Chapter 18
Virtual Prototyping of Metal Melt Filters: A HPC-Based Workflow for Query-Driven Visualization

Henry Lehmann and Bernhard Jung

18.1 Introduction

Recent advancements in additive manufacturing may be a game changer for the design of metal melt filters as the variety of 3D-printable filter geometries is vastly increased in comparison to conventional manufacturing processes. E.g., flow-guiding or surface-increasing elements of different shapes, sizes, and orientations may be added to the pores of foam-like filters, the strut shape may be freely varied, or filters may be designed with controlled variations of pore sizes. In the *Collaborative Research Center 920* (CRC 920), a combination of additive manufacturing with replication is successfully employed for the manufacturing of ceramic filters with hollow struts based on 3D-printed foam templates [1–4]. The particular foam templates for 3D-print are designed based on a geometric modeling approach for conventional open-cell PU foams, which allows for the generation of complete filters with adjusted strut thickness, pore density, and porosity based on periodic elements of hundreds of pores [5]. However, the new variety of filter designs, which can be targeted by 3D-printing technology, implies that a comprehensive exploration of the design space is not feasible using physical prototypes only. In this context, we propose a novel workflow for virtual prototyping of metal melt filters based on the state-of-the-art methods for geometrical modeling, *Computational Fluid Dynamics* (CFD), compression and indexing methods for scientific data, and query-driven visualization.

A major challenge are the large data volumes produced even by single CFD simulations of metal melt flow and even more so for the proposed virtual prototyping approach where many filter designs need to be evaluated inside a *High-Performance Computing* (HPC) environment. The large data problem is particularly severe during the analysis phase, where simulation results are assessed outside of the HPC envi-

H. Lehmann · B. Jung (✉)
Institute for Informatics, Technische Universität Bergakademie Freiberg,
Bernhard-von-Cotta-Straße 2, 09599 Freiberg, Germany
e-mail: jung@informatik.tu-freiberg.de

C. G. Aneziris and H. Biermann (eds.), *Multifunctional Ceramic Filter Systems for Metal Melt Filtration*, Springer Series in Materials Science 337,
https://doi.org/10.1007/978-3-031-40930-1_18

ronment using workstations with much lower computing resources. Furthermore, the analysis phase calls for interactive workflows. With data loading times being the main bottleneck, the amount of data loaded into memory should therefore be kept as small as possible.

Motivated by these requirements for interactive data analysis, we developed the LITE-QA (*Lossy In-Situ Tabular Encoding for Query-Driven Analytics*) framework for data management. During CFD simulations in the HPC environment, LITE-QA is used to compress and index the large volumes of simulation data. To support the data analysis phase, the framework adds search engine-like capabilities to data analysis tasks and allows, e.g., for focused visualizations of filter regions that meet the search criteria of the analyst, such as areas with high velocity, vorticity or backflow. Crucially, only very small parts of the simulation data, i.e. those parts that meet the analyst's search criteria, need to be loaded from disk. Figure 18.1 gives an overview of the proposed HPC-based virtual prototyping workflow and the embedded LITE-QA data management framework.

This chapter is organized as follows: Sect. 18.2 gives a brief overview of LITE-QA's data preparation methods (compression and indexing) when run 'in-situ', i.e. integrated into CFD simulations in the HPC environment. Then, Sect. 18.3 shows how the compressed and indexed data can be used in the analysis phase for the creation of query-driven visualizations. In order to demonstrate the feasibility of the proposed approach, Sect. 18.4 presents a virtual prototyping study involving a total of 84 unconventional filter designs. Finally, Sect. 18.5 gives a summary of the proposed

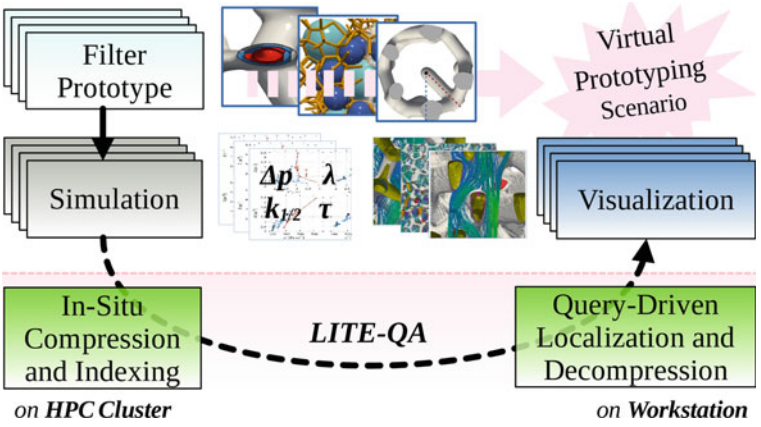


Fig. 18.1 HPC-based virtual prototyping workflow for visualization-assisted evaluation of virtual filter prototypes. CFD simulations of 84 filter prototypes are run in the HPC environment, where the resulting data is compressed and indexed using LITE-QA. The analysis phase outside the HPC environment is supported by LITE-QA's query mechanism for data-efficient, interactive creation of visualizations that provide further insights into the melt flow in the various filter variants. As a result of the virtual prototyping study, filter designs with improved performance and interesting flow characteristics are proposed

virtual prototyping approach and highlights the novel filter designs that performed best in the virtual prototyping study.

18.2 In-Situ Data Compression and Indexing

High-detail numerical simulations have become an increasingly important tool for the development of next-generation metal melt filters. For the CFD simulation of aluminum melts, the *Lattice-Boltzmann Method* (LBM) has been successfully applied at the pore-scale level inside porous filter structures [6–9]. The LBM allows for efficient parallelization in HPC systems, and thus allows the simulation of metal melt flow on high-resolution voxel grids with periodic boundaries [6, 8]. With increasing spatial and temporal resolution, LBM simulations are able to quickly generate large amounts of data in HPC clusters, easily in the order of hundreds of gigabytes or several terabytes and more. Conventional visualization workflows, where all data is stored on disk and loaded into main memory of less powerful workstations, lead to non-interactive workflows for data analysis and visualization or may even be impossible at all. Instead, the present research proposes novel methods for so-called in-situ data preparation, i.e. the reduction and indexing of simulation data while it is still residing in the HPC environment.

18.2.1 Early In-Situ Data Preparation

The simulation of a highly-porous filters requires large grids, i.e. at least 512^3 voxels in order to guarantee numerical stability and resolve fine geometric features like the struts in the required resolution. Due to the voxel-based discretization of the physical domain, the LBM can be setup for simulations of different filters. However, storing only the flow field for post-processing already requires 3×512 MB as 32bit floating point for each of the three components of the velocity vectors for each filter. For performing a local analysis and visualization of the aluminum melt flow inside a filter, a typical data set with five variables, e.g. the flow field and two additional properties, requires ≥ 2.5 GB for each filter and for every additional 40 time steps already additional 100 GB without compression.

Using fast algorithms early in the scientific workflow, the LITE-QA framework for HPC data management, as shown in Fig. 18.2, pursues two data preparation goals:

1. Reduction of storage required for high-resolution data during the running CFD simulations using fast error-bounded lossy compression, e.g. maximum relative error of 1% for decompressed contents, and
2. Creation of a compressed index for providing efficient access to compressed contents, i.e. query-based identification of regions and partial decompression of only the data needed for the visualization task.

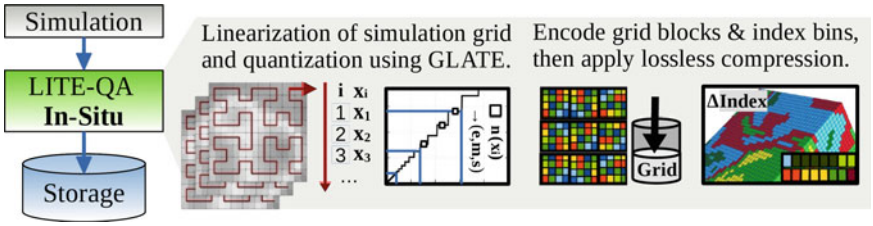


Fig. 18.2 The LITE-QA in-situ processing pipeline achieves high compression rates and is run as part of the simulations inside the HPC environment. The single steps of in-situ processing are grid linearization, data quantization, encoding of grid and index data, and lossless compression of the encodings

Like other in-situ compression methods, e.g. SBD [10], ISABELA [11], ZFP [12], SZ [13] and temporal extensions of them [14, 15], LITE-QA applies lossy compression and reduces the amount of data written out to the file system. LITE-QA applies lossy compression integrated with indexing directly in the simulation processes inside the HPC environment. Thus, temporary storage of large data is avoided already early in the workflow. By using fast algorithms for compression and indexing 'in-situ', i.e. in the HPC environment, the load on storage systems and the bandwidth requirements for network transfer are reduced, while also preparing the data set for later post-processing outside of the HPC environment. By using quantization with a small error of e.g. 1% maximum point-wise error, the full-resolution data sets are stored at a smaller memory footprint, while the full flexibility for post-hoc visualization and analysis is maintained.

18.2.2 Grid Compression and Index Generation

LITE-QA uses fast algorithms for grid compression and the generation of a compressed index, which are based on the sequential and differential encoding employed in the SBD compression algorithm. However, LITE-QA and SBD differ in the data quantization methods used for bounding the point-wise maximum error [16, 17]. Data quantization in SBD is based on a look-up table estimated from the data in the individual subgrids which causes compression artifacts on distributed grids in parallel simulations and on high-resolution temporal data [18]. Instead, LITE-QA compression employs the GLATE (*Grid Linearization and Tabular Encoding*) compressor which uses a step function for quantization, resulting in stable quantization across subgrid boundaries and on high-resolution data.

LITE-QA employs the data encoding of GLATE, i.e. the discrete quantization for linearized numerical data, for the compression of simulation grids in a block-wise manner and the generation of an index based on binning the quantized values. LITE-QA constructs a compact encoding for grid blocks and for index bins [18], which is designed to support partial decompression of simulation results, while being

adeuqate for high-degree data reduction using lossless compression techniques on those encodings as backend. For lossless compression of encoded grid blocks and index bins, the bit-packing codec `fastpfor` [19] and the general purpose lossless compressor `zstd`[1] are used.

The temporal extension t-GLATE yields an improved compression rate for temporal data by exploiting the temporal coherence between successive time steps. GLATE establishes a trade-off between data accuracy and compression rate. E.g. for a point-wise maximum error of 1%, the size of the grids is reduced by a factor of five, while data indices are reduced by a factor of three as compared to the uncompressed data assuming the 32bit floating point data type.

For the algorithmic details of LITE-QA, GLATE and t-GLATE we refer to the original publications [16–18]. In the following, representative results on the compression rates achieved for the metal melt filtration simulations conducted in the virtual prototyping study in Sect. 18.4 and, generally, HPC simulations in CRC 920 are presented.

18.2.3 In-Situ Compression Performance

In-situ compression performance is evaluated for an LBM simulation solving an incompressible isothermal flow of liquid aluminum inside a computer-generated monodisperse filter with porosity 90% and Reynolds number $Re = 90$ on a grid of 512^3 voxels. The fluid dynamics is based on the Navier-Stokes equations assuming a superficial velocity of 6 cm $\cdot$ s^{-1}. The simulation generates a typical data set for visualization, i.e. containing the flow field u, v, w and two additional properties, the velocity magnitude M and a vortex indicator Q. The GLATE quantization for grid and index compression in LITE-QA is operated at 1% maximum point-wise error, which is sufficient for the compression task with respect to local flow visualization.

Non-temporal Compression Performance

Figure 18.3 shows the performance for non-temporal compression of grids using GLATE. Figure 18.3 (1) compares GLATE's compression rates without additional indexing to ZFP, a state-of-the-art lossy floating point compression algorithm inspired by texture compression methods used in graphics hardware [12]. ZFP does not generate a compressed index. Figure 18.3 (2) shows the data size resulting from GLATE compression of the simulation variables with and without additional indexing of the variables u, M, Q. For the compression tests, GLATE restricts the relative error to 1% and ZFP is operated on level 15, which is comparable to a maximum point-wise error of 1% [17].

