

Extending the isoAdvector Geometric VOF Method to Flows in Porous Media

Konstantinos Missios^{1,*}, Niels Jacobsen², Kasper L. Moeller¹, and Johan Roenby¹

¹IMFUFA, Dep. Science & Environment, Roskilde University, Universitetsvej 1, 4000 Roskilde, DK

²Vattenfall Vindkraft A/S, Orestads Blvd. 108, 2300 Copenhagen, DK

Email address: missios.konstantinos@gmail.com

DOI: <https://doi.org/10.51560/ofj.v3.72>

Results with version(s): OpenFOAM® v2012

Repository: <https://github.com/InterFlowers/porousInterIsoFoam>

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 [1,2]. Here, we present an implementation of a porous medium interfacial flow solver based on the geometric VOF method, isoAdvector [3,4]. In our implementation, the porous medium 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.* [1]. 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 solution and 2) a porous dam break case by Liu *et al.* [5] 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 (code repository <https://github.com/InterFlowers/porousInterIsoFoam>) and as of OpenFOAM-v2112 the new functionality is integrated in the official interIsoFoam solver.

★ This article is an updated version of the conference paper Missios *et al.* 2022 [6] presented at the Marine2021 conference.

1. Introduction

Interfacial flow through porous media appears in numerous engineering problems. For instance coastal and marine as well as environmental applications such as waste management facilities. In this work we focus on coastal and marine applications; typical examples from this field are breakwaters, seawalls and other perforated marine structures such as aquaculture cages [7–9]. Given the complexity of these flows, analytical flow solutions are intractable. The predictive capabilities 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, velocity distribution and surface elevation.

Fully resolving the flow in the complex fluid domain inside the porous structure is often 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 [10], 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

* Corresponding author

equations. Later Liu *et al.* [5] conducted experiments in order to validate van Gent's work. More recently, Higuera *et al.* [2] 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. For interface advection they used a Multidimensional Universal Limiter with Explicit Solution (MULES) scheme, a description of which is available from Deshpande *et al.* [11]. Similarly, Jensen *et al.* [1] implemented the porous flow equations in the waves2Foam package [12] 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. An extensive validation of this implementation was carried out by Jacobsen *et al.* [13]. A recent study performed by Feichtner *et al.* explores the use of the VARANS approach for simulations of wave interaction with thin perforated structures [14].

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 advection accuracy, interface sharpness, volume conservation, and boundedness [3]. The method is well-documented and tested in multiple areas of application [15–18]. Especially the sharp interface attributes 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.* [5]. In both cases we compare against porousInterFoam, an extension of the interfacial flow solver interFoam that can model flows in porous media. The porousInterFoam solver is a refactored version of the porousWaveFoam solver (part of the waves2Foam package). To ensure fair comparison, the porousInterFoam and porousInterIsoFoam solver only differ in the interface advection where the former uses MULES and the latter uses isoAdvector. For implementation details, please see the accompanying code repository (<https://github.com/InterFlowers/porousInterIsoFoam>).

2. The governing equations

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

2.1. The VARANS equations. For the treatment of the momentum equation, we follow the derivation of Jensen *et al.* [1]. By applying a superficial volume average¹ 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 [10] and can also account for turbulent effects inside the porous region.

The volume averaged continuity equation is given as

$$\nabla \cdot \langle \bar{\mathbf{u}} \rangle = 0, \quad (1)$$

where $\mathbf{u}$ is the velocity field, the overbar represents ensemble averaging and the angle brackets represent the superficial volume average. The volume-averaged, Reynolds averaged momentum equation can be formulated as

$$(1 + C_m) \frac{\partial}{\partial t} \left(\frac{\rho \langle \bar{\mathbf{u}} \rangle}{n} \right) + \frac{1}{n} \nabla \cdot \left(\frac{\rho \langle \bar{\mathbf{u}} \rangle \langle \bar{\mathbf{u}} \rangle}{n} \right) = -\nabla \langle \bar{p}_f \rangle + \mathbf{g} \cdot \mathbf{x} \nabla \rho + \frac{1}{n} \nabla \cdot [\mu (\nabla \langle \bar{\mathbf{u}} \rangle + (\nabla \langle \bar{\mathbf{u}} \rangle)^T)] + \mathbf{F}, \quad (2)$$

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, $\langle \bar{p}_f \rangle$ is the intrinsic volume averaged pressure, $\mathbf{g}$ and $\mathbf{x}$ are the gravity and position vector, and μ is the dynamic viscosity. The last term, $\mathbf{F}$ contains the combined resistance force term exerted by the pore skeleton on the fluid. It is modeled as

$$\mathbf{F} = \rho \langle \bar{\mathbf{u}} \rangle (a + b |\langle \bar{\mathbf{u}} \rangle|), \quad (3)$$

where a and b are the resistance coefficients determined by van Gent [10].

