

DE LA RECHERCHE À L'INDUSTRIE



# Direct Numerical Simulation of bubbles with Adaptive Mesh Refinement (AMR) on parallel architecture

Low Velocity Flows, Paris

Arthur TALPAERT Grégoire ALLAIRE, Stéphane  
DELLACHERIE, Samuel KOKH, Anouar MEKKAS

CEA, École Polytechnique

2015-11-05

## Context

Direct Numerical Simulation

## Low-Mach flow

Physical model

Compressible two-phase Navier-Stokes equations

Low-Mach number approximation: DLMN model

## AMR strategy

Patch creation algorithms

## Simulation and speed-up

Interface advection and speed-up

## Simulation and LDC

Abstract Bubble Vibration model

Elliptic problem and multilevel AMR

Numerical results

## Conclusion

Perspectives

Acknowledgments

## References

## Back-up slides

AMR clockwork

Detail of patch-creation algorithms

## Direct Numerical Simulation

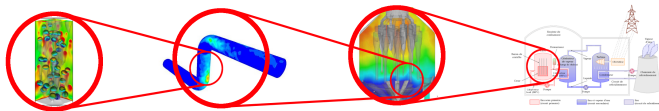


Figure: Successive zooms in representation (misc. sources incl. Bois, 2011)

The Direct Numerical Simulation is the most precise simulation of a thermal-hydraulic flow in terms of scale.

Strengths:

- potential for full knowledge of what happens at centimeter scale,
- no need for average-scale phenomenons as the drag force, void fraction based values, etc,
- can be used to recalculate the numerical value of some coefficients of closure laws used in larger scale models (needs to be validated).

Weaknesses:

- extremely high computational cost,
- extremely high data storage cost.

## Ways to reduce the computation effort

Though beneficial for precision, DNS is too costly as of now. Here are the ways we will explore to reduce the computation effort:

- **Physics** use a low-Mach number model neglecting some phenomenons,
- **Applied Maths** use an Adaptive Mesh Refinement for which only the most interesting areas are finely meshed,
- **Computer Science** use a parallel architecture to do simultaneous calculations.

## Physical model that we will use

In most situations, there are two very distinct time scales: *e.g.*  $\tau_{acoustic} \simeq 10^{-3}s$  and  $\tau_{matter} \simeq 1s$ .

This brings specific difficulties:

1. precision in schemes (*e.g.* the Godunov scheme has a poor precision),
2. robustness of solvers (for instance, inverting matrices with very different eigenvalues is hard with usual iterative methods, because of the bad conditioning).

In a nuclear reactor in particular, we are in low-Mach number conditions in the vast majority of cases, be it in nominal regime or in accidental regime (*Loss Of Flow Accidents*):  $M = \frac{|u|}{c} \simeq 10^{-3}$ .

We can say we can neglect shock waves and other acoustic phenomenons. However thermal phenomenons do remain important, so  $\nabla u \neq 0$  although  $M \ll 1$ .

$$M \ll 1 \\ |\nabla \cdot \lambda \nabla T| \ll 1$$

$$M \ll 1 \\ |\nabla \cdot \lambda \nabla T| = \mathcal{O}(1)$$

$$M = \mathcal{O}(1)$$

<i>Incomp. Navier-Stokes</i>	$\leq$	<i>Low-Mach asymptotic model</i>	$<$	<i>Comp. Navier-Stokes</i>
× <i>acoustic</i> , × <i>thermal</i>		× <b><i>acoustic</i></b> , ✓ <b><i>thermal</i></b>		✓ <i>acoustic</i> , ✓ <i>thermal</i>

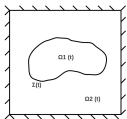
## The Diphasic Low-Mach Number model (DLMN)

This model takes its inspiration from previous work about combustion, at least since the seventies (see for instance Majda, 1982). Low-Mach number models have also been used in many others, as well as in cosmology (Almgren, Bell, Rendleman, & Zingale, 2006).

It was then proposed for Nuclear Reactors Thermal-Hydraulics (see Dellacherie, 2012) and its theory has been studied from an analytical point of view (Dellacherie, 2007; Penel, 2012; Gittel, Günther, & Ströhmer, 2014).

