

An *OpenFOAM* solver for computing suspended particles in water currents

Nils R. B. Olsen ^{a,*}, Subhojit Kadia ^a, Elena Pummer ^a and Gudrun Hillebrand ^b

^a Department of Civil and Environmental Engineering, The Norwegian University of Science and Technology, S. P. Andersens vei 5, N-7491 Trondheim, Norway

^b Federal Institute of Hydrology, Am Mainzer Tor 1, D-56068 Koblenz, Germany

*Corresponding author. E-mail: nils.r.olsen@ntnu.no

 NRBO, 0000-0001-8472-5292; SK, 0000-0002-9134-3222; EP, 0000-0002-0255-4715; GH, 0000-0003-1489-9443

ABSTRACT

A new OpenFOAM solver has been developed for computing the spatial variation of particle concentrations in flowing water. The new solver was programmed in C++ using OpenFOAM libraries, and the source code has been made openly available. The current article describes the coding of how the water flow and particle movements are computed. The solver is based on a Eulerian approach, where the particles are computed as concentrations in cells of a grid that resolves the computational domain. The Reynolds-averaged Navier–Stokes equations are solved by simpleFoam, using the $k-\epsilon$ turbulence model. The new solver uses a drift-flux approach to take the fall or rise velocity of the particles into account in a convection-diffusion equation. The model is therefore called sediDriftFoam. The results from the solver were tested on two cases with different types of particles. The first case was a sand trap with sand particles. The geometry was three-dimensional with a recirculation zone. The computed sediment concentrations in three vertical profiles compared well with earlier numerical studies and laboratory measurements. The second case was a straight channel flume with plastic particles that had a positive rise velocity. In this case, the results also compared well with the laboratory measurements.

Key words: CFD, numerical modelling, OpenFOAM, plastic particles, sediDriftFoam, sediments

HIGHLIGHTS

- Open source 3D sediment model.
- Based on OpenFOAM.
- Simple and easy to learn.
- Tested on computing trap efficiency of a sand trap.
- Tested on computing suspended plastic particles in a channel.

INTRODUCTION

The science of Hydroinformatics includes numerical modelling of water flow in lakes, rivers, canals and ducts (Maduka & Li 2021; Olsen 2022; Paschmann *et al.* 2022). Three-dimensional water flow fields are often found by solving the Navier–Stokes equations using a turbulence model in a science called computational fluid dynamics (CFD). The set of equations are complex to solve for a general geometry. The most used CFD programmes may have a source code with millions of lines that have taken many decades to make. There are several issues related to the complexity of codes with a large number of lines. Industry average experience is about 1–25 bugs pr. 1,000 lines of code (McConnell 2004). Also, the transfer of coding knowledge to new staff is very time-consuming. The risk of inexperienced staff introducing new bugs in the programme is considerable. Companies making CFD software are very dependent on specialised personnel with coding experience, and problems arise when key staff retire or move to other companies. Also, it is difficult for researchers outside the company to get access to the code, both for legal reasons and for the large amount of knowledge that needs to be transferred.

A solution to these problems is to use a high-level object-oriented computing language with open source access. The current article describes a solver made in *OpenFOAM*, which is based on the object-oriented C++ programming language. *OpenFOAM* is a library of different solvers that the users can extend by adding equations and thereby creating new solvers for specific flow problems. In the current study, it was possible to create a solver for computing particle transport in water

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

flow with only around 70 lines of new code. The rest of the code is in libraries that are written, maintained and tested by very experienced programmers. The new solver takes the fall/rise velocity of the particles into account by using a drift-flux approach. It is therefore called *sedDriftFoam*, as it is expected that it will mostly be used for computing sediment transport. However, it can also be used for computing transport of plastic particles. The code for the new solver is explained in detail in the current article, together with the equations that are solved. The source code for *sedDriftFoam* can be freely downloaded from the link provided in the Supplementary Material. Example input files are also given. The solver can thereby easily be used and further modified.

Earlier work using this approach includes [Sattar et al. \(2017\)](#), who programmed a sediment solver in *OpenFOAM* with a moving grid, and computed bed movements in a channel bend. Other examples in the field of sediment transport are *SediFoam* ([Sun & Xiao 2016](#)) and *sedFoam* ([Chauchat et al. 2017](#)). *SediFoam* models particles using a Lagrangian approach, where a large number of particles are released in the flow domain and their paths are tracked individually. The movements of the particles are a function of their acceleration, computed from Newton's second law of motion. The forces on the spherical particles are due to gravity, lift, drag, collision and lubrication. Several formulas with empirical coefficients were used for each force. The Lagrangian approach to modelling particles is popular within the field of fluid mechanics ([De Marchis & Milici 2016](#); [Schwarzmeier et al. 2023](#)).

sedFoam ([Chauchat et al. 2017](#)) is an Eulerian two-phase model, where fractions or concentrations of particles are computed for each cell in the grid. The model solves momentum equations both for the water and for the particles. Fluid stresses include both Reynolds stresses and granular stresses, including fluid-particle interactions on the grain scale. The model is fairly complex, with 47 equations describing momentum, continuity and different turbulence modelling options.