[1] `zstd` is a real-time compression algorithm providing high compression ratios and a very fast decoder, `zstd` is available at `https://facebook.github.io/zstd/`.

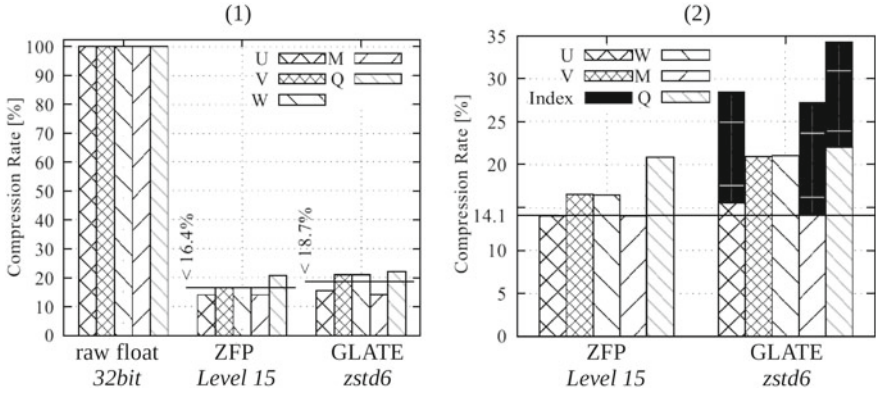
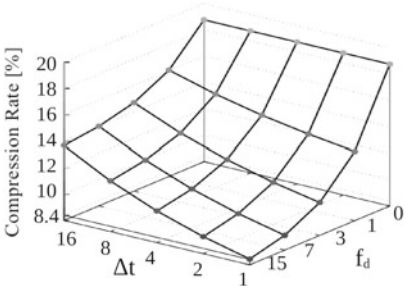


Fig. 18.3 Non-temporal compression performance on grids for variables u, v, w, M, Q and indices for u, M, Q. The data is generated using a LBM simulation of liquid aluminum through a computer-generated filter structure with 90% porosity, superficial velocity 6 cm $\cdot$ s^{-1} and Reynolds number of $Re = 90$. (1) GLATE and ZFP achieve at least five fold reduction for compressed grids in average for all variables u, v, w, M, Q. (2) During the more complicated index compression, a three fold reduction on compressed index bins is achieved in average, i.e. only ~12% more storage required as compared to the corresponding compressed grids

On the variables u, v, w, M, Q, GLATE and ZFP achieve an average compression rate of 18.7% and 16.4%. On the index variables u, M, Q, the average compression rate achieved by LITE-QA is 29%. The best index compression is achieved on M with 26%. The index bins for the variables u, M, Q are compressed at a lower rate, as compared to the grid blocks. However, the compressed index requires only ~12% more storage as compared to the corresponding compressed grids.

Temporal Compression Performance

The t-GLATE temporal compression scheme differentiates between so-called key-frames and difference-frames. While key-frames are compressed and decompressed independently, difference-frames reference the previous frame and are compressed with a higher efficiency. The compression efficiency is directly related to the time step size Δt of the simulation and to the amount of difference-frames inserted between key-frames. t-GLATE achieves a trade-off between temporal resolution of exported data, i.e. multiples of Δt, and the resulting compression rate by encoding the data differences to the last exported frame. The approach for difference encoding in t-GLATE falls back to encoding absolute values in the case differences become too large. Therefore, no decline of compression rate is observed, even when the data is exported with a low temporal resolution. t-GLATE is applied to 1024 time steps, which have been exported from the aforementioned simulation of liquid aluminum using multiples of the time step width $\Delta t = 9.13$ μs and increasing amounts of difference-frames between key-frames.



f_d	$1\Delta t$	$2\Delta t$	$4\Delta t$	$8\Delta t$	$16\Delta t$
0	18.7	18.7	18.7	18.7	18.7
1	13.2	13.7	14.3	15.0	16.0
3	10.5	11.1	12.0	13.2	14.6
7	9.1	9.9	10.9	12.3	14.0
15	8.4	9.3	10.4	11.8	13.6

Compression Rate [%]

Fig. 18.4 Temporal compression performance of t-GLATE for a sequence of 1024 time steps of liquid aluminum melt flow through a filter with 90% porosity and superficial velocity 6 cm $\cdot$ s^{-1}, Reynolds number $Re = 90$ and time step width $\Delta t = 9.13$ μs. t-GLATE establishes a trade-off between temporal resolution of decompressed data, i.e. multiples 1, 2, 4, 8, 16 of Δt, difference-frames per key-frame $f_d = 0, 1, 3, 7, 15$ and the resulting compression rate

As shown in Fig. 18.4, t-GLATE yields a compression rate of 8.4–18.7% for all time steps with data set variables u, v, w, M, Q in average. Without inserting any difference-frames, the compression rates correspond to non-temporal compression using GLATE and are equivalent to the rates shown in Fig. 18.3. For 15 difference-frames between key-frames, t-GLATE reduces the data set to 8.4% for exporting all 1024 time steps of the simulation run, i.e. the data set is reduced from 2.5 GB $\times$ 1024 = 2.5 TB to 215 GB.

18.2.4 Summary

In-situ data reduction and indexing methods aim at preparing large-scale simulation data for later analysis outside of the HPC environment. On non-temporal datasets, the GLATE method presented above compresses the LBM simulations to 18.7% in average, comparing to state-of-the-art methods such as ZFP. Further, GLATE was extended to t-GLATE for temporal datasets where compression rates are improved to 8.4% in average, outperforming existing methods. Moreover, LITE-QA combines the GLATE data compression with data indexing needed for query-driven analyses. Whereas uncompressed indices typically require storage amounts of $\geq 100\%$ in addition to the simulation data [20], GLATE compression with integrated indexing reduces the storage requirements for combined simulation and index data to ~27–35%, while guaranteeing a point-wise maximum error of 1%.

18.3 Query-Driven Visualization of Melt Flow

General challenges for the visual exploration of time-dependent scientific data sets and batches of CFD simulations include the non-interactive loading times already for single time steps of a simulation. Furthermore, a purely visual search for interesting features can be tedious and error-prone as flow-relevant features such as high vorticity, high-velocity or backflow occur rather sparsely, often only in a small percentage of all grid cells. Addressing these challenges, query-driven visualizations replace purely visual search in the full dataset with search engine-like capabilities to create visualizations of interesting areas only. The LITE-QA query mechanism, as illustrated in Fig. 18.5, operates on the compressed simulation grid data and index data generated using the HPC data preparation pipeline described in Sect. 18.2.

When searching the flow field for interesting regions with specific characteristics, e.g., areas with significant backflow or very fast flow, it is favorable to access the data in a query-driven manner. Queries make use of the compressed index and are steered using so-called range conditions on simulation variables, e.g., on the velocity magnitude M. Using the index, specific spatial locations are efficiently identified in the flow field where a range condition $a \leq M \leq b$ is true. Given the selected locations from the index, additional data is decompressed and local visualizations of the fluid flow are procedurally generated. Multiple visualizations of an selected region at different time steps can be created to observe the flow evolution. Similarly, comparative visualizations of the same region can be generated for design variants of a filter to gain insights into the effects of geometry modifications.

18.3.1 Querying of Regions in the Flow Field

The visualizations presented in this section use the unsteady simulation data described in Sect. 18.2.3 which contains five variables u, v, w, M, Q at each grid point of the the flow field. Compressed grids and indices were generated in-situ with LITE-QA

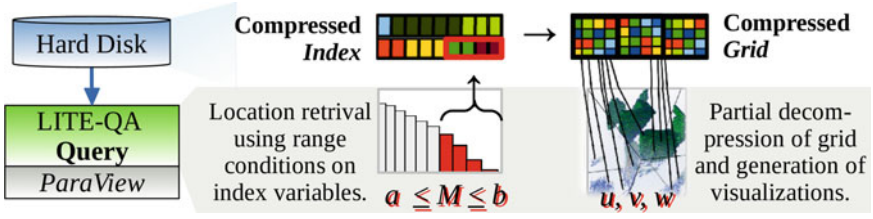


Fig. 18.5 The LITE-QA query processing pipeline integrates a compressed index for realization of a query mechanism used to locate grid cells based on range conditions on variables stored in the index, e.g. $a \leq M \leq b$. Given a set of locations, the flow field u, v, w is decompressed for visualization. The mechanism also minimizes data loading times as only data needed to answer the query is retrieved from disk

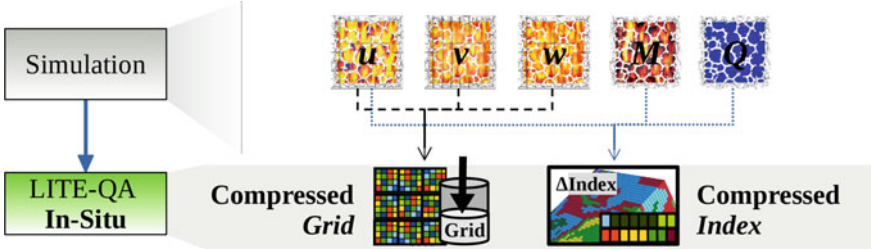


Fig. 18.6 Compressed data set for flow visualization purposes. The data produced by the numerical simulations is stored in a compressed grid for the grid variables u, v, w, and in a compressed index for the index variables u, M, Q accordingly. u is stored as compressed grid and index

during the running simulation, according to the setup as shown in Fig. 18.6. During the data preparation, the flow field u, v, w is stored as a compressed grid, hence u, v, w are called *Grid Variables*, and the variables u, M, Q are stored in a compressed index, hence u, M, Q are called *Index Variables*. The variable u is stored both in the grid and in the index.

Figure 18.7 shows three different regions in the flow field at time step 80. The regions are identified using range conditions on the index variables, i.e. u for identification of backflow, where the stream is redirected against the bulk flow, M for the identification of fast flow and Q for the identification of vortex-like flow. The regions are identified using the following range conditions on the index variables:

$$\begin{aligned} -\infty \leq u &\leq u_0 \quad \text{for backflow with } u_0 < 0, \\ m_0 &\leq M \leq +\infty \text{ for fast flow and} \\ q_0 &\leq Q \leq +\infty \text{ for vortex-like flow.} \end{aligned}$$

In the example, the concrete query parameters were determined as quantiles from value distributions of the index variables u, M, Q, i.e. u_0 as the 5% quantile of the value distribution for values $u < 0$, m_0 as the 85% threshold of the maximum velocity, and q_0 based on the 99% quantile of the value distribution of Q. On the simulation grid composed of 512^3 voxels, the range conditions on Q predictably returns 1% or about 1.34 million grid cells, whereas the amount of cells returned for the queries on u and M depend on the characteristics of their distributions at a given time step.

18.3.2 Implementation of Query Mechanism

The ParaView framework is used as a platform for the implementation of the LITE-QA query mechanism. ParaView is a powerful open-source application for scientific visualization. The central tool for modeling visualizations in ParaView is the so-called visualization pipeline, which arranges algorithms in a network graph and

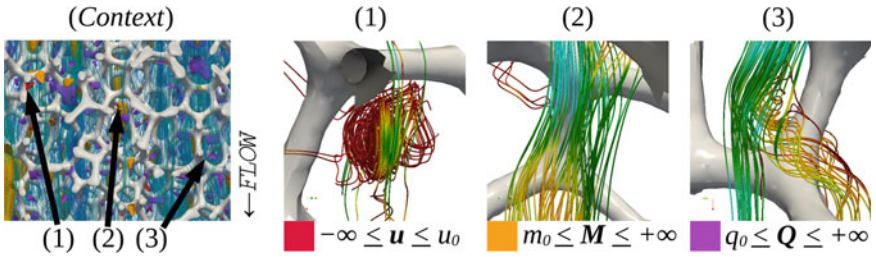


Fig. 18.7 Visualization of regions in the flow field of liquid aluminum melt inside a filter with 90% porosity, superficial velocity 6 cm $\cdot$ s^{-1} and Reynolds number $Re = 90$. Three different regions are located using range conditions on index variables u, M, Q, i.e. (1) backflow in form of a vortex in the slip stream behind a strut in red, (2) fast flow path through a pore window in orange, and (3) fast swirled and vortex-like flow on upstream surface of a strut joint in purple

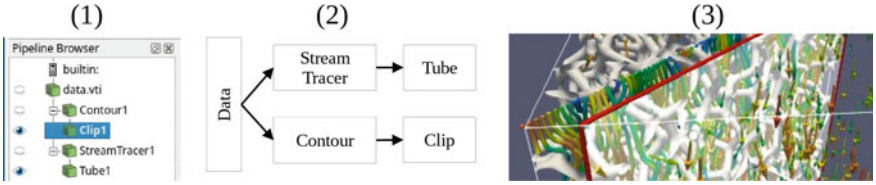


Fig. 18.8 ParaView visualization pipeline used to render the aluminum melt flow through a computer-generated filter structure with 90% porosity. The pipeline (1) as shown in the graphical user interface and (2) displayed as a network flow graph. (3) The pipeline produces the visualization consisting of a clipped iso-contour surface and streamlines as 3D tubes. The root of the pipeline is the data producer reading from a file

defines the data flow by mapping inputs and outputs between them. As shown in Fig. 18.8, algorithms import and transform data, e.g. obtained from simulation results, and perform rendering of 3D objects, e.g. clipped contour surfaces and streamlines with tubes wrapped around them for better visibility.

