Título The time derivative in the mass transport equation in a Drift-flux model

Tipo de Producto Ponencia resumen

Autores Axel Larreteguy, Pablo Caron y Francisco Barceló

Publicado en: XXIII Congreso de Métodos Numéricos y sus Aplicaciones, ENIEF 2017. Universidad Nacional de La Plata.

Código del Proyecto y Título del Proyecto

P17T03 - Simulación de Flujos Multifásicos en la Industria del Petróleo

Responsable del Proyecto

Axel Larreteguy

Línea

Fluidodinámica Computacional (FC)

Área Temática

Modelado y Simulación Computacional (MYS)

Fecha

Noviembre 2017

INTEC
Instituto de Tecnología

UADE
FUNDACIÓN UADE

The time derivative in the mass transport equation in a drift-flux model

Axel Larreteguy, Pablo Caron, Francisco Barceló

Instituto de Tecnología - Universidad Argentina de la Empresa

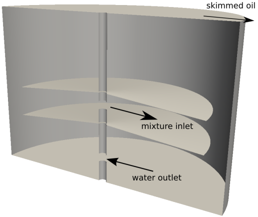
ENIEF 2017, 5-10 de noviembre, UNLP, La Plata, Argentina

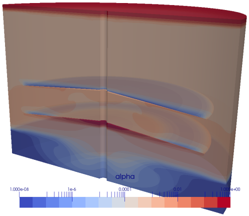
Table of Contents

- Introduction.
- Mathematical model.
- Numerical model.
- Issues and Remedies.
  - Issue I: keep fractions bounded
  - Issue II: balance your “mass” account
  - Issue III: use a “compressible” PISO loop
- Conclusions.

Introduction

- Interested in gravity separation of two immiscible fluids.





- Use finite volume (OpenFOAM) code, adapted to our needs.
- Had and have some mass conservation issues.
- Found solutions to some but no all of them.

Mathematical model

- Drift Flux Model, Ishii & Hibiki (Two Fluid Model, Ishii 1987).
- The mixture is a single pseudo-fluid.
- Two phases: 1-continuous and 2-disperse.
- The phase fractions are bounded fields!
- Diferential Equations:
  1. mixture continuity
  2. mixture momentum
  3. disperse phase transport
- Closure relation:
  4. drift velocity model

Mathematical model

- Main fields:
  - α_2 : disperse phase volumetric fraction
  - $\mathbf{v}_m$: center of mass (mixture) velocity
  - p_m : mixture pressure

Mathematical model

- Main fields:
  - α_2 : disperse phase volumetric fraction
  - $\mathbf{v}_m$: center of mass (mixture) velocity
  - p_m : mixture pressure
- Other important fields:
  - $\mathbf{v}_k$: velocity of phase $k = 1, 2$
  - $\alpha_1 = 1 - \alpha_2$: continuous phase volumetric fraction
  - $\rho_m = \alpha_1 \rho_1 + \alpha_2 \rho_2$: mixture density
  - $\mathbf{j} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2$: center of volume velocity
  - $\mathbf{v}_{2j} = \mathbf{v}_2 - \mathbf{j}$: “drift“ velocity

Mathematical model

Differential Equations:

1. mixture continuity

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{v}_m) = 0$$

Mathematical model

Differential Equations:

1. mixture continuity

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{v}_m) = 0$$

2. mixture momentum

$$\frac{\partial \rho_m \mathbf{v}_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{v}_m \mathbf{v}_m) =$$

$$-\nabla p_m + \nabla \cdot (\tau + \tau^t) - \nabla \cdot \left[\frac{\alpha_2 \rho_1 \rho_2}{(1-\alpha_2)\rho_m} \mathbf{v}_{2j} \mathbf{v}_{2j} \right] + \rho_m \mathbf{g}$$

Mathematical model

Differential Equations:

1. mixture continuity

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{v}_m) = 0$$

2. mixture momentum

$$\frac{\partial \rho_m \mathbf{v}_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{v}_m \mathbf{v}_m) =$$
$$- \nabla p_m + \nabla \cdot (\tau + \tau^t) - \nabla \cdot \left[\frac{\alpha_2 \rho_1 \rho_2}{(1-\alpha_2)\rho_m} \mathbf{v}_{2j} \mathbf{v}_{2j} \right] + \rho_m \mathbf{g}$$

3. disperse phase transport

$$\frac{\partial \alpha_2 \rho_2}{\partial t} + \nabla \cdot (\alpha_2 \rho_2 \mathbf{v}_m) = -\nabla \cdot \left(\frac{\alpha_2 \rho_1 \rho_2}{\rho_m} \mathbf{v}_{2j} \right)$$

Mathematical model

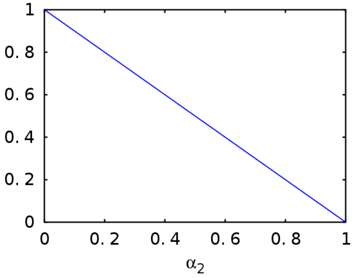
Closure relation:

4. drift velocity model (monodisperse)

$$\mathbf{v}_{2j} = \mathbf{V}_0(1-\alpha_2)$$

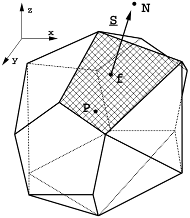
where

- $\mathbf{V}_0 = \frac{2}{9}\frac{g(\rho_1-\rho_2)r_d^2}{\mu_1}$: terminal velocity of a droplet
- $(1-\alpha_2)$: factor to model the hindering effect



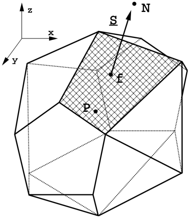
Numerical model

- Method: Finite Volumes, colocated polyhedrical cells, OpenFOAM (www.openfoam.org)



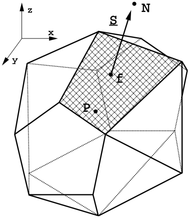
Numerical model

- Method: Finite Volumes, colocated polyhedrical cells, OpenFOAM (www.openfoam.org)
- Application: separationFoam (Larreteguy, Barceló, Caron), based on settlingFoam (Brennan, 2001)



Numerical model

- Method: Finite Volumes, colocated polyhedrical cells, OpenFOAM (www.openfoam.org)
- Application: separationFoam (Larreteguy, Barceló, Caron), based on settlingFoam (Brennan, 2001)



- Features of separationFoam 3.2 (2017)
    - model for the mixture viscosity
    - thermal effects (i.e.: FWKO with fire tubes)
    - control system (i.e.: to control fire tubes power)
    - pressure gradient driven drift (i.e.: rotating flow separators)

Numerical model: Mixture continuity

- Finite volume approximation: $\forall$ cell V^i with boundary ∂V^i

$$\int_{V^i} \frac{\partial \rho_m}{\partial t} dV + \int_{\partial V^i} (\mathbf{v}_m \rho_m) \cdot d\mathbf{S} = 0$$

$d\mathbf{S}$: infinitesimal surface element.

Numerical model: Mixture continuity

- Finite volume approximation: $\forall$ cell V^i with boundary ∂V^i

$$\int_{V^i} \frac{\partial \rho_m}{\partial t} dV + \int_{\partial V^i} (\mathbf{v}_m \rho_m) \cdot d\mathbf{S} = 0$$

$d\mathbf{S}$: infinitesimal surface element.

- For cells bounded by flat faces

$$\frac{\partial \rho_m^i}{\partial t} V^i + \sum_f \Phi_m^f = 0$$

$\Phi_m^f = \rho_m^f \mathbf{v}_m^f \cdot \mathbf{S}^f$: *convective mass flux*.
$(\cdot)^f$: magnitude interpolated at the face.
$\mathbf{S}^f$: face area vector.