Both *SediFoam* and *sedFoam* focus on fundamental research on fluid mechanics and particle movements. Much simpler models are often used to solve hydraulic engineering problems, for example *Telemac*, *Delft3D*, *SSIIM* and *Flow3D*. These models assume a one-way coupling where the particles will not affect the water velocity and turbulence, except when erosion and deposition changes the geometry. An Eulerian approach is used where the particles are computed as a passive concentration. Well-known formulas for particle pick-up rates are used as boundary conditions at the channel bed. These computer programmes have been used to model most sediment transport phenomena, including local scour ([Baranya et al. 2014](#)), turbidity currents ([Basani et al. 2014](#)), reservoir sedimentation ([Mool et al. 2017](#)), reservoir flushing ([Harb et al. 2014](#)), formation of braided rivers ([Olsen 2021](#)) and bedform modelling ([Goll & Kopmann 2012](#)).

The current work describes a particle model with a one-way Eulerian approach, implemented in the *OpenFOAM* library. This simpler solver uses a drift-flux approach to take the rise or fall velocity into account when computing the dispersion of the particles. It also includes particle pick-up rates for sediment flow over a fixed bed. The model is therefore called *sedDriftFoam*. The idea behind the drift-flux model is to superpose two velocity vector fields to produce the particle movements: one water velocity field from the solution of the Reynolds-averaged Navier–Stokes equations, and another field from the particle fall velocities. The algorithms are the same as used by [Olsen & Skoglund \(1994\)](#), but the current article provides much more details about the computational programme and its integration with *OpenFOAM*.

The new solver was tested on two cases: Vertical distribution of positively buoyant plastic particles in a straight channel, and sediment deposition in a sand trap. The reason for choosing these two cases was the availability of input and verification data, and their relevance for current research in water engineering. Problems caused by plastic particles have received considerable attention over the last years. Micro and macro plastics are transported from rivers to the ocean, where they can be harmful to marine life. A number of studies have been conducted to model plastic particle dispersion in the ocean by numerical models. Most of these studies used a 2D depth-averaged model with either an Eulerian ([Defontaine et al. 2020](#)) or a Lagrangian ([Liubartseva et al. 2016](#)) approach. A more theoretical study on how plastic particles with different densities and sizes move in a field of turbulence was provided by [Shamskhany & Karimpour \(2022\)](#). They used a Lagrangian approach with Large Eddy Simulation (LES) to resolve the turbulence and its effect on the particles in detail. [Shamskhany & Karimpour \(2022\)](#) used an *OpenFOAM* solver.

None of these studies tested the numerical models against measured plastic concentrations. This is done in the current study. Such comparisons are important to verify if the numerical model is correct and reliable.

Another hydraulic engineering problem involving particle movement is how silt and sand particles affect irrigation and hydropower projects. It is estimated that around 1% of the total global reservoir water volume is lost annually due to sedimentation ([Mahmood 1987](#)). Hydropower projects in sediment-carrying rivers need to be designed to minimise the

damages caused by the particles. A sand trap or a desilting basin is then often used to reduce the sediment concentration in the water before it causes wear on turbines or deposits in the channel system (Ivarsson *et al.* 2021). A typical design question is how large the sand traps need to be, to achieve sufficient particle removal. Not only the length and widths are important, but also the shape of expansion regions and the turbulence at the inflow section (Paschmann *et al.* 2017). Physical model studies have often been used to guide the design of the desilting basins. However, the scaling of the sediment particles is often challenging (Iyer & James 2022). For example, cohesive forces may occur when scaling down fine sand particles, causing practical problems in the laboratory model. A solution to this problem is to use a 3D numerical model that computes the water flow, turbulence characteristics and dispersion of the particles, as described in the current study.

IMPLEMENTATION OF THE *sediDriftFoam* SOLVER

The numerical model is based on an Eulerian approach, where the particles are computed as concentrations in a three-dimensional grid. The Reynolds-averaged Navier–Stokes equations are first solved on the same grid, together with a turbulence model. The standard k - ε model (Launder & Spalding 1974) was used in the current study, where k is the turbulent kinetic energy and ε is the dissipation rate of k . *OpenFOAM* provides a library where these equations can be solved by calling a function with four lines of code:

```
#include "uEqn.H"
#include "pEqn.H"
laminarTransport.correct();
turbulence->correct();
```

The lines include standard functions taken from the *simpleFoam* solver. The two first lines compute the water velocity and pressure, while the last function computes the turbulence.

The concentrations of the particles, c , were computed from the following convection-diffusion equation:

$$\frac{\partial c}{\partial t} + U_j \frac{\partial c}{\partial x_j} + w \frac{\partial c}{\partial z} = \frac{\partial}{\partial x_j} \left(\Gamma \frac{\partial c}{\partial x_j} \right) + F_R \quad (1)$$

The velocities, U_j ($j=1, 2$ and 3 to represent longitudinal velocity along x , lateral velocity along y and vertical velocity along z), and the turbulent diffusion Γ were obtained from the previously called Navier–Stokes solver from *simpleFoam*. The third term on the left side of Equation (1) is the drift-flux term. The fall or rise velocity of the particles is called w , and z is the coordinate in the vertical direction. F_R is a sediment pick-up rate given by van Rijn (1984). In the current numerical implementation, the velocity field from the Reynolds-averaged Navier–Stokes equations is altered by adding the fall or rise velocity. The resulting convection-diffusion equation is solved by a standard *OpenFOAM* solver.

The flux field from the fall or rise velocity is computed by interpolation from the cell centres to the cell surfaces. The flux field from the Reynolds-averaged Navier–Stokes equations is called ϕ . The flux field from the fall/rise velocity is called ϕ_c . This is computed with the following code:

```
forAll(Umod, i) {
    Umod[i].x() = 0.0;
    Umod[i].y() = 0.0;
    Umod[i].z() = fallvelocity;
}
phi_c = linearInterpolate(Umod) & mesh.Sf();
```

The variable `fallvelocity` is w in Equation (1). `Umod` is a vector field with velocities at the centre of the cells. The function *linearInterpolate* interpolates the velocities from the cell centres to the cell surfaces.

