Large Scale Lattice-Boltzmann Simulations: Computational Methods and Applications
Kandhai, B.D.

Citation for published version (APA):
Chapter 6

Lattice-Boltzmann and Finite-Element Simulations of Fluid Flow in a SMRX Mixer

6.1 Introduction

In the previous chapters we have studied various aspects of the lattice-BGK method. First, we presented some new results on the accuracy of the boundary conditions. Next, we proposed and validated a new relaxation technique to reduce the number of time-steps to reach steady state. Furthermore, efficient parallelization of the LBM scheme including load balancing methodologies based on the orthogonal recursive bisection technique were applied to a number of flow problems. Finally we studied grid refinement for 2D lattice-Boltzmann simulations.

In the remainder of the thesis we will focus our attention on applications. In this context our main interest is fluid flow in complex geometries focusing on fibrous porous media. Flow in fibrous media has many realistic applications as will become apparent in the next chapters. Before going into the details of the hydrodynamic properties of flow through fibrous media, we report in this chapter a rigorous comparison between the finite-element and the lattice-Boltzmann method applied to a realistic test case with complex geometries.

We studied 3D fluid flow in a SMRX static mixer. A SMRX static mixer is a piece of equipment composed of cylindrical pipes with excellent mixing performance and it is used as a highly efficient chemical reactor for viscous systems like poly-

*This chapter is based on the following publications:


Lattice-Boltzmann and Finite-Element Simulations of Fluid Flow in a SMRX Mixer

mers [72]. The complex geometry of this mixer makes 3D simulations certainly non-trivial. We have chosen for this problem as a benchmark case, as it is one of the very few applications related to flow in complex geometries with well documented results obtained by state-of-the-art traditional numerical methods and experimental measurements.

It is important to notice that our comparative study between LBM and other traditional numerical methods is not the first one since the introduction of LBM. In the literature several studies can be found in which some kind of validation of the LBM results is included. However, a large number of LBM simulations have been validated only on a qualitative level. This is partly due to the fact that these applications were highly complicated and have hardly been studied using traditional methods. To convince people not familiar with LBM, of its effectiveness, rigorous comparative studies with standard state-of-the-art traditional methods such as finite-element methods are extremely important. This is especially necessary because the Lattice-Boltzmann method seems to be, at the first glance, too simple to be effective for realistic problems. The need for comparative studies of LBM has also been noted by Luo in Ref. [51]:

"Since LBE is a newly developed method and is still in its infancy, most numerical results obtained with this technique are qualitative in nature. (See, for example, results in the collective works edited by Doolen [73] and the recent reviews by Benzi et al. [28] and by Qian et al. [52]) High precision results of numerical simulations by the LBE method (e.g. work by Hou et al. [24]) are rarely available. Also, since the LBE method can handle complex boundary geometry easily, many LBE simulations have no counterpart produced with traditional numerical methods with which to compare. Thus, there is a pressing need for high quality numerical benchmarks using the LBE method which can be compared with the results obtained by traditional methods or by experiments."

The most notable comparative efforts are that of Hou et al. [24] and that of Luo [51]. Our study is different from theirs in two respects. First of all we study flow in quite complex geometries in 3D and we additionally compare the computational requirements of FEM and LBM in more detail. The studies of Hou et al. [24] and that of Luo [51] have focussed on simple 2D geometries, e.g. the cavity flow problem and the backward facing step. However, their studies are certainly relevant as they have studied the complex behavior that may emerge as the Reynolds number is increased. We restrict to the laminar flow regime as our main interest is in low Reynolds number flow.

A comparison of a lattice-Boltzmann method and a finite-element method, applied to the complex SMRX test case, will be presented from a theoretical, methodological and experimental point of view.

### 6.2 The Static Mixer Reactor

The static mixer, a rather new technology introduced 15–20 years ago, has gained more and more in popularity within the chemical industry over recent years [72]. It consists of specially designed stationary obstacles inserted in a
pip in order to promote mixing of fluid streams flowing through it. Its mixing mechanism relies on splitting, stretching, reordering and recombination of the incoming fluid streams. Compared to traditional mechanical mixing equipment, the static mixer offers several advantages: it has low maintenance and operating costs, low space requirements and no moving parts.

