INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* **31**: 1019–1033 (1999)

# LATTICE-BOLTZMANN AND FINITE ELEMENT SIMULATIONS OF FLUID FLOW IN A SMRX STATIC MIXER REACTOR

D. KANDHAI[a,*], D.J.-E. VIDAL[b,1], A.G. HOEKSTRA[a,2], H. HOEFSLOOT[b,3], P. IEDEMA[b,4] AND P.M.A. SLOOT[a,5]

[a] *Faculty for Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, Netherlands*
[b] *Department of Chemical Engineering, University of Amsterdam, Nieuwe Achtergracht 166, 1018 WV Amsterdam, Netherlands*

## SUMMARY

A detailed comparison between the finite element method (FEM) and the lattice-Boltzmann method (LBM) is presented. As a realistic test case, three-dimensional fluid flow simulations in an SMRX static mixer were performed. The SMRX static mixer is a piece of equipment with excellent mixing performance and it is used as a highly efficient chemical reactor for viscous systems like polymers. The complex geometry of this mixer makes such three-dimensional simulations non-trivial. An excellent agreement between the results of the two simulation methods was found. Furthermore, the numerical results for the pressure drop as a function of the flow rate were close to experimental measurements. Results show that the relatively simple LBM is a good alternative to traditional methods. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: static mixer; finite element method; lattice-Boltzmann method

## 1. INTRODUCTION

The static mixer, a rather new technology introduced 15–20 years ago, has increased in popularity within the chemical industry over recent years [1]. It consists of specially designed stationary obstacles inserted in a pipe 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 with 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,

* Correspondence to: Faculty for Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, Netherlands. E-mail: kandhai@wins.uva.nl; http: //www.wins.uva.nl/pscs
[1] E-mail: vidal@chemeng.chem.uva.nl
[2] E-mail: alfons@wins.uva.nl
[3] E-mail: huubh@chemeng.chem.uva.nl
[4] E-mail: piet@chemeng.chem.uva.nl
[5] E-mail: peterslo@wins.uva.nl

CCC 0271–2091/99/221019–16$17.50

*Received July 1998*
*Revised November 1998*

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, a simplified version of the SMR-type mixer manufactured by Sulzer Chemtech Ltd. and used mainly in polymerization reactors [2]. It consists of a series of solid crossing tubes, placed inside a rectangular tubular reactor (see Figure 1). In this work, we will focus our attention on this static mixer and use the experimental results of van Dijck and van Dierendonck [3] as a validation for the numerical investigation.

Due to usually rather complex flows and geometries, only a few three-dimensional numerical simulations in static mixers have been performed in the past. The first one was in 1992 on a Kenics static mixer by Gyenis and Blickle [4] using stochastic simulations of steady state particle flows. The simulations that followed were all based on macroscopic momentum balance methods, like the finite element method (FEM), the finite difference method or the finite volume method (FVM). Using the finite volume package FLUENT™, Bakker and LaRoche [5] also studied flow and mixing in a Kenics static mixer. Later on, using the finite element program POLY3D™ from RheoTek, Bertrand *et al.* [6] looked at residence time distribution in low pressure drop (LPD) and interfacial surface generator (ISG) static mixers, from Ross Engineering, for Newtonian and power-law fluids. In the same way, Tanguy *et al.* [7] and Mickaily-Huber *et al.* [2] investigated flow and mixing in the complex SMRX static mixer. Recently, Avalosse and Crochet [8] have studied mixing of Newtonian and power-law fluids in a series of Kenics mixers using an FEM.

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 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 3).