A new flux function with the superposition of the two flux fields is coded as:

```
phi_d = phi + phi_c
```

The *OpenFOAM/C++* libraries automatically wrap a loop around this line, so no explicit loop statement is needed.

The solution of Equation (1) is done with the following code:

```
solve
(
  fvm::ddt(Conc)
  + fvm::div(phi_d, Conc)
  + fvm::Sp(S_implicit, Conc)
  =
  S_explicit
  + fvm::laplacian(turbulence->nut(), Conc);
);
```

The *OpenFOAM/C++* syntax in the code follows Equation (1). The first term including `fvm::ddt` is the transient term in Equation (1), with the time derivative. The second term with `fvm::div(phi_d, Conc)` is the convective term, which now is the sum of the two other terms on the left side of Equation (1). The last line with the Laplacian operator is the diffusive term, which is the first term on the right side of Equation (1).

There are two additional terms in the code above: an implicit and an explicit term. These two terms incorporate two different options of taking boundary conditions at the bed and the free surface into account. If the particle densities are larger than water and the bed shear stress is higher than the critical Shields value, then some particle resuspension will occur at the bed. This is given at the last term in Equation (1), F_R . The boundary condition at the bed follows the sediment equilibrium theory by Einstein (1950), implemented in the bed concentration formula by van Rijn (1984). This formula was coded in the solver and the resulting flux included in the `S_explicit` variable. The formula gives a maximum pick-up rate as a function of the bed shear stress. If the concentration exceeds this limit, deposition will occur. Lower concentrations will cause all the particles to resuspend, preventing deposition on the sand trap bed. This boundary condition is suitable for a situation with sediment transport over a fixed, non-erodible bed, as opposed to an alluvial channel where erosion can occur.

```
scalar bedshear = bedvelocity * 0.4 / Foam::log(30.0 * dist_to_wall / roughness);
bedshear = 1000.0 * bedshear * bedshear;
scalar critshear = 0.047 * 1650.0 * 9.8 * particlesize;
scalar Tstar = (bedshear - critshear) / critshear;
if(Tstar < 0.0) Tstar = 0.0;
scalar Dstar = particlesize * 25296.0;
scalar vanRijn = 0.015 * particlesize / dist_to_wall * Foam::pow(Tstar, 1.5) / Foam::pow(Dstar, 0.3);
if(bedshear > critshear) { // settling particles
  if(vanRijn > Conc[cellbed]) {
    S_explicit[cellbed] = -fallvelocity /
      (2.0 * dist_to_wall) * Conc[cellbed];
  } else {
    S_explicit[cellbed] = -fallvelocity /
      (2.0 * dist_to_wall) * vanRijn;
  }
} // end else if settling particles
```

Some modifications are required at the water surface when modelling plastic particles with a positive rise velocity. The drift-flux model needs to take into account that the particles will not move above the water surface. This was done by including an implicit source term in the convection-diffusion equation (Equation (1)) for the surface cells:

```
if(fallvelocity > 0) // plastic
  S_implicit[cellbed] = fallvelocity / (2.0 * dist_to_wall);
```

The standard k - ε turbulence model overpredicts the turbulence close to the water surface, unless a damping function is introduced (Celik & Rodi 1988). The following modification of epsilon in the surface cells was therefore included:

$$\varepsilon = \frac{k^{5/2}}{fD} \quad (2)$$

D is the water depth, k is the turbulent kinetic energy at the free surface and f is an empirical parameter in the damping function. Celik & Rodi (1988) recommended a value of $f=0.43$, which was used in the current study. This epsilon boundary condition was implemented using the `codedFixedValue` function in *OpenFOAM*, following Kadia *et al.* (2022). The code was given in the `epsilon` file used as boundary condition for *sediDriftFoam*.

The remaining details of the solver are given in the full source code of the programme, provided in the Supplementary Material. This also includes boundary conditions for the following cases.

MODELLING SEDIMENT FLOW THROUGH A SAND TRAP

The *sediDriftFoam* solver was first tested on a case modelling the settling of sediment particles in a sand trap. Measurements from a laboratory study (Olsen & Skoglund 1994) were used to assess the results from the simulations. A 2 m deep and 1 m wide flume was used in the experiment, where an entrance section was built in wood. Figure 1 shows the modelled geometry. The length of the domain was 23.5 m and the water depth in the downstream section was 1.54 m. The water discharge through the sand trap was $0.225 \text{ m}^3/\text{s}$ and the inflow sediment concentration was 420 ppm by volume. The sediments were mixed in a drum located above the upstream section. Nine plastic tubes lead the sediment/water mixture from the tank to the inlet section.

Three grid arrangements were used to model the case. The coarse grid had 0.1 m long and 0.023–0.07 m wide cells, with 20 cells uniformly distributed in the vertical direction. The total number of cells was 65,800. The finest grid had 0.075 m long cells that were 0.016–0.05 m wide. Thirty cells were uniformly distributed in the vertical direction, giving the total number of cells as 159,300. A third grid was also made, with 0.1 m long and 0.032–0.1 m wide cells. In this grid, the number of cells in the vertical direction was varied between 23 and 32, depending on the water depth. The total number of cells in this grid was 68,540. The lines in the third grid were skewed towards the bed, following the paths of the particles, to reduce false diffusion. This grid is seen in Figure 1.