Nowadays, over 200 different designs are available on the market. They are widely used in all kinds of chemical processes. Among those are gas/liquid reactors, polymerization reactors, blending units, heat exchangers and, to summarize, devices for promoting homogenization in concentration, temperature or velocity (e.g. for uniform residence time). Among all the static mixer designs, one of the most complex is the SMRX, an SMR-type mixer manufactured by Sulzer Chemtech Ltd. and used mainly in polymerization reactors [56].

![Figure 6.1: SMRX geometry.](image)

It consists of a series of solid crossing tubes, placed inside a rectangular tubular reactor (see Figure 6.1). In this work, we will focus our attention on this static mixer and use the experimental results of van Dijck et al. [74] as a validation for the numerical investigation.

Due to rather complex flows and geometries, only few 3D numerical simulations in static mixers were performed in the past. The first one was in 1992 on a Kenics static mixer by Gyenis and Blickle [75] using stochastic simulations of steady state particle flows. The simulations that followed were all based on macroscopic momentum balance methods like finite-element, finite difference, or finite volume methods. Using the finite volume package FLUENT™, Bakker and LaRoche [76] also studied flow and mixing in a Kenics static mixer. Later on, using RheoTek's finite-element program POLY3D™, Bertrand et al. [77] looked
at residence time distribution in Ross Engineering’s LPD and ISG static mixers for Newtonian and power-law fluids. In the same way, Tanguy et al. [78] and Mickaily-Huber et al. [56] investigated flow and mixing in the complex SMRX static mixer. Recently, Avalosse and Crochet [79] studied mixing of Newtonian and power-law fluids in a series of Kenics mixers using a finite-element method. A close review of these previous articles clearly shows that the major problem of the momentum balance methods is to generate a satisfactory body-fitted grid or mesh that does not require too much memory. Briefly stated, they have high memory requirements per grid element and as a consequence are quite rapidly limited by the available computer resources. Furthermore, complex geometries like the SMRX can lead to numerical inaccuracy due to poor body-fitted meshes, especially for the pressure field (see section III).

In this chapter we intend to show that the lattice-Boltzmann method [28, 22, 80, 12] can be used successfully as an alternative approach to traditional macroscopic momentum balance methods for computing fluid flows in complex geometries such as the SMRX static mixer.

### 6.3 The Galerkin Finite-Element method

In this section we present the basic fundamentals of the finite-element method (FEM). A description of the lattice-Boltzmann method (LBM) can be found in the introduction of this thesis. The aim is to underline the conceptual differences between both methods. For more exhaustive reviews, we refer to the book of Cuvelier et al. [81] for FEM and Refs. [28, 22, 80, 12, 16] for the lattice-Boltzmann method.

On a macroscopic scale, flow of an incompressible fluid in a given geometry \( \Omega \) can be described by the classical Navier-Stokes equations [2]

\[
\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) + \nabla \cdot \mathbf{\sigma} + \nabla p = \mathbf{f},
\]

\[
\nabla \cdot \mathbf{v} = 0
\]

where \( \rho \) is the fluid density, \( \mathbf{v} \) is the velocity, \( p \) is the pressure and \( \mathbf{f} \) is a body force, e.g. the gravitational force. The stress tensor \( \mathbf{\sigma} \) is a function of the rate-of-strain tensor \( \dot{\gamma} = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \), through a rheological model:

\[
\mathbf{\sigma} = -2\eta \dot{\gamma}
\]

where, depending on the rheological model chosen, the fluid viscosity \( \eta \) could be a function of \( |\dot{\gamma}| \) (non-Newtonian models) or simply equals to the Newtonian viscosity \( \mu \).

The finite-element method for solving fluid flow dynamics makes use of variational calculus which allows to transform a set of partial differential equations (in our case, the Navier-Stokes equations) into a system of linear algebraic
6.3 The Galerkin Finite-Element method

The Galerkin Finite-Element method is a numerical technique used to solve partial differential equations. For the sake of simplification and concision, let us consider the following differential equation

$$Lu = f$$  

(6.4)

where $u$ is the unknown function of the problem (in our case, there are 4 degrees of freedom: the three components of the velocity and the pressure) and $L$ represents the differential operator related to the partial differential equation treated.

The first step in FEM is to multiply the differential equation by a test function $\psi$ chosen in an appropriate vectorial space and to integrate the resulting equation in the studied domain $\Omega$. We subsequently obtain the following equation

$$\int_{\Omega} \psi (Lu - f) d\Omega = 0.$$  

(6.5)

![Diagram of discretization of the domain $\Omega$ by a set of triangular elements $\Omega_e$.](image)

Figure 6.2: Discretization of the domain $\Omega$ by a set of triangular elements $\Omega_e$.