¹A rigorous description of the averaging process can be found in Whitaker *et al.* [19]

In the present study we carry out laminar simulations and therefore we do not consider a turbulence closure. Furthermore, the surface tension term is disregarded since the capillary effects are assumed negligible in the cases of interest. This remains a valid approximation for many marine and coastal applications which is the focus of the present implementation. The reader is referred to Jensen *et al.* [1] where a more exhaustive account of the governing equations is given.

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,

$$\frac{d}{dt} \int_V \rho(\mathbf{x}, t) dV + \int_{\partial V} \rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) \cdot d\mathbf{S} = 0, \quad (4)$$

where $d\mathbf{S}$ is the differential area vector pointing out of the cell volume V while ∂V is the surface enclosing volume V . When employing a VOF method we define the volume fraction of phase-1 fluid within cell C_i as

$$\alpha_i(t) = \frac{1}{V_i} \int_{C_i} H(\mathbf{x}, t) dV, \quad (5)$$

where V_i is the volume of cell i and H is a three-dimensional Heaviside function equal to one inside the phase-1 region and zero elsewhere. The outcome of this definition is a volume fraction field taking the value $\alpha_i = 0$ in cells filled with phase-2, $\alpha_i = 1$ in cells filled with phase-1, and $\alpha_i \in (0, 1)$ when an interface is present in the cell.

By time integrating the discretized form of Eqn. (4) over the interval $[t, t + \Delta t]$ one can arrive at

$$\alpha_i(t + \Delta t) = \alpha_i(t) - \frac{1}{V_i} \sum_{f \in \mathcal{F}_i} \int_t^{t+\Delta t} \int_f H(\mathbf{x}, \tau) \mathbf{u}(\mathbf{x}, \tau) \cdot d\mathbf{S} d\tau, \quad (6)$$

where $\mathcal{F}_i$ the set of faces of cell i . The double integral on the RHS of Eqn. (6) is the outward advected phase-1 fluid from cell i through its face f during the time interval $[t, t + \Delta t]$. In practice, isoAdvector evaluates this integral, assuming a temporally constant advecting face flux, $\phi_f(t)$, on face f , yielding the following approximation,

$$\int_t^{t+\Delta t} \int_f H(\mathbf{x}, \tau) \mathbf{u}(\mathbf{x}, \tau) \cdot d\mathbf{S} d\tau \approx \frac{\phi_f(t)}{S_f} \int_t^{t+\Delta t} A_f(\tau) d\tau, \quad (7)$$

where S_f is the face area and $A_f(\tau)$ is the submerged area of face f at time τ . 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, f . The interface inside a cell is represented by an approximately planar polygonal face with a well-defined center $\mathbf{x}_i^I$ and a unit normal vector $\mathbf{n}_i^I$. The interface advection velocity $\mathbf{U}_i^I$ is approximated by interpolating the cell centred velocity field to $\mathbf{x}_i^I$. The aforementioned setup results in a unique description of how the interface travels within a cell and how it sweeps a given face f . Based on that model $A_f(\tau)$ is given as a second-degree polynomial in time that can be further integrated as per Eqn. (7). For the sake of clarity, further details of the isoAdvector method are left aside. The reader is referred to Roenby *et al.* [3] and Scheufler and Roenby [4] where the isoAdvector algorithm is described in detail.

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 i is only a fraction n_i of the total cell volume. As a result, the volume fraction definition in Eqn. (5) should be altered to

$$\alpha_i(t) = \frac{1}{n_i V_i} \int_{C_i} H(\mathbf{x}, t) dV. \quad (8)$$

In this way the volume fraction field retains the property to be in the range 0 to 1 with $\alpha_i = 1$ to refer to a cell with its void volume filled with phase-1 fluid and $\alpha_i = 0$ to a cell with its void volume filled with phase-2 fluid. Subsequently the above volume fraction modification directly affects Eqn. (6), which in order to accommodate the presence of a porous medium becomes

$$\alpha_i(t + \Delta t) = \alpha_i(t) - \frac{1}{n_i V_i} \sum_{f \in \mathcal{F}_i} \int_t^{t+\Delta t} \int_f H(\mathbf{x}, \tau) \mathbf{u}(\mathbf{x}, \tau) \cdot d\mathbf{S} d\tau. \quad (9)$$

The isoAdvector algorithm evaluates the second term on the RHS of Eqn. (9) based on geometric operations. Those operations require an estimate for the advection velocity vector $\mathbf{U}_i^I$ at interface centre $\mathbf{x}_i^I$. As mentioned, we interpolate the cell centred velocity field to $\mathbf{x}_i^I$ to approximate $\mathbf{U}_i^I$. However, when a porous medium is present, the advection velocity must 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 cell centred velocity vectors are an intrinsic representation of the velocity field. As a result, the interpolated velocity at the interface center $\mathbf{x}_i^I$ must be divided by the porosity value of the cell. Therefore the adjusted interface advection velocity in cell i will be $\mathbf{U}_i^{I'} = \mathbf{U}_i^I/n_i$.