The LITE-QA query mechanism is implemented using three algorithms for the ParaView visualization pipeline, which execute three different query types modeled after operations provided by the compressed index. The `lqaTable Source` algorithm performs the so-called *Count Query*, the `lqaIndex Source` performs the so-called *Index Query* and `lqaGrid Source` the so-called *Grid Query*. The LITE-QA query algorithms act as data producer. Instead of loading complete uncompressed data sets from files, they use the data index for localization and decompression of only those parts of the simulation grid that are required for the visualization. The queries are typically executed in a hierarchical order, as shown in Fig. 18.9:

1. `lqaTable Source` determines the value distribution of one index variable, e.g. M, and returns a histogram as a `vtkTable`.
2. `lqaIndex Source` evaluates one or more range conditions on index variables, e.g. $a \leq M \leq b$, and returns the grid cell indices I matching the range conditions as a `vtkPolyData` point cloud.

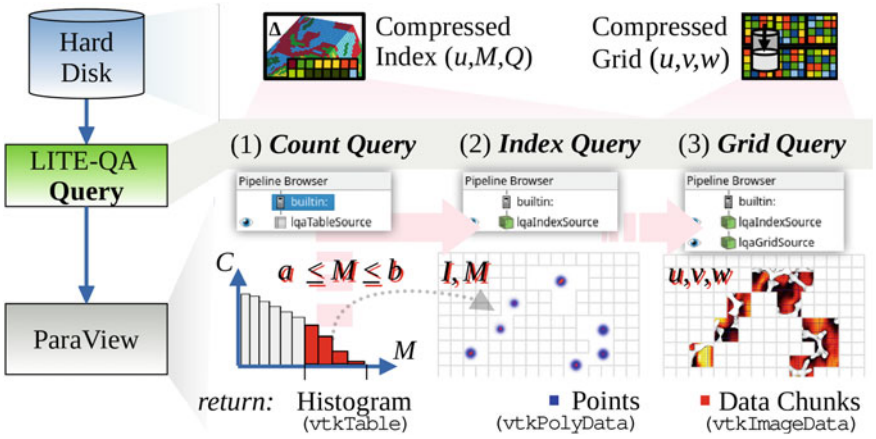


Fig. 18.9 LITE-QA query mechanism operating on compressed grids and compressed indices. Three query types are implemented as data producers for the ParaView visualization pipeline: (1) `lqaTable Source` performs a count query, which returns a histogram C, M as `vtkTable`, (2) `lqaIndex Source` performs an index query, which returns a `vtkPolyData` point cloud representing the indices I of grid cells whose data values M match the range condition $a \leq M \leq b$, and (3) `lqaGrid Source` performs a grid query, which returns a `vtkImageData` containing the decompressed flow field u, v, w at cell indices I

3. `lqaGrid Source` evaluates the range condition according to `lqaIndex Source` and, additionally, decompresses the simulation data at the respective grid cells to return the flow field u, v, w as a `vtkImageData`.

18.3.3 Query-Driven Local Visualization

The generation of the visualization scene is performed using the procedure as shown in Fig. 18.10, where the `lqaIndex Source` and `lqaGrid Source` algorithms are used as a data producer for the visualization pipeline. The range condition for the queries is formulated based on the data distribution of the index variables u, M, Q as explained in Sect. 18.3.1, which are obtained using a preceding count query using `lqaTable Source` as shown in Fig. 18.9. The locations obtained from the index using the range conditions on index variables u, M, Q usually form clusters, which correspond to local phenomena, e.g. backflow, fast preferential flow and vortex-like or swirled flow.

In the example visualization of Fig. 18.10, the point cloud obtained from the index is further decomposed into clusters by using an Euclidean clustering algorithm from the ParaView toolkit. By performing a threshold operation, one specific cluster is selected for visualization based on e.g. the cluster size or a sequential cluster index. Based on the points of the selected cluster a partial decompression of the

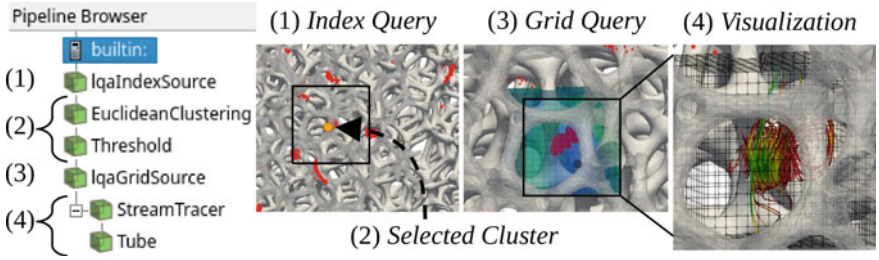


Fig. 18.10 ParaView pipeline for query-driven visualizations based on the LITE-QA framework. (1) determination of region in the flow field using an index query, e.g. $-\infty \leq u \leq u_0$ for backflow, (2) spatial clustering of the resulting point cloud and selection of one cluster, (3) partial decompression of the flow field u, v, w using a grid query on the selected cluster points, and (4) generation of visualization on decompressed data

flow field u, v, w is performed, as described in Sect. 18.3.2. Alternatively, in order to decompress a larger region of the flow field around the selected cluster, the grid query can perform the partial decompression based on a fixed decompression extent, which is defined as an axis aligned bounding box with edge length Δ placed at the geometric center point of the selected cluster. The actual visualization task is performed on the decompressed data, e.g. flow visualization with stream tracer and 3D tubes.

18.3.4 Export of Visualization Scenes

To further support the data analysis, the visualizations of local phenomena can be exported to the web and immersive *Virtual Reality* (VR) environments as shown in Fig. 18.11. Once a visualization has been generated using the procedure described in Sect. 18.3.3, the scene data can be exported to the web and to immersive VR environments using the `ExportScene()` and `SaveData()` function from the ParaView Python module respectively. For web export, ParaView generates a standalone HTML version of the scene, which uses WebGL for rendering. For VR, the scene is exported to files and imported into a distributed rendering system based on OpenSceneGraph.

Iterative Generation of Visualizations

Using the LITE-QA query mechanism, visualizations can be exported iteratively in ParaView, e.g. for creating flow animations of selected regions from temporal data sets. Once a region in the flow field has been located in one particular time step of the simulation, the visualization pipeline is evaluated using iterative calls to the `UpdatePipeline(time)` and `SaveScreenshot()` function from the ParaView Python module in order to generate frames for an animation at the identified

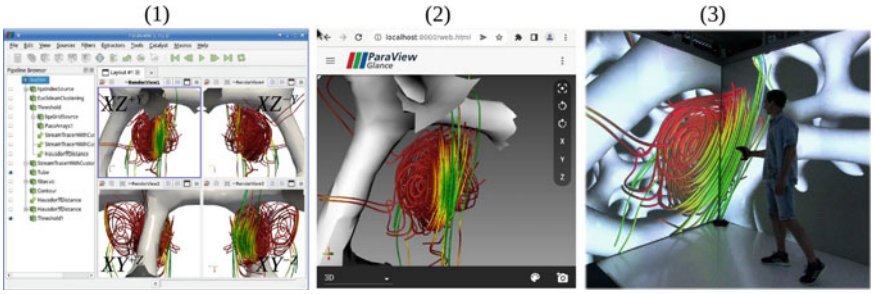


Fig. 18.11 Local visualization of a backflow vortex, which has been located using the LITE-QA query mechanism in (1) the ParaView graphical user interface. The scene is exported to (2) web and (3) immersive VR using the ParaView Python module. An immersive VR application based on OpenSceneGraph is run in the XSITE CAVE (eXtreme definition Spatial Immersion and interacTion Environment) [14, 21], an innovative surround-screen VR environment with an extremely high pixel resolution. The visualization is controlled intuitively using real-time optical tracking and hand-held interaction devices, e.g., adjustment of parameters and temporal and spatial navigation

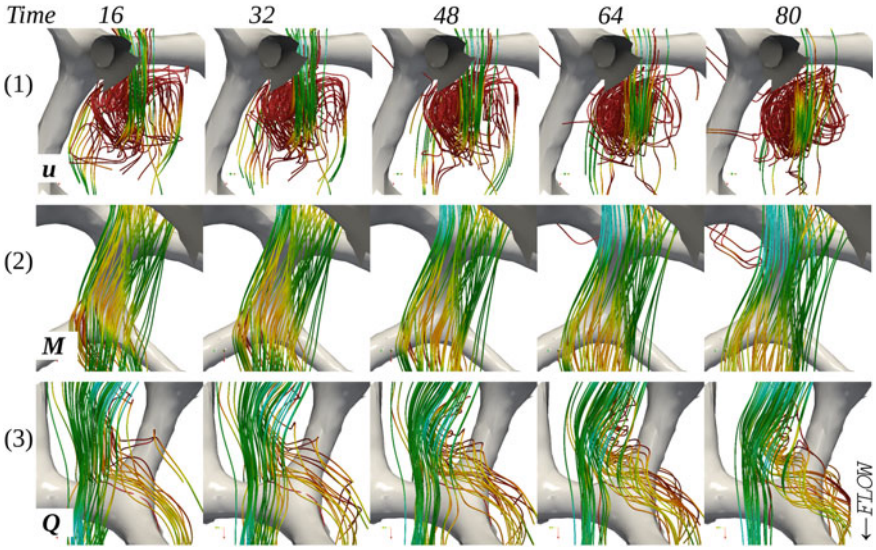


Fig. 18.12 Visualization of local flow evolution over multiple time steps in regions with interesting flow properties as identified by the query-mechanism: (1) strong backflow, i.e. $u \leq u_0 < 0$, (2) fast flow, i.e. $M \geq m_0$, and (3) vortex-like flow, i.e. $Q \geq q_0$

location in a temporal context. Figure 18.12 shows the temporal evolution of the flow of liquid aluminum for the three regions, which are identified in time step 80 using range conditions on the index variables u, M, Q, i.e. a large backflow vortex, fast preferential flow paths and vortex-like or swirled flow.

18.3.5 Summary

Data indexing as described in the previous section enables the efficient identification of regions in the simulation dataset that meet the criteria of user-specified queries. Query-driven visualization utilizes this by creating focused visualizations of large-scale simulations that only include subsets deemed relevant by the analyst such as regions with high vorticity or significant backflow. Query-driven visualization is also computationally very efficient as only the relevant subset of the full dataset needs to be loaded from disk. Besides supporting interactive visualizations of large-scale simulations in desktop, web, and immersive VR environments, the presented approach also supports the automated generation of animations.

While so far the presented methods for in-situ data preparation and query-driven analyses have been presented in the context of single simulations, the next section considers their application for virtual prototyping purposes, where a large number of filter designs is generated, simulated and analyzed.

