notus
Computational Fluid Dynamics

Immersed boundary methods in the Notus code: contribution, applications and perspectives

Stéphane Glockner
Joris Picot, Antoine Lemoine, Mathieu Coquerelle

I2M, Université de Bordeaux, Bordeaux INP, CNRS UMR 52 95

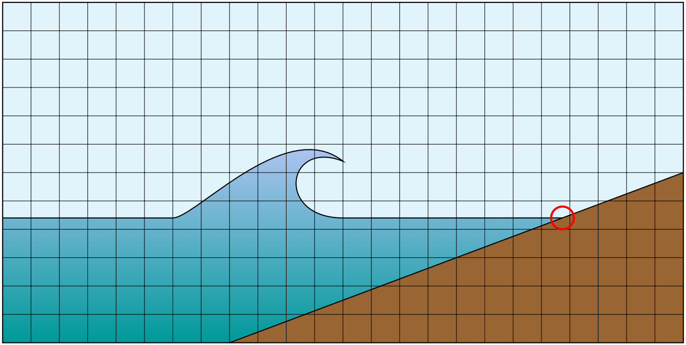
`glockner@bordeaux-inp.fr`
`https://notus-cfd.org/glockner`

9 avril 2018

Contents

1. Notus code
2. Immersed boundary methods and contribution
3. Simulations of solitary wave over immersed structures
4. Perspective: triple line management

Research guidelines for next years



What is (not) Notus

Open-source project started from scratch in 2015

- Modelisation and simulation of **incompressible fluid flows**
- **Massively parallel**
- 2D/3D Finite Volume methods on **Cartesian** staggered grids
- **Multiphysics**

Intended users

- **Mechanical** community: easy to use and adapt, proven state-of-the-art numerical methods
- **Mathematical** community: develop new numerical schemes, fast and efficient framework for comparative and qualitative tests
- Industrials, students

What is not Notus

- A concurrent of
- A commercial tool
- A click button code

Notus - some objectives

Objectives

- **Rationalize research efforts**
- Take advantage of synergies between Research projects / Teaching / Industry / HPC
- Benchmark methods on identified physical test cases
- **Towards numerical experiments**
- Numerical toolbox

Means

- A clear development environment
- A thoroughly **validated and documented** code
  → reference code
- **Mask parallelism** complexities for easy programming
- Porting on GENCI, PRACE, mesocentres

Some projects

- Wave breaking, tidal bore
- Wave attenuation by the rain, fluid/elastic solid interactions
- Decorated cave microclimate
- Supercritical flows
- Energy storage

Development environment and porting

Development framework

- **Fortran 2008**
- **MPI** parallel coding library
- **Git** distributed version control system
- **CMake** cross-platform build system
- **Doxygen** documentation generator from source code

Compilers and MPI libraries

- GNU compilers (7.2) and Open MPI (2.10)
- Intel compilers (18) and SGI MPT (2.11) and BullxMPI (1.2.8.3)
- IBM XL compilers (14.1) and MPI libraries (2.21.1)

Supercomputers

- Curie at TGCC
- Occigen at CINES
- Avakas at MCIA

Features - Domain

Domain

- 2D/3D Cartesian, immersed sub-domains

Incompressible Navier-Stokes equations

- Buoyancy force (Boussinesq approximation)
- Surface tension force (**CSF model**)
- Large Eddy Simulation model (mixed scale)

Multiphase immiscible flows

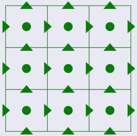
- **1-fluid model**
- Volume-of-Fluid, Moment-of-Fluid, Level-Set interface representations

Energy equation

Species transport equations

Features - Numerical methods

Discretisation



- 2D/3D **Cartesian Finite Volume** on staggered grids, automatic partitioning
- Time discretisation: **implicit & explicit**, up to 2nd order
- Spatial discretisation: up to 2nd order implicit schemes (advection and diffusion)
- Spatial discretisation: 3rd / 5th order WENO schemes (advection)
- 1st & 2nd order **immersed boundary method**