The sediment particles used in the laboratory model study were divided into four groups, according to Table 1. The numerical model was applied to each group and the concentrations were merged in the final result. The fall velocities in the table were taken from Rouse (1937).

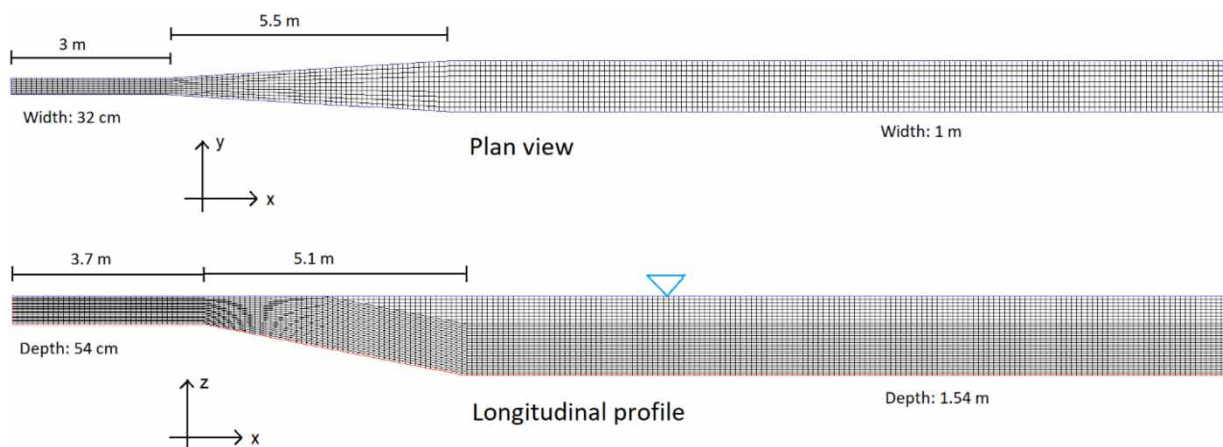


Figure 1 | Plan view and longitudinal profile of the sand trap showing the skewed grid. The flow direction is from left to right. The total length of the geometry is 23.5 m.

Table 1 | Sediment characteristics

Size number	Diameter (mm)	Fall velocity (mm/s)
1	0.45	70
2	0.3	40
3	0.2	25
4	0.1	8

Smooth wall laws were used on the side walls, and rough wall laws with a roughness of 1 mm were used on the bed. The distances from the bed cell centres to the bed were 1–3 cm, giving y^+ values between 200 and 700 for the bed. Detailed boundary conditions for this case are given in the Supplementary Material.

The computations with the *sediDriftFoam* solver used the first-order upwind scheme for the convective terms in all equations. The solution converged to a steady state with sufficiently low residuals ($\leq 10^{-4}$) after 900–6,600 time steps. The computational time of one particle size for the skewed grid was 30 min on one core of a PC with an i7 processor.

Figure 2 shows the computed sediment concentration distribution in a longitudinal profile along the centreline of the geometry. The uniform inflow of sediment concentration over the depth changed in the flow direction as the particles fell to the entrance bed and got picked up by the high-velocity flow. Overall, the maximum concentrations were found close to the bed due to settlement of the particles. This also caused lower sediment concentration towards the water surface. Sediment deposition occurred in the expansion region, where the flow velocity decreased. The resulting sediment concentration was reduced in the downstream section of the sand trap. Only a smaller fraction of the particles was transported through the geometry, indicating a high trap efficiency.

The concentration of sand particles was measured by Olsen & Skoglund (1994) in three vertical profiles of the flume. Brass pipes were mounted in the physical model, extracting water at four points in three vertical profiles. The water samples were then dried, and the weight of the sediments together with the sampled water volume gave the concentrations. The measured values are given in Figure 3, together with the computed particle concentrations in the same profiles. The results from the SSIIM model (Olsen & Skoglund 1994) are also provided to compare the results.

Figure 4 shows the water velocities in a longitudinal section along the centreline of the sand trap. The higher velocities are in the upstream entrance channel where the cross-sectional area is smaller than in the main sand trap. A recirculation zone formed in the expansion area. The length of this zone was overpredicted using the $k-\epsilon$ turbulence model with the default coefficients (Olsen & Skoglund 1994). However, the trap efficiency of the sand trap was not significantly affected.

The magnitude of the eddy-viscosity affects the vertical profile of the suspended sediments and thereby also the trap efficiency of the sand trap. Figure 5 shows that the higher values are in the region with the recirculation zone, downstream of the channel expansion, due to the higher velocity gradients in this area. The lower values of the eddy-viscosity in the entrance section are due to the specified upstream boundary conditions for k and epsilon. This can be seen in Figure 6.

Figure 6 shows the turbulent kinetic energy distribution along the centreline of the sand trap. Note the higher values at the start of the expansion region, especially close to the bed. The turbulent kinetic energy at the boundary is linearly proportional

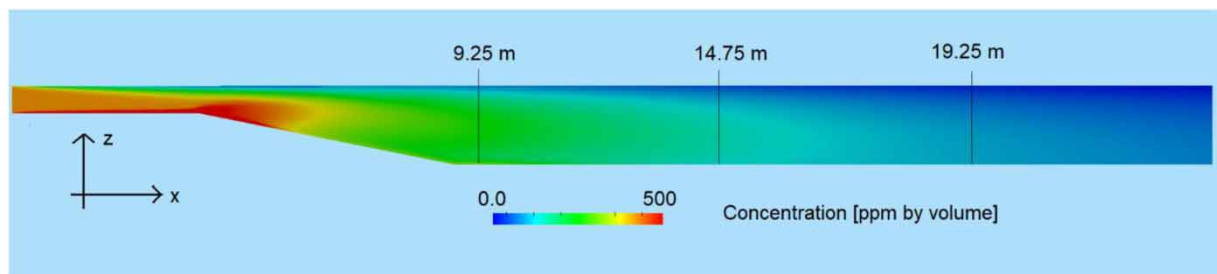


Figure 2 | Longitudinal profile along the centreline of the sand trap, showing computed sediment concentration. The vertical lines are the location of the measured values. The flow direction is from left to right.