18.4 Virtual Prototyping Study

In the virtual prototyping scenario presented in this section, a reference filter design is systematically varied in order to create a total of 84 filter variants. Metal melt flow in each of these filters is simulated using the LBM described in [6, 8, 9] and simulation data is compressed and indexed with LITE-QA as described in Sect. 18.2. Evaluation of the candidate filters involves a combination of statistical analyses of global melt flow properties derived directly from the compressed indices, query-driven visualizations that help to gain insights into the local effects of filter variations on the melt flow as well as calculations of global flow characteristics such as filtration efficiency and pressure drop directly obtained from the LBM simulations.

18.4.1 Overview of Investigated Filter Structures

A total of 84 computer-generated filters are screened for the virtual prototyping scenario. The geometry for the filters is obtained from the procedure of representative geometry generation based on a filter skeleton [5]. The filter skeletons are generated from a Laguerre tessellation of periodic sphere packings, which are iteratively adjusted until they reflect the topological properties of real *Ceramic Foam Filters* (CFFs), e.g. the number of faces per cell and the edges per face. The filter geometry is obtained using isotropic Gaussian smoothing on a sharp discretized representation of the strut network embedded into a 3D voxel grid. The reference structures are modeled using a strut aspect ratio of one corresponding to CFFs with circular strut

cross section shape as obtained from 3D-printed templates [1, 3, 5]. Based on the modeling procedure, the new filters are generated in the following three groups:

1. (fA) modification of strut shape:
   - f_a—elliptical elongation and flattening of the strut cross section with respect to the bulk flow direction controlled by a strut aspect ratio a,
   - f_{ab}—drop-like strut cross section controlled by a strut aspect ratios a for the upper and b for the lower half of the strut cross section shape, and
   - f_{ba}—reversed drop-like strut cross section shape.
2. (fB) insertion of flow-guiding features:
   - f_w—closing of a total amount of w randomly chosen pore windows,
   - f_α—insertion of finger-like struts on the downstream surface, downward-pointing, inclined by angle α with respect to the bulk flow direction, and
   - f_β—insertion of finger-like struts on the upstream surface, upward-pointing, inclined by angle β.
3. (fC) varying pore size and strut shape within a filter:
   - f_c—continuous thickening of struts from top to bottom, and
   - f_q—systematic arrangements of a total amount of q size-varying pores.

Modifications of Strut Shape and Insertion of Flow-Guiding Features

As shown in Fig. 18.13, all filters in groups (fA) and (fB) are designed by modification of the strut shape and pore geometry of a reference structure f_1^ε with 216 pores. Reference structures are generated for porosities $\varepsilon = 70, 80, 90\%$ which exhibit equal-sized pores with circular strut cross section shape, i.e. f_1^ε corresponds to f_a with aspect ratio $a = 1$ and porosity ε.

The geometric modifications of the strut cross section for f_a is directly integrated into the modeling procedure based on anisotropic Gaussian smoothing [22]. f_{ab} and f_{ba} are generated using a voxel-wise image blending operation, in order to merge flattened and elongated struts into a drop-like strut shape.

For modifications f_w, f_α and f_β, additional flow-guiding features, i.e. closed windows and finger-like struts, are inserted into the strut network prior to the generation of the actual filter surface [22]. The insertion of flow-guiding features reduces the porosity of f_w by 2–5% and for f_α and f_β by approximately 2–2.6%. As a result of the insertion of finger-like struts, the cumulative length of the strut network increases by approximately 14.5%. All filters in groups (fA) and (fB) are generated for porosities $\varepsilon = 70, 80, 90\%$.

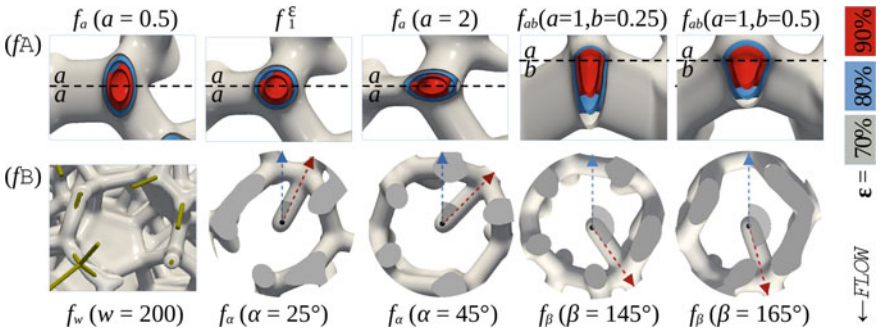


Fig. 18.13 Geometric modifications of the reference structure f_1^ε that has 216 equal-sized pores. Group (fA) with modifications of the strut shape f_a, f_{ab} and f_{ba}. For elliptical deformation f_a, the strut aspect ratio is controlled by parameter a, i.e. elongation for $a < 1$ and flattening for $a > 1$. The reference structure f_1^ε corresponds to $a = 1$. For f_{ab} and f_{ba}, an additional parameter b controls the aspect ratio of the lower part of the strut shape. Filters in group (fB), i.e. f_w, f_α and f_β, have flow-guiding features inserted. For f_w, a total of w windows are randomly closed. For f_α and f_β, downward- and, resp., upward-pointing finger-like struts are inserted inclined by angle α and β with respect to the bulk flow direction. Images for f_α are reproduced with permission [22]

Modifications with Varying Pore Size and Strut Shape

The filters in group (fC) vary pore size and, resp., strut shape along the filter depth. In group f_c, the strut width increases from top to bottom while in group f_q, the pore size is systematically varied in different layouts (see Fig. 18.14). Filters f_c are generated with the filter skeleton of the reference structure f_1^ε. Filters f_q are generated with three new filter skeletons arranging size-varying pores:

1. With $q = 200$ pores, a continuous transition from larger to smaller pores and reverse transition from small to large with 32 larger pores, 72 medium-sized pores and 96 smaller pores. The cumulative strut length decreases by 7.2% with respect to f_1^ε.
2. With $q = 265$ pores, a continuous transition from larger to smaller pores without reverse transition. The cumulative length of the strut network increases by 12.1%.
3. With $q = 320$ pores, an alternating pattern of 32 larger pores and clusters composed of nine smaller pores, which increase the cumulative strut length by 23.2%.

The filters f_q are generated for porosities $\varepsilon = 70, 80, 90\%$, while the filters f_c are generated for porosity $\varepsilon = 85\%$. The porosity value of f_c results from merging two reference structures $f_1^{90\%}$ and $f_1^{80\%}$ using a voxel-wise image blending operation along the filter depth. The procedure accomodates a smooth transition of the strut width and the strut shape across both parts of the filter, i.e. one with 90% porosity for the top and one with 80% porosity for the bottom. In addition to the transition from thin to thick struts, f_c integrates a smooth transition of the strut shape between the top and bottom parts of the filter, where the strut shape at the top is controlled by the aspect ratio $a = c$ and the aspect ratio in the bottom is set to $a = 4$. The filters f_q are

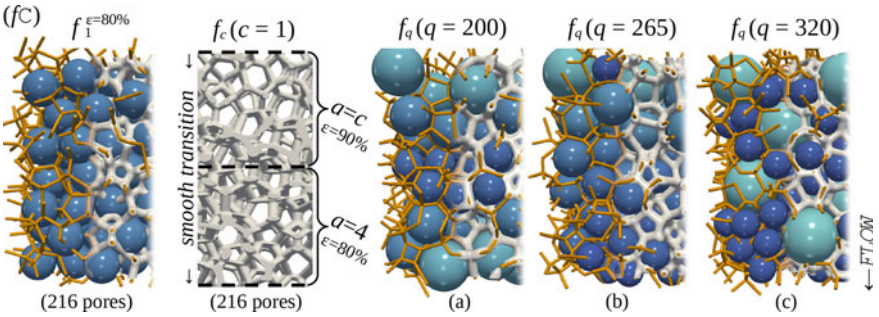


Fig. 18.14 Group (fC) of investigated filters with varying strut thickness f_c and varying pore size f_q. Filters f_c accomodate a continuous transition of thin struts with aspect ratio $a = c$ with $c \leq 2$ at the top to thick struts with fixed aspect ratio $a = 4$ in the bottom part of the filter. The filter f_c is shown for $c = 1$. The filters f_q are modeled with different layouts of $q = 200, 265, 320$ pores with varying size, based on new filter skeletons, i.e. transition of large to small pores (a) with and (b) without reverse transition, and (c) an alternating pattern of large pores and clusters of small pores. The filters f_1^{ε} and f_q are generated for porosities $\varepsilon = 70, 80, 90\%$ and shown for $\varepsilon = 80\%$. Dark blue spheres indicate smaller pores, whereas turquoise indicates larger pores with respect to f_1^{ε}

generated with circular strut cross section and uniform strut width for the complete filter.

As a consequence of the designs in group (fC), a variation of the porosity is induced in different spatial regions of the filter domain. For f_c (ix–xi) and f_q (ix) and (x), the porosity decreases along the filter depth in a uniform way according to the increase of strut thickness or the decrease of the pore size. In contrast, for f_q (xi) with $q = 320$ pores, where large pores alternate with clusters of smaller pores, locally increased and decreased porosity is induced on a raster of $4 \times 4 \times 4$ spatial regions in an interlaced manner.

18.4.2 Generation of Simulation Data Sets

A total of 84 simulation data sets are generated in LBM simulations using the parametric modifications as shown in Table 18.1. The 6 modifications in groups (fA) and (fB) are generated with 4 parameterizations and for porosities $\varepsilon = 70, 80, 90\%$ each, yielding 72 concrete filter designs. In group (fC), filters f_q are generated with 3 parameterizations and porosities $\varepsilon = 70, 80, 90\%$ to yield 9 concrete filter designs, while filters f_c are generated with 3 parameterizations and fixed porosity $\varepsilon = 85\%$ for 3 filter designs.

The filters are evaluated for process conditions present during the removal of alumina oxide inclusions from the aluminum melt inside CFFs with 30 PPI as reported for a pilot filtration line [23]. Therefore, the structures are scaled to meet the pore density of the 30 PPI filters [9, 22], resulting in a physical domain of $17.5 \times 17.5 \times$

Table 18.1 Overview of parametric variations for computer-generated filter structures used to generate the simulation data sets for the virtual prototyping scenario. All filters are generated for porosities $\varepsilon = 70, 80, 90\%$ except for f_c with fixed porosity $\varepsilon = 85\%$

Modification		Parameterization			
(fA) *Strut shape*		(i)	(ii)	(iii)	(iv)
– Elliptical struts	f_a	$a = 0.5$	1	2	4
– Drop-like struts	f_{ab}	$a = 1$ $b = 0.25$	1 0.5	2 0.25	2 0.5
– Drop-like reversed	f_{ba}	$a = 1$ $b = 0.25$	1 0.5	2 0.25	2 0.5
(fB) *Flow-guiding features*		(v)	(vi)	(vii)	(viii)
– Closed windows	f_w	$w = 50$	100	150	200
– Finger-like downwards	f_α	$\alpha = 15°$	25°	35°	45°
– Finger-like upwards	f_β	$\beta = 135°$	145°	155°	165°
(fC) *Varying geometry*		(ix)	(x)	(xi)	
– Size-varying pores	f_q	$q = 200$	265	320	
– Shape-varying struts	f_c	$c = 0.5$	1	2	

17.5 mm, which is discretized on a grid composed of 512^3 voxels with a spatial resolution $\Delta x = 34.5$ μm for all data sets.

Surface Area of Simulated Filters

The surface area S of the 84 generated filter samples from groups (fA), (fB) and (fC) is shown in Fig. 18.15. The filters f_a, f_{ab}, f_{ba}, f_α, f_β, f_w, f_c and f_q are generated with parameterizations (i–xi) according to Table 18.1. As can be seen, all modifications lead to an increase of the surface area, except f_q (ix), which has a lower surface area than the corresponding reference structure f_1^ε with the same porosity. Filters with drop-like strut shape f_{ab} and f_{ba} in parameterizations (i) and (ii), and filters with finger-like struts f_α and f_β exhibit the largest increase. Among filters with porosity $\varepsilon = 90\%$, filters with closed windows f_w (viii) have the largest surface. Although the filter f_q (xi) has the largest cumulative strut length, its surface area is smaller as compared to f_α and f_β.