# Starting point: compressible two-phase Navier-Stokes equations I

Compressible two-phase Navier-Stokes equations:



$$\Omega = \Omega_1(t) \cup \Omega_2(t)$$

Initial condition:

$$\begin{cases} \partial_t Y + \mathbf{u} \cdot \nabla Y = 0 \\ \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla P + \nabla \cdot \tau(\mathbf{u}) + \rho \mathbf{g} \\ \partial_t(\rho E) + \nabla \cdot [(\rho E + P)\mathbf{u}] \\ \quad = \nabla \cdot (\lambda \nabla T) + \nabla \cdot [\tau(\mathbf{u})\mathbf{u}] + \rho \mathbf{g} \cdot \mathbf{u} \end{cases} \quad (1)$$

$$Y(t=0, \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_g(t=0) \quad (\text{i.e. gas/vapor}), \\ 0 & \text{if } \mathbf{x} \in \Omega_l(t=0) \quad (\text{i.e. liquid}). \end{cases} \quad (2)$$

Boundary conditions:  $\Omega$  is bounded and we have adiabatic conditions.

$$\forall \mathbf{x} \in \partial\Omega, \mathbf{u}(t, \mathbf{x}) = 0 \quad \text{et} \quad \nabla T(t, \mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad (3)$$

Transport and thermodynamic coefficients:

$$\xi(Y, T, P) = Y\xi_g(T, P) + (1 - Y)\xi_l(T, P), \quad \xi \in \{\lambda, \alpha, C_p, \dots\}. \quad (4)$$

For instance, we have  $\lambda(Y, T, P) = \begin{cases} \lambda_g(T, P) & \text{if } \mathbf{x} \in \Omega_g(t) \text{ (i.e. } Y = 1) \\ \lambda_l(T, P) & \text{if } \mathbf{x} \in \Omega_l(t) \text{ (i.e. } Y = 0) \end{cases}$

## Low-Mach number approximation: DLMN model I

We can show that, when the Mach number is small, we can approximate the previous model with the DLMN model (see Dellacherie, 2005).

Transport equation for  $Y$ :

$$\partial_t Y + \mathbf{u} \cdot \nabla Y = 0 \quad (5)$$

Transport equation for internal enthalpy:

$$\rho C_p (\partial_t T + \mathbf{u} \cdot \nabla T) = \alpha T P'(t) + \nabla \cdot (\lambda \nabla T) \quad (6)$$

with  $\alpha = -\frac{1}{\rho} \frac{\partial \rho}{\partial T}(Y_1, T, P)$  the coefficient of thermal expansion,  
and  $P$  a pressure depending only on time and equations of state.

Conservation equation for momentum:

$$\rho (\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla \Pi + \nabla \cdot \tau(\mathbf{u}) + \rho \mathbf{g} \quad (7)$$

with  $\Pi$  a perturbation of the pressure, or a dynamic pressure. It not driven by thermodynamics but rather the velocity field, gravity, etc.



## Low-Mach number approximation: DLMN model II

Coupling equation of constraints with all other equations:

$$\nabla \cdot \mathbf{u} = G(t, \mathbf{x}) \quad (8)$$

with

$$\begin{cases} G(t, \mathbf{x}) = -\frac{P'(t)}{\Gamma P(t)} + \frac{\beta}{P(t)} \nabla \cdot (\lambda \nabla T) \\ \beta = \frac{\alpha P}{\rho C_p}, \quad \Gamma = \frac{\rho c^2}{P} \end{cases} \quad (9)$$

and

$$P'(t) = \frac{\int_{\Omega} \beta(Y, T, P) \nabla \cdot \lambda \nabla T dx}{\int_{\Omega} \frac{dx}{\Gamma(Y, T, P)}} \quad (10)$$

Initial condition: same, with in addition  $\mathbf{u}^0(\mathbf{x})$  matching  $\nabla \cdot \mathbf{u}^0 = G(t=0, \mathbf{x})$ .  
Boundary conditions (external and internal on  $\Sigma(t)$ ) and transport and thermal coefficients: same.

## Relevance of an Adaptive Mesh Refinement (AMR)

Objective: Improve the precision of the calculation without making too much additional costly calculation.

