17th Australasian Fluid Mechanics Conference
Auckland, New Zealand
5-9 December 2010

Hybrid volume-of-fluid and discrete particle solver for oil-mist filter simulations

A. J. C. King[1], B. J. Mullins[1] and G. Ladenburger[1,2]

[1]Fluid Dynamics Research Group
Curtin University of Technology,
Bentley, WA 6102, Australia

[2]Institut für Mechanische, Verfahrenstechnik und Mechanik,
Karlsruher Institut für Technologie (KIT)
D-76131 Karlsruhe, Germany

Abstract

This paper presents a combined volume-of-fluid and discrete Lagrangian particle tracking method which can be used to simulate oil-mist or coalescing filters, and gauge their effectiveness. The solver is built on the open-source OpenFOAM CFD libraries, which are extended to include a physically accurate model for motion of small particles, coupled with a volume-of-fluid treatment as particles coalesce, or are collected on the fibres. A key part of the method is accounting for particles as they change between a discrete Lagrangian treatment to a volume-of-fluid treatment. Results from an investigation into an idealised filter geometry are used to demonstrate that the method is feasible, robust and that realistic behaviour of an oil-mist filter can be captured. Successful implementation of the method should lead to a better understanding of the behaviour of oil mist filters, and will allow design options to be explored through simulation, in addition to experimentation.

Introduction

Many industrial processes create oil-mist laden exhaust air during operation, which is often detrimental to the environment or can cause safety and health issues. Typically, to reduce these effects the oil aerosol present in the exhausted air is removed by mechanical methods, including filtration. The fundamentals of particulate filtration are well-understood and can be used for some filter designs, namely those where an idealised geometry can be determined, and where the particles are solid. Oil-mist filters behave in a considerably different manner, and the use of these fundamental principles is generally unsuccessful. This means that currently, the performance of oil-mist filters must be determined experimentally, and the design process for oil-mist filters is predominantly by 'trial and error'. To gain insight into oil-mist filter behaviour, and to allow accurate prediction of their performance it is desirable that simulation can be used. This will allow the filter design process to be undertaken in a more systematic manner, and reduce development costs.

Unfortunately, oil-mists or oil aerosols are complex to model with conventional Computational Fluid Dynamics (CFD) techniques. To start with, initial droplet sizes are small (in the order of 1 μm typically) and behave like solid particles. However as droplets agglomerate and interact with the fibres in the filter, exclusive treatment as solid particles becomes inaccurate. The behaviour in this region can be better represented using a volume-of-fluid model, which accounts for the fluid mechanics of the two phases (oil and air) in a coupled manner.

The work presented in this paper seeks to combine these modelling techniques in one simulation, to gain the advantages of both. Additionally, the transition between each treatment should be both accurate and robust. The OpenFOAM CFD software is used as the basis for this work[6]. As it is open-source, the source code is readily available and extensions and modifications to models can be easily implemented at a low level. At this point in time the work is ongoing, however a substantially useful solver has been created. One-way transition from a Lagrangian representation (solid particle) to a volume-of-fluid representation of the oil has been accomplished, while coalescence and break-up are yet to be implemented. At this stage the model should be able to give accurate characterisation of long-term oil-mist filter behaviour. A sample case representing an idealised filter geometry is used to demonstrate the effectiveness of the technique.

Governing Equations

Fluids

The current work builds on the volume-of-fluid model [3], a well established multiphase CFD technique for flows where two immiscible fluids are present, and a distinct surface interface can be defined. In this model an additional variable, α_l is used to represent the volume fraction of a liquid throughout the domain, while the volume fraction of the alternative phase can be determined from $\alpha_a = 1 - \alpha_l$. Representative fluid properties (such as density, ρ, and viscosity, μ) are reconstructed within the domain, based on the individual fluid properties and the local volume fraction of each fluid.