Modeling the Fluid Flow

The flow through the open-cell structures is assumed to be periodic with a constant flow rate driven by an imposed pressure gradient in the x-direction and adjusted by a controller, which monitors the flow rate through the inlet and outlet of the filter, while maintaining the prescribed superficial velocity [9]. All process simulations lie within the steady state, where the viscous losses contribute to at most 60% of

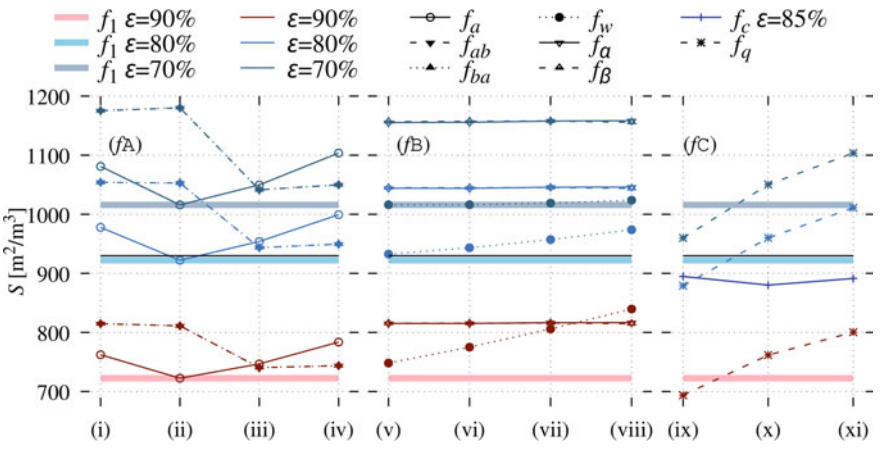


Fig. 18.15 Surface area S of the 84 filter prototypes. The filters are referenced (i–xi) according to Table 18.1. Left (i–iv): group (fA) with modified strut shape f_a, f_{ab} and f_{ba}. Middle (v–viii): group (fB) with with flow-guiding features f_α, f_β and f_w. Right (ix–xi): group (fC) with size-varying pores f_q and shape-varying, thickening struts f_c. The reference structures f_1^ε for porosities $\varepsilon = 70, 80, 90\%$ are shown as horizontal lines. Colors correspond to the porosity of the respective reference structure used for modification, except f_c that has porosity $\varepsilon = 85\%$

Table 18.2 Characteristics of the investigated aluminum filtration process for the reference structure $f_1^{80\%}$, where the average interstitial velocity is given by $u = u_D/\varepsilon$ and the strut width is $d_s = 636\ \mu$m

Dimensionless number	Value	Definition
Reynolds number	1.85×10^1	$Re = \rho u d_s/\mu$
Forchheimer number	1.07×10^0	$Fo = \rho u_D k_1/(\mu k_2)$
Interception number	3.15×10^{-2}	$d_P^* = d_P/d_s$
Stokes number	1.69×10^{-3}	$St = \rho_P d_P^2 u/(18\mu d_s)$
Gravitational number	4.80×10^{-2}	$N_G = (\rho_P - \rho)\, d_P^2 g/18\mu u$

the total pressure drop losses [9, 22]. The process is assumed with an aluminum density of $\rho = 2356\ \text{kg} \cdot \text{m}^{-3}$ at a temperature of 730°C, superficial velocity of $u_D = 1\ \text{cm} \cdot \text{s}^{-1}$, dynamic viscosity $\mu = 1.01 \cdot 10^{-3}\ \text{Pa} \cdot \text{s}^{-1}$ and Reynolds number $Re = 18.5$ as defined in Table 18.2.

Application of LITE-QA In-Situ Data Preparation

The data sets are generated using GLATE data compression for the flow field u, v, w, and data indices u, M, Q as described in Sect. 18.3.1. This choice of data indexing allows for the efficient retrieval of flow regions with backflow, high-velocity flow and vortex-like flow. GLATE compression is configured to bound the point-wise

maximum error at 1%. As process simulations lie within the steady state it suffices to store only one time step per simulation.

The in-situ data preparation pipeline is applied during the LBM simulations as described in Sect. 18.2. The data sets are compressed to 24.1% in average, where the flow fields u, v, w are compressed to 15.3% in average and the data indices for u, M, Q are compressed to 25.2%. Compared to the unsteady data set used for compression testing, as described in Sect. 18.2.3, the steady solutions are compressed stronger due to the lower Reynolds number [15]. The overall data size for the simulations of 84 computer-generated filters is reduced from $84 \times 2.5\text{GB} = 212.5\text{GB}$ to 51.1 GB including the additionally generated indices, i.e. the flow fields u, v, w account for 19.3 GB and the data indices u, M, Q for 31.8 GB.

18.4.3 Query-Driven Statistical Analyses of Melt Flow

Several statistical analyses of the melt flow can be performed by queries that only rely on the meta-information stored in the header of the LITE-QA compressed indices. As the data amounts stored in the header account for only a very small fraction of the total index size, data loading times are negligible [16].

Particularly, count queries, as described in Sect. 18.3.2, return the data inside the index header for an index variable, including the number of occurrences of each data value in the voxel grid. This data can be used to estimate various statistical quantities such as average, median, minimum, maximum, relative frequencies, and histograms. Following the approach of [24], also the hydraulic tortuosity of the melt flow can be calculated from the velocity distribution, accessible over index variable M, and the velocity distribution in principal flow direction, accessible over index variable u.

Effect of Filter Design on Flow Characteristics

Figure 18.16 shows the tortuosity and maximum velocity for all 84 filter designs. The maximum velocity $M_{\max}$ is presented normalized with respect to the superficial velocity u_D. As can be seen, flattened struts f_a with $a > 1$ and closed windows f_w substantially increase both maximum velocity and flow tortuosity with respect to the underlying reference structure f_1^{ε}. The only filters which decrease both quantities are the filters with drop-like strut shape f_{ab} and f_{ba}. Different arrangements of size-varying pores f_q only have marginal effects on the tortuosity, however, the maximum velocity increases substantially as the flow is passing through smaller pores with a higher porosity. The insertion of finger-like struts slightly reduces the maximum velocity for f_α and f_β while also the flow tortuosity is decreased indicating a smoothing effect on the flow. The filters f_c, that have porosity $\varepsilon = 85\%$ and implement a continuous variation of strut width and strut shape, lead to the opposite effect, i.e. the maximum velocity and the tortuosity are increased.

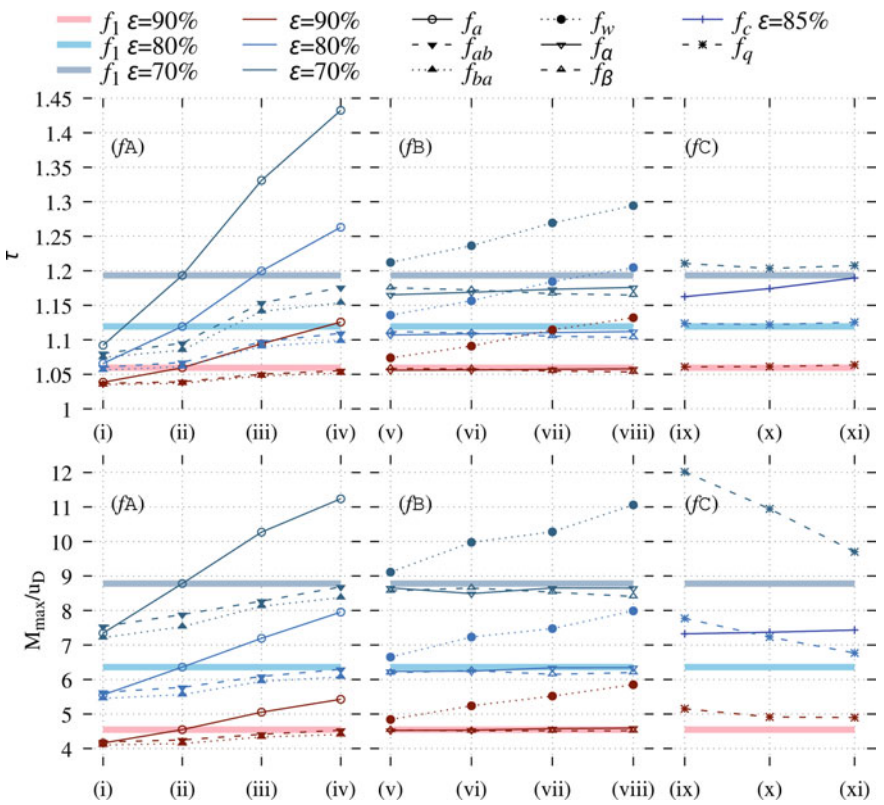


Fig. 18.16 Hydraulic tortuosity τ and normalized maximum velocity $M_{\max}/u_D$ computed using count queries on index variables M and u of the 84 filter prototypes. See Table 18.1 for filter parameterizations (i–xi). The reference structures f_1^ε for porosities $\varepsilon = 70, 80, 90\%$ are shown as horizontal lines. The filter modifications (i–xi) are colored according to the porosity class of the respective reference structure, except f_c that has a porosity of $\varepsilon = 85\%$

Statistical analyses of the flow field reveal global relationships between design modifications and flow characteristics across the complete fluid domain. In order to gain insights, how the design modifications influence the flow characteristics locally, additional visualizations of the melt flow are generated using the LITE-QA query mechanism as described next.

18.4.4 Query-Driven Comparative Visualization of Melt Flow

The LITE-QA query mechanism as described in Sect. 18.3 can be used for the automated creation of melt flow visualizations in specific areas of the filter that exhibit interesting flow characteristics. By visualizing the same filter regions for different fil-

ter variants, insights can be gained about the effects of geometric filter modifications on the melt flow.

In particular, comparative visualizations for the flow in specific filters from group (fA), (fB) and (fC) with modified strut shape, inserted flow-guiding features and varying pore geometries are generated. The locations for decompression and visualization are obtained from index queries, which are used to identify specific regions in the flow using range conditions, and from the filter skeleton directly, e.g. the geometric center of a strut or a pore. The visualizations are generated by rendering streamlines and iso-contour surfaces using the ParaView visualization pipeline described in Sect. 18.3.3.

Flow Regions in Computer-Generated Filters and CFF Sample

Figure 18.17 shows visualizations of vortices, which are generated as examples for characteristic flow in two datasets, i.e. the reference structure $f_1^{80\%}$ and a real CFF sample with 216 pores and 77.3% porosity obtained from computer tomography scanning in a previous study [9]. The CFF sample exhibits a pore density, specific surface area and strut width similar to the reference structure. However, the CFF sample also shows several defects, e.g., closed pore windows, cracked struts and deformed pores, due to imperfect manufacturing conditions. For each filter, the LITE-QA query mechanism, as shown in Fig. 18.10, is used to locate vortex regions, i.e. clusters of voxels with high vorticity, inside the flow fields. Five exemplary visualizations are generated for comparing the locations of high-vorticity clusters inside the filter geometries.

For both the computer-generated filter and the real CFF sample, vortices appear close to the upstream and downstream surfaces as well as in pore windows that are oriented in parallel to the bulk flow direction. As the upstream surface is exposed to the direct momentum of the flow, the velocity of vortices is higher as compared to vortices on the downstream surface or in pore windows. On the upstream, the flow crawls up on the surface of struts or closed windows before passing in another direction. On the downstream and in the pore windows, slower vortices appear in the slipstream of struts exhibiting less mass exchange with the bulk flow.

Deformation of Vortex-Like Flow Region

Figure 18.18 shows a vortex on the upstream surface of the reference structure $f_1^{80\%}$ and its deformation due to a different porosity $f_1^{90\%}$, a filter with elliptical strut shape f_a, and a filter with an upward-pointing finger-like strut f_β. After locating the vortex inside the reference structure f_1^{ε} using an index query, the visualization is regenerated using grid queries for the determined location for the four filter datasets. The camera is placed at three different angles showing the flow deformation due to the shape modifications in front, back and side view.