Numerical model: Disperse phase transport

- Finite volume approximation:

$$\int_{V^i} \frac{\partial \alpha_2}{\partial t} dV + \int_{\partial V^i} (\mathbf{v}_m \alpha_2) \cdot d\mathbf{S} + \int_{\partial V^i} \left(\frac{\rho_1}{\rho_m} \mathbf{v}_{2j} \alpha_2 \right) \cdot d\mathbf{S} = 0$$

Numerical model: Disperse phase transport

- Finite volume approximation:

$$\int_{V^i} \frac{\partial \alpha_2}{\partial t} dV + \int_{\partial V^i} (\mathbf{v}_m \alpha_2) \cdot d\mathbf{S} + \int_{\partial V^i} \left(\frac{\rho_1}{\rho_m} \mathbf{v}_{2j} \alpha_2 \right) \cdot d\mathbf{S} = 0$$

- For flat faces

$$\frac{\partial \alpha_2^i}{\partial t} V^i + \sum_f \alpha_2^f \mathbf{v}_m^f \cdot \mathbf{S}^f + \sum_f \frac{\rho_1}{\rho_m^f} \alpha_2^f \mathbf{v}_{2j}^f \cdot \mathbf{S}^f = 0$$

Numerical model: Disperse phase transport

- Finite volume approximation:

$$\int_{V^i} \frac{\partial \alpha_2}{\partial t} dV + \int_{\partial V^i} (\mathbf{v}_m \alpha_2) \cdot d\mathbf{S} + \int_{\partial V^i} \left(\frac{\rho_1}{\rho_m} \mathbf{v}_{2j} \alpha_2 \right) \cdot d\mathbf{S} = 0$$

- For flat faces

$$\frac{\partial \alpha_2^i}{\partial t} V^i + \sum_f \alpha_2^f \mathbf{v}_m^f \cdot \mathbf{S}^f + \sum_f \frac{\rho_1}{\rho_m^f} \alpha_2^f \mathbf{v}_{2j}^f \cdot \mathbf{S}^f = 0$$

- For consistency, we want to use the mass flux Φ_m^f in this equation.

Numerical model: Disperse phase transport

- To do so, we define the *disperse mass fraction* $\tilde{\alpha}_2 = \frac{\rho_2}{\rho_m}\alpha_2$.

Numerical model: Disperse phase transport

- To do so, we define the *disperse mass fraction* $\tilde{\alpha}_2 = \frac{\rho_2}{\rho_m}\alpha_2$.
- We write the disperse phase mass transport equation as

$$\frac{\partial \rho_m^i \tilde{\alpha}_2^i}{\partial t} V^i + \sum_f \Phi_{\tilde{\alpha}}^f \tilde{\alpha}_2^f = 0$$

$\Phi_{\tilde{\alpha}}^f = \Phi_m^f + \Phi_d^f$: total mass flux
$\Phi_d^f = \rho_1 \mathbf{v}_{2j}^f \cdot \mathbf{S}^f$: drift mass flux.

Numerical model: Solution procedure

- Standard solution procedure for $\mathbf{v}_m$ and p (PISO loop).
- $\tilde{\alpha}_2$ obtained from the disperse phase transport equation, *which is not as simple as it might seem*.

Issue I: keep fractions bounded

Problems with the original settlingFoam applied to water-oil separators:

- Symptom: most of the oil mass is lost!

[1] *A bounded downwind-upwind semi-discrete scheme for finite volume methods for phase separation problems*, Axel E. Larreteguy, Luis F. Barceló, Pablo A. Caron, 1/Oct/2017, App. Math. Mod., Vol 50, pp 118-134; doi:10.1016/j.apm.2017.05.003

Issue I: keep fractions bounded

Problems with the original settlingFoam applied to water-oil separators:

- Symptom: most of the oil mass is lost!
- Cause: the disperse phase keeps filling an already full cell, thus requiring truncation of disperse fraction to the range $[0, 1]$.