We will adapt the mesh by refining it in the most relevant areas. Two mostly used techniques for multi-level AMR on cartesian grids:

1. tree-based AMR (*aka* cell-based), one non-conformal mesh
  2. patch-based AMR (*aka* block-based), multi-level conformal meshes
- easy to use for parallelization

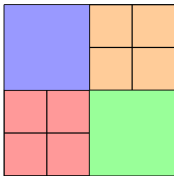


Figure: Tree-based AMR

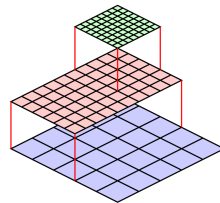


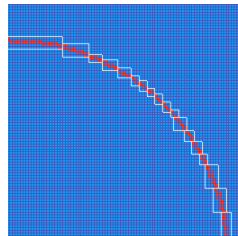
Figure: Patch-based AMR

(illustrations from Fikl, 2014)

## Different patch-creation algorithms

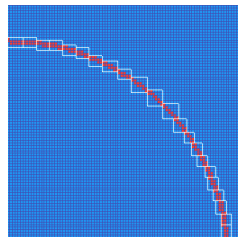
### Berger-Rigoutsos algorithm

- Introduced by Berger & Rigoutsos (Berger & Rigoutsos, 1991) and analyzed by Livne (Livne, 2006a).
- Covering the region of interest is the sole constraining goal.
- No constraint set on geometry.



### $l_{min} - l_{max}$ algorithm

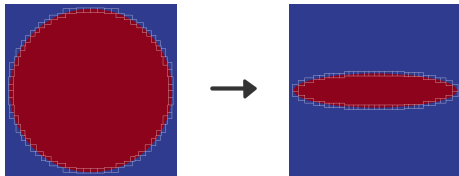
- New algorithm, inspired by and similar to the Livne algorithm (Livne, 2006b).
- Also includes geometrical constraints.
- Aims at improving load balance and minimizing communication between patches.



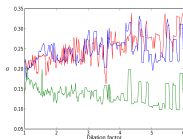
## Statistics for the relevance of patch algorithm

Series of test cases: ellipsoidal bubbles for which the ratio between longest and shortest radius (the dilation factor) goes from 1 to 6.

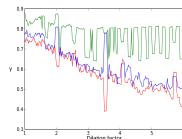
The problem is set in a grid of  $300\Delta x \times 300\Delta x$ .



*Normalized standard deviation* as a function of the dilation factor and of the patch creation algorithm (lower is better)



*Average squareness* as a function of the dilation factor and of the patch creation algorithm (higher is better)



$$\sigma = \sqrt{\frac{M(\{S_i^2\}_i) - M(\{S_i\}_i)^2}{\max(\{S_i^2\}_i) - \min(\{S_i\}_i)^2}}$$

$$\gamma = M\left(\left(\frac{\text{length of shortest side of patch } i}{\text{length of longest side of patch } i}\right)_i\right)$$

## Interface advection

We represent the two-phase environment with a discrete field:

$$Y(t, \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_g(t) & \text{(i.e. gas/vapor),} \\ 0 & \text{if } \mathbf{x} \in \Omega_l(t) & \text{(i.e. liquid).} \end{cases} \quad (11)$$

We want to implement the following advection equation:

$$\partial_t Y + \mathbf{u} \cdot \nabla Y = 0 \quad (12)$$

We will use the Després-Lagoutière anti-diffusive scheme since it is extremely well fitted for interface transport (see for instance Lagoutière, 2000).

Computational implementation:

Kothe-Rider test, with a periodic origin-centered advection flux. The spatial domain is a unit cube  $[0, 1] \times [0, 1] \times [0, 1]$  and the velocity field is periodic in time. It advects a spherical bubble back and forth. If the numerical schemes are ideal, then the initial and final positions of the sphere should coincide.

## Evolution of a 3D advected bubble

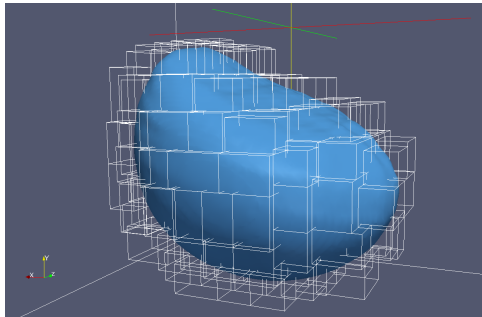


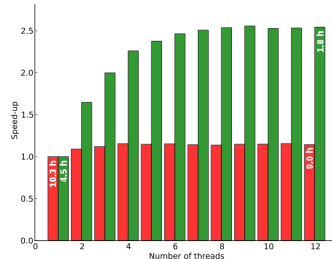
Figure: 3D simulation of a bubble with Kothe-Rider advection

Online video: <https://youtu.be/Ixgge4h6eF8>.