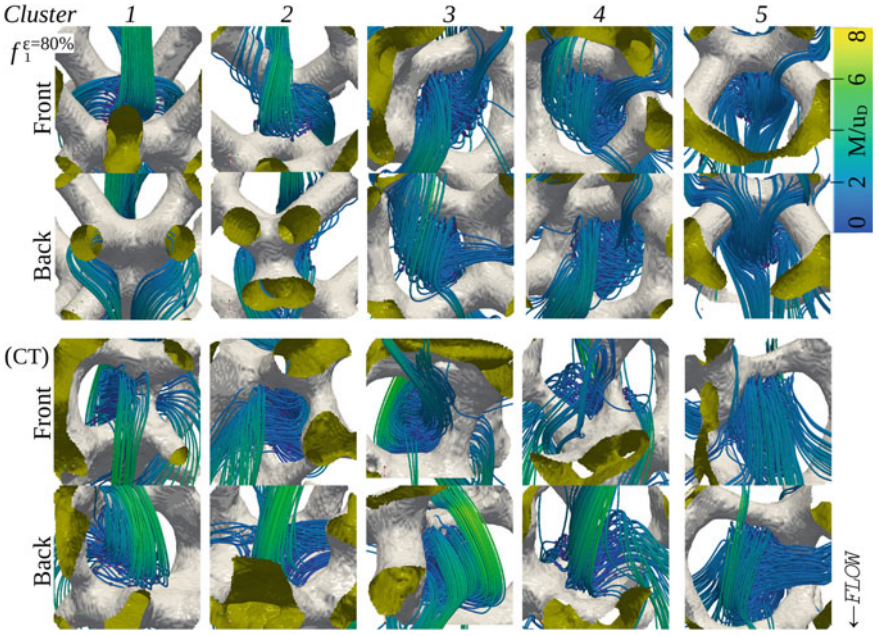


Fig. 18.17 Comparative visualization of vortices in reference structure $f_1^{80\%}$ and (CT) a real CFF sample with porosity of 77.3% obtained from computer tomography scanning. Both filters exhibit the same pore density, specific surface area and strut width. Fast vortices appear close to the upstream surface. Vortices close to the downstream surface and in pore windows are slower

The vortex is found next to a small closed pore window in the reference structure $f_1^{80\%}$, where the flow is redirected and crawling up the strut on the upstream surface. While for f_1^{ε} with $\varepsilon = 90\%$ the window is open, for $\varepsilon = 80\%$ the window is closed and the vortex region has a larger size. The flow splits in two opposing streams, which merge with the bulk flow while passing two opposing lateral pore windows. The flattened struts in f_a cause the closing of a second pore window, resulting in a stretch of the vortex region along the new upstream surface and an increase of the flow velocity. In contrast, the finger-like struts in f_β result in a decrease of the vortex volume, as the flow is redirected before it enters the vortex region. This local observation may explain the reduced tortuosity as observed in the global statistical analyses of Fig. 18.16.

Flow Around Struts with Modified Shape

Figure 18.19 shows the melt flow around a strut of the reference structure $f_1^{80\%}$ in comparison to filters f_a, f_{ab} and f_{ba} with modified strut shape (elliptical, drop-like, drop-like reversed). A grid query is used to decompress a cubic region of the flow field around the geometric center of a strut which is orthogonal to the bulk flow

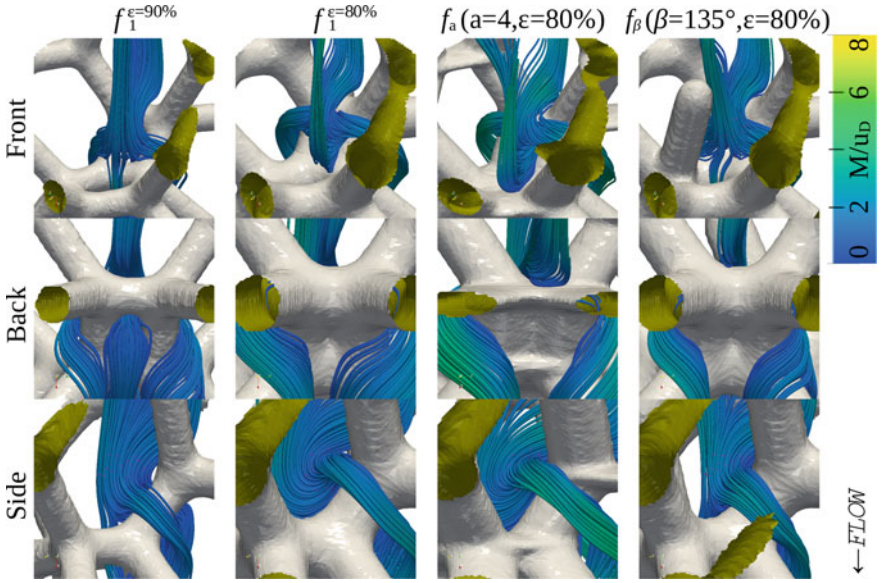


Fig. 18.18 Visualization of a vortex found by a LITE-QA query on the upstream surface of the reference structure $f_1^{80\%}$ and visualizations of the same region in three filter variants. All variants deform the vortex region

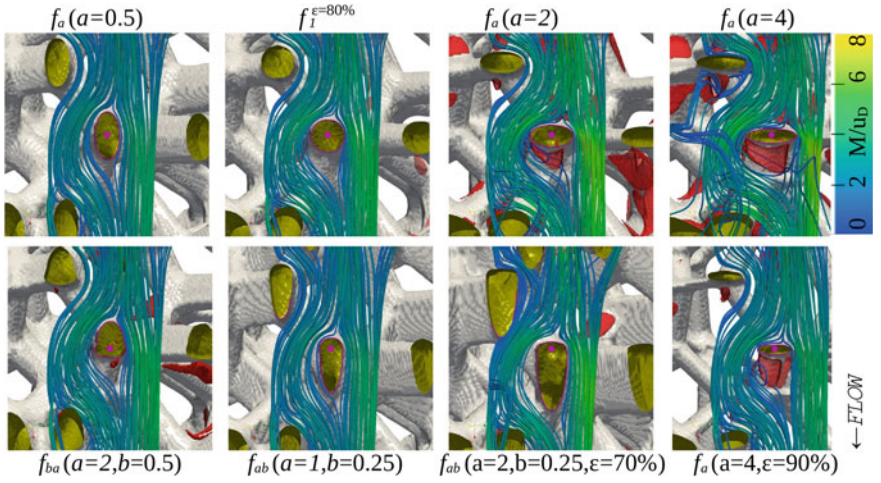


Fig. 18.19 Visualization of the flow around struts with modified shape for filters f_a (elliptical), f_{ab} (drop-like) and f_{ba} (drop-like reversed) in comparison to the reference structure f_1^{ε} (circular cross section). The iso-contour surfaces in red show fast backflow in the slipstream of the struts. All filters have a porosity of $\varepsilon = 80\%$, unless indicated otherwise

direction. The camera focuses on the strut center and is directed along the strut axis. The iso-contour surfaces in red highlight the fast backflow in the slipstream which is computed as the 5% quantile of the distribution for all values $u < 0$ in each data set.

As can be seen, for flat elliptical struts f_a with $a > 1$, the flow on the upstream surface of the struts exhibits increased velocity, as it is forced around the strut. The flow in the slipstream of the strut is mostly isolated from the bulk flow constituting a backflow region which increases the flow tortuosity. In contrast, for the elongated elliptical shape f_a ($a = 0.5$) and the drop-like strut shapes f_{ab}, the flow is smoothly directed around the strut, decreasing velocity and tortuosity. Interestingly, the reversed drop-like strut shapes f_{ba}, also exhibit decreased velocity and tortuosity. Due to the elongation of the upstream surface without sharp corners, the flow velocity on the upstream surface of f_{ba} is lower as compared to f_1^{ε}.

Flow Through Pores with Flow-Guiding Features

Figure 18.20 shows the melt flow through a pore of the reference structure f_1^{ε} in comparison to filters f_a, f_{ab}, f_{ba} with modified strut shape and filters f_α, f_β, f_w that have flow-guiding features inserted. A grid query is used to decompress a cubic region of the flow field around the geometric center of a pore. The camera is rotated towards one pore window and directed orthogonal to the bulk flow. The iso-contour surfaces in orange highlight the fast preferential flow, which is computed for each filter using the 85% threshold of the maximum velocity of $f_1^{80\%}$.

Consistent with the global statistical analyses indicating an increased velocity for flattened elliptical struts f_a with $a > 1$, the local visualization of f_a ($a = 2$) shows a higher-velocity flow (orange iso-contour) at the influx of the pore as compared to the reference structure f_1^{ε}. However, also consistent with the global analyses, when the porosity is increased from $\varepsilon = 80\%$ to $\varepsilon = 90\%$, even in case of very flat struts f_a ($a = 4$, $\varepsilon = 90\%$) velocity is still smaller as compared to the reference structure (absence of orange iso-contour). Similarly, drop-like strut shapes f_{ab} and f_{ba} decrease the velocity.

For the filters f_α and f_β with finger-like struts, the area of high velocity is deformed and the tortuosity is slightly lowered. The insertion of upward-pointing finger-like struts can cause a redirection of the flow through lateral windows of the pore. For filters f_w with closed windows, statistical analyses of global melt flow indicate an substantial increase of tortuosity and velocity. On the one hand, the flow is forced through a smaller number of pore windows, while on the other hand, regions behind closed windows are formed which are mostly isolated from the bulk flow. The latter effect can be observed for f_w ($w = 200$) in Fig. 18.20.

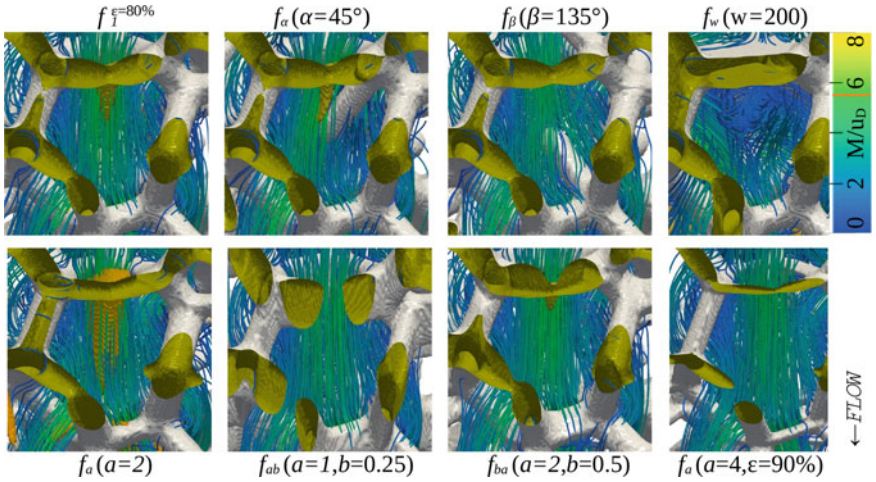


Fig. 18.20 Visualization of the flow through a pore of reference filter f_1^{ε}, filters f_a, f_{ab}, f_{ba} with modified strut cross section (elliptical, drop-like, drop-like reversed) and filters f_α, f_β, f_w with flow-guiding features (finger-like struts downwards, upwards, closed windows). Regions with high velocity are shown as iso-contour surfaces in orange. All shown filters have porosity $\varepsilon = 80\%$, except f_a $(a = 4)$ with $\varepsilon = 90\%$

Flow Through Size-Varying Pores

Figure 18.21 compares regions with backflow, high-velocity flow and high-vorticity flow for two filters with equal-sized pores and four filters with size-varying pores. The filters with equal-sized pores are the reference structure $f_1^{80\%}$ and the filter f_β $(\beta = 135°)$ with upward-pointing finger-like struts. The three filters f_q are laid out with pores of different diameters across the filter. In filter f_c $(c = 2)$, the effective size of pore cavities decreases from top to bottom due to the continuous transition from thin to thick struts.

A grid query is used to decompress a grid slice with a depth of 128 voxels for one complete filter height and width showing the flow field along the bulk flow direction. Regions with strong backflow, high velocity, and high vorticity in the flow field are found with range queries as described in Sect. 18.3.1 and rendered as iso-contour surfaces. Additionally, streamlines are seeded inside the flow field regions and traced both forwards and backwards along the complete height of the filter.