[1] *A bounded downwind-upwind semi-discrete scheme for finite volume methods for phase separation problems*, Axel E. Larreteguy, Luis F. Barceló, Pablo A. Caron, 1/Oct/2017, App. Math. Mod., Vol 50, pp 118-134; doi:10.1016/j.apm.2017.05.003

Issue I: keep fractions bounded

Problems with the original settlingFoam applied to water-oil separators:

- Symptom: most of the oil mass is lost!
- Cause: the disperse phase keeps filling an already full cell, thus requiring truncation of disperse fraction to the range $[0, 1]$.
- Remedy: new interpolation scheme (*downwind-upwind*[1]) for the drift mass flux $\Phi_d^f = \rho_1 \mathbf{v}_{2j}^f \cdot \mathbf{S}^f$

[1] *A bounded downwind-upwind semi-discrete scheme for finite volume methods for phase separation problems*, Axel E. Larreteguy, Luis F. Barceló, Pablo A. Caron, 1/Oct/2017, App. Math. Mod., Vol 50, pp 118-134; doi:10.1016/j.apm.2017.05.003

Issue I: keep fractions bounded

Problems with the original settlingFoam applied to water-oil separators:

- Symptom: most of the oil mass is lost!
- Cause: the disperse phase keeps filling an already full cell, thus requiring truncation of disperse fraction to the range $[0, 1]$.
- Remedy: new interpolation scheme (*downwind-upwind*[1]) for the drift mass flux $\Phi_d^f = \rho_1 \mathbf{v}_{2j}^f \cdot \mathbf{S}^f$

Not all of our problems were gone, however.

[1] *A bounded downwind-upwind semi-discrete scheme for finite volume methods for phase separation problems*, Axel E. Larreteguy, Luis F. Barceló, Pablo A. Caron, 1/Oct/2017, App. Math. Mod., Vol 50, pp 118-134; doi:10.1016/j.apm.2017.05.003

Issue II: balance your “mass” account

Because..

- Symptom: we still have **slight** losses of disperse phase.
- Cause?
- Remedy?

We are using conservative fluxes and a bounded scheme, so… how can it be?

Issue II: balance your “mass” account

The problem is tracked down to the following:

- Let us recall the disperse phase mass transport equation:

$$\frac{\partial \rho_m^i \tilde{\alpha}_2^i}{\partial t} V^i + \sum_f \Phi_{\tilde{\alpha}}^f \tilde{\alpha}_2^f = 0$$

- The hidden problem: the time derivative is applied to the product of two variables that are not independent, for $\rho_m = \rho_1 \rho_2 / \left(\rho_2 - (\rho_2 - \rho_1)\tilde{\alpha}_2\right)$.
- In other words: the time derivative is applied to a non-linear function of $\tilde{\alpha}_2$
- We need to linearize it.

Issue II: balance your “mass” account

Original settlingFoam algorithm for solving for $\tilde{\alpha}$ and ρ_m:

- The linearization is done by splitting the derivative of the product

$$\rho_m^i \frac{\partial \tilde{\alpha}_2^i}{\partial t} V^i + \cancel{\tilde{\alpha}_2^i \frac{\partial \rho_m^i}{\partial t}} V^i + \sum_f \Phi_{\tilde{\alpha}}^f \tilde{\alpha}_2^f = 0$$

and neglecting the second term (i.e.: consider $\rho_m^i = \rho_m^{io} = const$).

Issue II: balance your "mass" account

Original settlingFoam algorithm for solving for $\tilde{\alpha}$ and ρ_m:

- The linearization is done by splitting the derivative of the product

$$\rho_m^i \frac{\partial \tilde{\alpha}_2^i}{\partial t} V^i + \cancel{\tilde{\alpha}_2^i \frac{\partial \rho_m^i}{\partial t}} V^i + \sum_f \Phi_{\tilde{\alpha}}^f \tilde{\alpha}_2^f = 0$$

  and neglecting the second term (i.e.: consider $\rho_m^i = \rho_m^{io} = const$).

- The solution algorithm is the following:
  1. solve equation to get the updated mass fraction $\tilde{\alpha}_2$;
  2. evaluate the updated density as $\rho_m = \rho_1 + (\rho_1/\rho_2 - 1)\tilde{\alpha}_2$; and
  3. evaluate the updated disperse volume fraction $\alpha_2 = \rho_m \tilde{\alpha}_2 / \rho_2$.

