

Algebraic momentum preserving method with a fully coupled 3D parallel solver for the simulation of two-phase incompressible flows

Mohamed EL OUAFA

Supervisors: Stéphane VINCENT, Vincent LE CHENADEC, Benoit TROUETTE

MSME, Univ. Gustave Eiffel, UMR CNRS 8208, Univ. Paris-Est Créteil

September 28 2022

OUTLINE

1. Introduction

Part A

2. Numerical methods

- Incompressible two phase flows with interface tracking
- Algebraic momentum preserving method
- Pressure-velocity coupling algorithms
- Fully-coupled solver and block preconditioning

Part B

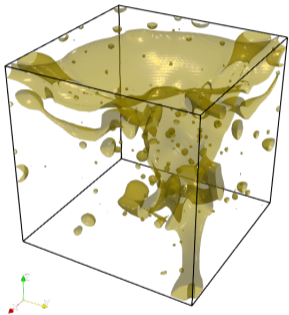
3. Solver validations

- Three dimensional test cases
- Strong scalability

4. Conclusion and perspectives

Introduction

Multi-scale flows



Phase inversion between two immiscible liquids (to be submitted to JCP 2022)



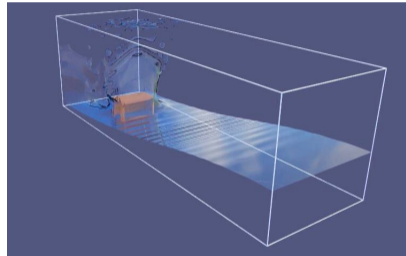
Impact of a liquid on a surface [<http://physique.unice.fr/sem6/2012-2013/PagesWeb/PT/Splash/Html/Index.html>]

Introduction

Multi-scale flows



Breaking wave (LEGI Laboratory)
[images.cnrs.fr/en/photo/20150001_0066]



Dam-Break flow in the presence of obstacle (to be submitted to JCP 2022)

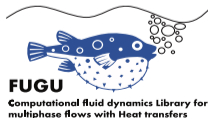
Challenges



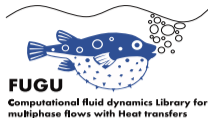
- Presence of different scales in space and time.
- Complex interactions between capillary, inertial and viscous forces.
- Simulations may be **unstable** and velocities may be wrongly evaluated when **high density/viscosity ratios** are involved.
- Treatment of velocity-pressure **coupling** in incompressible flow
- Numerical difficulties with **ill-conditioned linear systems**.

Objectives

In the framework of the homemade code Fugu, develop a parallel and robust solver dealing with two-phase flows, with high density and viscosity ratios.



General features of Fugu code



- Implicit finite volumes discretized on staggered meshes.
- Three pressure-velocity coupling methods are considered, a **fully-coupled solver** published in El Ouafa et al. CICIP2021, an augmented Lagrangian approach of Vincent et al. JCP2011 and a standard Goda JCP1978 projection method.
- The solver uses a **BiCGStabII solver** of Vorst et al. IAM1998, preconditioned with **MG** (HYPRElibrary) and home made preconditioner for fully-coupled approach (which also uses HYPRElibrary).
- Centered schemes are utilized for all spatial derivatives except inertial term.
- The non-linear terms in the momentum equation are discretised with a **consistent mass-momentum transport** and linearized with an Adams-Bashforth scheme.
- The **VOF method** used is the one of Yue and Weymouth JCP2011.
- CSF models for surface tension with VOF smoothing Pianet et al. IJMF2010.
- **MPI** based parallelization.