In order to solve equation 6.5 the domain $\Omega$ is split up in many sub-domains $\Omega_e$ called finite elements (Figure 6.2). Each element is made of geometrical nodes and is constructed in such a way that it fits as good as possible the geometry $\Omega$ and respects a certain continuity with its neighbors. The element set is called a mesh. Normally, the quality of the solution is improved as the mesh resolution is increased. The repartition and the number of nodes determine the type of element to be used. Briefly stated, there are triangular and quadrilateral elements in 2D and tetrahedral and hexahedral elements in 3D. Due to their flexibility, triangular and tetrahedral elements are more suitable to fit non-trivial geometries. Especially in 3D, mesh creation requires the use of automatic and robust methods.
mesh generators. If we now apply equation 6.5 to each element $Q_e$, we get the following equation,

$$\sum_{i=1}^{ne} \int_{Q_e} \psi_i (Lu - f) dQ_e = 0,$$

(6.6)

where $ne$ is the total number of elements. Then, in each domain element, an approximation of the unknown function is introduced in the following form:

$$u = \sum_{j=1}^{nn} \phi_j u_j$$

(6.7)

where $nn$ is the number of nodes in the element, $u_j$ is the value of the unknown function at node $j$ and $\phi_j$ is called a basis function. A Lagrange polynomial is usually taken as basis function.

Combining equations 6.6 and 6.7 and using the Galerkin method, which consists simply of taking $\psi_j = \psi_i$, we obtain

$$\sum_{i=1}^{ne} \sum_{j=1}^{nn} (u_j \int_{Q_e} \phi_j \phi_i dQ_e - \int_{Q_e} \phi_i L dQ_e) = 0$$

(6.8)

In order to lower the order of differentiation of the first integral in the previous equation, integration by parts (Green’s formula) can be performed. This usually leads to an expression where Neumann type boundary conditions can be introduced [81]. It is important to note that to avoid the definition of different sets of basis functions for each element, each integral is actually first evaluated in a reference element with fixed geometry.

Finally, equation 6.8 leads to an overall sparse matrix system of the form

$$Au_j = b$$

(6.9)

which can be solved using a LU decomposition or by means of iterative algorithms. The latter usually requires preconditioning.

In the context of the Navier-Stokes equations, it is important to introduce a few extensions. Since the pressure is related to the velocity, the Uzawa algorithm [82] can be implemented in order to decouple the pressure and the velocity. This iterative algorithm is based on the penalization of the incompressibility constraint through,

$$p^{l+1} = p^l + \alpha \nabla \cdot v^{l+1}$$

(6.10)

where $\alpha$ is the descent parameter, $p^l$ and $v^l$ are the pressure and the velocity at iteration $l$, respectively.

The inertial term of the Navier-Stokes equations is non-linear. For low Reynolds number, it can be added to the right hand side of the matrix system. It is then evaluated by means of the velocity solution found at the previous iteration using the Picard algorithm [9]. When the Reynolds number is rather high (laminar-transient regime), linearization of the Navier-Stokes equations using the Newton-Raphson algorithm is required for convergence. For non-Newtonian fluid flows, the problem becomes non-linear in shear rate. In this
case, a Lagrangian augmented method is used that allows to decouple the treatment of the non-linearity from the pressure and the velocity fields [83]. It is obvious that FEM and LBM are two very different numerical approaches. FEM is based on approximations of flow equations that are governed by basic physical conservation laws on a macroscopic scale, whereas LBM is based on evolution rules which obey the same conservation laws on a mesoscopic scale. In LBM the physical evolution rules are discrete while in FEM methods the discretization is performed on the level of the macroscopic flow equations.

6.4 Simulation results

In this section we present a detailed comparison between the two methods using the SMRX test case. We compare the simulation results such as the velocity and the pressure along the SMRX mixer of both methods, as well as their computational requirements. The experimental results of van Dijck et al. [74] for the pressure drop at different flow rates are used to validate both numerical methods. The simulation parameters correspond to those used in these experiments (Table 6.1). The rectangular reactor consists of an inlet section, followed by one or two mixer elements and an outlet section.

