Using the isoAdvector Geometric VOF Method for Interfacial Flows Through Porous Media

MARINE 2021

Konstantinos Missios[1,*], **Niels G. Jacobsen**[2], **Kasper Moeller**[1] **and Johan Roenby**[1]

[1] IMFUFA, Department of Science and Environment, Roskilde University, Roskilde, Denmark

[2] Vattenfall, Copenhagen, Denmark

[*] Corresponding author: Konstantinos Missios, missios@ruc.dk

ABSTRACT

We consider the interfacial flow in and around porous structures in coastal and marine engineering. During recent years, interfacial flow through porous media has been repeatedly simulated with Computational Fluid Dynamics (CFD) based on algebraic Volume Of Fluid (VOF) methods (Jensen et al., 2014; Higuera et al., 2014). Here, we present an implementation of a porous medium interfacial flow solver based on the geometric VOF method, isoAdvector (Roenby et al., 2016; Roenby et al., 2017). In our implementation, the porous media is treated without resolving the actual pore geometry. Rather, the porous media, pores, and rigid structure are considered a continuum and the effects of porosity on the fluid flow are modelled through source terms in the Navier-Stokes equations, including Darcy-Forchheimer forces, added mass force and accounting for the part of mesh cells that are occupied by the solid material comprising the skeleton of the porous medium. The governing equations are adopted from the formulation by Jensen *et al.* (2014). For the interface advection using isoAdvector, we also account for the reduced cell volume available for fluid flow and for the increase in the interface front velocity caused by a cell being partially filled with solid material. The solver is implemented in the open source CFD library OpenFOAM®. It is validated using two case setups: 1) A pure passive advection test case to compare the isolated advection algorithm against a known analytical soltuion and 2) a porous dam break case by Liu *et al.*(1999) where both numerical and experimental results are available for comparison. We find good agreement with numerical and experimental results. For both cases the interface sharpness, shape conservation as well as volume conservation and boundedness are demonstrated to be very good. The solver is released as open source for the benefit of the coastal and marine CFD community.

Keywords: interfacial flow; porous medium; breakwater; isoAdvector; Geometric VOF; OpenFOAM.

NOMENCLATURE

$\langle\ \rangle$	Superficial volume average
$\langle\ \rangle^f$	Intrinsic volume average
$\bar{\ }$	Ensemble average
C_m	Added mass coefficient [-]
n	Porosity [-]
ρ	Fluid density [kg m^{-3}]
g_i	Gravitational acceleration vector [m s^{-2}]
μ	Fluid dynamic viscosity [kg m^{-1} s^{-1}]
CFD	Computational Fluid Dynamics
VOF	Volume Of Fluid
FVM	Finite Volume Method
VARANS	Volume Averaged Reynolds Averaged Navier–Stokes
MULES	Multidimensional Universal Limiter with Explicit Solution

1 INTRODUCTION

Interfacial flow through porous media appears in numerous engineering problems. For instance coastal, marine, oil and gas as well as environmental applications such as waste management facilities. In this work we focus on coastal and marine applications; two typical examples from this field are breakwaters and seawalls. These structures often contain porous regions intended to absorb the kinetic energy of the incoming waves. Given the complexity of these flows, analytical flow solutions are intractable. The predictable power of experimental model tests, on the other hand, are limited by scaling effects. Numerical simulation provides an alternative engineering tool from which important flow features can be extracted such as pressure and velocity distribution and surface elevation.

Fully resolving the flow in the complex fluid domain inside the porous structure is typically too computationally expensive. A more convenient modeling strategy is to treat the porous medium in the volume averaged spirit of the Finite Volume Method (FVM). The first systematic study within an FVM framework was carried out by van Gent (1995), who utilized the Volume Averaged Reynolds Averaged Navier Stokes (VARANS) and added explicit source terms to the equations to account for the resistance forces exerted on the fluid by the porous material. Those explicit forces were based on the Darcy-Forchheimer equations. Later Liu *et al.* (1999) conducted experiments in order to validate van Gent's work. More recently, Higuera *et al.* (2014) applied van Gent's approach in their implementation, denoted IHFOAM, of an intrinsic VARANS set of equations along with a two-equation turbulence closure model to take into account the turbulent stresses. Their interface advection used a Multidimensional Universal Limiter with Explicit Solution (MULES) scheme in OpenFOAM. Similarly Jensen *et al.* (2014) implemented the porous flow equatoins in the waves2Foam package (Jacobsen et al., 2012) also using MULES for interface advection. Their implementation contained a corrected interface advection taking properly into account the reduced cell volume available for fluid flow when it is partially filled with porous material. Turbulence was not modelled and all turbulent effects were assumed incorporated in the body force in the momentum equation.

