ON NEW CHALLENGES FOR CFD SIMULATION IN FILTRATION

Michael Dedering, Wolfgang Stausberg,
IBS Filtran, Industriestrasse 19
D-51597 Morsbach-Lichtenberg, Germany.

Oleg Iliev(*), Zahra Lakdawala,
Fraunhofer Institut für Techno- und Wirtschaftsmathematik,
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany.

Raimondas Ciegis, Vadimas Starikovicius,
Vilnius Gediminas Technical University, Sauletekio av. 11
10223 Vilnius, Lithuania

1. Introduction

Computational fluid dynamics (CFD) simulation is increasingly used in studying various filtration processes and for designing filter elements. Algorithms and software for CFD simulations of filtration processes were earlier presented by Fraunhofer ITWM [1]. This paper discusses new challenges and recent developments in CFD simulations used in filtration processes. New trends such as the subgrid resolution approach, computer simulation of the efficiency test, CFD support for designing filter elements with multiple porous media, and parallel computing are discussed. The subgrid resolution is a challenging approach, which allows to account for certain details of the solution, still working on relatively coarse grid. To address the challenge of simulating filter efficiency tests, we combine parameter identification from measurements carried out on a simplified filter, with CFD simulation, in order to predict filter efficiency for newly designed filter elements. However, the design of such elements is still a real challenge, and we show how CFD simulations efficiently assist such designs. New results with the recently developed parallel algorithms for simulation of industrial filtration processes are presented.

2. Simulation of flow through filters

We are interested mainly in oil filtration, thus we restrict the considerations to the laminar incompressible flows. Flow through filters may be described by the Navier Stokes-Brinkmann equations:

$$\frac{\partial \vec{u}}{\partial t} - \nabla \cdot (\tilde{\mu} \nabla \vec{u}) + (\rho \vec{u}, \nabla)\vec{u} + \mu K^{-1}\vec{u} + \nabla p = \vec{f} \quad \text{(momentum balance equation)}$$

$$\nabla \cdot \vec{u} = 0 \quad \text{(continuity equation)}$$

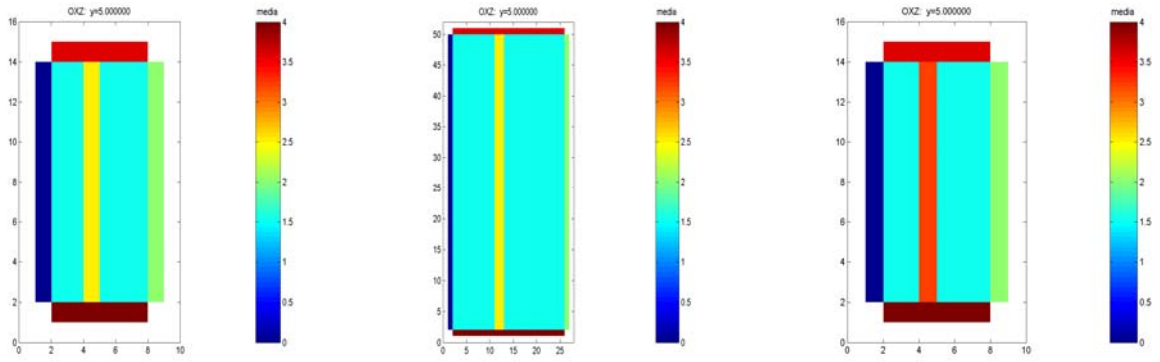
$$\frac{\partial C}{\partial t} + (\vec{u}, \nabla C) = D\Delta C - \alpha C \quad \text{(transport equation)}$$

Here $\vec{u}$, *p* and *C* are the velocity, the pressure and the concentration of dirt particles, respectively. Further on, $\tilde{\mu}$, *K,* ρ are the viscosity, the permeability of the porous (filtration) media, and the fluid density, whereas *D* and α represent the diffusion coefficient and absorption rate for the filter media. The equations are solved on Cartesian grid using finite volume discretization and collocated arrangement of the unknowns. Special attention is paid to the efficiency of the algorithm, see [1,2] for more details