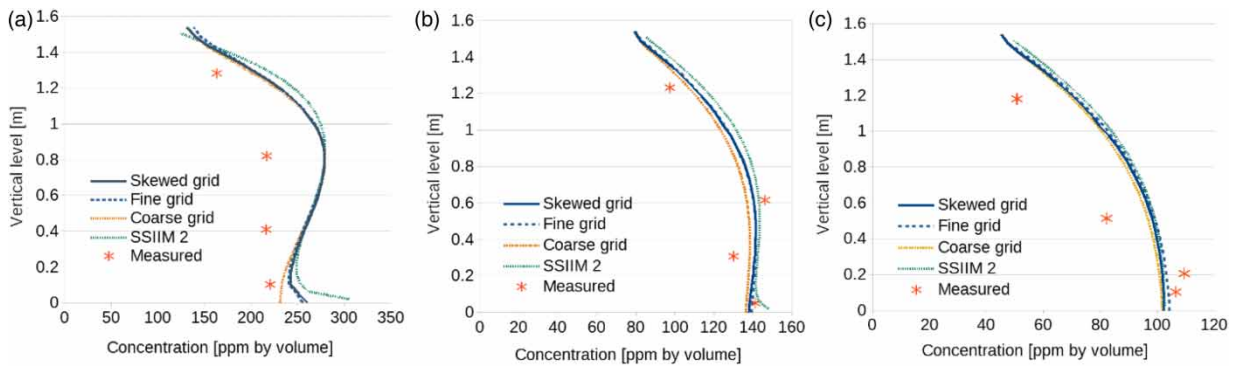


Figure 3 | Computed and measured concentration profiles along the centre of the sand trap located at different distances from the upstream end of the flume: (a) 9.25 m, (b) 14.75 m and (c) 19.25 m.

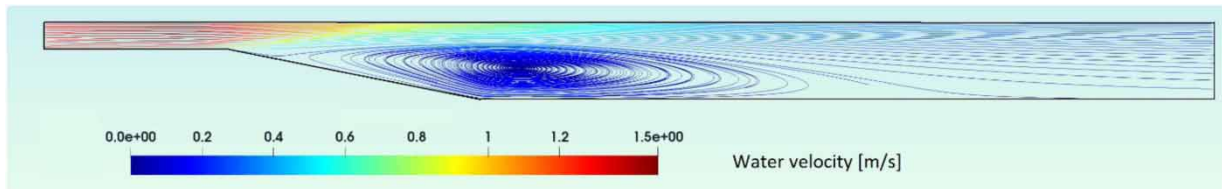


Figure 4 | Velocity distribution in a longitudinal section along the centreline of the sand trap showed by streamlines. The flow direction is from left to right.

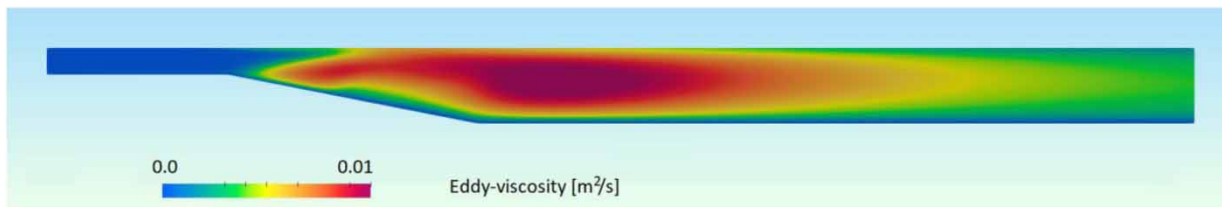


Figure 5 | Eddy-viscosity distribution in a longitudinal section along the centreline of the sand trap.

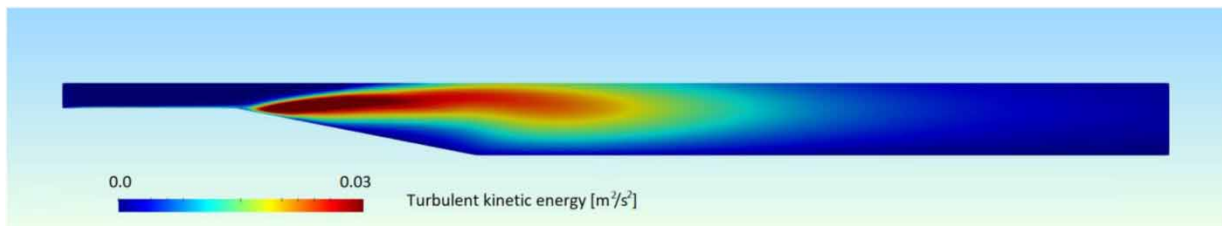


Figure 6 | Turbulent kinetic energy in a longitudinal section at the middle of the sand trap.

to the bed shear stress (Rodi 1980). The high shear stress at this location causes no sediment deposition here. This can also be seen in the photograph of the expansion region from the laboratory study on the web page <http://pvv.org/~nilsol/cases/doris>.