In contrast to the previous studies that used algebraic VOF for the interface advection part, in this paper we employ a geometric VOF method called isoAdvector. The algorithm performs very well in terms of interface sharpness, volume fraction conservation, and boundedness, Roenby *et al.* (2016). The method is well-documented and tested in multiple set-ups (Vukcevic et al., 2019; Laurila et al., 2019; Meredith et al., 2017). Especially the sharp interface of isoAdvector is a desirable feature when simulating waves interacting with porous structures in coastal and marine engineering. Our newly developed solver, porousInterIsoFoam, is essentially an extension of the existing interfacial flow solver, interIsoFoam, in OpenFOAM enabling it to model flows in and around porous structures.

In the remaining manuscript we briefly describe the theoretical background for the implemented solver. We then present two validation cases using porousInterIsoFoam for 1) pure passive advection of a disc through a porous region and 2) reproducing the porous dam break experiment conducted by Liu *et al.* (1999).

2 THE GOVERNING EQUATIONS

In this section we present the porous interfacial flow equations implemented in our new solver.

2.1 The VARANS equations

For the treatment of the momentum equation, we follow the derivation of Jensen *et al.* (2014). By applying a superficial volume average to the Reynolds Averaged Navier–Stokes (RANS) equations, the VARANS equations are derived. In these equations, the effect of the porous region on the flow is included based on the Darcy-Forchheimer equation via the linear and non-linear resistance forces and an added mass force proportional to the fluid acceleration. The resistance coefficients are determined according to van Gent (1995) and also account for turbulent effects.

The volume averaged continuity equation is given as

$$\frac{\partial \langle \overline{u}_i \rangle}{\partial x_i} = 0, \tag{1}$$

where u_i is the i'th component of the velocity field, the overbar represents ensemble averaging and the brackets represent the superficial volume average. The volume-averaged, Reynolds averaged momentum equation can be formulated as

$$(1 + C_m) \frac{\partial}{\partial t}\left(\frac{\rho \langle \overline{u}_i \rangle}{n}\right) + \frac{1}{n}\frac{\partial}{\partial x_j}\left(\frac{\rho \langle \overline{u}_i \rangle \langle \overline{u}_j \rangle}{n}\right) = -\frac{\partial \langle \overline{p} \rangle^f}{\partial x_i} + g_j x_j \frac{\partial \rho}{\partial x_i} + \frac{1}{n}\frac{\partial}{\partial x_j}\mu\left(\frac{\partial \langle \overline{u}_i \rangle}{\partial x_j} + \frac{\partial \langle \overline{u}_j \rangle}{\partial x_i}\right) + F_i, \tag{2}$$

Here the added mass coefficient, C_m is modelled as $C_m = 0.34(1-n)/n$, where n is the effective porosity field. This is given by $n = V_f/V$, where V_f is the void volume within a cell and V is the total volume of the cell. ρ is the fluid density, $\langle \overline{p} \rangle^f$ is the intrinsic volume averaged pressure, $\vec{g}$ and $\vec{x}$ are the gravity and position vector, and μ is the dynamic viscosity. The last term, F_i contains the combined resistance force term exerted by the porous region on the flow. It is modeled as

$$F_i = \rho \langle \overline{u}_i \rangle \left(a + b \left| \langle \overline{u}_i \rangle \right|\right), \tag{3}$$

where a and b are the resistance coefficients determined by van Gent (1995). The reader is referred to Jensen *et al.* (2014) where a more complete account is given of these equations.