Navier-Stokes

- Velocity/pressure coupling: **time splitting methods** (Goda, Timmermans)
- Surface tension: Closest-Point method to compute curvature (→ Level-set only)

Fluid / fluid interface representation and transport

- Volume-of-Fluid method / PLIC / directional splitting
- Moment-of-Fluid method 2D / 3D / backward advection
- Level-set / WENO

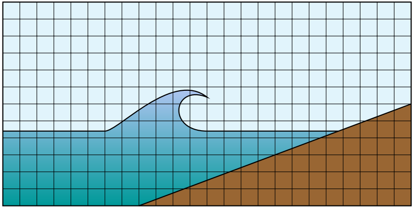
HYPRE library

- BiCGStab, GMRES iterative solvers
- Preconditioners: **geometric & algebraic multigrid**

MUMPS direct solver

- Mainly for 2D matrix
- PORD, Metis graph partitioners

Research guidelines for next years



Interfaces

- Fluid / fluid interfaces (advection, surface tension)
- Fluid / solid boundaries (with or wihtout wetting)
- Fluid / porous media interface
- Fluid / solid phase change

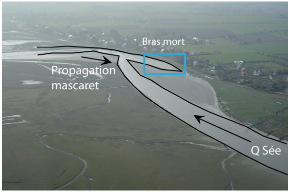
2nd order “everywhere” ? Efficiency ?

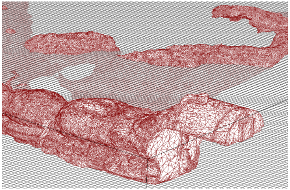
- 2nd order advection scheme, one-fluid model ?
- 2nd order immersed boundaries, but scalable ?
- 2nd order interface reconstruction, even if immersed boundaries ?
- 2nd order interface reconstruction, and curvature ?
- ...

Immersed Boundaries Method

Immersed complex objects into a Cartesian mesh

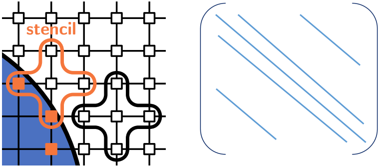
- Microclimate study of decorated caves
- Breaking wave and submersion
- Tidal bore





Motivations / objectives

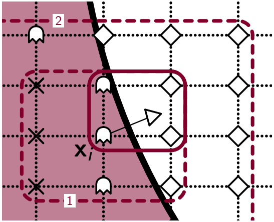
- Avoid complex unstructured grids
- Use the quickest iterative solvers and preconditioners
- → Compact stencils (9 or 25 pts in 2D)
- Band matrix
- 2nd order for Dirichlet and Neumann BC

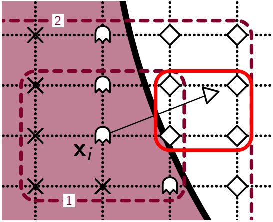


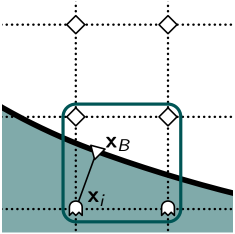
Immersed Boundaries Method

Direct forcing method

- **Extrapolation** of the solution on ghost nodes **compatible with the boundary condition (Dirichlet/Neumann)**
- 2nd order methods: Mittal [JCP2008] (*linear method*)
  Linear relation between the ghost nodes and their image nodes (△)
  → **non-compact stencils**
  Dirichlet : stencil size = 2
  Neumann : stencil size = 3
- 2nd order methods: Coco [JCP2013] (*direct method*)
  Use of the boundary point x_B
  Dirichlet : stencil size = 1 → **compact stencil** for square cells
  Neumann : stencil size = 2



