuperscript{b}Safran Tech, Rue des Jeunes Bijoux, Chateauroux, 73114 Magny-Le-Hongre, France

DLR jet-in-cross-flow [1]

- Flow topology analysis

\[ \Delta x = 10 \, \mu m, \ \text{1B cells, 15 600 liquid structures, 10 000 cores of Irene AMD, TGCC, CEA} \]

- Eulerian/Lagrangian coupling in the works (I. El Yamani PhD thesis)

Modeling of fire resistance tests for composite materials

• Kerosene spray burner
  – BFER mechanism for kerosene, 6 species, 2 reactions
• Coupling of 3 solvers
  – combustion, conduction, radiative heat transfer
    (Discrete Ordinate Method)

Courtesy R. Letournel, SAFRAN TECH
Wind turbines

- Impact of yaw on wake development behind offshore wind turbines
- Collaboration with SIEMENS/GAMESA Renewable Energies

$t = 0.00 \text{ s}$

F. Houtin Mongrolle, P. Benard, G. Lartigue, V. Moureau

4 cases: 2 inflows and 2 yaw angles
Resources: 8448 cores/case on Irene AMD Rome
Mesh sizes: 1.5B – 1.7B

$\gamma = 0^\circ$

$\gamma = 30^\circ$
Simulations in the framework of the COVID pandemic

Time: 3.000498

Droplet diameter
1.0e-07 1e-5 2e-5 3e-5 4e-5 5.0e-05
Performance optimization
Importance of memory accesses in unstructured FV codes

- Code performance on a CPU can be limited by:
  - Processor speed (compute bound)
  - **Memory access speed** (memory bound)
  - The roofline model

![Diagram showing the relationship between arithmetic intensity and attainable performance in CFD methods.](image-url)
CPU: Parallel computation and domain decomposition

- Large problems cannot be computed by a single process
- Domain decomposition to divide the problem amongst many processes
  - More memory available
  - More computational power
  - Communication needed

Data on these nodes have to be exchanged between processes.
MPI/OpenMP + cache-blocking: 3-level domain decomposition

MPI + OpenMP + in-thread domain decomposition

Full MPI [1]

- Cell groups for cache-blocking
- Cell group size ~ 2000
- **Overhead: node duplication**

Coarse-grain Hybrid OpenMP + MPI

- Threads substitute MPI ranks
- Fewer MPI ranks but thread-safety
- **Overhead:**
  - Threads must communicate
  - Node duplication

Hierarchical domain decomposition in YALES2

• Explicit partitioning

Level #0
Fluid domain

Level #1
MPI #0
MPI #1
MPI #2
MPI #3

Level #2
TH #0
TH #1
TH #0
TH #1
TH #0
TH #1
TH #0
TH #1

- Task sharing

Level #3
El. groups

- Multi-block IOs
- Profiling
- Block adaptation

- Cache-blocking
- GPU
- Load balancing
- Preconditioning

• Implicit partitioning

Level #4
Sub. el. groups

- Interpolation
- Part. localization

Level #5
Pair/node/particle groups

- Vectorization
- Graph operators
Dynamic load balancing for Euler/Lagrange models

- A difficult task which requires: mesh coloring, data movement, ...
- Balancing cells and particles requires 2-constraints optimization [1]

Dynamic load balancing for Euler/Lagrange models

GPUs & APUs

- Working on OpenACC port since 2017
  - Benefits from array objects \((r_1_t, r_2_t, \ldots, i_1_t, i_2_t, \ldots)\)
  - GPU memory management in the data structures

```fortran
if (acc_is_present(r2%val)) then
  !$acc update device(r2)
  !$acc enter data create(r2%val)
  !$acc parallel loop collapse(2) present(r2%val,r2_tmp) copyin(new_allocdim1,new_allocdim2)
  do j=1,new_allocdim2
    do i=1,new_allocdim1
      r2%val(i,j) = r2_tmp(i,j)
    end do
  end do
  !$acc exit data delete(r2_tmp) finalize
end if
```
GPUs & APUs

- Status of OpenACC port in YALES2_2024.04
- One solver is ported (SCS), two main solvers on-going (ICS, ECS)
- Supported on
  - NVIDIA, Jean-Zay with gfortran and nvfortran
- Support to come
  - Adastra AMD MI250: SCS port is done with cce17 but results are still wrong
  - AMD MI300A (APU): CPU port done during ECFD7 Jan. 2024
Conclusions
Conclusions

• Data partitioning is the key of efficient unstructured adaptive mesh refinement.

• Many algorithms still need improvement
  – Multi-level contiguous parallel graph coloring
  – Collapse/face swap parallel kernels

• On-going work: anisotropic AMR
Mesh generation

- Arbitrary STL mesh adaptation