2.2 Interface advection in porous media

The evolution of the fluid interface can be described by a cell volume integrated form of the continuity equation,

$$\frac{\partial}{\partial t}\int_V \rho(\vec{x},t)dV + \int_{\partial V} \rho(\vec{x},t)u_i(\vec{x},t)dS_i = 0, \tag{4}$$

here dS_i is the differential area vector pointing out of the volume V. When employing a VOF method we define the volume fraction of phase-1 fluid within cell j as,

$$\alpha^j(t) = \frac{1}{V^j}\int_{C^j} H(\vec{x},t)dV, \tag{5}$$

where H is a three-dimensional Heaviside function equal to unit inside the phase-1 regions and zero elsewhere. The outcome of this definition is a volume fraction field that takes values in the range from 0 to 1, $\alpha^j \in [0,1]$; when $\alpha^j = 0$ the corresponding cell is filled with phase-2 fluid, on the contrary $\alpha^j = 1$ implies that the cell is totally occupied by phase-1. Finally, when $\alpha^j \in (0,1)$ an interface is present within this cell.

By time integrating the discretized form Equation 4 over the interval $[t, t+\Delta t]$ one can arrive at

$$\alpha^j(t+\Delta t) = \alpha^j(t) - \frac{1}{V^j}\sum_{f^j}\int_t^{t+\Delta t}\int_f H(\vec{x},t)u_i(\vec{x},t)dS_i d\tau, \tag{6}$$

where, V^j is the total volume of cell j and f^j the set of faces of cell j. The double integral of the RHS of Equation 6 is essentially the outward advected phase-1 fluid from cell j through its face f during the time interval $[t, t+\Delta t]$. Without getting into much detail, isoAdvector evaluates this integral, assuming a temporally constant advecting face flux, $\phi_k(t)$, on face k, yielding the following approximation,

$$\int_t^{t+\Delta t}\int_k H(\vec{x},t)u_i(\vec{x},t)dS_i d\tau \approx \frac{\phi_k(t)}{S_k}\int_t^{t+\Delta t} A_k(\tau)d\tau, \tag{7}$$

where S_k is the face area and $A_k(\tau)$ is the submerged area of face k. To calculate this area, it is important to know how the interface moves inside a cell during the interval $[t, t + \Delta t]$ and hence how it sweeps the face of interest k. The interface inside a cell is by an approximately planar polygonal face with a well-defined center x_i^I and a unit normal vector n_i^I. The interface advection velocity U_i^I is modelled to be equal to the interpolated velocity from the neighbouring cell velocities at x_i^I. The aforementioned setup results in a unique description of how interface travels within a cell and how it sweeps a given face k. Based on that model $A_k(\tau)$ is given as a second-order polynomial in time that can be further integrated as per Equation 7. In favour of clarity, all the details of the isoAdvector method are left aside. The reader is referred to Roenby *et al.* (2016) and Scheufler and Roenby (2019) that describe in detail how isoAdvector works.

In order to extend isoAdvector to flows in porous media we must make a number of modifications to the existing algorithm. First, the available void volume within cell j is only a portion n^j of the total volume. As a result, the volume fraction definition in Equation 5 should be altered to

$$\alpha^j(t) = \frac{1}{n^j V^j} \int_{C^j} H(\vec{x}, t) dV, \tag{8}$$

in this way the volume fraction field retains the property to be in the range 0 to 1 with $\alpha^j = 1$ to refer to a cell with its void volume filled with phase-1 fluid and $\alpha^j = 0$ to a cell with its void volume filled with phase-2 fluid. Subsequently the above volume fraction regulation directly affects Equation 6, which in order to accommodate for the presence of a porous medium becomes

$$\alpha^j(t + \Delta t) = \alpha^j(t) - \frac{1}{n^j V^j} \sum_{f^j} \int_t^{t+\Delta t} \int_f H(\vec{x}, t) u_i(\vec{x}, t) dS_i d\tau. \tag{9}$$

The isoAdvector algorithm evaluates the second term of the RHS in Equation 9 based on geometric operations. Those operations require an estimate for the advection velocity vector U_i^I at interface center $\vec{x}^I$. This estimation is essentially an interpolation of the velocity values stored at the neighbouring cell-centers to the interface center. However when a porous medium is present, the advection velocity has to be adjusted accordingly. This is because the interface in a cell is a Lagrangian surface and thus it will be advected with the superficial velocity. While the velocity values stored in the neighbouring cell centers are an intrinsic representation of the velocity field. As a result, the interpolated velocity at the interface center $\vec{x}^I$ has to be divided by the porosity value of the cell; therefore the adjusted interface advection velocity will be $U_i^{I\prime} = U_i^I / n$.