2: Numerical methods

- Incompressible two phase flows with interface tracking
- Algebraic momentum preserving method
- Pressure-velocity coupling algorithms
- Fully-coupled solver and block preconditioning

Incompressible two phase flows with interface tracking (Kataoka IJMF 1986)

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \overline{\overline{B_{penu}}} (f(\mathbf{u}) - \mathbf{u}_\infty) = -\nabla p + \nabla \cdot \overline{\overline{T}} + \underline{\underline{\sigma \kappa \delta \mathbf{n}}} + \rho \mathbf{g} \quad (2)$$

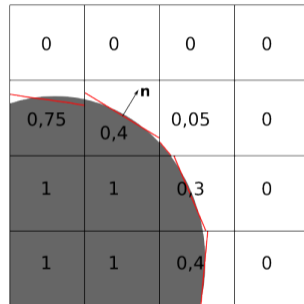
$\overline{\overline{B_{penu}}}$ for specifying BC Angot et al. NM 1999 Continuum Surface Force Brackbill et al. JCP1992

Temporal evolution of volume fraction given by

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0 \quad (3)$$

Recasting in the “conservative form Weymouth et al. JCP2011 ” as

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u} C) = C \nabla \cdot \mathbf{u} \quad (4)$$



Eulerian Implicit (WY)

Material properties defined on basis of interface position

$$\rho(C) = \rho_1 C + (1 - C)\rho_2,$$

$$\mu(C) = \mu_1 C + (1 - C)\mu_2,$$

(5)

Algebraic momentum preserving method

The Navier-Stokes equations (conservative form) can be recast in operator notation

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + L \left(\overline{\overline{B}}_{penu}, \overline{\overline{T}}, \sigma, \mathbf{g} \right) \quad (7)$$

$$L = L \left(\overline{\overline{B}}_{penu} \right) + L \left(\overline{\overline{T}} \right) + L_{\sigma} + L_{\mathbf{g}}$$

Operator-split framework

$$\frac{\rho^{**} \mathbf{u}^{**} - \rho^n \mathbf{u}^n}{\Delta t} + \nabla \cdot \left(\rho^{*,n+1} \mathbf{u}^{*,n+1} \otimes \mathbf{u}^{*,n+1} \right) = 0 \quad (8)$$

$$\frac{\rho^{n+1} \mathbf{u}^{n+1} - \rho^{**} \mathbf{u}^{**}}{\Delta t} = \nabla p + L \left(\overline{\overline{B}}_{penu}, \overline{\overline{T}}, \sigma, \mathbf{g} \right) \quad (9)$$

- ρ^{n+1} is obtained from the resolution of

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (10)$$

- strong stability preserving Runge-Kutta 3 (SPP-RK3) time integrator + WENO or CUI Cubista TVD schemes

Rudmann (IJNM 1998), Desjardins Moureau (CTR SummerProgram 2010), Rhaessi, Pitsch, (CaF 2012), V. LeChenadec, H. Pitsch, (JCP 2013), Ghods, Herrmann, (PS 2013), Vaudor et al. (CaF2016) Owkes, Desjardins (JCP 2017), Nangia et al JCP (2019), Estivaleres et al JCP (2021), Arrufat et al C&F (2021)

Pressure-velocity coupling algorithms

Find (u, p) such as the conservation equations are satisfied

Segregated approach

- Chorin type projection (MC1968) and its variants
- Kinematics Scalar Projection Method (KSP) (Caltagirone et al. MFD2015)
- Fractional time-step technique (Guermond et al. JCP2009)
- SIMPLE, SIMPLEC, PISO..... (Patankar et al. 1983)

Monolithic solver

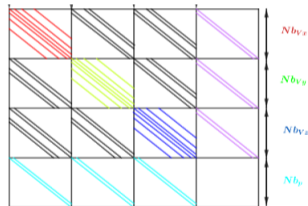
- Augmented Lagrangian (Fortin and Glowinski. JAAM1985, Vincent et al. JCP 2011)
- Uzawa algorithm
- Fully-coupled solver (Bootland et al. JCP 2017, El Ouafa et al. CACP2021)

Fully-coupled solver and block preconditioning (El Ouafa et al. CICIP2021)

Suitable temporal and spatial discretization leads to:

$$\begin{pmatrix} M_u^{(\rho^{n+1})} + A_u^{(\mu^{n+1})} & B_p^T \\ B_u & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \quad \text{or} \quad Ax = b$$