MODELLING VERTICAL CONCENTRATIONS OF PLASTIC PARTICLES IN A CHANNEL

The second test case for the *sediDriftFoam* solver was to compute the vertical distribution of plastic particles with a buoyancy smaller than water in a channel. The positive rise velocity caused an increase in the concentration towards the water surface.

The model was tested on a case from RWTH Aachen University, where concentration profiles were measured in a laboratory flume (Born *et al.* 2023). The flume had a length of 12.4 m and the experiments were conducted for a water depth of 0.067 m with a mean velocity of 1.4 m/s. The plastic particles had a diameter of 1.5 mm and a rise velocity of 40 mm/s. The upstream 8 m of the flume was modelled using two-dimensional (2D) width-averaged grids, containing 1,000–1,600 cells in the longitudinal direction and 40–50 cells in the vertical direction. The measured and computed concentration profiles at a location 7 m from the upstream end are compared in Figure 7. The computed plastic particle concentrations agree well with the measured data throughout the depth. Most of the plastic particles from the physical model study move in the upper 20% of the water column. Even though the measurements integrate discrete observations of about 2,000 particles over a period of about 30 s, the resulting values fluctuate over the depth. The numerical model provides a smoother profile by solving a partial differential equation in the grid.

Figure 8 shows a longitudinal profile of the computed plastic concentrations. The particles are uniformly distributed over the depth at the inflow boundary. As the particles move downstream, the positive rise velocity then causes an increased concentration close to the water surface. The figure indicates that the concentration gradient in the horizontal direction is negligible in the middle and downstream part of the simulated channel. This indicates that the flume and the computational geometry is sufficiently long to develop the flow and to produce a uniform particle concentration distribution over the depth. This is also indicated in Figure 9, showing the turbulent diffusion/eddy-viscosity in the same profile. The values at the downstream end of the geometry are uniform along the streamwise direction.

DISCUSSION

The science of Hydroinformatics aims at improving computational tools for water engineering and research. Engineers today often use commercial CFD programmes to solve practical problems in hydraulics. The programmes have intuitive and friendly user interfaces that reduce the time spent on projects. Open source alternatives, like *OpenFOAM*, requires more training for the users. But an advantage of the open source alternative is the possibility to learn and understand all the details on how the solution algorithms work. Commercial CFD programmes do normally not give the users access to the source code. Does a programme actually solve the exact same equations as given in the Users Manual? Are there undocumented stabilisation algorithms that introduce systematic errors? In an open source programme, the user is able to see all algorithms that are coded.

Another main advantage with open source software is the ability to change the code. The current article describes a new solver that has been made from the libraries of a high-level CFD language. The relatively few lines of code make the solver easy to learn for new users. Also, it should not be too complicated to modify the solver further. *OpenFOAM* contains several

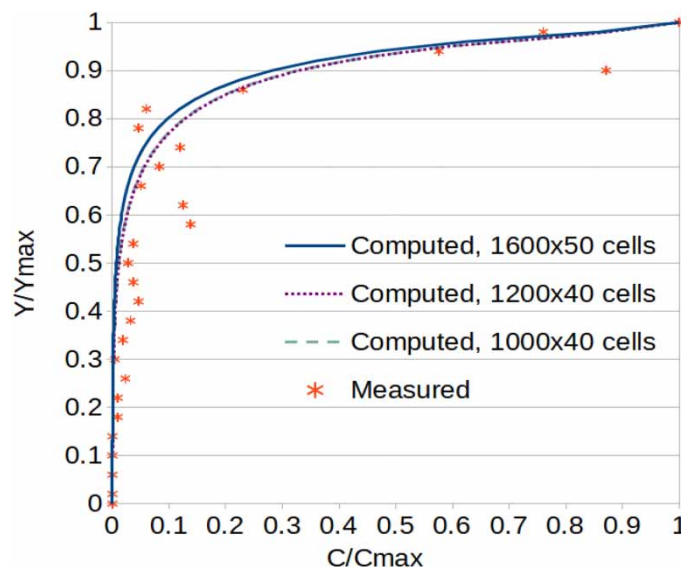


Figure 7 | Measured and computed profiles of plastic particle concentrations with different grid sizes.

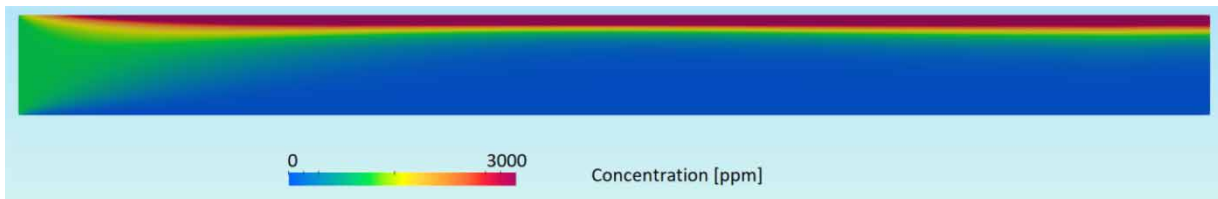


Figure 8 | Longitudinal profile of plastic particle concentrations. The flow direction is from left to right. The figure is enlarged 10× in the vertical direction. The length of the section is 8 m and the depth is 0.067 m.