## Parallelization speed-up

We use the shared-memory technology OpenMP to set the parallelization. Let the speed-up be defined as  $su = \frac{\text{sequential computation time}}{\text{parallelized computation time}}$ . We use a  $200 \times 200 \times 200$  grid with one refinement level.

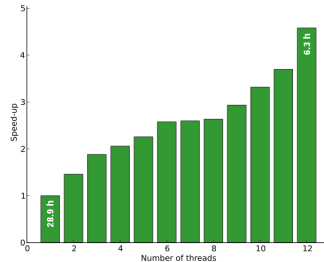
First test case:  
refinement factor = 6.



- 7113 patches for **Berger-Rigoutsos**
- 172 patches for  $l_{min} - l_{max}$

## Parallelization speed-up

Second test case:  
refinement factor = 8.



- 172 patches for  $l_{min} - l_{max}$  again
- the simulation with the Berger-Rigoutsos algorithm did not even pass the iteration 0, because of a lack of hardware memory (although we had 15.6 GiB of RAM).

Notice that this simulation is strictly equivalent to having more than four billion ( $4 \times 10^9$ ) fine cells.



## Abstract Bubble Vibration model

In addition to the advection equation, we coupled it with an elliptic equation which solution gives the velocity field  $\mathbf{u}(\mathbf{x}, t)$ . We represented this model, for which  $\mathbf{u}(\mathbf{x}, t)$  is potential (non-linear hyperbolic/elliptic coupling), with the following system of equations:

$$\begin{cases} \partial_t Y + \mathbf{u} \cdot \nabla Y = 0 \\ \Delta \phi = \psi(t) \left( Y - \frac{1}{\Omega} \int Y dx \right) \\ \nabla \phi \cdot \mathbf{n}|_{\partial\Omega} = 0 \\ \mathbf{u} = \nabla \phi \end{cases} \quad (13)$$

where  $\psi(t)$  is a given function and is the modelization of a vibration phenomenon. This model had already been theoretically analyzed and implemented to a large extent (Dellacherie & Lafitte, 2005), (Penel, Dellacherie, & Lafitte, 2013), (Mekkas, 2008).



Figure: Initial condition for  $Y$  field in 2D

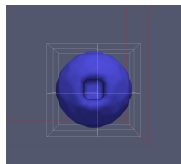


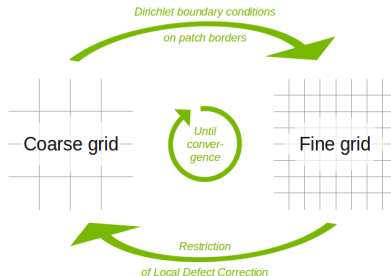
Figure: Initial condition for  $Y$  (and  $\phi$ ) fields in 3D

## Elliptic problem and multilevel AMR with LDC

In order to compute  $\phi$  in equation 13, we want to approximate the following equation on both the coarse and fine level, and that the information of one level helps with the resolution of the other level:

$$\begin{cases} \Delta\phi = \psi(t) \left( Y - \frac{1}{\Omega} \int Y dx \right) = f(x, t) \\ \nabla\phi \cdot \mathbf{n}|_{\partial\Omega} = 0 \end{cases} \quad (14)$$

The Local Defect Correction method (Hackbusch, 1984) is an efficient way to benefit from AMR (Anthonissen, Mattheij, & ten Thije Boonkamp, 2003) (Barbié, 2013):



## Numerical results for ABV in 2D

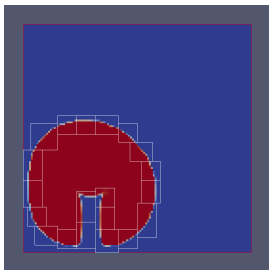


Figure:  $Y$  field

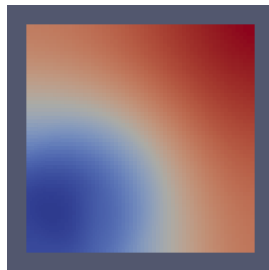


Figure:  $\phi$  field

Results at  $t = 3.12$ , i.e. at iteration 13, here with  $\psi(t)$  a cosine function.

We displayed the location of the patches on the left figure.

⇒ Online video: <https://youtu.be/Xyxfv3w88AQ>.