- $M_u^{(\rho^{n+1})} \mathbf{u}^{n+1} = \int_{\Omega} \left[\frac{\rho^{n+1}}{\Delta t} \mathbf{u}^{n+1} + \bar{\bar{B}}_{penu} f(\mathbf{u}^{n+1}) \right] dV$
- $A_u^{(\mu)} \mathbf{u}^{n+1} = - \int_{\Sigma} \left[\mu^{n+1} (\nabla \mathbf{u}^{n+1} + \nabla^T \mathbf{u}^{n+1}) \right] \cdot \mathbf{n} dS$
- $B_p^T p^{n+1} = \int_{\Omega} \nabla p^{n+1} dV$
- $B_u \mathbf{u}^{n+1} = \int_{\Omega} \nabla \cdot \mathbf{u}^{n+1} dV$
- $\mathbf{f} = \int_{\Omega} \left(\frac{\rho^{**} \mathbf{u}^{**}}{\Delta t} + \bar{\bar{B}}_{penu} \mathbf{u}_{\infty} + \mathbf{F}_s \right) dV$



Structure of the fully-coupled matrix

Iterative solver: BICGStabl (Vorst et al. IAM1987)

Preconditionner

Instead of solving, $Ax = b$ we prefer to solve $P^{-1}Ax = P^{-1}b$ for which $cond(P^{-1}A) \ll cond(A)$.

Block preconditioning techniques

build the preconditioner \mathbf{P} of the problem by taking a block LU decomposition of the original matrix \mathbf{A}

$$\begin{pmatrix} F_u & B_p^T \\ B_u & 0 \end{pmatrix} = \begin{pmatrix} I_u & 0 \\ B_u F_u^{-1} & I_p \end{pmatrix} \begin{pmatrix} F_u & B_p^T \\ 0 & S_p \end{pmatrix} \text{ with } F_u = M_u + A_u$$

Here s_p is the Schur complement of the pressure block: $S_p = -B_u F_u^{-1} B_p^T$

Block triangular preconditioner: $\mathbf{P} \approx \begin{pmatrix} F_u & B_p^T \\ 0 & S_p \end{pmatrix}$

not feasible to use the exact Schur complement S_p and the velocity block F_u as a part of the preconditioning operator !!!!!

Block preconditioning techniques for the velocity block (El Ouafa et al. CICP2021)

using the Neumann series expansion at the order zero, the form of our velocity preconditioner reads as follows:

$$\hat{F}_u = \begin{pmatrix} F_{uu} & F_{uv} & F_{uw} \\ 0 & F_{vv} & F_{vw} \\ 0 & 0 & F_{ww} \end{pmatrix} \quad (11)$$

Block preconditioning techniques for the inverse of Schur complement

\hat{S}_p^{-1} by the pressure convection diffusion (PCD) preconditioning (Bootland et al. JCP 2017)

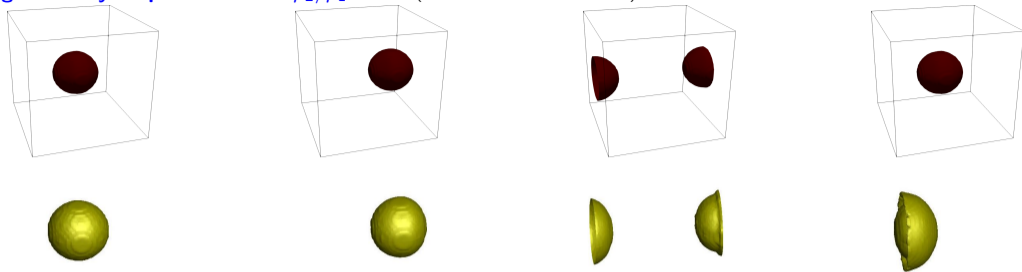
$$\hat{S}_{PCD}^{-1} = (M_p^{1/\mu})^{-1} + (A_p^{1/\rho})^{-1} (N_p^{(1)} + \frac{1}{\Delta t} M_p^{(1)}) (M_p^{(1)})^{-1} \quad (12)$$