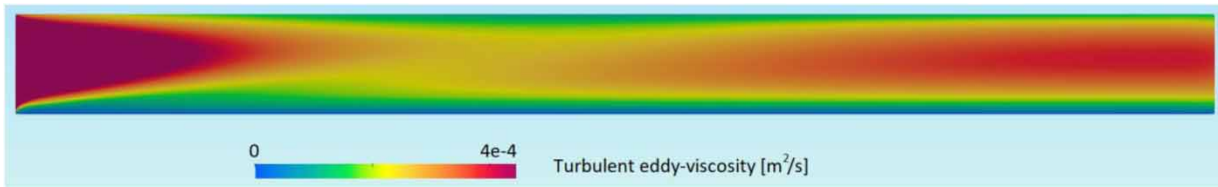


Figure 9 | Longitudinal profile of turbulent eddy-viscosity. The flow direction is from left to right. The figure is enlarged 10× in the vertical direction. The length of the section is 8 m and the depth is 0.067 m.

libraries for changing the grid, so it should be possible to extend *sediDriftFoam* with algorithms that compute bed level changes over time as a function of sediment erosion and deposition. It would also be possible to compute particles that have different characteristics than sediments or plastics. An example is gas bubbles formed by air entrainment in hydropower plants, causing mortality for fish.

The two examples in the current article show that it is possible to use the new solver for computing plastic particle concentrations in a flume and sand deposition in a sand trap. Tests with different grid sizes show that there is some variation in the results. However, [Figures 3 and 7](#) show that the variations in the measurements are much larger. The largest deviation between the computed and measured values is seen in [Figure 3](#), section 9.25 m, about 0.8 m above the bed. Improved results for this section were found by [Olsen & Skoglund \(1994\)](#) when modifying a coefficient in the $k-\epsilon$ turbulence model. In the current study, the RNG (Re-Normalized Group) and Realisable $k-\epsilon$ turbulence models were tested, but this did not give a converged solution. A converged solution for the sand trap was also only possible with first-order upwind discretization schemes. A second-order discretization scheme is often recommended to reduce false diffusion in CFD computations. However, false diffusion is also reduced by grid refinement. The relatively small variations in the results when using different grids is an indication that false diffusion did not have a large effect on the results in the current case.

Convergence problems for the sand trap may be due to the physics of the flow field. The upstream boundary has a small cross-sectional area with relatively high velocity. A jet is formed, and it is difficult to predict its location in the sand trap. [Almeland et al. \(2019\)](#) found very different flow fields in a sand trap, depending on grids, discretization schemes and turbulence models. The flow field in the sand trap may not be steady, and then it is very difficult to get a converged solution. [Almeland et al. \(2019\)](#) even found that the second-order scheme gave the incorrect flow field for one particular sand trap case, whereas the first-order scheme gave the better result.

CONCLUSIONS

A new open-source CFD solver for calculation of suspended particles in water currents is presented. The programme is made in the *OpenFOAM* framework. It has been tested for deposition of sand in a three-dimensional sand trap and for the vertical distribution of plastic particles with a positive rise velocity in a flume. The computed results show good agreement with vertical concentration profiles obtained previously from the two laboratory studies. In the sand trap case, the particles deposit in the expansion zone and the concentration decreases towards the water surface and in the downstream direction. The rise velocity of the plastic particles creates an opposite trend with the higher concentrations located near the free surface. The developed code is relatively simple and can therefore be applied and modified by a wide range of users, including students and researchers working on sediment transport cases.

The main limitation of the model is its one-way coupling between the particles and the water. The water will affect the particles, but the particles will not affect the water flow field. This assumption is only valid for low particle concentrations, including the examples shown in the current article.

DATA AVAILABILITY STATEMENT

All relevant data are available from an online repository or repositories: <http://pvv.org/~nilsol/sediDriftFoam>.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