3 BENCHMARK CASES

Here we present validation of the porousInterIsoFoam solver in two different cases. Initially, a passive advection case is simulated, and afterwards, we reproduce the experimental results of Liu *et al.* (1999). These simulations aim to evaluate/illustrate some critical aspects of this new solver, like interface sharpness, shape preservation, boundedness, and volume conservation.

3.1 Passive Advection of a Disc Through a Porous Region

In this section, the passive advection of a circular water blob through a uniform porous region is simulated. The term *passive* implies that the flow field remains constant, and undisturbed by the presence of the water blob or the porous region throughout the simulation. As a result, this case will allow us to have a closer look at the interface advection algorithm isolated from the complex porous flow phenomena. As depicted in Figure 1 the computational set-up consists of a rectangular domain with length $L = 5$[m] and height $H = 3$[m]. The rectangular domain contains an inner porous region of length $L_p = 2$[m] with porosity $n = 0.5$. The circular blob with center $(x_0, y_0) = (0.5, 0.5)$ [m] and radius $R = 0.25$[m] is advected diagonally with a velocity $U = (1, 0.5)$[m s^{-1}] through the porous

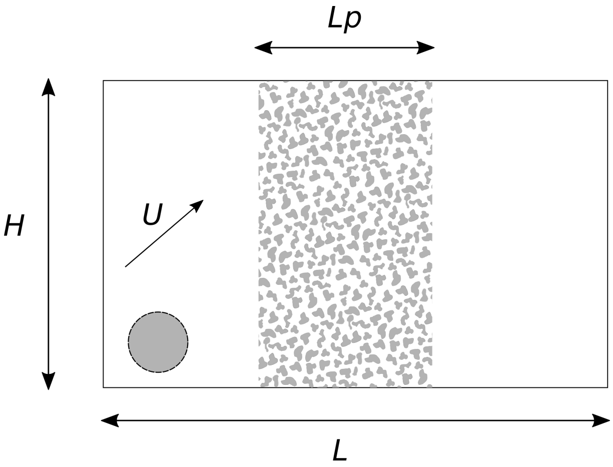


Figure 1: Sketch of the set up for the disc in constant flow through porous region case. The disc is shown in grey with a dashed outline, while the porous region located in the middle of the figure is depicted in a grey-white pattern. U refers to the constant velocity vector, L is the total length of the domain, H is the total height of the domain, and L_p is the length of the porous region in the horizontal direction.