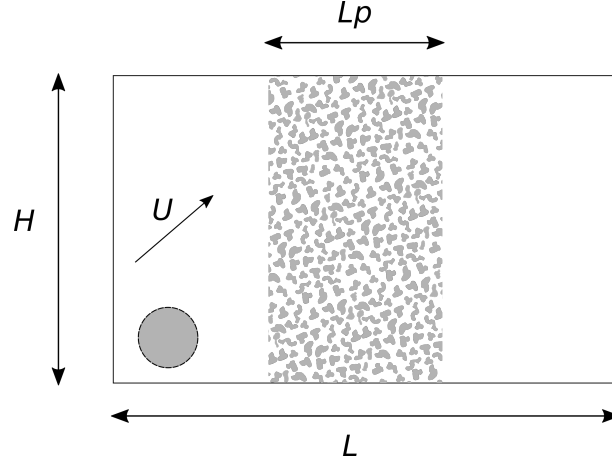


Figure 1. Sketch of the set up for the disc in constant flow through porous region case. The disc is shown in grey with a dashed outline, while the porous region located in the middle of the figure is depicted in a grey-white pattern. U refers to the constant velocity vector, L is the total length of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

3. Benchmark cases

Here we present validation of the porousInterIsoFoam solver in two different cases. Initially, a pure passive advection case is simulated, and afterwards, we reproduce the experimental results of Liu *et al.* [5]. These simulations aim to evaluate/illustrate some critical aspects of this new solver, such as 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 disc through a uniform porous region is simulated. The term *passive* implies that the flow field is prescribed and unaltered by the presence of the disc 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 Fig. 1 the computational set-up consists of a rectangular domain of length $L = 5[\text{m}]$ and height $H = 3[\text{m}]$. The domain is discretised by a uniform mesh with cell size $\Delta x = \Delta y = 0.05[\text{m}]$. The rectangular domain contains an inner porous region of length $L_p = 2[\text{m}]$ with porosity $n = 0.5$. The circular blob with center $(x_0, y_0) = (0.5, 0.5)[\text{m}]$ and radius $R = 0.25[\text{m}]$ is advected diagonally with a velocity $\mathbf{U} = (1, 0.5)[\text{m s}^{-1}]$ outside the porous region. Inside the porous region, fluid incompressibility dictates a velocity $\mathbf{U}' = \mathbf{U}/n$, hence a doubling of the interface velocity with the chosen porosity. The result is that the circular blob is stretched as it passes the left boundary into the porous region. The blob regains its circular shape as it leaves the porous region through its right boundary. If we call the left and right limits of the porous domain x_L and x_R , respectively, then the analytical trajectories for the particles comprising the circular blob can be written

$$\mathbf{x}(t) = \mathbf{x}(0) + \begin{cases} \mathbf{U}t, & t < t_L = |x_L - x(0)|/U_x \\ \mathbf{U}(t_L + (t - t_L)/n), & t_L \leq t \leq t_R = t_L + |x_L - x_R|/U_x \\ \mathbf{U}(t_L + (t_R - t_L)/n + (t - t_R)), & t_R < t \end{cases} \quad (10)$$

We stop the simulation at $t = 2.5[\text{s}]$, leaving the disc time to enter, pass through, and exit the porous region. Lastly, the time step selection is adjustable in order to keep the maximum Courant number, maxCo , at 0.25. The Courant number of a cell i with porosity n_i is defined as

$$\text{Co}_i = \frac{\Delta t}{2n_i V_i} \sum_{f \in \mathcal{F}_i} |\phi_f|. \quad (11)$$

which is identical to the existing definition in OpenFOAM when $n_i = 1$. The porous modification retains the interpretation of the Courant number as the fraction of the total fluid volume in the cell ($n_i V_i$) that

is fluxed out of the cell during the time step, Δt . We note that this porous Courant number definition poses a more strict limitation on the time step size than the non-porous version. In the simulation we also restrict the maximum *interfacial* Courant number, `maxAlphaCo`, to 0.25. In accordance with the existing `maxAlphaCo` implementation in OpenFOAM, the maximum is taken over all cells with $0.01 < \alpha < 0.99$.