In the volume-of-fluid model implemented in OpenFOAM a single momentum equation is solved for the system, as follows

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla \mathbf{P} + \mu \nabla^2 \mathbf{U} + \mathbf{F} \quad (1)$$

in addition to a continuity equation,

$$\nabla \cdot \mathbf{U} = 0 \quad (2)$$

and a transport equation for the volume fraction of one phase, which includes interface compression

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{U}\alpha_l) + \nabla \cdot [\mathbf{U}_r \alpha_l (1 - \alpha_l)] = 0 \quad (3)$$

where $\mathbf{U}_r$ is designated as the *compression velocity*, though unrelated to compressible flows[1].

The solver is developed as a three-dimensional unsteady solver. The *Pressure Implicit Splitting of Operators* (PISO) algorithm [4] is used to solve the coupled pressure-velocity fields, while the *Multidimensional Universal Limiter with Explicit Solution* (MULES) algorithm is used for the volume fraction transport equation.

Discrete Particles

The motion of each influent oil particle is individually tracked through the air phase until either the particle is collected in the oil phase, or until the particle passes through a domain boundary, which includes being captured by a fibre. Tracking is conducted in a Lagrangian frame of reference, with explicit integration of the particle velocity and position conducted at least once per timestep.

The forces acting on each particle are due to buoyancy, drag, and thermal diffusion (Brownian Motion). Buoyant forces are determined from

$$\mathbf{F}_b = \frac{\pi d_p^3}{6} \left(\rho_{\text{oil}} - \rho_{\text{fluid}}\right) \mathbf{g} \tag{4}$$

where d_p is the particle diameter, ρ_{oil} the density of the oil droplets, and ρ_{fluid} is the density of the combined phase, which for this work will always be the density of the air phase.

Drag forces are calculated from

$$\mathbf{F}_d = \frac{18\mu}{\rho_{\text{oil}} d_p^2} \frac{C_D}{24} \text{Re} \tag{5}$$

where $C_D = f(\text{Re})$ is the particle drag coefficient, and Re is the particle Reynolds number based on the relative velocity between the particle and the fluid. The drag coefficient was determined based on the model of Haider and Levenspiel [2].

Thermal diffusion is neglible unless the particle diameter is less than approximately 1 μm. For these particles forces due to thermal diffusion were calculated from

$$\mathbf{F}_t = \mathbf{G}\sqrt{\frac{\pi S_0}{\Delta t}} \tag{6}$$

where $\mathbf{G}$ is a vector whose components are independent zero-mean, unit-variance Gaussian random numbers.

S_0 is given by

$$S_0 = 216 \frac{\nu \rho k_b T_{\text{ref}}}{\pi^2 d_p^5 \rho_p^2 C_c} \tag{7}$$

where C_c is the Cunningham Correction factor, λ is the particle mean free path, given by

$$\lambda = 0.385 \sqrt{\frac{\rho_{\text{oil}} d_p k_b T_{\text{ref}}}{\mu^2 \pi^2}} \tag{8}$$

In the above, k_b is the Boltzmann constant and the reference temperature, T_{ref} was taken as 293.15 K.

Particle Tracking Algorithm

A reasonably robust particle tracking algorithm is implemented in OpenFOAM and a detailed description of the algorithm is presented in [5], however a brief overview is presented here.

For each particle, the following steps are undertaken.

1. Fluid and flow properties at the current cell location are determined by interpolation.
2. The particle forces are evaluated.
3. The particle trajectory is calculated, based on the fraction of time remaining for the current time step.
4. Intersection of the trajectory with any cell face is determined.
   (a) If the particle's trajectory intersects a boundary face, a flag is set, and no further calculation is made for this particle in the simulation.
   (b) If the particle's trajectory intersects an internal face, the particle is transferred to the neighbouring cell, and the fraction of the trajectory completed is deducted from remaining time, and the process is continued from step 1.
   (c) If the particle trajectory does not intersect a face, update the particle's position and move onto next particle

There are a number of additional parts to this algorithm that deal with parallelisation issues, particle rebound, and poor quality meshes, however the reader is referred to [5] for these details.