### Fluid flow properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newtonian viscosity</td>
<td>1.46 Pa.s</td>
</tr>
<tr>
<td>Fluid density</td>
<td>1053 kg/m³</td>
</tr>
<tr>
<td>Flow rate</td>
<td>0 – 250 l/h</td>
</tr>
</tbody>
</table>

### Reactor dimensions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height/width</td>
<td>8.5 cm</td>
</tr>
<tr>
<td>Length</td>
<td>17.0 cm (1 SMRX)</td>
</tr>
<tr>
<td></td>
<td>25.5 cm (2 SMRX)</td>
</tr>
</tbody>
</table>

### Mixer dimensions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height/width/length</td>
<td>8.5 cm</td>
</tr>
<tr>
<td>Number of tubes per SMRX</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 6.1: The fluid flow properties and the SMRX dimensions adopted from van Dijck’s experiments and used to validate the simulations.

Concerning the FEM simulations, the first step is to generate a satisfactory mesh of the SMRX geometry. Generating a body-fitted mesh for such a complex geometry is quite a challenge [56]. Using the mesh generator included in the commercial I-DEAS package [78], we succeeded in creating two adequate meshes (with adequate we just mean meshes that look satisfactory) made of roughly 35,000 and 45,000 tetrahedral elements (Figure 6.3). However, even with this powerful software, some problems occurred at certain crossing points between the tubes. To overcome these problems, 4 tubes with the same diameter as the mixer tubes were placed perpendicular to the flow direction at the
crossing points. It can be argued that they have minor influence on the flow since there are located in dead or low velocity zones. In our meshes, we used two types of tetrahedral elements, namely the $P_1^+ - P_0$ elements (called linear elements) and the $P_2^- - P_1$ elements (called quadratic elements). These element types satisfy the so-called Brezzi-Babuska condition [84], a theoretical compatibility condition which ensures reliable computations, especially for the pressure. The $P_1^+ - P_0$ linear element is an 8 nodal points element where the velocity is approximated linearly, the pressure is taken to be constant and extra degrees of freedom are added at the middle of each face to satisfy the Brezzi-Babuska condition. The $P_2^- - P_1$ quadratic element is a 15 nodal points element where the velocity and pressure are approximated quadratically and linearly, respectively, and extra degrees of freedom are added at the middle of each face and edge and also at the centroid. We refer the reader to the paper of Bertrand et al. [85] for more details. The flow simulations were performed on one RISC6000 77 MHz node of an IBM 9076 SP2 with 512 MB of nodal memory using POLY3D™. The boundary conditions are summarized in Table 6.2. The memory space usage was 129 MB, 480 MB, and 165 MB respectively for the 35,000 element $P_1^+ - P_0$ and $P_2^- - P_1$ meshes and for the 45,000 element $P_1^+ - P_0$ mesh. We did not consider simulations on a 45,000 element $P_2^- - P_1$ mesh, because the simulation results obtained on the 35,000 element $P_2^- - P_1$ mesh were quite satisfactory, as will be shown in the next section. The computational time was 40 CPU min, 190 CPU min and 57 CPU min respectively.

In LBM the geometry is represented on a uniform Cartesian grid. Each grid point is marked as a solid point when it belongs to an obstacle and otherwise it is marked as a fluid point. To obtain a satisfactory discretization of the SMRX element we have used lattices of dimension 112x56x56 and 224x112x112 grid

Figure 6.3: Finite-element mesh for the SMRX reactor (35000 elements).
points, based on a tube radius discretization of 4 and 8 grid points respectively. Tube radii of 4 and 8 grid points respectively are based on previous studies using the lattice-Boltzmann method (see for example Refs [40, 41]). Compared to FEM, the grid generation is much easier for LBM, especially due to the uniformity of the lattice. Of course, it must be emphasized that the uniform nature of the lattice has its own limitations. For example, in this specific SMRX case the number of grid points for representing the inlet and the outlet sections is of the same amount as the number of grid points for representing the mixer element itself. A grid refinement based on a coarser grid resolution for the inlet and the outlet sections and a finer resolution for the element discretization could be very useful. However, the formulation and application of LBM schemes based on non-uniform lattices is still an important research topic (see for example addendum A and Refs. [69, 29, 86]).