## Numerical results for ABV in 3D

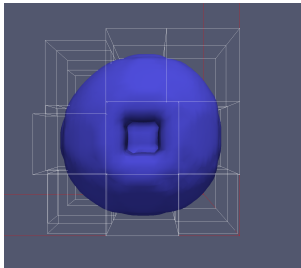


Figure: Contour of  $Y$  field

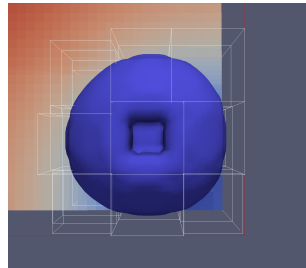


Figure: Both  $Y$  and  $\phi$  fields

Results at  $t = 3.046$ , i.e. at iteration 13, here with  $\psi(t)$  a cosine function.

We displayed the location of the patches on the left figure.

⇒ Online video: <https://youtu.be/-V2NmaUWAJM>.

## Verification and usefulness of AMR

We have a verification formula for the volume of the ABV bubble as a function of time  $t$  and of the pulsation  $\psi(t)$ :

$$C_0 = \log \left( \frac{V_{bubble}(t=0)}{V_\Omega - V_{bubble}(0)} \right), \quad \Psi(t) = \int_{t'=0}^{t'=t} \psi(t') dt' \quad (15)$$

$$V_{bubble}(t) = V_\Omega \frac{\exp(\Psi(t) + C_0)}{1 + \exp(\Psi(t) + C_0)} \quad (16)$$

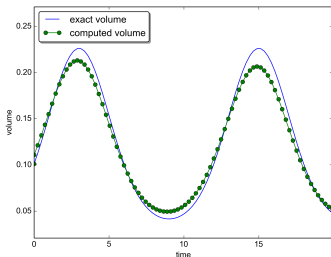


Figure: 30 × 30, no AMR

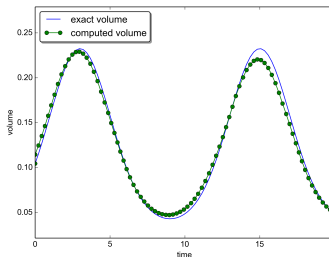


Figure: 60 × 60, no AMR

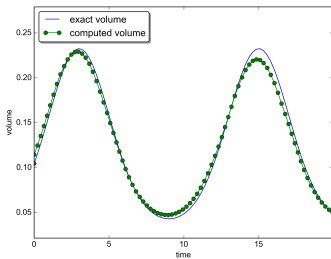


Figure:  $60 \times 60$ , no AMR

1 min 2 s of computation

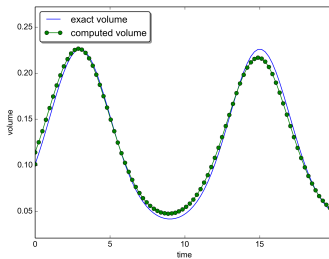


Figure:  $30 \times 30$ , AMR of factor 2

41 s of computation

This shows that with AMR, we get a similar precision to a simulation where the whole space is highly refined.

⇒ gain of 30% of computation time, although only 2D and without parallelization

## Perspectives

Short-term objectives: multi-phase incompressible Navier-Stokes simulation, with prediction-correction on staggered grid.

Test cases:

- inclusion of hot bubbles in a cold liquid environment; the bubbles should shrink. In a closed environment (pressure-cooker), the other bubbles should dilate,
- bubble column reactor, in order to compute closure laws,
- sloshing test cases.

Long-term objectives: full DLMN model simulation; multi-phase, incompressible but thermally expandable, in low-Mach number conditions.

## Conclusion

We have now set the first steps for a complete the Direct Numerical Simulation of the dilation of bubbles in a less costly manner:

- we have set some of the successive layers of how to model a bubble dilation and movement in low-Mach conditions,
- we set the base for a successful patch-based AMR,
- we obtain sizable speed-up of the computation thanks to parallelization.