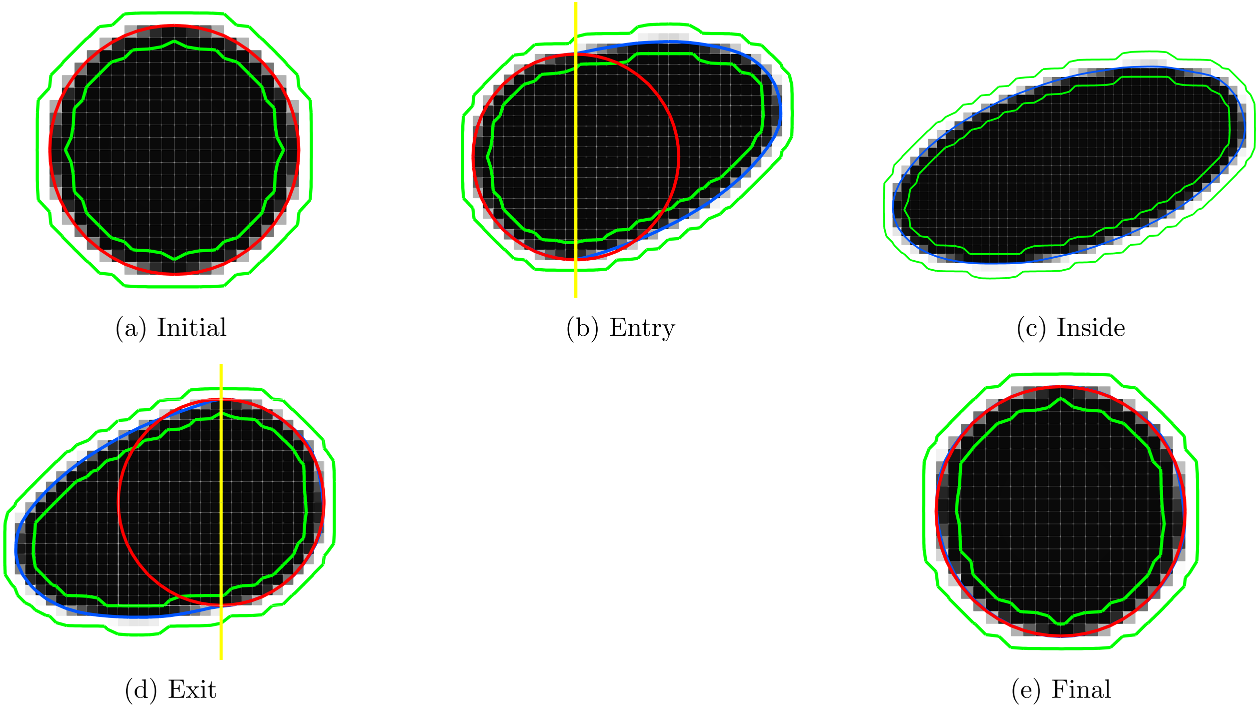
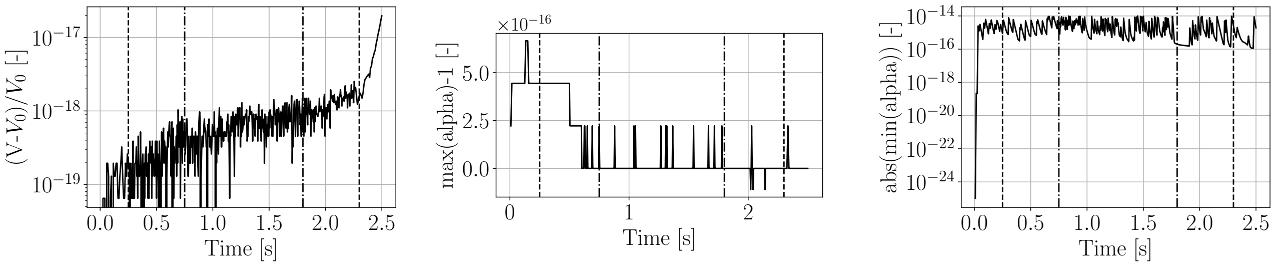


Figure 2: Disc in uniform flow at various phases (Initial, during the entry, inside, during the exit and outside of the porous region). Volume fractions are shown in grey scale while the initial $\alpha = 0.5$ contour is illustrated in red. $\alpha = 0.5$ contour is shown in blue, and $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown in green. In yellow is shown the porous region boundary.

region. The simulation lasts $t = 2.5[\,\mathrm{s}\,]$. During this time the disc enters, passes through, and exits the porous region. Lastly, the timestep selection is adjustable in order to keep the maximum interfacial Courant number below 0.5. The different stages of the disc during the simulation are illustrated in Figure 2. As the disc enters the porous region, it elongates in the advection direction, and its advection velocity becomes double. Afterwards, the elongated ellipsoidal blob is transformed back to its original shape as it leaves the porous domain. This initial visual inspection indicates that the advection algorithm works as expected. The interface sharpness is illustrated with the $\alpha = 0.01$ and



(a) Volume fraction error

(b) Upper

(c) Lower

Figure 3: Absolute normalized error, upper and lower bounding of water volume fraction as a function of time. The dashed lines represent the time instances that the disc is externally tangent to the porous region, in contrast the dash-doted lines show the time instances that the disc is internally tangent to the porous region.

$\alpha = 0.99$ contours. They should be as close to each other as possible with the given mesh resolution. They are observed to be three cells apart during the whole course of the simulation. The blue and red lines in each panel on the figure show the actual and theoretical circular initial and final interface shape, respectively. In the part of the simulation where some of the disk is inside the porous region, its shape gets distorted because of the different intrinsic velocities inside and outside the porous region. Nevertheless, the disk is observed to regain its circular shape very accurately upon exit from the porous region.