<table>
<thead>
<tr>
<th>Boundary</th>
<th>FEM</th>
<th>LBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet</td>
<td>$v_x$ using a quasi-parabolic velocity distribution given by a series approximation [87]; $v_y = v_z = 0$</td>
<td>Periodic</td>
</tr>
<tr>
<td>Outlet</td>
<td>$-p + \rho \frac{\partial v_x}{\partial x} = 0; v_y = v_z = 0$</td>
<td>Periodic</td>
</tr>
<tr>
<td>Solid walls</td>
<td>$v_x = v_y = v_z = 0$</td>
<td>Bounce-back rule</td>
</tr>
</tbody>
</table>

Table 6.2: Boundary conditions used in both methods ($v_x$, $v_y$ and $v_z$ are the components of the velocity vector and the flow is in the $x$ direction).

The boundary conditions used in the LBM simulations are summarized in table 6.2. Notice that the flow boundaries and especially the no-slip boundaries can be implemented quite easily due to the particle based approach. The in- and outlet are periodic and the flow is driven by a local body force. The use of periodic boundaries is based on the assumption that the velocity profiles at the in- and outlet are fully developed (the inlet and outlet sections are long enough to guarantee that). More sophisticated pressure and velocity boundaries can be used when these conditions are not valid (see e.g. Ref. [35]). The solid walls have been modeled by the bounce-back boundary condition; the velocities of the particles that reach the wall are simply reversed. The bounce-back rule generates a no-slip boundary which is located somewhere between the solid and the adjacent fluid nodes (see chapter 3 and Refs. [38, 55]). As the grid resolution is increased, the agreement between the actual geometry and the locations of the no-slip boundaries is improved. Careful determination of the appropriate lattice dimensions for a certain simulation is therefore very important. The flow rate in the simulations is controlled by the magnitude of the body force and the viscosity is tuned by the relaxation parameter.

The LBM simulations were performed on a Parsytec CC parallel machine with 128 MB memory per node (133 MHz PowerPC 604). The total memory space usage was 50 MB and 400 MB for the 112x56x56 and 224x112x112 lattice respectively. The computational time on one node of the parallel machine was
Table 6.3: Computational requirements of FEM and LBM for different grid/mesh resolutions. In the case of FEM the meshes I, II and III denote the \(35000 \, P^+ - P_0\) element mesh, \(45000 \, P^+ - P_0\) element mesh and the \(35000 \, P^+_2 - P_1\) elements mesh, respectively. In the case of LBM the \(112\times56\times56\) and \(224\times112\times112\) lattices are meant by I and III, respectively. We have included the CPU times and the total memory requirements of both methods. As both methods are quite different, it does not make sense to compare the computational requirements of both methods obtained on the various mesh/grid types. However, with additional information concerning the accuracy obtained on the different mesh/grid types one can still gain some insight in the computational requirements of both method for a certain accuracy range. In the next section, we will see that for FEM accurate results are only obtained on the type III mesh, whereas the results obtained with LBM on the type I grid is quite accurate.

110 CPU min and 3300 CPU min (estimated, because this simulation could not be executed on a single node of the machine due to memory constraints) respectively.

However, the final LBM simulations were performed on 16 nodes of the parallel machine and then the computation time was 10 CPU min (parallel efficiency [88] of 0.7) and 210 CPU min (parallel efficiency close to 1) for the two grid resolutions. The computational requirements of both methods are summarized in Table 6.3. It is evident that the sequential computation time of the fine grid LBM simulations is quite high. The reason is that although the computation time for each LBM iteration increases linearly with the number of lattice points, the number of time iterations to reach the steady state depends quadratically on the lattice spacing (provided that the relaxation parameter is kept constant). Notice that in LBM a complete time dependent flow is simulated (see chapter 4 for details) in contrast to FEM where in our simulations the stationary Navier-Stokes equations are solved.