Acknowledgments for supervision and funding:





## Bibliography I

- Almgren, A., Bell, J., Rendleman, C., & Zingale, M. (2006). Low mach number modeling of type ia supernovae. i. hydrodynamics. *The Astrophysical Journal*, 637(2), 922.
- Anthonissen, M., Mattheij, R., & ten Thije Boonkamp, J. (2003). Convergence analysis of the local defect correction method for diffusion equations. *Numerische Mathematik*, 95(3), 401–425.
- Barbié, L. (2013). *Raffinement de maillage multi-grille local en vue de la simulation 3d du combustible nucléaire des réacteurs à eau sous pression* (Unpublished doctoral dissertation). Aix-Marseille Université.
- Berger, M. J., & Rigoutsos, I. (1991, Sep/Oct). An algorithm for point clustering and grid generation. *IEEE Transactions Systems, Man and Cybernetics*, 21(5), 1278–1286.
- Bois, G. (2011). *Heat and mass transfers at liquid/vapor interfaces with phase-change: pro-posal for a large-scale modeling of interfaces* (Unpublished doctoral dissertation). Université de Grenoble.
- Dellacherie, S. (2005). On a diphasic low Mach number system. *ESAIM: M2AN*, 223, 151–187.
- Dellacherie, S. (2007). Numerical resolution of a potential diphasic low Mach number system. *Journal of Computational Physics*, 39(3), 487–514.

## Bibliography II

- Dellacherie, S. (2012, March). On a low-Mach nuclear core model. In ESAIM: PROCEEDINGS (Vol. 35, p. 79-106). EDP Sciences, 17, Avenue du Hoggar, Parc d'Activité de Courtabuf, BP 112, F-91944 Les Ulis Cedex A, France: EDP Sciences, SMAI.
- Dellacherie, S., & Lafitte, O. (2005). *Existence et unicité d'une solution classique à un modèle abstrait de vibration de bulles de type hyperbolique-elliptique* (Tech. Rep. No. CRM-3200). CRM, Montréal, Canada: Centre de Recherches Mathématiques.
- Fikl, A. (2014, October). *Adaptive mesh refinement with p4est* (Tech. Rep.). Digiteo Labs - bât. 565 - PC 190, CEA Saclay, 91191 Gif-sur-Yvette cedex: Sup Galilée, CEA, Maison de la Simulation.
- Gittel, H.-P., Günther, M., & Ströhmer, G. (2014). Remarks on a nonlinear transport problem. *Journal of Differential Equations*, 256(3), 957 - 988. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0022039613004452> doi: <http://dx.doi.org/10.1016/j.jde.2013.10.002>
- Hackbusch, W. (1984). Local defect correction and domain decomposition techniques. *Defect Correction Methods*, Springer Vienna, 89(113).
- Lagoutière, F. (2000). *Modélisation mathématique et résolution numérique de problèmes de fluides compressibles à plusieurs constituants* (Unpublished doctoral dissertation). Université Pierre-et-Marie-Curie, Paris-VI.

## Bibliography III

- Livne, O. E. (2006a, January). *Clustering on single refinement level: Berger-Rigoustos algorithm* (Tech. Rep. No. UUSCI-2006-001). Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, USA: University of Utah.
- Livne, O. E. (2006b, January). *Minimum and maximum patch size clustering on a single refinement level* (Tech. Rep. No. UUSCI-2006-002). Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, USA: University of Utah.
- Majda, A. (1982). *Equations for low Mach number combustion* (Tech. Rep. No. 112). Berkeley, California: University of California at Berkeley.
- Mekkas, A. (2008). *Résolution numérique d'un modèle de vibration de bulle abstraite* (Unpublished master's thesis). SupGalilée, Centre Mathématique et Informatique, École Centrale Marseille. (CEA, ONERA)
- Penel, Y. (2012). Well-posedness of a low mach number system. *C. R. Acad. Sci.*, 1(350), 51–55.
- Penel, Y., Dellacherie, S., & Lafitte, O. (2013). Theoretical study of an abstract bubble vibration model. *Zeitschrift für Analysis und Ihre Anwendungen*, 32(1), 19–36.

## Tools

We develop and use the programming interface CDMATH. It is aimed at Thermal-Hydraulicists who want to quickly develop with a higher level of abstraction. CDMATH is written in C++ and is open source.

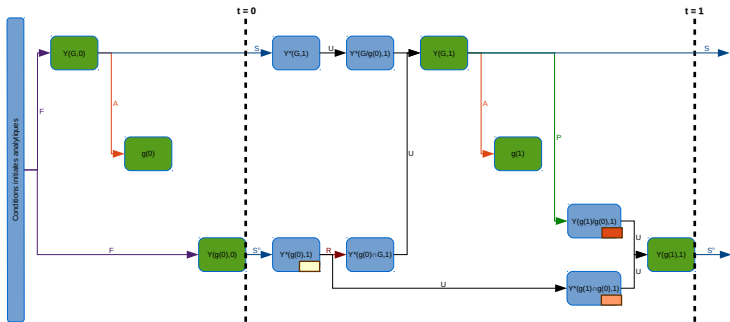
Download CDMATH on <https://github.com/PROJECT-CDMATH/CDMATH>.