In Figure 3a a plot of the normalized absolute error of total water volume fraction is shown. The error is observed to be in the order of the machine precision. One more outcome that follows from the properties of the isoAdvector algorithm is the satisfying upper and lower bounding behaviour. Here is evaluated by monitoring the maximum overshoot ($\max\left(\alpha^j\right) - 1$) and undershoot ($\left|\min\left(\alpha^j\right)\right|$). It is clearly visible in Figure 3b & 3c that both maximum undershoot and overshoot values of the water volume fraction are confined close to machine precision.

3.2 Porous Dam Break

The second simulation case intends to reproduce an experiment of a porous dam break conducted by Liu *et al.* (1999), this particular experiment has been a reference point for many past developments in the field. Here we evaluate the solver behaviour as a whole including the combined interface advection and porous momentum equation. Figure 4 illustrates the simulation set up. On the left there is a column of water that is released at $t = 0$[s]. Then water flows through the uniform porous block placed in the middle of the domain, until at some point the system finds equilibrium. The full description of the experimental setup can be found in Liu *et al.* (1999).

The water elevation profiles for different time instances are illustrated in Figure 5. It is clear that porousInterIsoFoam predicts the water elevation accurately in most of the simulation. There is a deviation in the initial phase, this is presumably due to modelling of the water gate; in the simulation the whole water column is released instantly while in reality it is a finite process that takes around 0.1[s].

Figure 6 illustrates the interface sharpness in the simulation. Here the $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown to be separated by three cells which is the best one can do with a VOF representation.

In Figure 7a we show the normalized volume fraction evolution over the course of the simulation (i.e. 4[s]). Here it is shown that the new solver has excellent volume fraction conservation properties since the maximum normalized volume fraction deviation remains below the order of 10^{-10}. The accumulation of the water volume fraction error is associated with the iterative pressure solver and its convergence criteria. We have verified numerically that when stricter tolerance is applied this error

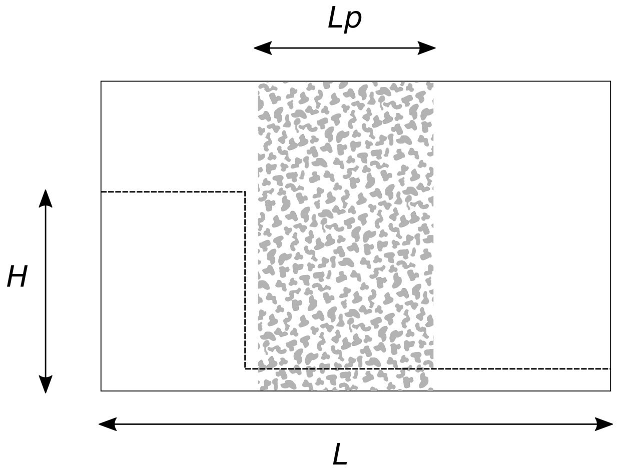


Figure 4: Sketch of the set-up for the porous dam break case. The free surface of the initial water distribution is shown in a dashed line, while the porous region located in the middle of the figure is depicted with a grey-white pattern. L is the total length of the domain, H is the height of the initial water column, and L_p is the length of the porous region in the horizontal direction.