Issue II: balance your “mass” account

We propose the following

Predictor-Corrector algorithm:

1. obtain a *predicted* $\tilde{\alpha}_2^{i*}$ by aplying step i) of the original algorithm,

Issue II: balance your "mass" account

We propose the following

Predictor-Corrector algorithm:

1. obtain a *predicted* $\tilde{\alpha}_2^{i*}$ by aplying step i) of the original algorithm,
2. compute the updated mass as $m_2^i = \rho_m^{io} \tilde{\alpha}_2^{i*} V^i$,

Issue II: balance your "mass" account

We propose the following

Predictor-Corrector algorithm:

1. obtain a *predicted* $\tilde{\alpha}_2^{i*}$ by aplying step i) of the original algorithm,
2. compute the updated mass as $m_2^i = \rho_m^{io} \tilde{\alpha}_2^{i*} V^i$,
3. compute the updated volume fraction as $\alpha_2^i = m_2^i / (\rho_2 V^i)$,

Issue II: balance your "mass" account

We propose the following

Predictor-Corrector algorithm:

1. obtain a *predicted* $\tilde{\alpha}_2^{i*}$ by aplying step i) of the original algorithm,
2. compute the updated mass as $m_2^i = \rho_m^{io} \tilde{\alpha}_2^{i*} V^i$,
3. compute the updated volume fraction as $\alpha_2^i = m_2^i / (\rho_2 V^i)$,
4. update the mixture density as $\rho_m^i = (1 - \alpha_2^i)\rho_1 + \alpha_2^i \rho_2$, and finally

Issue II: balance your "mass" account

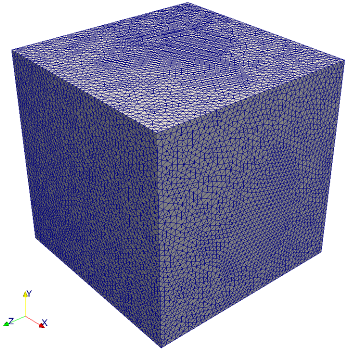
We propose the following

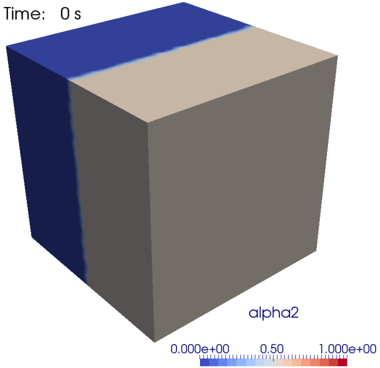
Predictor-Corrector algorithm:

1. obtain a *predicted* $\tilde{\alpha}_2^{i*}$ by aplying step i) of the original algorithm,
2. compute the updated mass as $m_2^i = \rho_m^{io} \tilde{\alpha}_2^{i*} V^i$,
3. compute the updated volume fraction as $\alpha_2^i = m_2^i / (\rho_2 V^i)$,
4. update the mixture density as $\rho_m^i = (1 - \alpha_2^i)\rho_1 + \alpha_2^i \rho_2$, and finally
5. update the mass fraction as $\tilde{\alpha}_2 = \rho_2 \alpha_2 / \rho_m$.

Issue II: balance your “mass” account

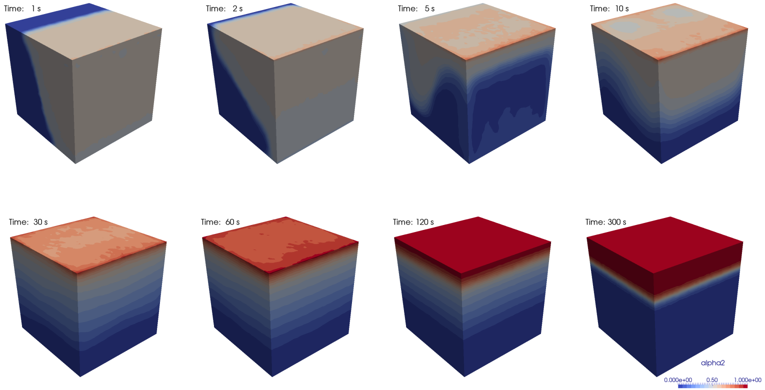
Predictor-corrector algorithm: test - density driven stirring and separation of a water-oil mixture in a cube (in a *horrible* mesh)





Issue II: balance your “mass” account

Hard test in a *horrible* mesh: results



Unit cube - snapshots of volume fraction at selected times

Issue II: balance your “mass” account

- The proposed predictor-corrector algorithm has been found to conserve mass up to machine precision *as long as there is no need for truncation of the disperse phase fraction field*.