Figures 6.10.a to 6.10.f show the velocity at three slices along the reactor. We conclude that qualitatively there is a good agreement for the local velocities obtained by the two methods. We have to underline the fact that the software used to render the FEM contour plots uses only the geometrical nodes (vertex nodes), leading to a loss of accuracy (especially when simulations are performed with \(P^+_2 - P_1\) element) and to coarser contour plots than the LBM. Nevertheless the agreement is good. Furthermore, it is clear that the hot spots in the velocity profiles obtained by both methods agree with each other.
The average of the norm of the velocity along the reactor multiplied by the void factor, is shown in Figure 6.4. The reason for multiplying by the void factor is to avoid small fluctuations due to the 4 tubes added in the cross flow direction in the FEM mesh. In this figure we have included the results obtained on the two LBM grids and on the two types of FEM meshes (the $P_1^+$-$P_0$ and $P_2^+$-$P_1$ elements, respectively). The qualitative shape of the profile mimics the distribution of the void space along the reactor, as one would expect. For FEM the average and maximum difference between the velocity field of the two meshes is approximately 1.7% and 4.2%, respectively. For LBM we have found an average and maximum difference of around 1.6% and 3.1% for the velocity field, respectively. Moreover, we clearly see that there is a very good agreement (maximum difference 1%) between the solution obtained on the fine LBM grid and the $P_2^+$-$P_1$ FEM mesh.

The pressure along the reactor, which is much more sensitive to numerical accuracy than the velocity [56], is depicted in Figure 6.5. Here we see that there are clear differences in the pressure field obtained on the two FEM meshes, despite the good agreement found for the velocity field. The difference in pressure drop between the $P_1^+$-$P_0$ and $P_2^+$-$P_1$ meshes is around 44%. The difference between the LBM simulations however is around 7.6%. Moreover, it is evident that there is a very good agreement for the average pressure along the reactor between $P_2^+$-$P_1$ FEM simulations and the LBM simulations on the fine grid. The difference in
Lattice-Boltzmann and Finite-Element Simulations of Fluid Flow in a SMRX Mixer

0.00 0.04 0.08 0.12 0.16
Distance along flow direction (m)

0.00 0.04 0.08 0.12
Distance along flow direction (m)

Figure 6.5: Pressure along the reactor (flow rate 250 l/h).

The pressure drop as a function of the flow rate is shown in Figure 6.6. Here we have included the results obtained by the two methods on the different grids and meshes. We see that for FEM there is a large discrepancy between the $P^+_1 P_0$ and the $P^+_2 P_1$ meshes for all flow rates. Even for increasing number of elements the results obtained by the $P^+_2 P_1$ are not that accurate. An approximately 30% increase in the number of $P^+_1 P_0$ elements leads to an improvement in the pressure drop by around 8% only, which is normal for such type of element. The LBM results on the two grids are quite close to each other. We clearly see that indeed the FEM and the LBM solutions converge to each other as the grid or mesh-element type is refined. These results are also good in agreement with the experimental data of van Dijck et al. The error in the experimental data is approximately 7% [74]. For low flow rates the simulations are in the estimated error range of the experimental data. For higher flow rates the simulations overestimate the pressure drop. The maximum difference between simulations and experiment is around 15%. These differences may be caused by experimental uncertainties in the calibration of the flow rate and in viscosity measurements. Notice that in both simulations, we have assumed that the fluid is Newtonian, whereas from experimental measurements it was evident that the fluid is not purely Newtonian [74]. Furthermore, more detailed experimental measurements (at least more data points) are required in order to judge the ac-
6.4 Simulation results

Figure 6.6: Pressure drop versus flow rate.

Figure 6.7: LBM discretization of two SMRX reactors (tube radius of 4 lattice points).
We have also simulated fluid flow in a reactor consisting of 2 SMRX elements with the LBM. The geometry of this reactor is shown in Figure 6.7. In this setup the second SMRX element is rotated 90 degrees. The results for the pressure drop on the different grids are shown in figure 6.8. We clearly see that also in this case there is a good agreement between simulation and experiment. Here we found a deviation in the order of 15% between simulation and experiment. This case could not be simulated using FEM due to memory requirements. For LBM, the total amount of memory usage depends linearly on the number of mixer elements. In principle a reactor consisting of more mixer elements (which corresponds to the actual configuration in industrial reactors) can be simulated on the full domain of the parallel machine.

From all these results, it appears that the FEM has more difficulties in predicting the right pressure drop. With the FEM, a too coarse mesh will most probably lead to a poor estimate of the pressure field, despite a relative good estimate of the velocity field. Especially for the current test case, we argue that a \( P^+_2 - P_1 \) type of element is absolutely required to get a satisfactory estimate of the pressure field.