For filter f_c $(c = 2)$ with a continuous transition from thin to thick struts along the filter depth, the flow is smoother in the top part, as compared to the bottom, where the flow is forced into smaller pores. Almost all backflow, high-velocity and high-vorticity regions are concentrated in the lower half of the filter.

In the filters f_q $(q = 200)$ and f_q $(q = 265)$ with continuous pore size transition along the filter height, the flow is forced into regions of small pores, i.e. middle region of f_q $(q = 200)$ and bottom region of f_q $(q = 265)$, where the fast preferential flow and vortex-like flow concentrates. In contrast, for the filter f_q $(q = 320)$ with an

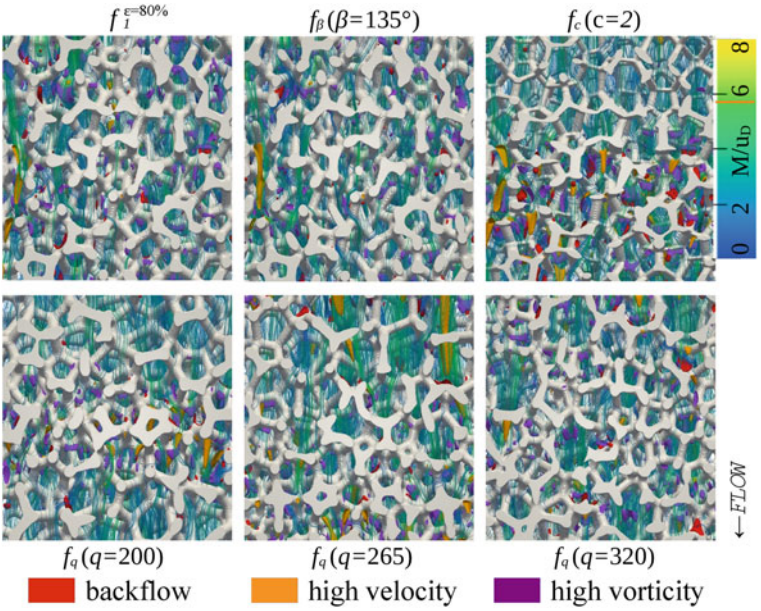


Fig. 18.21 Visualizations of regions in the flow field for filters f_q with size-varying pores and f_c with shape-varying struts in comparison to f_β with upward-pointing finger-like struts and the reference structure f_1^ε. All shown filters have porosity $\varepsilon = 80\%$

alternating layout of large and small pores, the regions of fast flow shrink in size as the flow can escape into larger pores more easily. Areas of high vorticity are scattered across the whole filter.

The filter f_c with increasingly thicker struts shows the largest amount of backflow areas among the filters compared in Fig. 18.21 which explains its high tortuosity. In contrast to f_c, the filters f_q do not increase the amount of fast backflow regions. Similar to f_q, the flow in the filters f_β with additional finger-like struts becomes smoother, due to the flow redirection through lateral pore windows as shown in Fig. 18.20, which can effects the size of the backflow regions.

18.4.5 Evaluation of Filtration Performance

The filtration performance of the filters is evaluated based on the filtration coefficient λ and pressure drop p' which are both directly determined from the LBM simulations using the Darcy-Forchheimer law and the conventional depth filtration law.

The Darcy-Forchheimer law describes a simplified relationship between the pressure drop $p' = \mathrm{d}p/\mathrm{d}x$ and the superficial velocity u_D for flow through porous media given by

$$\frac{\mathrm{d}p}{\mathrm{d}x} = -\frac{\mu}{k_1} u_D - \frac{\rho}{k_2} |u_D| u_D.$$

The relationship depends on the Darcy permeability k_1 and the Forchheimer coefficients k_2, which are required to characterize the contribution of the viscous and inertial losses to the overall pressure drop during the process simulations. The parameters are estimated using a computationally efficient approach, which permits comparative assessment with a sufficient accuracy [9].

The filtration coefficient λ is obtained by fitting the simulated inclusion distribution with the conventional depth filtration law given by

$$\frac{\mathrm{d}c}{\mathrm{d}x} = -\lambda c.$$

The inclusion distribution is obtained by tracking 10^5 spherical inclusions with diameter $d_\mathrm{P} = 20$ μm and density $\rho_\mathrm{P} = 3900$ kg $\cdot$ m^{-3} randomly inserted on the filter inlet. The particle transport, characterized as shown in Table 18.2, is simulated coupled with the LBM under consideration of the drag force for 3 residence times, i.e. an effective filter depth of 52.6 mm [9].

Effect of Filter Design on Filtration Process

Figure 18.22 shows the filtration coefficient λ and pressure drop p' obtained from simulations of all 84 filters. See Table 18.1 for geometric filter modifications and parameterizations (i–xi). An improvement of filtration performance is assessed when the filtration efficiency, measured by λ, increases without substantial impact on the pressure drop p' as compared to the reference structure f_1^ε of corresponding porosity.

From all 84 filters, the filter f_{ab} (iv) with drop-like struts is the only filter which both increases the filtration coefficient λ and at the same time reduces the pressure drop p' with respect to the reference structure f_1^ε with corresponding porosity. Interestingly, the filter f_a (iv) with very flat elliptical strut shapes $a = 4$ and with porosity $\varepsilon = 90\%$ outperforms the reference structure $f_1^{80\%}$ with a higher λ and a lower p'.

All filters yield an increase of λ w.r.t. the respective reference structure f_1^ε with same porosity except f_a (i), i.e. elongated elliptical struts, f_{ba} (i–iv), i.e. inverse drop-like struts, and f_q (ix), i.e. size-varying pores with large-small-large transition. However, a disproportionate increase of the pressure drop p' is observed for f_a (iii–iv), i.e. flat elliptical struts, and f_w (vii–viii), i.e. 150 or more closed windows.

The pressure drop increases only moderately for the filters with finger-like struts f_α and f_β and filters with size-varying pores f_q. The filters f_c with transition from thin to thick struts and porosity $\varepsilon = 85\%$ increase the filtration coefficient λ, while only moderately increasing p' with respect to the reference structure $f_1^{80\%}$.

Concerning the symmetric shape modifications, the following conclusions can be drawn:

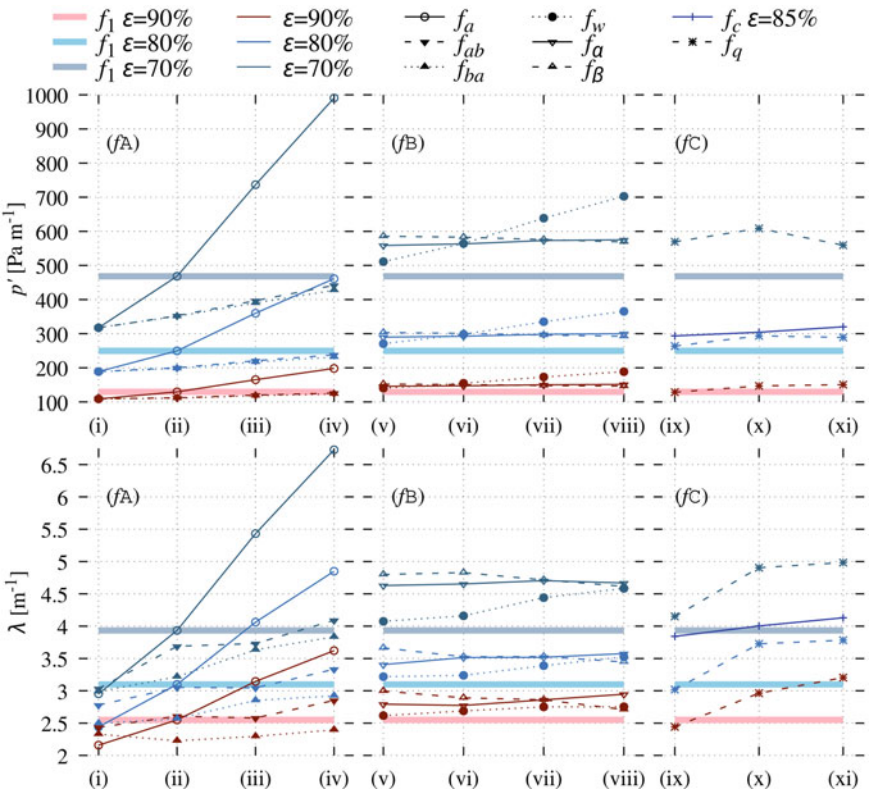


Fig. 18.22 Pressure drop p' and filtration coefficient λ of 84 filter prototypes. See Table 18.1 for geometric filter modifications and parameterizations (i–xi). Group (fA): strut shape modifications f_a, f_{ab}, f_{ba} with parameterizations (i–iv). Group (fB): insertion of flow-guiding features f_α, f_β, f_w with parameterizations (v–viii). Group (fC): varying pore size f_q and strut shape f_c within the filter with parameterizations (ix–xi). The reference structures f_1^ε for porosities $\varepsilon = 70, 80, 90\%$ are shown as horizontal lines. The filters (i–xi) are shown in colors matching the porosity class of the corresponding reference structure, except f_c with fixed porosity $\varepsilon = 85\%$

- filters with flat elliptical struts f_a ($a > 1$) outperform filters with vertically elongated struts f_a ($a < 1$)
- filters with drop-like struts f_{ab} outperform filters with reversed drop-like struts f_{ba}
- filters with upward-pointing finger-like struts f_β outperform filters with downward-pointing finger-like struts f_α.

Selection of Top-Performing Filters

Based on the data shown in Fig. 18.22, a set of top-performing modified filters is determined by comparison to the filtration coefficient λ of the reference structure $f_1^{80\%}$ and the pressure drop p'_{CFF} of the 30 PPI CFF sample shown in Fig. 18.17.

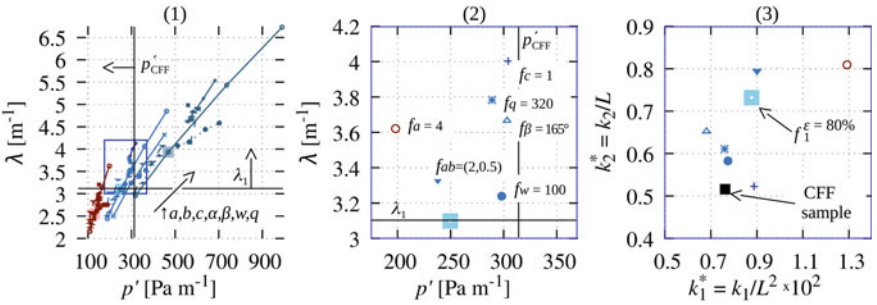


Fig. 18.23 Correlation of pressure drop p' and filtration coefficient λ for (1) all filters and, zooming in into the blue rectangular area and filtering out less well performing parameterizations, (2) the six top-performing filters whose filtration coefficient λ exceeds that of $f_1^{80\%}$ and whose pressure drop is lower than the pressure drop p'_{CFF} of the CFF sample. The six filters are f_c, f_q, f_β, f_a, f_{ab} and f_w with parameterizations as shown in the figure and summarized in Table 18.3. (3) Correlation of dimensionless viscous and inertial permeability k_1^* and k_2^* for the six top-performing filters

Figure 18.23 (1) shows the correlation of the pressure drop p' and the filtration coefficient λ for all filters. The vertical straight line shows the pressure drop of the CFF sample p'_{CFF}.

In order to select a group of filters with increased performance, first, all filters are considered whose filtration coefficient λ is higher as compared to $f_1^{80\%}$ and whose pressure drop p' is lower than the pressure drop p'_{CFF} of the CFF sample. Second, from the considered filters, for each modification one filter is selected, which has the highest filtration coefficient. Figure 18.23 (2) shows six top-performing filters that are also summarized in Table 18.3. The top-performing filters increase λ by 4.5–29% w.r.t. $f_1^{80\%}$. The filters f_a ($a = 4, \varepsilon = 90\%$) and f_{ab} ($a = 2, b = 0.5, \varepsilon = 80\%$) decrease p' by 4.7–20.6%, while the other four top-performing filters increase p' by 15.9–21.9%.