Issue II: balance your "mass" account

- The proposed predictor-corrector algorithm has been found to conserve mass up to machine precision *as long as there is no need for truncation of the disperse phase fraction field*.
- But, what about accuracy?

Issue II: balance your "mass" account

The bottle experiment:

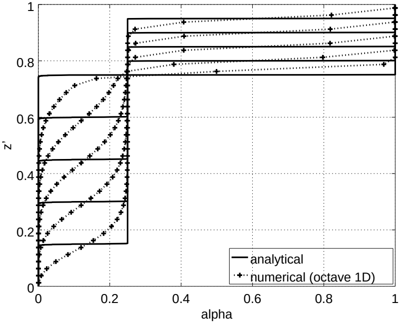
Issue II: balance your "mass" account

The *bottle experiment*: an academic 1D verification example

- Uniform mixture in a tank, left to separate (height $H = 10m$).
- $\mathbf{j} = 0 \Rightarrow$ mixture mass and momentum equations not requiered.
- Analytical solution available.
- 1D numerical solution also available (`sim1D`: 1D Octave version of `separationFoam`).
- Case: oil in water ($\rho_1 = 1000kg/m^3, \rho_1 = 900kg/m^3$)
- Discretization details for both `separationFoam` and `sim1D`
  - uniform spatial discretization (40 cells)
  - fixed time step ($\Delta t = \Delta z / V_0 \Rightarrow Co_{drift} = 1$).

Issue II: balance your "mass" account

The bottle experiment: analytical and `sim1D` solutions.

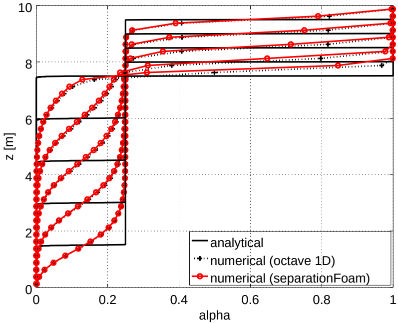


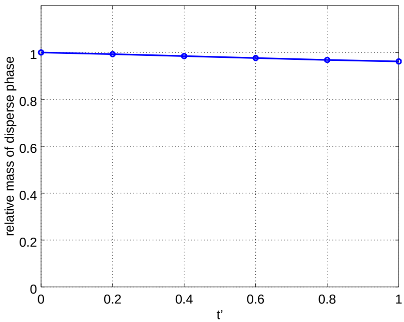
α_2 profiles for $t' = 0.2, 0.4, 0.6, 0.8, 1.0$

Dimensionless solution (valid for all mixtures, drop size and tank height).

Issue II: balance your "mass" account

The bottle experiment (oil in water): **original** algorithm



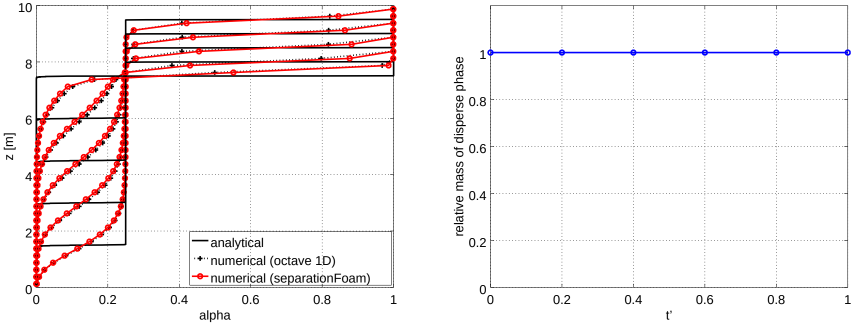


Disperse fraction profiles & mass fraction

Mass NOT conserved (despite that fraction truncation is not required).