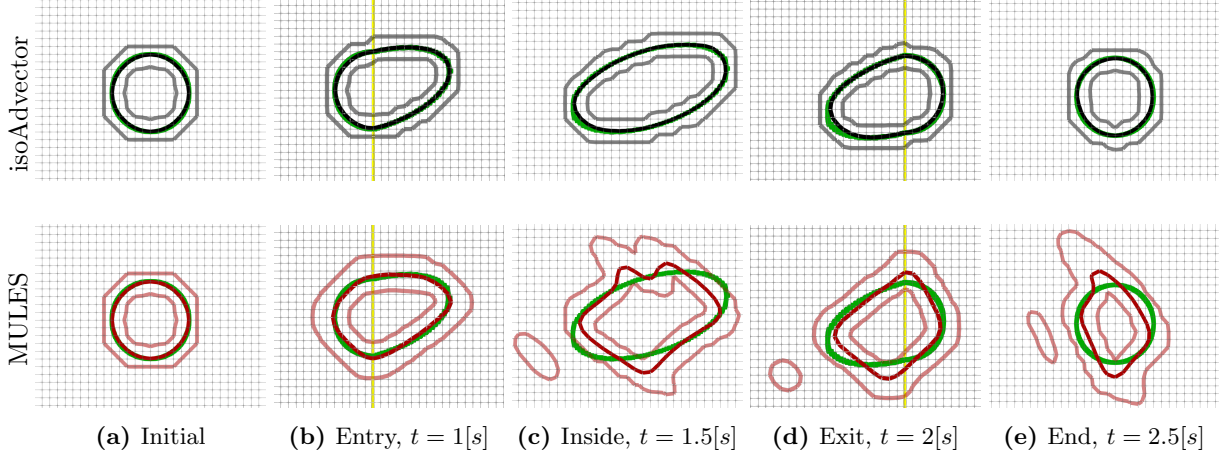


Figure 2. Disc in uniform flow at various phases (Initial, during the entry, inside, during the exit and outside of the porous region). Comparison of the shape preservation and interface sharpness for the two solvers. `porousInterIsoFoam` results are shown in the top row of panels in grey-scale ($\alpha = 0.5$ contour is illustrated in black while $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown in grey) whereas `porousInterFoam` results are shown in the bottom row of panels in red-scale ($\alpha = 0.5$ contour is illustrated in red while $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown in light red). In green is illustrated the analytical solution from Eqn. (10). Finally, the limits of the porous region are shown as yellow vertical lines.

The different stages of the disc during the simulation are illustrated in Fig. 2. As the disc enters the porous region, it elongates in the advection direction, and its advection velocity doubles. When exiting the porous region, the elongated elliptic blob transforms back to its original shape. The interface sharpness is illustrated with the $\alpha = 0.01$ and $\alpha = 0.99$ contours. Those contours should be as close as possible to each other with the given mesh resolution. In the `porousInterIsoFoam` case the contours are observed to be three cells apart during the whole course of the simulation. For `porousInterFoam` the interface width is approximately 5 cells. The green closed curve in each panel on the figure shows the theoretical interface shape from Eqn. (10). `porousInterIsoFoam` recovers an almost circular interface shape upon its exit from the porous region whereas `porousInterFoam` returns a distorted shape.

In Fig. 3A a plot of the volume conservation error is shown. For both solvers the error is observed to be around machine precision with `porousInterFoam` exhibiting a lower error compared to `porousInterIsoFoam`. In Fig. 3B the upper bounding error is depicted for both solvers. Here the `porousInterIsoFoam` error is around machine precision, whereas the `porousInterFoam` maximum α value settles around 10^{-7} below 1. This is due to the smear of the volume fraction field that subsequently leads to a maximum value below one. In Fig. 3C we show the lower bounding error, which stays below machine precision during the whole simulation for both solvers.

3.2. Porous Dam Break. In the second benchmark case we reproduce an experiment of a porous dam break conducted by Liu *et al.* [5]. This particular experiment has been a reference point for many past developments in the field (e.g. [1], [2] and [20]). Here we evaluate the solver behaviour as a whole including the combined interface advection and porous momentum equation. Fig. 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.* [5]. The domain is discretised by a uniform mesh with cell size of $\Delta x = \Delta y \approx 0.006[m]$. This simulation is carried out with a maximum interfacial Courant number of 0.5.

The water surface profiles for different time instances are illustrated in Fig. 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