Moreover, the LBM uses roughly 10 times less memory than the FEM to reach a similar accuracy since the solution given by the LBM coarse grid reaches already a satisfactory precision. Although the simulations were not executed on the same computer, it appears also that the LBM coarse grid simulation re-
6.5 Methodological Comparison

In the previous section, we have demonstrated that LBM is indeed efficient for simulating a single-phase, isothermal, incompressible and laminar fluid flow through an SMRX reactor. In this section, we will discuss the possibilities to simulate more complicated flow problems by the two methods. Also more practical aspects like for example code development and memory usage will be considered. Table 6.4 summarizes the different aspects.

<table>
<thead>
<tr>
<th></th>
<th>FEM</th>
<th>LBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code implementation</td>
<td>long</td>
<td>easy</td>
</tr>
<tr>
<td>Memory Usage (Number of “elements”) / MB</td>
<td>$P_1^+-P_0$: 274</td>
<td>6800</td>
</tr>
<tr>
<td></td>
<td>$P_3^+-P_1$: 74</td>
<td></td>
</tr>
<tr>
<td>CPU Time ($\sim$ number of “elements”) / $X$</td>
<td>$X = 1.4$</td>
<td>$X = 1.7$</td>
</tr>
<tr>
<td>Parallelization</td>
<td>not trivial</td>
<td>inherent locality</td>
</tr>
<tr>
<td>Local mesh refinement</td>
<td>available</td>
<td>in development</td>
</tr>
<tr>
<td>Transient flow</td>
<td>available</td>
<td>inherent</td>
</tr>
<tr>
<td>Heat transfer</td>
<td>available</td>
<td>research topic</td>
</tr>
<tr>
<td>Mass transfer</td>
<td>available</td>
<td>available</td>
</tr>
<tr>
<td>Multi-phase</td>
<td>not as straightforward</td>
<td>research topic</td>
</tr>
<tr>
<td></td>
<td>as in LBM</td>
<td></td>
</tr>
<tr>
<td>Non-Newtonian/elastic fluid rheology</td>
<td>available</td>
<td>research topic</td>
</tr>
<tr>
<td>Turbulence</td>
<td>partially successful</td>
<td>partially successful</td>
</tr>
</tbody>
</table>

Table 6.4: A methodological comparison between FEM and LBM.

The FEM is in use in many commercial CFD packages like POLY3D™, POLYFLOW™ or FIDAP™. This in contrary to the LBM, which is a rather new method and it is still in its development phase. The only commercial LBM code at the moment is POWERFLOW™ developed by EXA Corporation. The implementation of a LBM code is quite straightforward, whereas the implementation of a FEM code is long and tedious (a matter of years). For this specific application, the LBM can reach the same level of accuracy with a memory usage roughly 10 times lower than the FEM. In Table 6.4, we have included the average requirements (at least for the codes used) for both methods in terms of elements or grid cells per MB of memory, although the nature of one “element” requires roughly the same computational time (on a sequential machine) compared to the FEM fine mesh simulation. However, the computational time required by the LBM fine grid simulation shows a drastic increase compared to the coarse grid simulation.
Lattice-Boltzmann and Finite-Element Simulations of Fluid Flow in a SMRX Mixer

differs from one method to the other and therefore these numbers should be interpreted with care. Nevertheless, it can provide some insights in the amount of memory needed for a specific application given the number of elements or grid points which are required for a satisfactory discretization.

As we mentioned earlier, it was possible for our test case to achieve the same level of accuracy using roughly the same computational time. However, the computational time as a function of the number of elements or grid cells behaves quite differently for both methods. It appears that as the mesh or the grid is refined the computational time as a function of the number of “elements” increases with the power 1.4 for the FEM [78] and 1.7 for LBM as explained below. Notice that in LBM a decrease of the lattice spacing by a factor 2 results into 8 times more lattice points. The lattice spacing is thus proportional to $N^{\frac{1}{2}}$ ($N$ is the number of lattice points) and the number of time steps to reach a steady-state depends quadratically on the lattice spacing [55]. Therefore the total computational time is proportional to $N^{\frac{1}{2}}N \approx N^{1.7}$, which explains the sharp rise in computational time noticed for the LBM fine grid simulations. Nevertheless, the total computational time depends on the problem. Notice that a more efficient relaxation scheme (see chapter 3) may be used to accelerate the convergence of the LBM simulations.

The inherent locality of the update rules in the LBM makes efficient parallelization straightforward (see chapter 5), whereas parallelization of FEM codes may be more complicated especially when implicit methods are used.