Particle Collection

Particle collection takes place by two mechanisms — particles may be collected by the fibres in the filter (or other boundaries); or, particles may be collected by the existing oil. In the first case, capture is covered by the preceding algorithm, namely, if the particle's trajectory is within $d_p/2$ of a boundary face, either at the end of its trajectory, or at the point at which it intersects a face, then the particle is considered to have impacted on the boundary face, and therefore captured by the filter. The particle is tagged with the identity of the boundary that it has impacted on, to allow post-processing. Cyclic boundaries are treated slightly differently in that the particle trajectory is continued from the linked face on the connected boundary.

In the second case, a particle is considered to be collected by the oil when the cell it occupies has a fluid volume fraction, α_l, above 0.5. Where this value was nominally chosen to identify a cell that the contains the oil-air interface. The particle is tagged as collected, and is no longer considered in the simulation. To maintain mass continuity, it is necessary to add a corresponding "mass" source to the α_l transport equation. The added volume fraction is determined from

$$\alpha_{l,\text{source}} = \frac{\pi d_p^3}{6} \frac{1}{V_{\text{cell}}} \tag{9}$$

which converts the particle volume into a cell volume fraction. Figure 1, shows a droplet and the fluid interface before and after a particle is collected by the oil.

In addition to a volume fraction source term, a source term is added to the momentum equation to account for the momentum of the particle. The added momentum is determined from

$$M_{\text{source}} = \rho_p U_{\text{p}} \tag{10}$$

where U_p is the particle velocity.

Demonstration Case

As a test of the functionality and robustness of the algorithm, a simple test case was created, as shown in Figure 2. This represents a mesh screen with a fibre diameter of 10μm, and a fibre centre-to-centre spacing of 50μm.

Boundary Conditions

A uniform velocity of 0.1 m/s was applied to the inlet of the domain, while a zero normal gradient on the velocity was used for the outlet boundary. For pressure, a constant static pressure

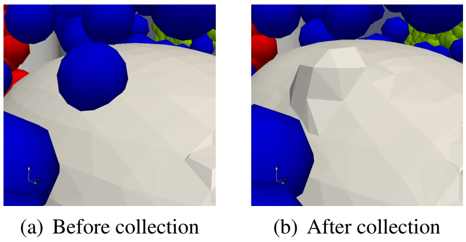


Figure 1: Collection of oil particle by oil

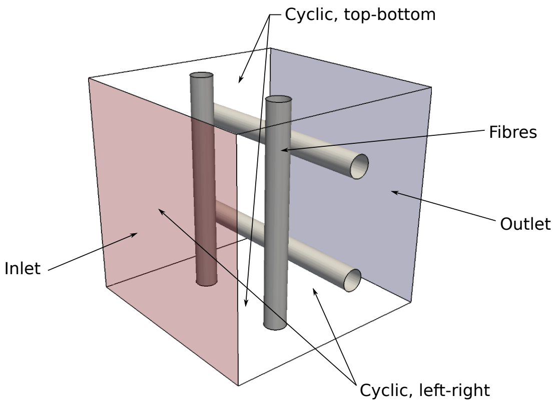


Figure 2: Test case for combined model

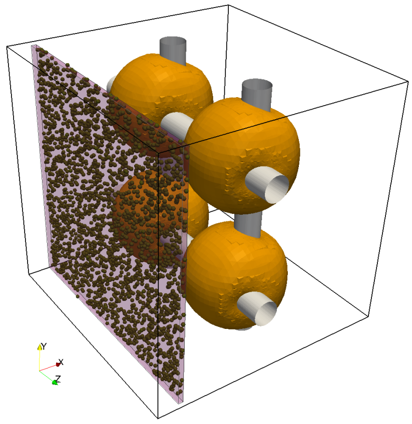


Figure 3: Initial conditions for simulation

Table 1: Particle status at simulation end

Particle Status	Particles	
	Number	%
Still in Motion	767	30.6
Collected by Fibre	244	9.7
Collected by oil	13	0.6
Exited through outlet	1480	59.1
Total	2504	100

was used for the outlet, while the inlet pressure was determined by the solver. For the remaining sides of the domain two linked cyclic boundaries were created, one linking the top and the bottom of the domain, and the other linking the left and right sides of the domain. The fibre boundaries were treated as walls, with an imposed no-slip condition for velocity and a zero normal gradient condition enforced for the pressure.