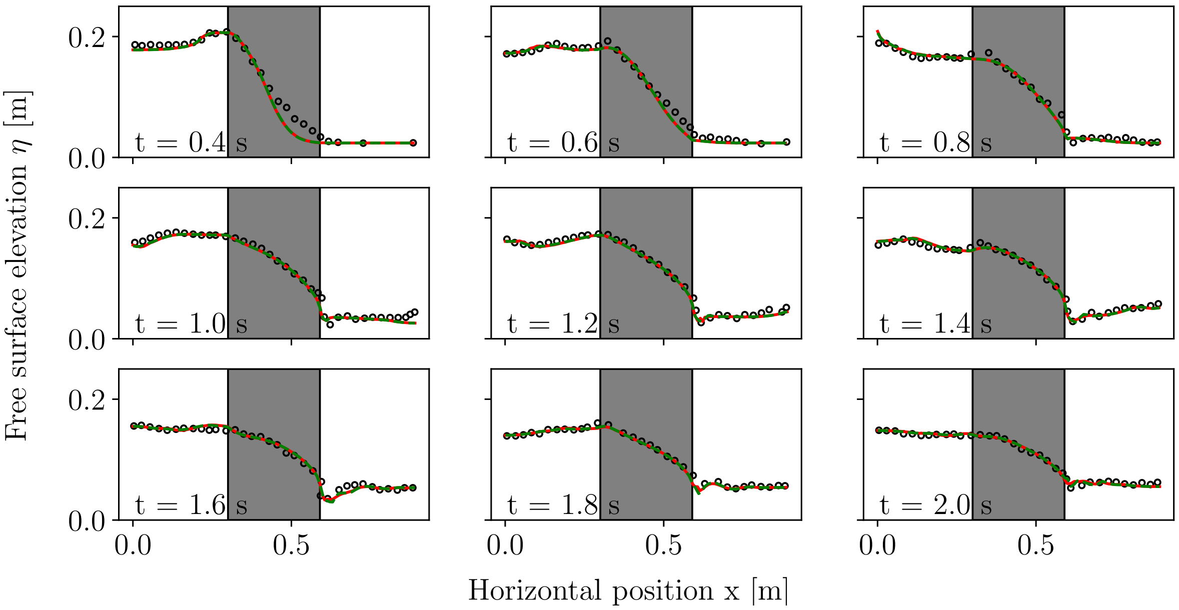


Figure 5: Comparison of free surface elevation profiles at 9 different time instances. The experimental data, shown in block circles, are measurements presented in Liu *et al.* (1999) while the numerical values, produced by porousWaveFoam, are shown in green dashed line (interFoam-based solver using MULES advection scheme in waves2Foam). The free surface values acquired by our new implementation are shown in red line. The gray rectangle in the middle of the domain highlights the porous region.

becomes smaller.

Regarding the upper and lower bounding of the volume fraction, one can observe in Figure 7b & 7c, again, very promising results. The maximum overshoot of the volume fraction field is confined below 10^{-10} while the undershoot is retained down to machine precision.

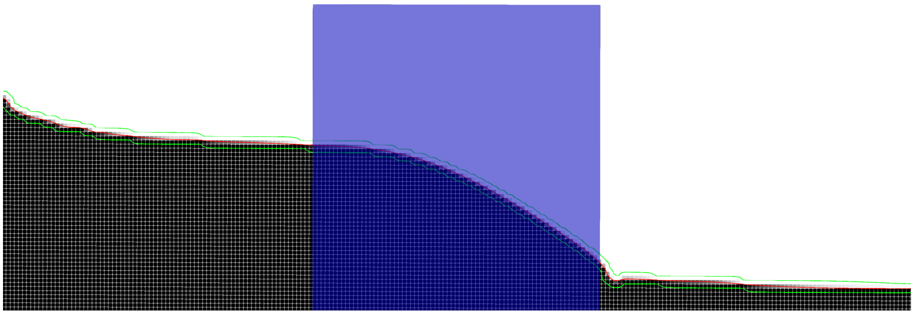
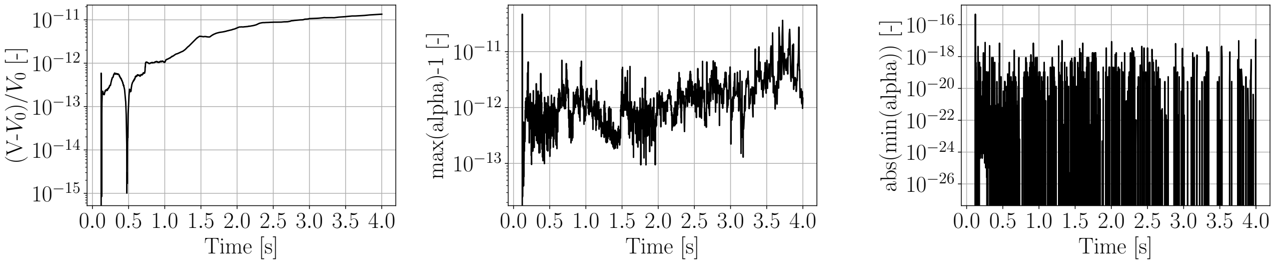


Figure 6: Porous dam break at $t = 0.8\,[\mathrm{s}]$. The simulation is performed with porousInteIsoFoam. Volume fractions shown in grey scale while the $\alpha = 0.5$ contour is illustrated in red. $\alpha = 0.01$ and $\alpha = 0.99$ contours shown in green. The porous region is shown in blue.