$\hat{P}^{-1} \mathbf{A} \mathbf{x} = \hat{P}^{-1} \mathbf{b} \rightarrow$ each matrix-vector product $\mathbf{A} \mathbf{x} = \mathbf{z}$ will be followed by the calculation of $\hat{P}^{-1} \mathbf{z} = \mathbf{y}$

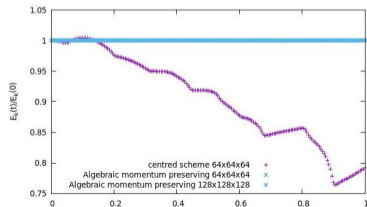
3: Solver validations

- Three dimensional test cases

High density droplet advection $\rho_2/\rho_1 = 10^6$ (to be submitted to JCP 2022)



High density droplet advection without viscous, gravity and capillary effect. Comparison between **Algebraic momentum preserving** and **centred scheme with VOF** on the mesh $64 \times 64 \times 64$. The interface is plotted at time 0s (first column) 0.25s (second column) 0.5s (third column) and 1s (fourth column).

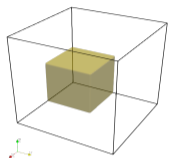


- Conservation of shape and kinetic energy
- The domain integrals of the kinetic energy: $E_k = \frac{1}{2} \int_{\Omega} \rho U^2 dV$

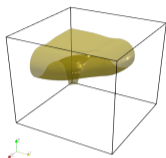
Simulation of the inversion of the phase for $Re=221$ and $We=121$ (to be submitted to JCP 2022)

• $Re = \frac{\rho_1 H U_g}{\mu_1}$ and $We = \frac{\rho_1 H U_g^2}{\sigma}$ with $U_g = \frac{\rho_2 - \rho_1}{\rho_1} \sqrt{\frac{gH}{2}}$

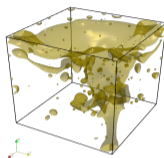
• HPC DNS calculations $\approx 10^9$ cells – 12800 processors of the HPC Occigen cluster of CINES



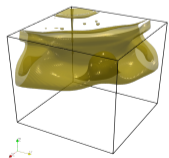
0s



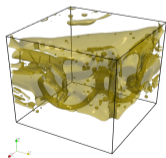
3s



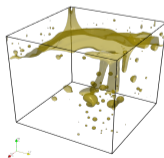
7s



1.5s



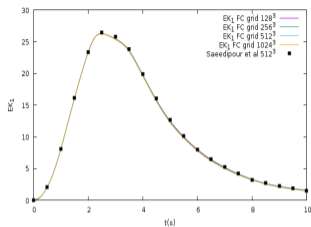
5s



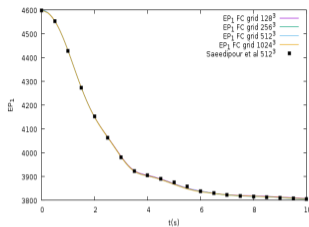
8s

Comparison of macroscopic quantities of interest with the reference data of Saeedipour et al. AM 2021

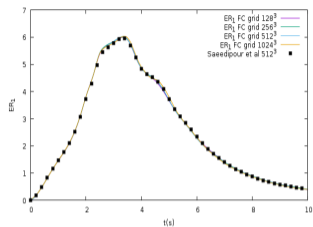
● HPC DNS calculations $\approx 10^9$ cells – 12800 processors of the HPC Occigen cluster of CINES