Home page: <http://cdmath.jimdo.com/>.

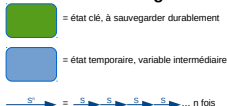
CDMATH is an abstraction based on the library medCoupling, which is part of the SALOME platform (co-developed by the CEA, EDF and OpenCascade). We included our improvements to AMR – in particular the  $l_{min} - l_{max}$  algorithm – to SALOME's file formats.



# AMR diagram

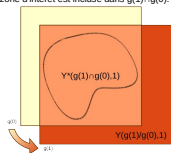


## Schéma de la stratégie AMR



$Y$  = grandeurs physiques  
 $Y^*$  = valeur intermédiaire  
 $G$  = maillage grossier défini sur tout l'espace  
 $g$  = maillage fin défini dans les patches  
 $n$  = maximum des coefficients de raffinement  
 $F$  = calcul à partir d'une formule analytique  
 $A$  = adaptation de maillage  
 $S$  = solveur mécanique  
 $S^2$  = solveur mécanique, utilisé  $n$  fois  
 $U$  = recopie  
 $R$  = restriction de domaine  $g \rightarrow G$   
 $P$  = prolongation  $G \rightarrow g$

Si  $A$  est bien conçue, alors à l'instant 1 la zone d'intérêt est incluse dans  $g(1) \cap g(0)$ .



$Y(G,1) = Y^*(G,1)$  enrichie des valeurs de  $Y^*(g(0),1)$  dès que possible  
 $Y(g(1),1) =$  projection de  $Y(G,1)$  sur  $g(1)$ , enrichie des valeurs de  $Y^*(g(0),1)$  dès que possible

## Data exchange between AMR levels

For two levels:



For more than two levels:

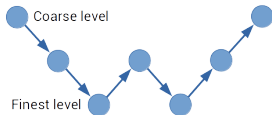


Figure: W strategy

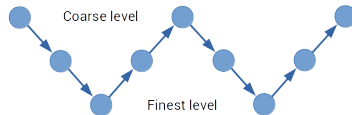


Figure: Two times V strategy

## Local Defect Correction algorithm

The Local Defect Correction method (Hackbusch, 1984) is an efficient way to benefit from AMR (Anthonissen et al., 2003) (Barbié, 2013) to solve elliptic problems as the following:

$$\begin{cases} L\phi = s, \\ \text{physical BCs.} \end{cases} \quad (17)$$

### Initialization

Calculation of  $\phi_{coarse}^0$  from  $L_{coarse}(\phi_{coarse}) = f_{coarse}$ .

### Iterations, as long as no convergence

For all patches, coarse as well as fine:

- define matrix  $A$  for linear problem
- $b = s + \text{contrib}(\text{physical BCs})$
- if (fine level)
  - ▶  $b = b + \text{contrib}(\text{Dirichlet BCs on patch borders from coarse level})$
- if ( $iter \geq 1$ )
  - ▶  $\text{correction}^{iter} = A\phi^{iter-1} - b$  if on area to be refined, 0 otherwise
  - ▶  $b = b + \text{correction}^{iter}$
- solve  $A\phi^{iter} = b$  to get the unknown  $\phi^{iter}$
- restriction of fine level onto the coarse level

Here we detailed the algorithm for 2 levels.

Note: the local defect correction  $\text{correction}^{iter}$  is not a residue and does not tend to zero.

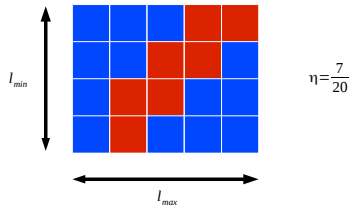
## How to create patches

We will consider the following input parameters:

- $\eta_{min}$ : minimum ratio between flagged area total area that a patch may have – this ensures that the refined grid calculation is concentrated on the areas of interest that were flagged –,
- $n_{min}$ : minimum number of cells on any length in any direction for a patch,
- $n_{max}$ : maximum number of cells on any length in any direction for a patch.

The two geometrical parameters will ensure that the patches remain comparable as far as processor load and processor communication are concerned.

The three parameters are the inputs of a general algorithm. The two following algorithms are the most relevant particular cases.