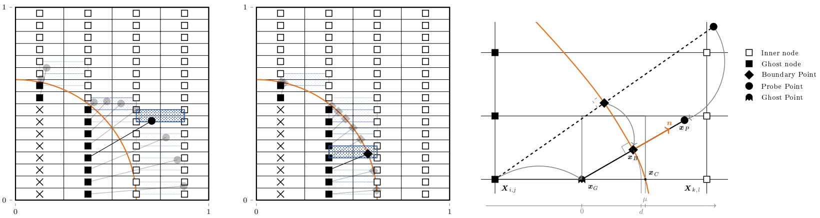




Immersed Boundaries Method - contribution

Support irregular (streched) grids → the ghost node shifting method

- stencil sizes increase with the mesh ratio
- loss of precision
- slower solvers (algebraic multigrid vs. geometric)
- direct method (Coco) not applicable
- → **shift the ghost node toward the boundary**
- same stencil as for rectangular cell
- direct method applicable
- **Compact stencil**: 2nd order for Dirichlet, 1st order for Neumann

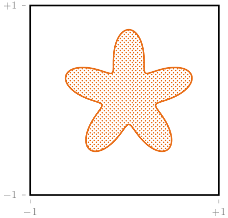


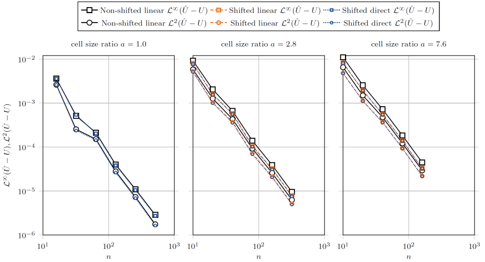
J. Picot, S. Glockner, Discretization stencil reduction of direct forcing immersed boundary methods on rectangular cells: the ghost node shifting method, *Journal of Computational Physics*, vol. 364, pp. 18–48, 2018.

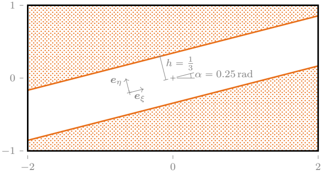
Immersed Boundaries Method - validation

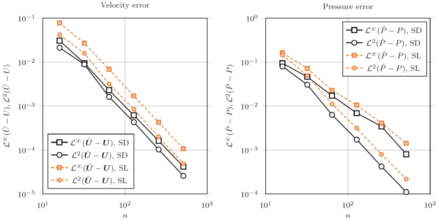
Verification

- Laplacian : circle, flower shapes, Dirichlet & Neumann
- Navier-Stokes : Couette and Poiseuille flows









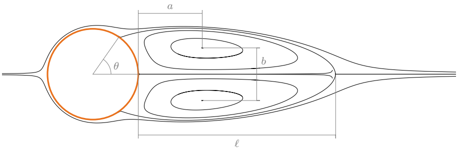
Immersed Boundaries Method - validation

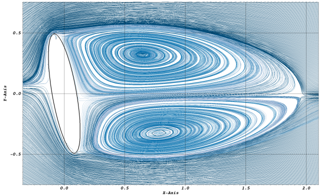
Validation

- Flows around circular (Re=40) and elliptic (Re=20) cylinders

Reference	ℓ / d	a / d	b / d	θ	C_D
Coutanceau (exp. 1977)	2.13	0.76	0.59	53.5	—
Linnick (ordre élevé 2005)	2.28	0.72	0.60	53.6	1.54
Le (immersed 2006)	2.22	—	—	53.6	1.56
Taira (immersed 2007)	2.30	0.73	0.60	53.7	1.54
present linear/direct	2.27	0.72	0.59	53.1	1.56

Reference	C_D	C_L
Dennis and Young	2.116	0.256
D'alessio and Dennis	2.089	0.255
Yoon et al.	2.102	0.252
present linear/direct	2.130	0.246