Figure 18.23 (3) shows the viscous and inertial permeability k_1 and k_2 for the top-performing filters. As can be seen, the filters f_a and f_{ab} increase both permeabilities, the filter f_c only decreases the inertial permeability and the remaining filters f_w, f_β and f_q decrease both k_1 and k_2. Notably, the permeabilities of f_β and f_q decrease only slightly although their cumulative strut lengths, due to new struts and many smaller pores, lead to a considerable increase of the surface area by 13.3% and, resp., 9.6% as compared to the reference structure $f_1^{80\%}$.

18.4.6 Discussion

This virtual prototyping study has investigated eight basic modifications of a reference geometry with several parameterizations each, see Table 18.1, yielding 84 virtual filter prototypes in total. For six of the eight modifications, Table 18.3 shows the filter

Table 18.3 Top-performing filters whose filtration coefficient λ exceeds that of f_1^{ε} and whose pressure drop is lower than the pressure drop p'_{CFF} of the CFF sample. Filters are ranked by λ

Ranked Modification	Filter	$\varepsilon^{[\%]}$	$p'^{[\mathrm{Pa\cdot s^{-1}}]}$	$\lambda^{[\mathrm{m^{-1}}]}$
1. Shape-varying, thickening struts	f_c (x)$^{c=1}$	85	304.2	4.00
2. Size-varying pores, alternating	f_q (xi)$^{q=320}$	80	289.0	3.78
3. Finger-like struts upwards	f_β (v)$^{\beta=135°}$	80	303.2	3.66
4. Flattened strut cross section	f_a (iv)$^{a=4}$	90	198.1	3.62
5. Drop-like strut cross section	f_{ab} (iv)$^{a=2}_{b=0.5}$	80	237.9	3.33
6. Randomly closed windows	f_w (vi)$^{w=100}$	80	298.1	3.24
Reference structure	f_1^{ε}	80	249.6	3.10

with the best performing parameterization. Two modifications are not included as they perform less well than their counterparts: Upward-pointing finger-like struts outperform downward-pointing finger-like struts and drop-like strut shapes outperform struts with inverse drop-like shape.

Table 18.3 ranks the modifications according to the filtration coefficient λ. The four best-ranked modifications yield an increase of the filtration coefficient λ by 16.7–29%, while the other two modifications, i.e. drop-like struts and closed windows, yield substantially lower improvements of 4–6%.

Nonetheless, also the latter modifications show interesting effects when not only the filtration coefficient λ is considered. Drop-like deformations of the strut shape can be parameterized such that pressure drop is substantially lower than in the reference structure without noteworthy decline of λ, e.g. f_{ab} (ii) and (iii) in Fig. 18.22. The deliberate closing of pore windows, which are oriented nearly parallel with respect to the bulk flow, hinders the formation of large slow vortices in the slipstream of struts, as shown in Fig. 18.17. Such observations could inform, e.g., the design of filters with more elaborate window-closing policies as compared to the randomized window closing of this study.

Increased complexity of filter manufacturing is a further issue for some of the modifications. Filters with very thin struts or articulate flat elliptical or even drop-like strut cross sections require a higher fidelity for 3D-printing of the polyurethane templates needed for filter replication. Coating of struts in an evenly manner also becomes more challenging for non-circular strut cross sections.

Positive results regarding filtration performance but without the potential drawbacks concerning their manufacturability are obtained for filter f_β with upward-

pointing finger-like struts and the filter f_q with an alternating pattern of small and large pores.

The finger-like struts in f_β in the parametrization shown in Table 18.3 increase filtration efficiency by 18% while only marginally reducing the maximum velocity and the hydraulic tortuosity of the flow. As illustrated in Fig. 18.20, upward-pointing finger-like struts redirect flow through lateral pore windows. This decreases the size of vortices in the slipstream behind struts inside pore windows that are approximately parallel to the bulk flow. Overall, the insertion of upward-pointing finger-like struts substantially increases filtration efficiency, ranked at third place in Table 18.3, while also smoothing the flow.

In f_q of Table 18.3, clusters of nine smaller pores and single larger pores are arranged alternately, inducing a variation of locally increased and decreased porosity in an interlaced manner. As the flow can escape into nearby larger pores, the maximum flow velocity only increases slightly with respect to the reference structure, while the tortuosity remains unchanged. In contrast, in the other investigated layouts of size-varying pores the flow must pass through distinct layers of small pores which causes larger high-velocity regions and higher velocity peaks as shown in Fig. 18.21. The alternating layout of large pores and clusters of smaller pores yields an improvement of λ by 22% as compared to the reference filter. This is the second-best value of all investigated filters, while avoiding potential manufacturing complexities.

The filter f_c, which shows the best improvement of the filtration coefficient λ by 29% within this study, has thin struts in its upper part and further integrates an elliptical flattening. Possibly, other techniques than replication of 3D-printed polyurethane templates are needed to manufacture this filter with high fidelity.

18.5 Conclusions

The advent of additive manufacturing has vastly increased the design space of filters for metal melt filtration. Novel filter geometries can be conceived that vary, e.g., in strut shape and pore size distribution across the filter or that add flow-guiding or surface-increasing features to the pores. Furthermore, all such modifications can be parameterized in numerous ways. In order to assess this large space of possible filter designs before they are actually manufactured, a virtual prototyping workflow based on HPC simulations was presented.

A major challenge are the large data volumes produced even by single CFD simulations of metal melt flow and even more so for virtual prototyping settings where many filter designs need to be evaluated. The large data problem is particularly severe during the analysis phase, where simulation results are assessed outside of the HPC environment, using workstations with much lower computing resources. The LITE-QA framework was developed to address this large data challenge. Integrated into code executed in the HPC environment, the resulting simulation data is compressed and indexed. The data reduction rates of combined data compression and compressed indexing significantly improve on other state-of-the-art methods. As support for the

analysis phase, LITE-QA offers query-driven access to the compressed and indexed data. This adds search engine-like capabilities to data analysis tasks and allows, e.g., for focused visualizations of filter regions that meet the search criteria of the analyst, such as areas with high velocity, vorticity or backflow. Crucially, only very small parts of the simulation data, i.e. those parts that meet the analyst's search criteria, need to be loaded from disk. In this way, interactive data analyses are enabled even for very large scientific data sets.

Furthermore, a virtual prototyping study including 84 new filter designs for metal-melt filtration was conducted. The filter designs are created by geometric modification of a reference model that emulates the topology, pore density and strut width of conventional foam filters used for aluminum filtration. Eight basic geometric modifications were investigated that alter the strut shape, add flow-guiding features or vary pore and strut size within a filter. The modification parameters were systematically varied and filter geometries were generated for three porosities. An evaluation of filter performance based on HPC simulations and LITE-QA data management identified the six best-performing filter designs with higher filtration coefficient than the reference filter while only having a moderate effect on melt flow pressure. Local visualizations of regions with interesting flow properties such as backflow, high velocity or high vorticity provide further insights on how the investigated modifications influence the melt flow. The largest improvement of the filtration coefficient by 29% was achieved for a filter with thin struts at the top that increasingly become thicker towards the bottom. The thin struts however require high-fidelity 3D-printing technology. The next-best performing filter designs do not include particularly thin struts or struts of articulate shapes, thus placing lower demands on additive manufacturing processing: A layout where large pores alternate with clusters of small pores with 22% improvement and the insertion of upward-pointing finger-like struts with 18% improvement of the filtration efficiency. These best-performing virtual filter prototypes can be seen as promising candidates for additive manufacturing and further testing as physical prototypes.

Acknowledgements The authors would like to thank the German Research Foundation (DFG) for supporting this investigation carried out by the subproject S02 of the Collaborative Research Centre 920, "Multi-Functional Filters for Metal Melt Filtration—A Contribution towards Zero Defect Materials" (Project-ID 169148856). The authors are also grateful to the Centre for Information Services and High Performance Computing (ZIH) at TU Dresden for providing its facilities for high throughput calculations.

References

1. T. Wetzig, M. Neumann, M. Schwarz, et al., Adv. Eng. Mater. **24**, 2100777 (2022). https://doi.org/10.1002/adem.202100777
2. B. Bock-Seefeld, T. Wetzig, J. Hubálková, et al., Adv. Eng. Mater. **24**, 2100655 (2022). https://doi.org/10.1002/adem.202100655

3. A. Herdering, M. Abendroth, P. Gehre, et al., Ceram. Int. **45**, 153–159 (2019). https://doi.org/10.1016/j.ceramint.2018.09.146
4. A. Herdering, J. Hubálková, M. Abendroth, et al., Int. Ceram. Rev. **68**, 30–37 (2019). https://doi.org/10.1007/s42411-019-0013-z
5. M. Abendroth, E. Werzner, C. Settgast, S. Ray, Adv. Eng. Mater. **19**, 1700080 (2017). https://doi.org/10.1002/adem.201700080
6. E. Werzner, M.A.A. Mendes, S. Ray, D. Trimis, Adv. Eng. Mater. **15**, 1307–1314 (2013). https://doi.org/10.1002/adem.201300465
7. H. Lehmann, E. Werzner, M.A.A. Mendes, et al., Adv. Eng. Mater. **15**, 1260–1269 (2013). https://doi.org/10.1002/adem.201300129
8. C. Demuth, E. Werzner, M.A.A. Mendes, et al., Adv. Eng. Mater. **19**, 1700238 (2017). https://doi.org/10.1002/adem.201700238
9. E. Werzner, M. Abendroth, C. Demuth, et al., Adv. Eng. Mater. **19**, 1700240 (2017). https://doi.org/10.1002/adem.201700240
10. J. Iverson, C. Kamath, G. Karypis, Fast and effective lossy compression algorithms for scientific datasets, in *Euro-par 2012 Parallel Processing* (Springer, 2012), pp. 843–856
11. S. Lakshminarasimhan, N. Shah, S. Ethier, et al., Concur. Comput. Pract. Exp. **25**, 524–540 (2013). https://doi.org/10.1002/cpe.2887
12. P. Lindstrom, Fixed-rate compressed floating-point arrays. IEEE Trans. Visual Comput. Graph. **20**, 2674–2683 (2014)
13. S. Di, F. Cappello, Fast error-bounded lossy HPC data compression with SZ, in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2016), pp. 730–739
14. H. Lehmann, B. Jung, In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations, in *4th IEEE Symposium on Large Data Analysis and Visualization (LDAV)* (2014), pp. 51–58
15. H. Lehmann, E. Werzner, C. Degenkolb, Optimizing in-situ data compression for large-scale scientific simulations, in *Proceedings of the 24th High Performance Computing Symposium*, Society for Computer Simulation International, San Diego, CA, USA, pp. 5:1–5:8 (2016)
16. H. Lehmann, E. Werzner, C. Demuth, et al., Efficient visualization of large-scale metal melt flow simulations using lossy in-situ tabular encoding for query-driven analytics, in *2018 IEEE International Conference on Computational Science and Engineering (CSE)* (2018), pp. 123–131
17. H. Lehmann, B. Jung, Temporal in-situ compression of scientific floating point data with t-GLATE, in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (2018), pp. 1386–1391
18. H. Lehmann, Temporal lossy in-situ compression for computational fluid dynamics simulations. PhD thesis, Faculty for Mathematics & Informatics, Technical University Bergakademie Freiberg, 2018
19. D. Lemire, L. Boytsov, N. Kurz, Softw. Pract. Exp. **46**, 723–749 (2016). https://doi.org/10.1002/spe.2326
20. K. Wu, S. Ahern, E. W. Bethel, et al., J. Phys. Conf. Ser. **180**, 012053 (2009). https://doi.org/10.1088/1742-6596/180/1/012053
21. A. Asad, H. Lehmann, B. Jung, R. Schwarze, Adv. Eng. Mater. **24**, 2100753 (2022). https://doi.org/10.1002/adem.202100753
22. H. Lehmann, E. Werzner, A. Malik, et al., Adv. Eng. Mater. 2100878 (2021). https://doi.org/10.1002/adem.202100878
23. C. Voigt, E. Jäckel, F. Taina, et al., Metall. Mater. Trans. B **48**, 497–505 (2017). https://doi.org/10.1007/s11663-016-0869-5
24. A. Duda, Z. Koza, M. Matyka, Physi. Rev. E **84**, 036319 (2011). https://doi.org/10.1103/PhysRevE.84.036319