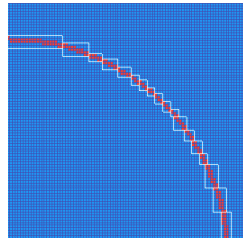




## Patch-covering algorithms I

### Berger-Rigoutsos algorithm

- Introduced by Berger & Rigoutsos (Berger & Rigoutsos, 1991) and analyzed by Livne (Livne, 2006a).
- Set  $\eta_{min} > 0$  as the sole constraining goal.
- No constraint set on geometry.

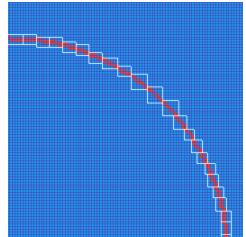


In the following example,  
 $\eta_{min} = 0.4$

## Patch-covering algorithms II

### $l_{min} - l_{max}$ algorithm

- New algorithm, inspired by and similar to the Livne algorithm (Livne, 2006b).
- Set  $\eta_{min} > 0$ , but it is not the priority objective.
- Set  $n_{min} > 1$  and  $n_{max} < n_{system}$ .  
They are strictly constraining objectives and have the priority when in competition with the  $\eta_{min}$  objective.



In the following example,

$$\eta_{min} = 0.4$$

$$n_{min} = 5$$

$$n_{max} = 10$$

## Quantitative comparison between algorithms

Definition of quality functions:

We will note  $(S_i)_{i \in [0, N]}$  the list of the  $N$  patches and  $M(\{X_i\}_i) = \sum_{i=0}^N X_i$  the average of variables  $X_{i \in [0, N]}$  where  $i$  is the subscript of a patch.

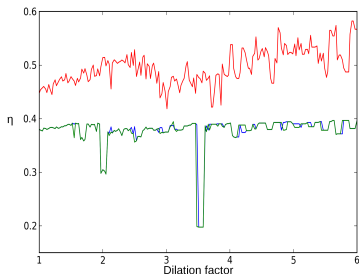
- Minimize the (unnecessary) computation: we want the calculation to be done on flagged cells and the least possible on non-flagged cells.  
So we want the resulting efficiency  $\eta = M(\{\eta_i\}_i)$  as close to 1 as possible.
- Minimize surface differences between patches, for good CPU load balance in case of multiprocessing.

So we want the normalized standard deviation  $\sigma = \sqrt{\frac{M(\{S_i^2\}_i) - M(\{S_i\}_i)^2}{\max(\{S_i^2\}_i) - \min(\{S_i\}_i)^2}}$  as close to 0 as possible.

- Minimize the communication location between patches, i.e. the segment that is common between patches. For a given area, the rectangle with the smallest perimeter is the square.

So we want the average squareness  $\gamma = M(\left(\frac{\text{length of shortest side of patch } i}{\text{length of longest side of patch } i}\right)_i)$  to be as close to 1 as possible.

## Resulting efficiency $\eta$



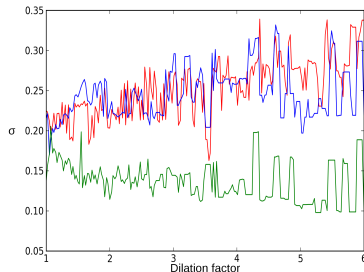
Resulting efficiency  
as a function of the dilation factor  
and of the patch creation algorithm  
(higher is better)

$$\eta = M(\{\eta_i\}_i)$$

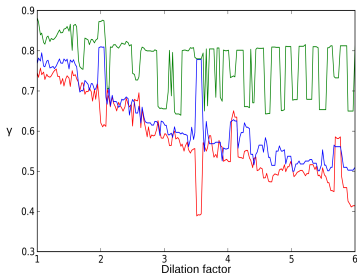
## Normalized standard deviation $\sigma$

Normalized standard deviation  
as a function of the dilation factor  
and of the patch creation algorithm  
(lower is better)

$$\sigma = \sqrt{\frac{M(\{S_i^2\}_i) - M(\{S_i\}_i)^2}{\max(\{S_i^2\}_i) - \min(\{S_i\}_i)^2}}$$



## Average squareness $\gamma$



Average squareness  
as a function of the dilation factor  
and of the patch creation algorithm  
(higher is better)

$$\gamma = M\left(\left(\frac{\text{length of shortest side of patch } i}{\text{length of longest side of patch } i}\right)_i\right)$$