Issue II: balance your “mass” account

The bottle experiment (oil in water): **predictor-corrector** algorithm



Disperse fraction profiles & mass fraction

Mass IS conserved (as long as fractions are not truncated).

Issue II: balance your "mass" account

We seem to have solved Issue II: "we still have slight losses of disperse phase"

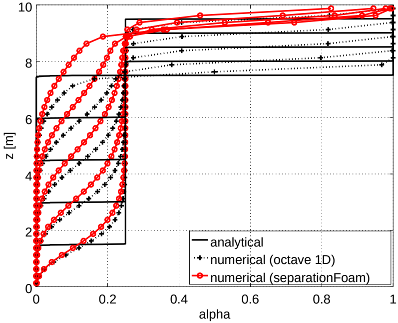
- Cause: bad accounting of mass inventory.
- Remedy: predictor-corrector algorithm.

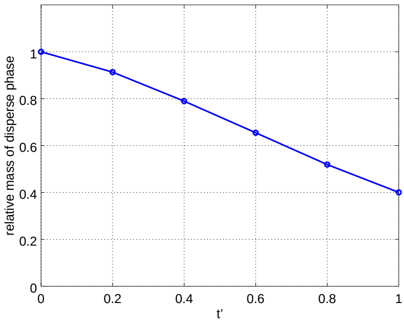
But, is it really so?

Let us consider "air" in water ($\rho_1 = 1000 kg/m^3, \rho_2 = 1 kg/m^3$).

Issue II: balance your "mass" account

The bottle experiment (air in water): **original** algorithm



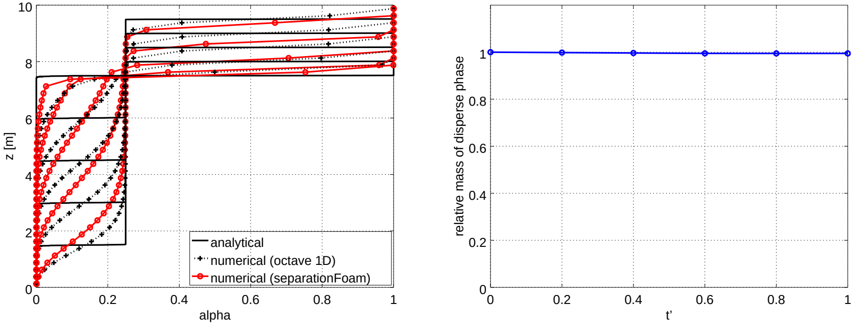


Disperse fraction profiles & mass fraction

- **Huge mass loss** (due to both bad mass accounting AND truncation).
- `separationFoam` and `sim1D` **solutions differ** (they shouldn't!)

Issue II: balance your "mass" account

The bottle experiment (air in water): **predictor-corrector** algorithm



Disperse fraction profiles & mass fraction

- **Mass IS conserved**, but...
- `separationFoam` and `sim1D` **solutions still differ!**

Issue III: use a “compressible” PISO loop

So, we are still in trouble ⇒ **Issue III**.

Hint: It has to be related to the equations **not solved in** `sim1D` (i.e. those of the PISO loop).

- Symptom: `separationFoam` and `sim1D` solutions differ.

Issue III: use a "compressible" PISO loop

So, we are still in trouble ⇒ **Issue III**.

Hint: It has to be related to the equations **not solved in** `sim1D` (i.e. those of the PISO loop).

- Symptom: `separationFoam` and `sim1D` solutions differ.
- Cause: PISO loop is not dealing well with the "compressible" mixture

$$\sum_f \left(\frac{1}{a_i}\right)_f (\nabla p)_f \cdot \mathbf{S}^f = \underbrace{\frac{\partial \rho_m^i}{\partial t} V^i}_{\text{how to deal with this?}} + \sum_f \Phi_U^f$$

leading to a solution that has $\mathbf{j} \neq 0$.