Concerning the mesh or grid generation for complex geometries we can make the following remarks. It is obvious that it is much easier to generate a grid for the LBM than to generate a mesh for the FEM. However, recently a new method, the so-called virtual finite-element method (VFEM), has been proposed by Bertrand et al. [89]. This method belongs to the class of Fictitious Domain Methods based on Lagrange multipliers (Glowinski et al. [90]). It allows to impose fictitiously the inner part of the geometry using kinematic constraints introduced into the mathematical formulation by means of Lagrange multipliers. Then only one volumetric mesh representing the enclosure without its internal parts (in our test case, the reactor without the SMRX element) has to be generated. To gain insight in the performance of LBM compared to the promising VFEM approach, we have performed a preliminary comparison between LBM and VFEM. The results for the pressure drop as a function of the flow rate are shown in Fig. 6.9. We clearly see that there is a good agreement between LBM and VFEM. However, in return for the gain in terms of mesh generation, a much longer computational time was observed to reach equivalent convergence criteria. The VFEM method allowed us to simulate the SMRX test case without the introduction of the 4 tubes in the cross flow direction as mentioned earlier and it may be really useful in cases where usually remeshing is required or meshing is impossible by conventional methods. Otherwise, in classical 3D FEM meshing, only the use of tetrahedral elements allows the creation of a suitable mesh for geometries like our test case. Fortunately, mesh refinement techniques are fully available for the FEM, which support mesh refinement only where needed,
Simulation of transient flows is a fully available feature for both methods and it is even inherent in the LBM. Heat transfer, mass transfer, non-Newtonian and elastic fluid rheology are other features that have already been studied for many years by using FEM, but are quite recent developments in LBM [80, 91]. Extensions of both models to deal with turbulence have been studied for several test cases (see e.g. [92]). It is however important to notice that the complex phenomena which are inherent to turbulence makes both methods only partially successful. However, simulation of multi-phase flow and suspension flow using FEM is not as straightforward as in LBM. The LBM method appears to be suitable for modeling these complex flows in some hydrodynamic regimes as demonstrated by e.g. Grunau et al. [93] and Ladd [40, 41]. This discussion clearly suggests that both methods are good in their own respect and also that there may be developments in the future which may change the range of tractable applications for both methods.
6.6 Conclusions

Our results based on flow in a SMRX reactor show a good agreement between FEM simulations, LBM simulations, and experimental measurements. This suggests that the LBM is an accurate method for flow through complex geometries. It appears that the LBM is less memory consuming and uses computational times comparable to the FEM (for the same accuracy of the simulations), although there may be cases where the FEM method is more efficient, e.g. due to the uniform nature of the LBM grids. However, the execution times of the LBM methods show a sharp increase on very fine meshes. We discovered also that the LBM shows similar accuracy between pressure and velocity fields, whereas the FEM could exhibit a rather good estimate of the velocity field combined with a bad estimate of the pressure field due to mesh coarseness. Clearly, the choice between the two methods relies on the type of problem to solve, the computer resources available and time. For instance, starting to build a code from scratch, having a problem requiring a lot of memory and/or parallelization and/or dealing with multi-phase, the LBM will turn out to be a faster and easier method. On the other hand, having a flow problem involving heat transfer and/or non-Newtonian and elastic fluid rheology, the FEM would be a better choice. However, as mentioned earlier, the LBM is a rather new method and those features could probably become tractable for LBM in the coming years.
Figure 6.10: Contour plots of the velocity profile at different cross sections in the reactor (a and b at $x = 3.1$ cm, c and d at $x = 6.4$ cm, e and f at $x = 6.8$ cm where $x = 0$ is at the beginning of the static mixer element) for FEM (a, c and e) and LBM (b, d and f) simulations (finest meshes and flow rate is 250 l/h).
6.6 Conclusions

Our results show that the coupling between LBM and FEM simulations is crucial for accurate fluid flow predictions. This suggests that the two methods should be integrated in complex geometries. It appears that LBM has the potential to bridge computational gaps and provide efficient FEM simulations, although it may suffer from reinitialization issues at certain times, e.g., due to the use of the LBM. We have discovered that the LBM results are more sensitive to the level of accuracy, whereas the FEM results are more robust but may be limited by the number of nodes. We recommend using a hybrid approach with a good estimate of the pressure field due to mesh convergence. Further research is needed to improve the coupling method and enhance the simulation accuracy.