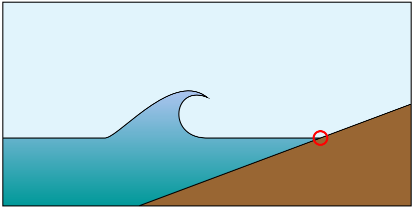
Immersed Boundaries Method - CPU times

Flower test case, dirichlet BC

- Hypre GMRES, tolerance 10^{-10}, SMG & BoomerAMG multigrid preconditioners
- MUMPS direct solver
- cell ratio 1 & 7.6
- SMG 4 times quicker than BoomerAMG
- Should be even better in 3D
- MUMPS direct solver is competitive
- but SMG fail to solve with Neumann BC and high cell ratio

		Méthode directe			*Méthode linéaire*		
		GMRES+SMG		*MUMPS*	*GMRES+BoomerAMG*		*MUMPS*
a	*Maillage*	*Iter.*	*Temps*/s	*Temps*/s	*Iter.*	*Temps*/s	*Temps*/s
1	160 × 160	15	0.06	0.05	18	0.04	0.09
	320 × 320	19	0.13	0.14	19	0.12	0.31
	640 × 640	23	0.35	0.56	21	0.65	1.32
	1280 × 1280	24	1.35	2.71	19	4.01	9.93
7.6	608 × 80	18	0.08	0.07	23	0.07	0.12
	1216 × 160	17	0.16	0.22	25	0.34	0.53
	2432 × 320	21	0.54	0.88	28	1.7	3.5
	4864 × 640	29	6.25	4.2	31	24.6	19.7

Interface methods in notus 0.2.0



Completly immersed subdomain: 2nd order

- go on with validation
- extension to 3D under progress

Partially immersed subdomain → triple ligne → only 1st approximation

- → “step” broken line boundary

Interface transport

- Linear construction (VOF-PLIC or MOF) (with 1st order immersed boundary)
- Level-Set + WENO (without immersed boundary)

Contents

1. Notus code
2. Immersed boundary methods and contribution
3. Simulations of solitary wave over immersed structures
4. Perspective: triple line management

User Interface: `.nts` file example

```
include std "physical_properties.nts";

system {
    test_case tc_solitary_wave_wall;
}

domain {
    spatial_dimension 2;
    corner_1_coordinates (0.0, 0.0);
    corner_2_coordinates (5.0, 0.2);

    immersed_boundary {
        double j = 0.005;
        while (j<0.08) {
                double i = 0;
                while (i<1.5) {
                        rectangle {
                            corner_1_coordinates (1.05 + i, j);
                            corner_2_coordinates (1.09 + i, j + 0.01);
                        }
                        if (j + 0.015 < 0.08)  {
                                rectangle {
                                    corner_1_coordinates (1    + i, j + 0.015);
                                    corner_2_coordinates (1.04 + i, j + 0.025);
                                }
                        }
                        i = i + 0.1;
                }
                j = j + 0.03;
        }
    }
}

grid {
    grid_type regular;
    number_of_cells (512, 128);
    number_of_ghost_cells 2;
}
```

User Interface: `.nts` file example

```
modeling {
    materials {
        fluid "air";
        fluid "water";
    }

    equations {
        navier_stokes {
            gravity_term (0, -9.81);

            boundary_condition {
                left slip;
                right wall;
                top slip;
                bottom slip;
            }
            immersed_boundary_condition {
            wall;
            }
        }

        phase_advection {
            fluid "water"{}
        }
    }
}
```