Issue III: use a "compressible" PISO loop

So, we are still in trouble $\Rightarrow$ **Issue III**.

Hint: It has to be related to the equations **not solved in** `sim1D` (i.e. those of the PISO loop).

- Symptom: `separationFoam` and `sim1D` solutions differ.
- Cause: PISO loop is not dealing well with the "compressible" mixture

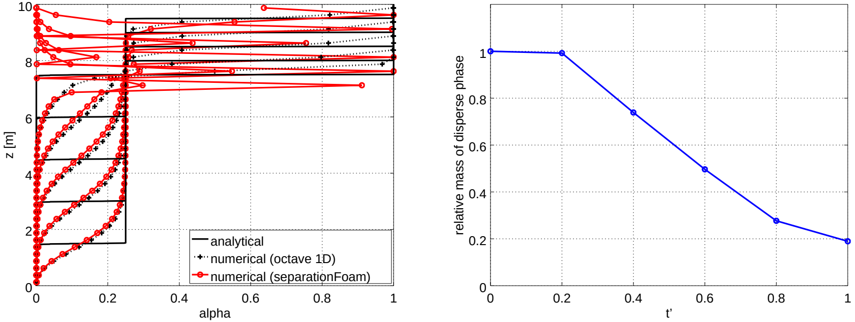
$$\sum_f \left(\frac{1}{a_i}\right)_f (\nabla p)_f \cdot \mathbf{S}^f = \underbrace{\frac{\partial \rho_m^i}{\partial t} V^i}_{\text{how to deal with this?}} + \sum_f \Phi_U^f$$

leading to a solution that has $\mathbf{j} \neq 0$.

- Remedy: a possible approach is to use $\partial \rho_m^i / \partial t$ from the previous step.

Issue III: use a “compressible” PISO loop

The bottle experiment (air in water): **predictor-corrector** algorithm



Disperse fraction profiles & mass fraction

- The **water front** from `separationFoam` and `sim1D` are **much closer**,
- but the **air front** is a **total disaster!**

Issue III: use a “compressible” PISO loop

To conclude this issue:

- The “compressible” PISO explains the differences between the two numerical solutions, and is the right way to go, although some problems remain.

Issue III: use a “compressible” PISO loop

To conclude this issue:

- The “compressible” PISO explains the differences between the two numerical solutions, and is the right way to go, although some problems remain.
- We have identified that the problems are related to the way OpenFOAM imposes a reference pressure at some point, which is required for closed domains to overcome the “up to an aditive constant” indefinition of the pressure field.

Issue III: use a “compressible” PISO loop

To conclude this issue:

- The “compressible” PISO explains the differences between the two numerical solutions, and is the right way to go, although some problems remain.
- We have identified that the problems are related to the way OpenFOAM imposes a reference pressure at some point, which is required for closed domains to overcome the “up to an aditive constant” indefinition of the pressure field.
- We tracked down the problem to either inadequate use of the available boundary conditions (meaning US) or inadequate handling of these boundary conditions in the coding (meaning THEM).

Issue III: use a “compressible” PISO loop

To conclude this issue:

- The “compressible” PISO explains the differences between the two numerical solutions, and is the right way to go, although some problems remain.
- We have identified that the problems are related to the way OpenFOAM imposes a reference pressure at some point, which is required for closed domains to overcome the “up to an aditive constant” indefinition of the pressure field.
- We tracked down the problem to either inadequate use of the available boundary conditions (meaning US) or inadequate handling of these boundary conditions in the coding (meaning THEM).
- We owe you the final answer... Still working on it.

Conclusions

- We have found some issues and developed and implemented remedies for a drift flux model as applied to separation problems.
- The resulting code is able to give solutions even with extreme density ratios (i.e. 1/1000) and high Courant numbers.
- There are unsolved issues related to the use and/or coding of boundary conditions and their interaction with the "compressible" PISO loop.

Acknowledgments

The authors wish to thank

- Universidad Argentina de la Empresa (support under Grant P17T03)
- ENIEF 2017 organizers
- You, for your kind attention