Initial Conditions

The domain was initialised with 4 "droplets" of oil at the 4 fibre intersection points, as shown in Figure 3. The initial velocity for the air phase was set to 0.1 m/s normal to the inlet boundary, while for the oil phase the initial velocity was set to 0 m/s. In the first time step of the simulation, 2500 oil particles (nominally) with a diameter of 1.5 μm were injected into the domain, with their velocity matched to the surrounding flow. Due to limitations in injecting particles in parallel operation the actual number of injected particles may vary slightly with the nominal number of injected particles. The injection region is shown in Figure 3.

Simulation Parameters

Flow through the mesh was simulated for 0.0007 s with a constant timestep size of $\Delta t = 1.0 \times 10^{-8}$ s, selected to maintain the maximum Courant number below 0.2.

Results

The simulation was completed in parallel using a 16 computational cores across 4 machines, for a total execution time of approximately 10 hours. The actual number of particles injected to the domain was 2504. Table 1 shows the particle status at the end of the simulation, while Figure 4 shows the surface interface between the air and oil phases and the particle locations between 5×10^{-4} s and 7×10^{-4} s.

Discussion

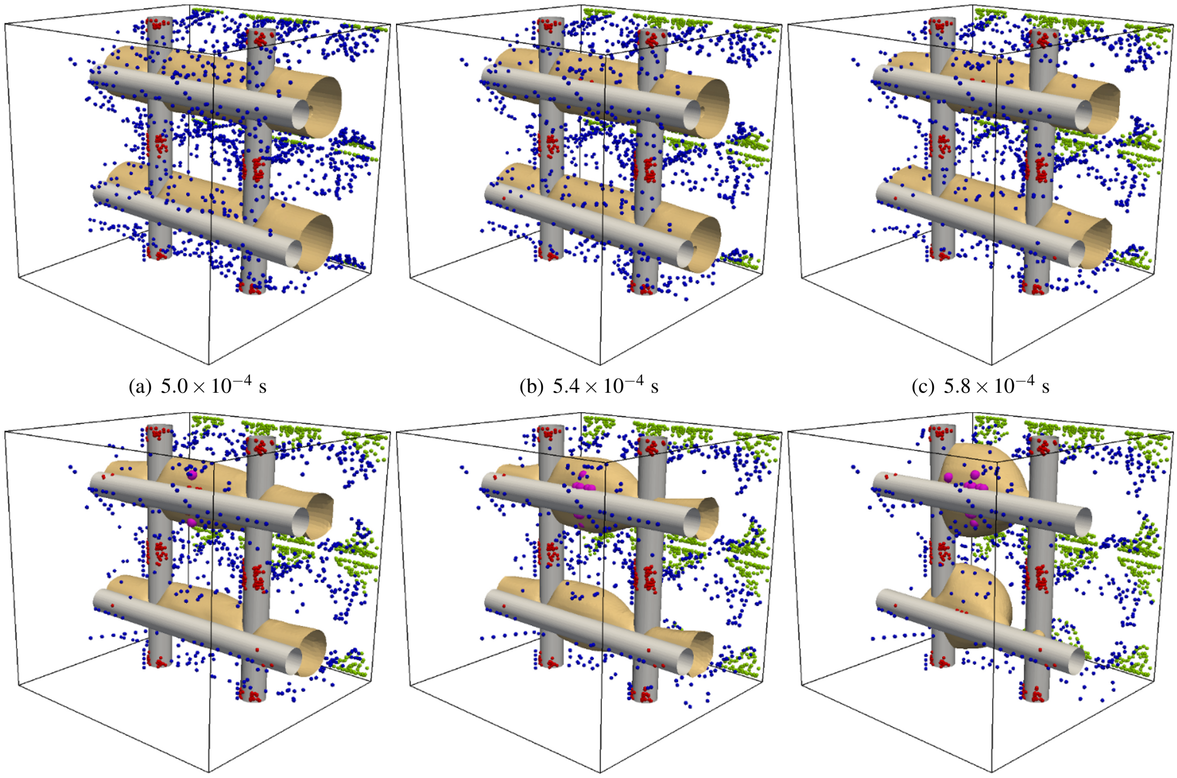
The results from the demonstration case indicate that a combined volume-of-fluid and Lagrangian treatment is realisable, though the model will require further improvements to account for particle collision and agglomeration. The implementation is reasonably robust, with a number of simulations completed successfully (though not presented in this paper).