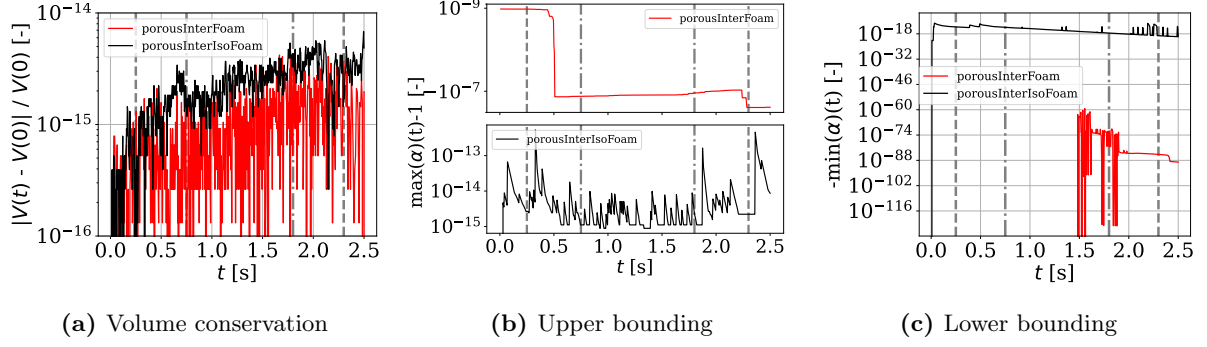


Figure 3. Comparison of volume conservation error as well as upper and lower bounding of the volume fraction for the two solvers. The two vertical dashed lines mark the time interval where the disc is in contact with the porous region. The vertical dash-dotted lines mark the time interval where the disc is fully inside the porous region.

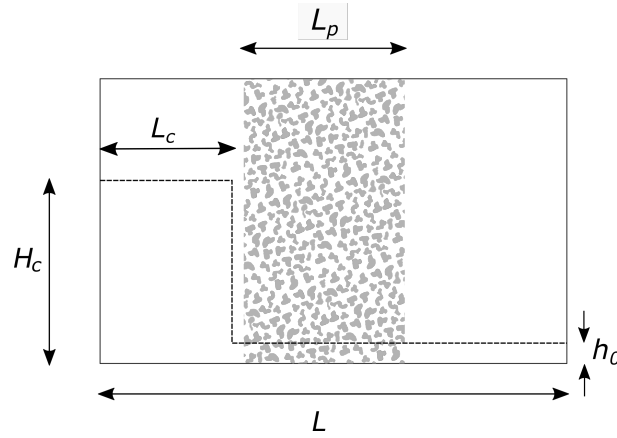


Figure 4. Sketch of the set-up for the porous dam break case. The free surface of the initial water distribution is shown in a dashed line, while the porous region located in the middle of the figure is depicted with a grey-white pattern. $L = 0.892[m]$ is the total length of the domain, $H_c = 0.24[m]$ and $L_c = 0.28[m]$ is the height and width of the initial water column, and $L_p = 0.29[m]$ is the length of the porous region in the horizontal direction. Lastly, the height of the short column is $h_0 = 0.022[m]$.

column is released instantly while in reality it is a finite process that takes around $0.1[s]$, as also noted by Liu *et al.* [5] (similar observations have been made in a different case study by Fekken *et al.* [21]).

Fig. 6 illustrates the interface sharpness capabilities of isoAdvector compared to MULES. For the case of isoAdvector the $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown to be separated by three cells which is the best a VOF representation can do. On the other hand, porousInterFoam results in a smeared interface that expands over more cells, while at the same time producing numerical vapor. Both the numerical vapor issue and the smeared interface of MULES has been reported in earlier investigations [22, 23].

In Fig. 7A we show the evolution in the volume conservation error over the course of the simulation (i.e. $4[s]$). Here it is observed that both solvers exhibit excellent volume conservation and retain the error below 10^{-13} . Fig. 7B shows that the upper bounding error is kept below 10^{-9} for both solvers. As for the lower bounding error Fig. 7C shows that both solvers retain the error below machine precision.