User Interface: `.nts` file example

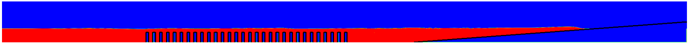
```
numerical_parameters {
    time_iterations 2000;
    time_step 0.35d-3;
    cfl 0.3d0;

    navier_stokes {
        advection_scheme implicit hybrid_o2_centered_o1_upwind;

        solver_momentum hypre_parcsr_gmres {
            max_iteration 200;
            tolerance 1.0d-10;
            initial_preconditioner left_jacobi;
}
        solver_pressure hypre_struct_gmres {
            max_iteration 200;
            tolerance 1.0d-10;
            initial_preconditioner left_jacobi;

            preconditioner smg {
                max_iteration 1;
                tolerance 0.0d0;
    }
}
        immersed_boundary {
            method penalization;
        }
    }
    phase_advection {
        vof_plic {
            smooth_volume_fraction 2;
}
    }
}
post_processing {
    output_library adios;
    output_frequency 50;
    output_fields volume_fraction, velocity, pressure, divergence, ibd_distance;
}
```

Simulations : 2D solitary wave on immersed boundaries

Rectangular immersed structure

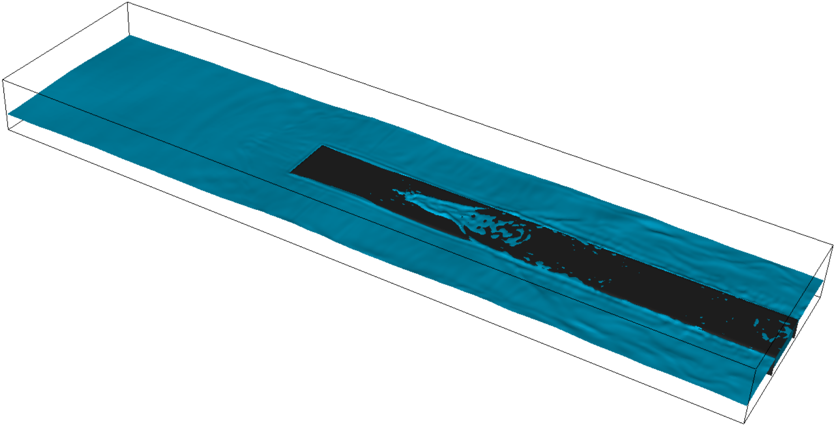
Breaking on a beach

Damping of the wave



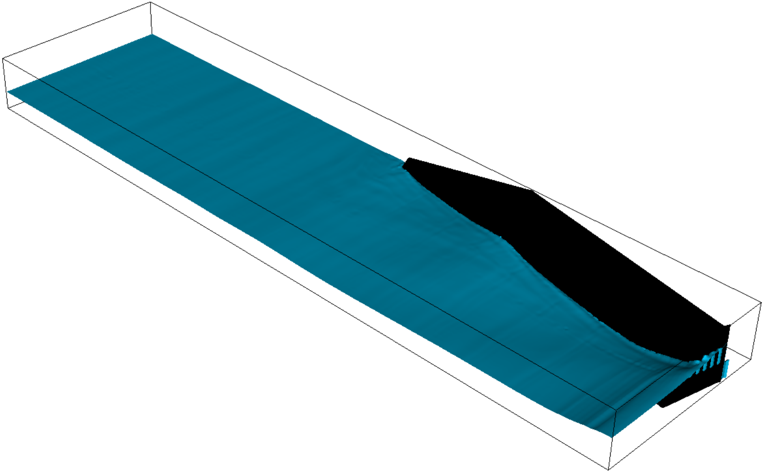
Simulations : 3D submersion example

Submersion



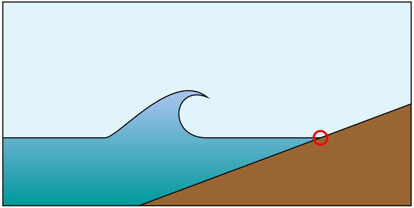
|◁ < ◁ ▷ > ▷| − ▸◂ +

Simulations : 3D sloping bank



|◁ < ◁ ▷ > ▷| − ▸◂ +

Interface methods in notus 0.2.0



Triple line management

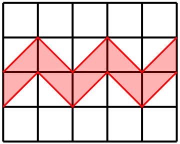
- coupling **within a single cell** of immersed boundary method, interface representation and transport
- **3 material** problem → Moment-of-Fluid method
- eventually wetting phenomena