3. Subgrid approach

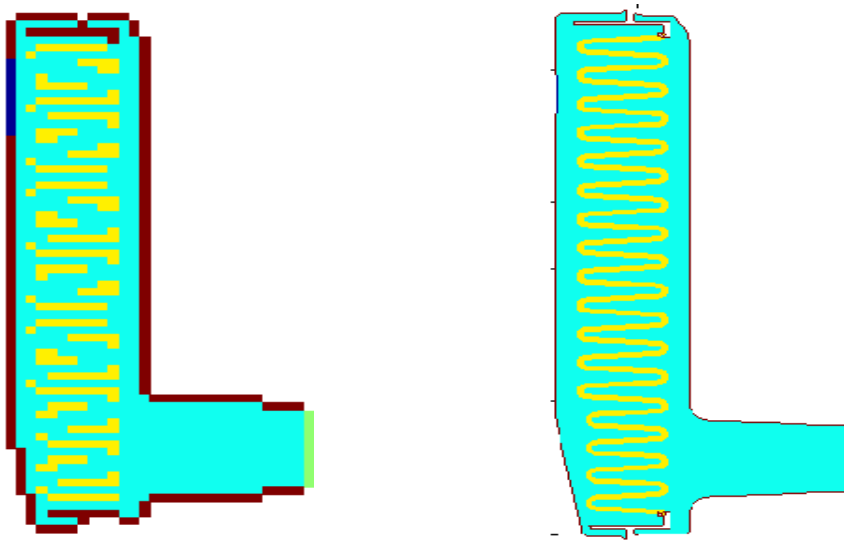
For many real filters, the filter medium or the filter element geometry is either too fine or too complicated. A naive approach is to use a fine computational grid at the expense of extensive memory and computational time, but this becomes infeasible. We suggest using a subgrid approach in order to capture the fine details. Auxiliary problems are solved in appropriately chosen grid cells on a relatively coarse computational grid. The solutions of the auxiliary problems are used within a systematic and a careful procedure of modifying and updating the coefficients of the Navier Stokes Brinkmann system in the chosen cells. More precisely, if some details of the filtration medium or of the filter element geometry are too fine to be resolved by the current computational grid, they are effectively accounted for by solving on a finer grid auxiliary problem in the proper grid cells, and modifying the coefficients of the Navier-Stokes-Brinkman equation there. Unlike the local refinement approach, the solution is not iterated between the coarse and the fine subgrids here. The approach is especially efficient if a series of computations have to be done on one fixed geometry, varying flow rate, viscosity, etc. In this case, the coefficients of the Navier-Stokes-Brinkman equation are updated only once, and all consecutive computations are performed fast on the selected coarser grid.

The idea of the procedure is presented with the help of a test example and of a real example. The test example represents a simple geometry with a thin flat filtering medium. This problem can be easily solved in various ways, here we demonstrate the idea of the subgrid approach for solving this problem. If we use fine enough grid, resolving the filtering medium, and having several cells within the filtering medium, we get accurate results. On the other hand, if we use in naïve way a coarse grid, where the cells are thicker than the filter medium, we get a very much wrong result. One way to account for the fine details of the geometry, and still to have a reasonable computational time, is to use the subgrid approach. In this case this means that all the coarse cells which contain piece of the thin filtering medium will be considered as cells where the permeability has to be changed. Auxiliary problem on finer grid is solved only in these cells (and only once for a fixed geometry), after that the basic simulations with varying flow rate, viscosity, etc., are solved on the coarse grid. In this simple case we get a very accurate solution using the subgrid approach.



Grid Size	Pressure drop (mbar)	Time (s)
2mm	578	2,00
0.5mm	289	85,00
Subgrid	289	15,00

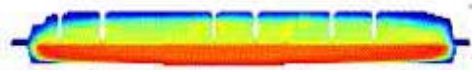
The second example which we consider is already with a complicated geometry. On the right picture, a fine grid is used. The thin filtering medium is well resolved from the grid, but computations on such a grid are very much time consuming. On the left picture, one sees the same filter geometry, but represented on a coarse grid. In this case, the grid can not capture the details of the filtering medium, and the results produced on such a grid are useless. At the same time, using the subgrid approach on this coarse grid, allows to compute the solution with 5% accuracy, using 10 times less computational time.