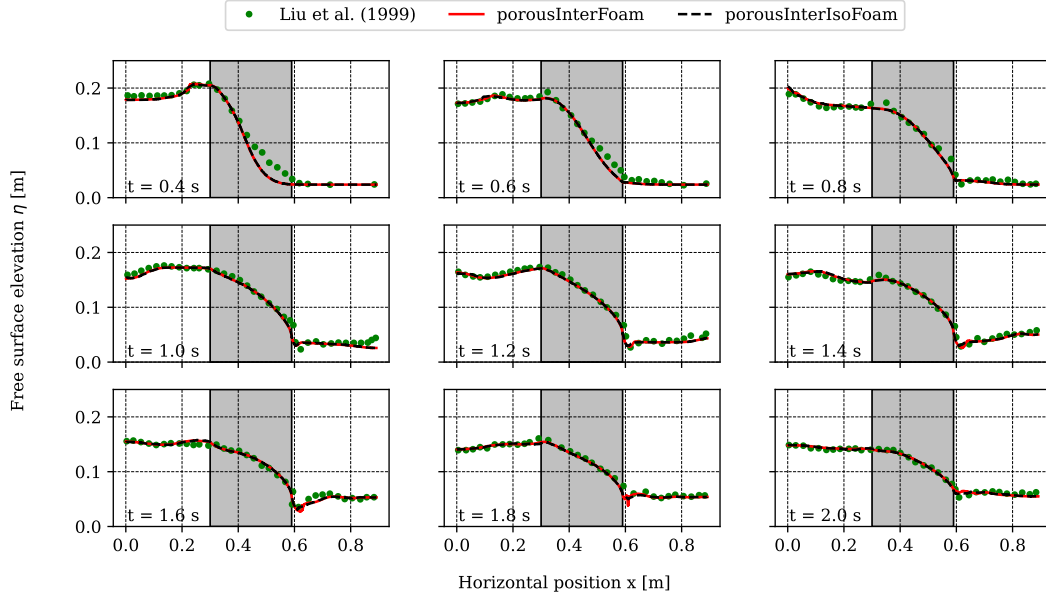


Figure 5. Comparison of free surface elevation profiles at 9 different time instances. The experimental data, shown in green dots, are measurements presented in Liu *et al.* [5] while the numerical values, produced by porousInterFoam, are shown in red line (interFoam-based solver using MULES advection scheme). The free surface values acquired by our new implementation are shown in black dashed line. The grey rectangle in the middle of the domain marks the porous region.

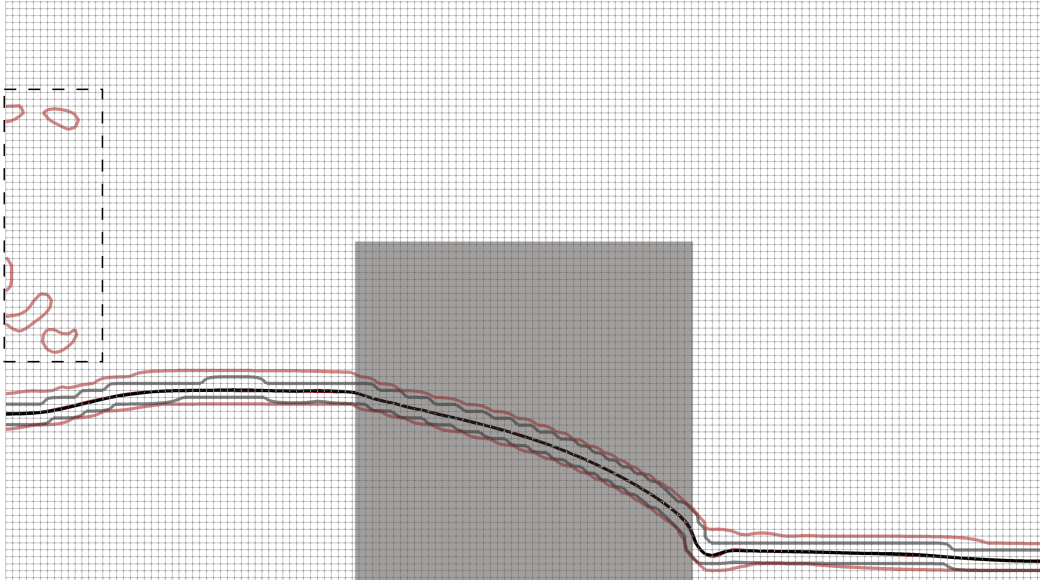


Figure 6. Comparison of the interface sharpness for the two solvers. Porous dam break at $t = 1$ [s]. The porousInterIsoFoam results are shown in black palette, $\alpha = 0.5$ contour is illustrated in black while $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown in grey. Results acquired with porousInterFoam are represented by the red palette, $\alpha = 0.5$ contour is illustrated in red (this line is overlapped by the black $\alpha = 0.5$ contour) while $\alpha = 0.01$ and $\alpha = 0.99$ contours are shown in light red. The porous region is shown in light grey. The rectangle in dashed outline highlights the area where porousInterFoam produces numerical vapor.

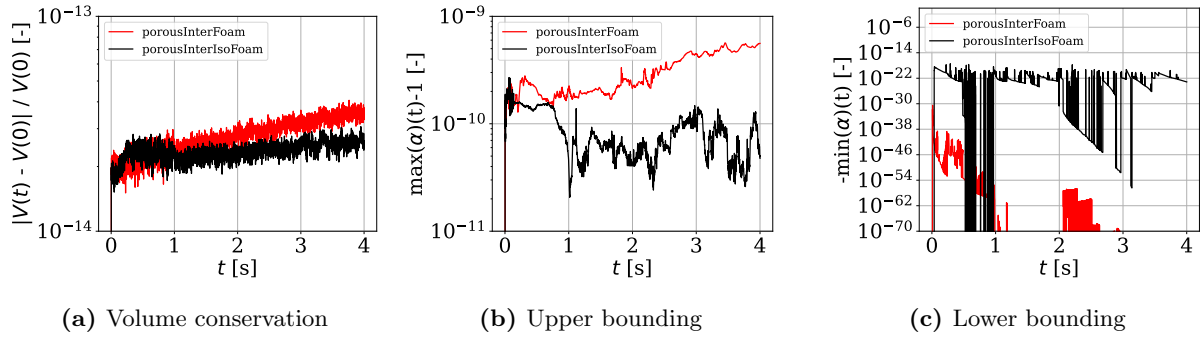


Figure 7. Comparison of errors in volume conservation and in upper and lower bounding for the two solvers.