- Almeland, S. K., Olsen, N. R. B., Bråtveit, K. & Aryal, P. R. 2019 Multiple solutions of the Navier–Stokes equations computing water flow in sand traps. *Engineering Applications of Computational Fluid Mechanics* **13** (1), 199–219. doi:10.1080/19942060.2019.1566094.
- Baranya, S., Olsen, N. R. B., Stoesser, T. & Sturm, T. 2014 A nested grid based computational fluid dynamics model to predict bridge pier scour. *Water Management* **167** (5), 259–268. doi:10.1680/wama.12.00104.
- Basani, R., Janocko, M., Cartigny, M. J. B., Hansen, E. W. M. & Eggenhuisen, J. T. 2014 MassFLOW-3D as a simulation tool for turbidity currents: some preliminary results. *International Association of Sedimentologists Special Publication* **46**, 587–608.
- Born, M., Brill, C., Schaefer, D., Hillebrand, G. & Schüttrumpf, H. 2023 Determination of microplastics' vertical concentration transport (Rouse) profiles in flumes. *Environmental Science and Technology* doi:10.1021/acs.est.2c06885.
- Celik, I. & Rodi, W. 1988 Modeling suspended sediment transport in nonequilibrium situations. *ASCE Journal of Hydraulic Engineering* **114**, 1157–1191.
- Chauchat, J., Cheng, Z., Nagel, T., Bonamy, C. & Hsu, T.-J. 2017 SedFoam-2.0: a 3-D two-phase flow numerical model for sediment transport. *Geoscientific Model Development* **10** (12), 4367–4392.
- Defontaine, S., Sous, D., Tesan, J., Monperus, M., Lenoble, V. & Lanceleur, L. 2020 Microplastics in a salt-wedge estuary: vertical structure and tidal dynamics. *Marine Pollution Bulletin* **160**, 111688. doi:10.1016/j.marpolbul.2020.111688.
- De Marchis, M. & Milici, B. 2016 Turbulence modulation by micro-particles in smooth and rough channels. *Physics of Fluids* **28** (11), 115101. <http://dx.doi.org/10.1063/1.4966647>.
- Einstein, H. A. 1950 *The Bed-Load Function for Sediment Transportation in Open Channel Flows*. US Dept. of Agriculture Tech Bull 1026, Washington, DC.
- Goll, A., Kopmann, R., 2012 Dune simulation with TELEMAC-3D and SISYPHE: A parameter study. In: *Proceedings of the XIXth TELEMAC-MASCARET User Conference* (Bourban, S., Durand, N. & Hervouet, J.-M. eds.). TELEMAC-MASCARET, pp. 19–25.
- Harb, G., Haun, S., Schneider, J. & Olsen, N. R. B. 2014 Numerical analysis of synthetic granulate deposition in a physical model study. *International Journal of Sediment Research* **29** (1), 110–117.
- Ivarsson, M. M., Trivedi, C. & Vereide, K. 2021 Investigations of rake and rib structures in sand traps to prevent sediment transport in hydropower plants. *Energies* **14**, 3882. doi:10.3390/en14133882.
- Iyer, J. C. & James, E. J. 2022 Model studies for the design of inlet transition of settling basins of hydropower projects in high sediment yield areas: a review. *ISH Journal of Hydraulic Engineering* doi:10.1080/09715010.2022.2084350.
- Kadia, S., Rüther, N., Albayrak, I. & Pummer, E. 2022 Reynolds stress modeling of supercritical narrow channel flows using OpenFOAM: secondary currents and turbulent flow characteristics. *Physics of Fluids* **34** (12), 125116.
- Lauder, B. E. & Spalding, D. B. 1974 The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering* **3** (2), 269–289.
- Liubartseva, S., Coppini, G., Lecci, R. & Creti, S. 2016 Regional approach to modeling the transport of floating plastic debris in the Adriatic Sea. *Marine Pollution Bulletin* **103** (1–2), 115–127. doi:10.1016/j.marpolbul.2015.12.031.
- Maduka, M. & Li, C. W. 2021 Numerical study of ducted turbines in bi-directional tidal flows. *Engineering Applications of Computational Fluid Mechanics* **15** (1), 194–209.
- Mahmood, K. 1987 *Reservoir Sedimentation: Impact, Extent and Mitigation*. World Bank Technical Paper 71.
- McConnell, S. 2004 *Code Complete*. Pearson Education, London. ISBN:978-0-7356-1967-8.
- Mool, P., Popescu, I., Giri, S., Omer, A., Sloff, K., Kitamura, Y. & Solomatine, D. 2017 Delft3D morphological modelling of sediment management in daily peaking run-of-the-river hydropower (PROR) reservoirs in Nepal. In: *85th Annual Meeting of International Commission on Large Dams*, Prague, Czech Republic.
- Olsen, N. R. B. 2021 3D numerical modelling of braided channel formation. *Geomorphology* **375**, 107528. doi:10.1016/j.geomorph.2020.107528.
- Olsen, N. R. B. 2022 Explaining the formation of sedimentary structures under antidunes using a 2D width-averaged numerical model. *Norwegian Journal of Geology* **102**. doi:10.17850/njg102-3-2.
- Olsen, N. R. B. & Skoglund, M. 1994 Three-dimensional numerical modeling of water and sediment flow in a sand trap. *Journal of Hydraulic Research* **32** (6), 833–844.

- Paschmann, C., Fernandez, J. N., Vetsch, D. F. & Boes, M. 2017 Assessment of flow field and sediment flux at alpine desanding facilities. *International Journal of River Basin Management* doi:10.1080/15715124.2017.1280814.
- Paschmann, C., Vetsch, D. F. & Boes, M. 2022 Design of desanding facilities for hydropower schemes based on trapping efficiency. *Water* **14**, 520. doi:10.3390/w14040520.
- Rodi, W. 1980 *Turbulence Models and Their Application in Hydraulics*. IAHR, Beijing.
- Rouse, H. 1937 *Nomogram for the Settling Velocity of Spheres*. Division of Geology and Geography, Exhibit D of the Report of the Commission of Sedimentation, National Research Council, Washington, DC, pp. 57–64.
- Sattar, A. M. A., Jasak, H. & Skuric, V. 2017 Three dimensional modeling of free surface flow and sediment transport with bed deformation using automatic mesh motion. *Environmental Modelling & Software* **97**, 303–317. doi:10.1016/j.envsoft.2017.08.005.
- Schwarzmeier, C., Rettinger, C., Kemmler, S., Plweinsik, J., Nuñez-Gonzalez, F., Köstler, H. & Vowinkel, B. 2023 Particle-resolved simulation of antidunes in free-surface flow. *Journal of Fluid Mechanics* **961**. doi:10.1017/jfm.2023.262.
- Shamskhany, A. & Karimpour, S. 2022 Entrainment and vertical mixing of aquatic microplastics in turbulent flow: the coupled role of particle size and density. *Marine Pollution Bulletin* **184**, 114160. doi:10.1016/j.marpolbul.2022.114160.
- Sun, R. & Xiao, H. 2016 SediFoam: a general-purpose, open-source CFD–DEM solver for particle-laden flow with emphasis on sediment transport. *Computers & Geosciences* **89**, 207–219. doi:10.1016/j.cageo.2016.01.011.
- van Rijn, L. C. 1984 Sediment transport, part II, suspended sediment transport. *Journal of Hydraulic Engineering* **110** (11), 1613–1641.

First received 27 April 2023; accepted in revised form 31 July 2023. Available online 9 August 2023