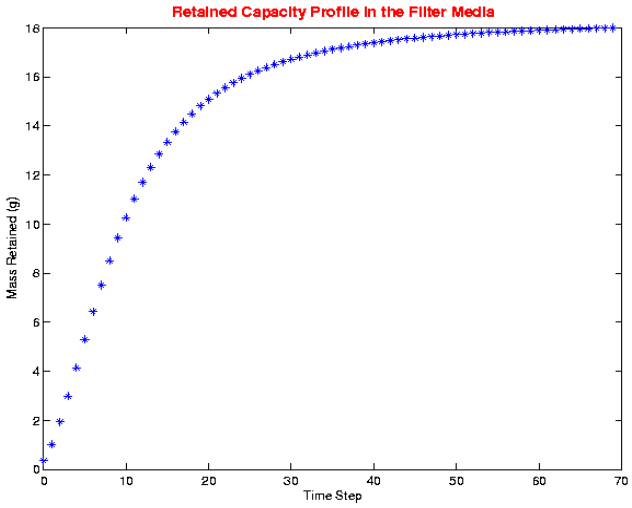


4. Efficiency test simulations

Solving the concentration of particles equation, equipped with absorption rate identified from the measurements, one can simulate an efficiency test. Pictures below show the

concentration of particles at certain time moment (left), as well as retained capacity profile (right). In these simulations, the flow was not influenced by the captured particles, what is obviously true at the early stages of the filtration process, but it is an approximation at the late stage of filtration. Here we obtained a good accuracy due to the fact that in identifying the absorption rate from measurements we used information for the full operating time of the filter. Currently we are working on a further improvement of the model, where the influence of the particles capturing on the flow will be accounted for, and the absorption rate will be identified as a time dependent function.





5. Parallel computing

A significant step towards reducing CPU time and increasing the problem size of CFD simulations is to run the simulations on parallel machines/processors. In SuFiS, the linear solver requires up to 90% of total CPU time and memory for most problems. Therefore, the main goal of parallel computing becomes two fold and both of them are extremely crucial for optimal CFD simulations. Firstly, it is the distribution of the discretization part of the algorithm, and secondly, it is the parallelization of the linear solver. Additionally, load balancing to scale the problem size according to the increased number of processed becomes crucial for optimal performance.

The domain decomposition algorithm takes into account that the orthogonal 3D structured grid is used as reference grid for definition of computational grid. Therefore, a standard 3D decomposition of processors, $p_1 \times p_2 \times p_3$, is used. This simplifies implementation of data exchange algorithms, since it is easy to define the neighbours of each processor and overlapping elements. METIS proves to be a good software for optimal load balancing. The grid is generated for real industrial filters. The figure shows an example, for which the graph of this grid has 596094 nodes, 1507732 edges and auxiliary grid load of 542800 nodes. An interface is provided for communicating between SuFiS ad METIS.

The parallelization of the linear solver, namely the BiCGSTAB algorithm, requires inter communication between processors. Implementation of matrix vector multiplication step needs additional information when the boundary nodes of local part of the vector are updated. Such information is obtained by exchanging data with neighbouring processors in

the specified topology of processors. Additionally, inner product computations require global communication between all processors. We make use of MPI routines, such as MPI_ALLREDUCE that computes the sum and distributes to all processors. The following table shows results of computational efficiency simulation tests for parallel computing on SP5 cluster with special IBM processors and a special network.

Number of Processors	Total time (s)	Speed up	Efficiency
	554.9	1.00	1.00
2	263.7	2.10	1.05
4	128.0	4.33	1.08
8	64.6	8.59	1.07
12	43.9	12.64	1.05
16	33.3	16.66	1.04
24	22.7	24.44	1.02
32	17.6	31.52	0.98

6. Conclusions

We presented new trends for simulating filtration processes efficiently. In the subgrid approach, it is apparent that the precision of pressure drop is strongly dependent on the accuracy of updated local permeability prescribed for the filter media. It is shown that an update using the Darcy law can give fairly accurate results. On the parallel computing end, it is shown that the parallel algorithm, using METIS for optimal load balancing, and MPI routines for intercommunication between processors, scales well in computational time and memory. It is clear that it is ready for efficient simulation of filtration processed with complicated geometries.

References

[1] O.Iliev, V.Laptev, D.Vasileva, Algorithms and software for flow through oil filters. Filtech Europa, Volume I, p. I-327 - I-334, October 2003.

[2] R.Ciegis, O.Iliev, Z.Lakdawala, On Parallel Numerical Algorithms for Simulating Industrial Filtration Problems, Computational Method in Applied Mathematics, Volume 7 (2007), No. 2, p. 118-134