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.* [1], 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 shape preservation while retaining excellent volume conservation and boundedness. The solver was validated with a pure advection test case demonstrating the superior shape preservation compared to a similar solver using MULES for interface advection. The porousInterIsoFoam solver was also benchmarked with 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 capable of facilitating coastal and marine studies of interfacial flows involving porous regions (code repository <https://github.com/InterFlowers/porousInterIsoFoam>). As of OpenFOAM-v2112 the porousInterIsoFoam capabilities have been merged into the official interIsoFoam solver (the reader is referred to the v2112 release notes for further details on usage).

Acknowledgements

The authors are grateful for partial funding for the presented work from Deltares and from Independent Research Fund Denmark through a DFF Sapere Aude Research Leader grant (grant number 9063-00018B).

Author Contributions: Conceptualisation, K.M, N.J. and J.R.; methodology, K.M., N.J. and J.R.; software, K.M, N.J. and J.R.; validation, K.M. and J.R.; investigation, K.M and J.R.; writing—original draft preparation, K.M. and J.R.; writing—review and editing, K.M., N.J., K.L.M. and J.R.; supervision, J.R.; funding acquisition, J.R. All authors have read and agreed to the published version of the manuscript.

References

- [1] B. Jensen, N. G. Jacobsen, and E. D. Christensen, “Investigations on the porous media equations and resistance coefficients for coastal structures,” *Coastal Engineering*, vol. 84, pp. 56–72, 2014, publisher: Elsevier B.V. [Online]. Available: <http://dx.doi.org/10.1016/j.coastaleng.2013.11.004>
- [2] P. Higuera, J. L. Lara, and I. J. Losada, “Three-dimensional interaction of waves and porous coastal structures using OpenFOAM®. Part I: Formulation and validation,” *Coastal Engineering*, vol. 83, pp. 243–258, 2014, publisher: Elsevier B.V. [Online]. Available: <http://dx.doi.org/10.1016/j.coastaleng.2013.08.010>
- [3] J. Roenby, H. Bredmose, and H. Jasak, “A computational method for sharp interface advection,” *Royal Society Open Science*, vol. 3, no. 11, 2016. [Online]. Available: <https://doi.org/10.1098/rsos.160405>
- [4] H. Scheuffer and J. Roenby, “Accurate and efficient surface reconstruction from volume fraction data on general meshes,” *Journal of Computational Physics*, vol. 383, pp. 1–23, Apr. 2019. [Online]. Available: <https://doi.org/10.1016/j.jcp.2019.01.009>
- [5] P. L.-F. Liu, P. Lin, K.-A. Chang, and T. Sakakiyama, “Numerical Modeling of Wave Interaction with Porous Structures,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 125, no. 6, pp. 322–330, Nov. 1999. [Online]. Available: [https://doi.org/10.1061/\(ASCE\)0733-950X\(1999\)125:6\(322\)](https://doi.org/10.1061/(ASCE)0733-950X(1999)125:6(322))
- [6] K. Missios, N. Jacobsen, J. Roenby, and K. Moeller, “Using the isoAdvector geometric VoF method for interfacial flows through porous media,” *The 9th Conference on Computational Methods in Marine Engineering (Marine 2021)*, vol. 1, no. 1, Jan. 2022. [Online]. Available: <https://doi.org/10.2218/marine2021.6811>