Moment-of-Fluid method

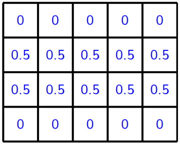
Volume-of-Fluid - PLIC

- Volume fraction + **normal** to the interface → linear construction of the interface
- Requires a **9 pts stencil** (2D)

0.0	0.4	0.9
0.3	1.0	1.0
0.6	1.0	1.0

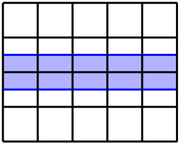


Original interface



0	0	0	0	0
0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5
0	0	0	0	0

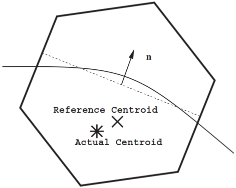
VOF representation



PLIC reconstruction

Moment-of-Fluid

- Volume fraction + **centroid** → linear reconstruction that:
  - matches the volume fraction
  - minimises the discrepancy between the specified centroid and the centroid of the reconstructed polygon
- → **1 pt stencil**, 2nd order
- Generalised to n materials

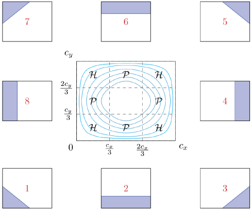


Source: Dyadechko & Shashkov (JCP 2006)

Moment-of-Fluid method - contribution

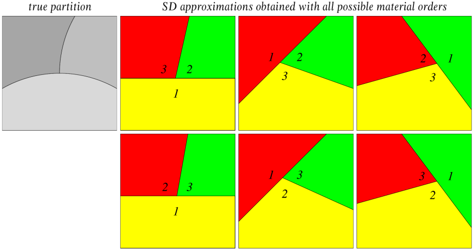
Remove minimisation for Cartesian grids

- analytic form of the centroid curve (for a given volume fraction)
- from 20% to 300% faster



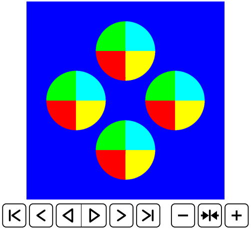
Multimaterial construction: serial dissection

The best solution minimizes the sum of the centroïd defects


true partition

SD approximations obtained with all possible material orders

Source: Dyadechko, V., Shashkov, M. (2008)

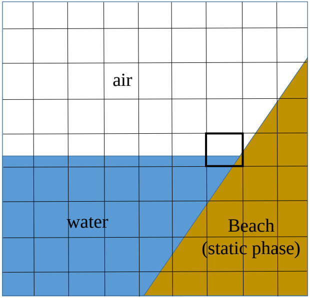


A. Lemoine, S. Glockner, J. Breil, Moment-of-Fluid Analytic Reconstruction on 2D Cartesian Grids, *Journal of Computational Physics*, vol. 328, pp. 131–139, 2017.

Moment-of-Fluid method coupling with immersed boundary methods

3 phase problem : 2 fluid phases (water & air) + 1 solid one (beach)

- → consider the beach phase as static
- → bluid the interface in the remaining part of the cell



Conclusion, projects

Notus 0.2.0

- 2D/3D massively parallel code for multiphase flows
- 1st order immersed domain
- Still a lot of work to ensure 2nd order “eveywhere”

MOF 3D

Remove minimization: under progress

3D immersed boundary methods

Direct extension of 2D (under progress)

MOF and immersed boundary methods

Project, should start last trimester 2018

Related projects

PhD Florian Desmond (2017-20): wave breaking
PhD microclimate of decorated caves: immersed interface (heat and mass transfer)
ANR Superfon: supercritical flows (immersed boundary, optimization of the code)
Fluid/Elastic solid interaction
...