One thing that becomes clear from the demonstration case is the complicated dynamics that occur within an oil-mist filter. Initially, the 4 oil droplets at the fibre intersections coalesce behind the horizontal fibres into long columns of fluid, after which these become unstable and droplets are again formed on the fibre intersections. This particular motion is predominantly determined by the initial conditions of the simulation, and for realistic simulations the sensitivity to the initial conditions will need to be determined, especially since the motion significantly affects the trajectories of the discrete oil particles and the resultant collection efficiencies observed for an oil-mist filter.

While the above are readily apparent from the present model, additional work is required to validate the generated results.

Conclusions

This paper presented a combined volume-of-fluid and discrete Lagrangian tracking algorithm, implemented as an extension to the OpenFOAM CFD software. The small influent oil particulate (in the order of 1 μm diameter) are initially treated as discrete particles, and are tracked within a Lagrangian frame of reference. These particles interact with the filter geometry and any previous collected oil, which is treated using a volume-of-fluid approach. Where the particles interact with the collected oil, a source term is included in the volume-fraction transport equation, and the particle is removed from the simulation. The solver has been used successfully on an idealised filter geometry



(a) 5.0×10^{-4} s (b) 5.4×10^{-4} s (c) 5.8×10^{-4} s

(d) 6.2×10^{-4} s (e) 6.6×10^{-4} s (f) 7.0×10^{-4} s

Figure 4: Simulation results. Particles coloured by status: blue – in motion; red – collected by fibre; magenta – collected by oil; green – exited through outlet.

to demonstrate that this technique is feasible and robust.

Continued development of the solver will allow simulation of realistic oil-mist filter geometries to be undertaken, and hence, prediction of performance earlier in the design phase. In addition, further insight into of the dynamics that occur within oil-mist filters may be gained due to the extra data that may be extracted from the numerical model. Both of these points should allow design of oil-mist filters to be undertaken with more certainty, and as a result, with reduced costs.

Future work involves adding models to include particle coalescence in the free-stream, formation of new droplets from oil mist collected on bare fibres and break-up of large droplets into particles. In addition experiments and further simulations will need to be conducted to validate the accuracy of the solver.

Acknowledgements

The authors would like to asknowledge the support of ARC Linkage Grant LP0883877, and the Ernest-Solvay-Stiftung.

References

[1] Edin Berberović, Nils P. van Hinsberg, Suad Jakirlić, Ilia V. Roisman, and Cameron Tropea. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Phys. Rev. E*, 79(3):036306–, 2009.

[2] A. Haider and O. Levenspiel. Drag coefficient and terminal velocity of spherical and nonspherical particles. *Powder Technology*, 58(1):63–70, May 1989. ISSN 0032-5910.

[3] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, January 1981. ISSN 0021-9991.

[4] R I Issa. Solution of implicitly discretized fluid flow equations by operator splitting. *Journal of Computational Physics*, 62:40 – 65, 1986.

[5] Graham B. Macpherson, Niklas Nordin, and Henry G. Weller. Particle tracking in unstructured, arbitrary polyhedral meshes for use in cfd and molecular dynamics. *Communications in Numerical Methods in Engineering*, 25(3): 263–273, 2009.

[6] OpenCFD Ltd. *OpenFOAM - User Guide*. OpenCFD Ltd., July 2009. Version 1.6, http://www.openfoam.com/.