- [7] P. Lin and S. A. Karunaratna, "Numerical Study of Solitary Wave Interaction with Porous Breakwaters," *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 133, no. 5, pp. 352–363, Sep. 2007. [Online]. Available: [https://doi.org/10.1061/\(ASCE\)0733-950X\(2007\)133:5\(352\)](https://doi.org/10.1061/(ASCE)0733-950X(2007)133:5(352))
- [8] A. Koraim, E. Heikal, and A. Abo Zaid, "Hydrodynamic characteristics of porous seawall protected by submerged breakwater," *Applied Ocean Research*, vol. 46, pp. 1–14, Jun. 2014. [Online]. Available: <https://doi.org/10.1016/j.apor.2014.01.003>
- [9] H. Chen and E. D. Christensen, "Investigations on the porous resistance coefficients for fishing net structures," *Journal of Fluids and Structures*, vol. 65, pp. 76–107, Aug. 2016. [Online]. Available: <https://doi.org/10.1016/j.jfluidstructs.2016.05.005>
- [10] M. R. A. v. van Gent, "Wave interaction with permeable coastal structures," Ph.D. dissertation, Delft University, 1995. [Online]. Available: <http://resolver.tudelft.nl/uuid:7bbff8e4-215d-4bfc-a3af-51cdec754bd>
- [11] S. S. Deshpande, L. Anumolu, and M. F. Trujillo, "Evaluating the performance of the two-phase flow solver interFoam," *Computational Science & Discovery*, vol. 5, no. 1, p. 014016, Nov. 2012. [Online]. Available: <https://dx.doi.org/10.1088/1749-4699/5/1/014016>
- [12] N. G. Jacobsen, D. R. Fuhrman, and J. Fredsøe, "A wave generation toolbox for the open-source CFD library: OpenFoam®," *International Journal for Numerical Methods in Fluids*, vol. 70, no. 9, pp. 1073–1088, Nov. 2012. [Online]. Available: <https://doi.org/10.1002/fld.2726>
- [13] N. G. Jacobsen, M. R. van Gent, and G. Wolters, "Numerical analysis of the interaction of irregular waves with two dimensional permeable coastal structures," *Coastal Engineering*, vol. 102, pp. 13–29, Aug. 2015. [Online]. Available: <https://doi.org/10.1016/j.coastaleng.2015.05.004>
- [14] A. Feichtner, E. Mackay, G. Tabor, P. R. Thies, L. Johanning, and D. Ning, "Using a porous-media approach for CFD modelling of wave interaction with thin perforated structures," *Journal of Ocean Engineering and Marine Energy*, vol. 7, no. 1, pp. 1–23, Feb. 2021. [Online]. Available: <http://link.springer.com/10.1007/s40722-020-00183-7>
- [15] V. Vukcevic, R. Keser, H. Jasak, M. Battistoni, H. Im, and J. Roenby, "Development of a CFD Solver for Primary Diesel Jet Atomization in FOAM-Extend," Sep. 2019, pp. 2019–24–0128. [Online]. Available: <https://doi.org/10.4271/2019-24-0128>
- [16] E. Laurila, J. Roenby, V. Maakala, P. Peltonen, H. Kahila, and V. Vuorinen, "Analysis of viscous fluid flow in a pressure-swirl atomizer using large-eddy simulation," *International Journal of Multiphase Flow*, vol. 113, pp. 371–388, Apr. 2019. [Online]. Available: <https://doi.org/10.1016/j.ijmultiphaseflow.2018.10.008>
- [17] K. V. Meredith, X. Zhou, and Y. Wang, "Towards Resolving the Atomization Process of an Idealized Fire Sprinkler with VOF Modeling," no. September, pp. 6–8, 2017. [Online]. Available: <http://dx.doi.org/10.4995/ILASS2017.2017.5014>
- [18] T. Mukha, S. K. Almeland, and R. E. Bensow, "Large-Eddy Simulation of a Classical Hydraulic Jump: Influence of Modelling Parameters on the Predictive Accuracy," *Fluids*, vol. 7, no. 3, p. 101, Mar. 2022. [Online]. Available: <https://doi.org/10.3390/fluids7030101>
- [19] S. Whitaker, "The Forchheimer equation: A theoretical development," *Transport in Porous Media*, vol. 25, no. 1, pp. 27–61, Oct. 1996. [Online]. Available: <https://dx.doi.org/10.1007/BF00141261>
- [20] X. Sun, M. Sun, K. Takabatake, C. C. Pain, and M. Sakai, "Numerical Simulation of Free Surface Fluid Flows Through Porous Media by Using the Explicit MPS Method," *Transport in Porous Media*, vol. 127, no. 1, pp. 7–33, Mar. 2019. [Online]. Available: <https://dx.doi.org/10.1007/s11242-018-1178-z>
- [21] G. Fekken, A. Veldman, and B. Buchner, "Simulation Of Green Water Loading Using The Navier-Stokes Equations," Jun. 2001. [Online]. Available: <https://research.rug.nl/en/publications/be0032c1-a67c-46c9-bc28-2a0182397181>
- [22] J. Li, Y. You, K. Chen, and X. Zhang, "Numerical computations of resonant sloshing using the modified isoAdvector method and the buoyancy-modified turbulence closure model," *Applied Ocean Research*, vol. 93, p. 101829, Dec. 2019. [Online]. Available: <https://doi.org/10.1016/j.apor.2019.05.014>
- [23] I. Gatin, V. Vukčević, H. Jasak, J. Seo, and S. H. Rhee, "CFD verification and validation of green sea loads," *Ocean Engineering*, vol. 148, pp. 500–515, Jan. 2018. [Online]. Available: <https://doi.org/10.1016/j.oceaneng.2017.10.026>