(a) Volume fraction error

(b) Upper

(c) Lower

Figure 7: Absolute normalized error, upper and lower bounding of water volume fraction as a function of time for the porous dam break case.

4 CONCLUSION

An extension of the interfacial solver interIsoFoam has been derived and implemented. The extended solver, porousInterIsoFoam, is capable of simulating two-phase flows inside and around porous regions. The momentum equation implementation is based on the analysis of Jensen *et al.* (2014), while the geometric VOF method, isoAdvector, is extended to account for a porous medium occupying part of the volume of computational cells. Using isoAdvector for interface advection gives rise to improved interface shape sharpness as well as good volume conservation and boundedness. The solver was validated with a pure advection test case demonstrating excellent shape preservation. It was also compared against numerical and experimental data in a porous dam break case where it accurately matched the interface shape measurements while also conserving fluid mass to a high precision with a sharp and well-bounded volume fraction field. The porousInterIsoFoam solver is released as an open-source tool that can facilitate coastal and marine studies of interfacial flows that involve porous regions.

REFERENCES

Higuera, P., Lara, J.L., Losada, I.J., 2013. Realistic wave generation and active wave absorption for Navier–Stokes models. Coastal Engineering 71, 102–118. https://doi.org/10.1016/j.coastaleng.2012.07.002

Higuera, P., Lara, J.L., Losada, I.J., 2014. Three-dimensional interaction of waves and porous coastal structures using OpenFOAM®. Part I: Formulation and validation. Coastal Engineering 83, 243–258. https://doi.org/10.1016/j.coastaleng.2013.08.010

Jacobsen, N., Fuhrman, D., Fredsoe, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFoam (R). International Journal for Numerical Methods in Fluids. 70. 10.1002/fld.2726.

Jensen, B., Jacobsen, N.G., Christensen, E.D., 2014. Investigations on the porous media equations and resistance coefficients for coastal structures. Coastal Engineering 84, 56–72. https://doi.org/10.1016/j.coastaleng.2013.11.004

Laurila, E., Roenby, J., Maakala, V., Peltonen, P., Kahila, H., Vuorinen, V., 2019. Analysis of viscous fluid flow in a pressure-swirl atomizer using large-eddy simulation. International Journal of Multiphase Flow 113, 371–388. https://doi.org/10.1016/j.ijmultiphaseflow.2018.10.008

Liu, P.L.F., Lin, P., Chang, K.A., Sakakiyama, T., 1999, "Numerical modeling of wave interaction with porous structures". J. Waterw. Port Coast. Ocean Eng. 125, 322–330.

Meredith, K.V., Zhou, X., Wang, Y., 2017. Towards Resolving the Atomization Process of an Idealized Fire Sprinkler with VOF Modeling 6–8. https://doi.org/10.4995/ilass2017.2017.5014

Roenby, J., Bredmose, H., Jasak, H., 2016. A computational method for sharp interface advection. Royal Society Open Science 3. https://doi.org/10.1098/rsos.160405

Roenby, J., Bredmose, H., Jasak, H., 2018. IsoAdvector: Geometric VOF on general meshes, in: OpenFOAM - Selected Papers of the 11th Workshop. Springer.

Scheufler, H., Roenby, J., 2019. Accurate and efficient surface reconstruction from volume fraction data on general meshes. Journal of Computational Physics 383, 1–23. https://doi.org/10.1016/j.jcp.2019.01.009

Van Gent, M., 1995. Wave interaction with permeable coastal structures. Delft University of Technology, Delft.

Vukcevic, V., Keser, R., Jasak, H., Battistoni, M., Im, H., Roenby, J., 2019. Development of a CFD Solver for Primary Diesel Jet Atomization in FOAM-Extend. Presented at the 14th International Conference on Engines & Vehicles, pp. 2019-24–0128. https://doi.org/10.4271/2019-24-0128