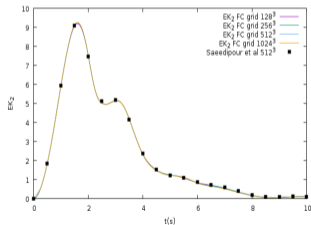
Water kinetic energy



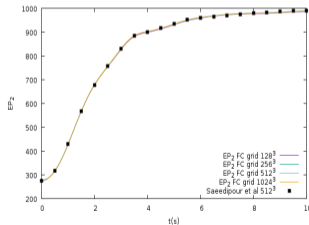
Water potential energy



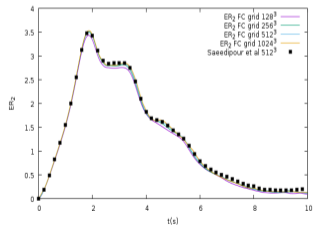
Water enstrophy



Oil kinetic energy



Oil potential energy

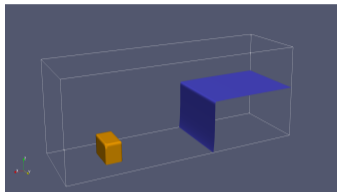


Oil enstrophy

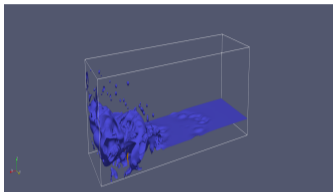
Dam-Break Flow in the Presence of Obstacle $\rho_2/\rho_1 = 10^3$ and $\mu_2/\mu_1 = 10^2$ (to be submitted to JCP 2022)

• Domain $D=[3.22 \times 1 \times 1]$

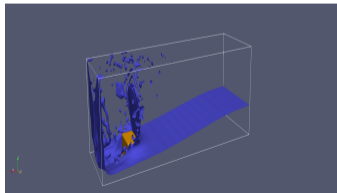
• mesh $200 \times 80 \times 80$



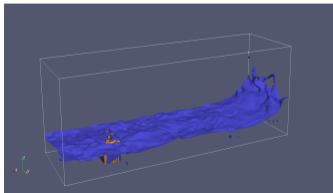
0s



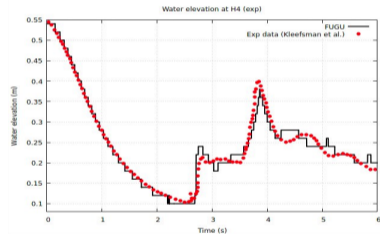
1.1467s



0.8747s



3.4585s



Water elevation

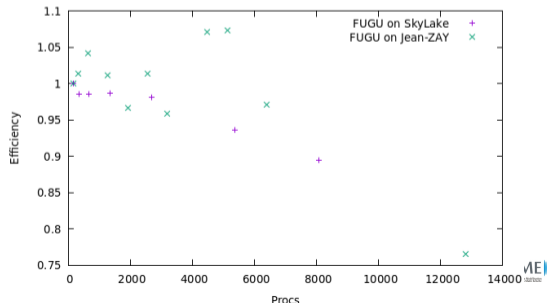
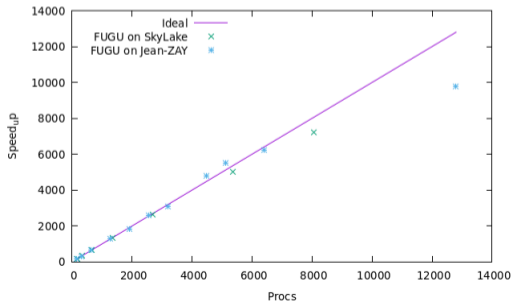
Goal

-Find the point at which a reasonable computation time is achieved but still limit the overhead induced by a parallel process (Skylake and Jean Zay for a number of processors increasing from 160 to 12800 over 30 time iterations on 134 million cells).

Speed-up and Efficiency

$$S(p) = \frac{T_{ref}}{T_p} N_{proc_{ref}} \quad (13)$$

$$E(p) = \frac{S(p)}{p} \quad (14)$$



Conclusions and perspectives:

- Fully-coupled solver 2D/3D
- High density and viscosity ratios
- Simulations massively parallel
- Momentum preserving
- Extension to LES of turbulent two-phase flows and compressible flows

Thank you for your attention