In this paper we intend to show that there is an alternative approach, the so-called lattice-Boltzmann method (LBM) [9–12], to traditional macroscopic momentum balance methods for computing fluid flows in complex geometries, such as that of the SMRX static mixer. This method is based on a different concept and has been proven successful in simulating complex fluid dynamics problems, where conventional methods may be difficult to apply [12–15]. The key idea behind the LBM is to model fluid flow by distributions of particles, which, at each time step, *propagate* to a neighboring lattice points and subsequently re-distribute their momenta in a local *collision* phase. The inherent locality of this method makes it also ideal for parallel computing [16]. Of course, this method has its own limitations and the choice of one or another type of method will mainly depend on the problem to be

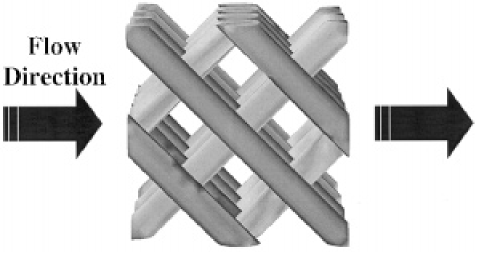

Figure 1. SMRX geometry.

solved. The main objective of this article is to show, rigorously, that this method can indeed be used for realistic engineering applications.

A comparison of the LBM with a Galerkin FEM, applied to the complex SMRX test case, will be presented from a theoretical, methodological and experimental point of view.

## 2. NUMERICAL METHODS

In this section we present the basic fundamentals of the FEM and the LBM. The aim is to underline the conceptual differences between the two methods. For more exhaustive reviews, we refer to the book of Cuvelier *et al.* [17] for the FEM and References [9–12] for the LBM.

### *2.1. The Galerkin finite element method*

On a macroscopic scale, the flow of an incompressible fluid in a given geometry $\Omega$ can be described by the classical Navier–Stokes equations [18]

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right)+\nabla\cdot\sigma+\nabla p=\mathbf{f}, \tag{1}$$

$$\nabla\cdot\mathbf{v}=0, \tag{2}$$

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 gravity force. The stress tensor $\sigma$ 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

$$\sigma=-2\eta\dot{\gamma}, \tag{3}$$

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 FEM for solving fluid flow dynamics makes use of the variational calculus, which allows the transformation of a set of partial differential equations (PDEs), the Navier–Stokes equations, into a system of linear algebraic equations that can be solved after preconditioning using a simple LU decomposition or by means of iterative algorithms.

### *2.2. The lattice-Boltzmann method*

The LBM originated from the lattice gas automata that are discrete models of hydrodynamics [12,19]. In these models, the computational grid consists of a number of lattice points (similar to a uniform finite difference grid) which are connected with some of their neighbors (depending on the model) by a bond or link. Basically, particles move synchronously along the bonds of the lattice and interact locally according to a given set of rules in the following two phases:

1. *Propagation*. In this phase particles move along lattice bonds from one lattice node to one of its neighbors.
2. *Collision*. Particles on the same lattice node shuffle their momenta locally, subject to mass and momentum conservation.

The major drawbacks of early lattice-gas models were statistical noise, exponential complexity of the collision operator, and the lack of Galilean invariance [9]. Later, many modifications have helped to resolve most of these problems and lattice-gas methods have proven to be correct models for the Navier–Stokes equations [12].

Nevertheless, the apparent deficiencies of the early lattice-gas methods inspired the formulation of the LBM. In the LBM, a population of particles is being tracked instead of a single particle and there is relatively more freedom in the definition of the collision operator [20–22]. The latest major modification to date is the lattice-Bhatnagar–Gross–Krook (BGK) model, where the collision operator is based on a single-time relaxation to the local equilibrium distribution [13]. This model is the simplest one in the hierarchy of LBMs and is currently widely used in practical fluid flow simulations. In this paper, the so-called $D_3Q_{19}$ model is used, where each lattice point is connected to its six nearest and 12 diagonal neighbors on distance $\sqrt{2}$. Rest particles are also included here.

The time evolution of the lattice-BGK (LBGK) model is given by [13]

$$f_i(\mathbf{r}+\mathbf{c}_i, t+1) = f_i(\mathbf{r}, t) + \frac{1}{\tau}\,(f_i^{(0)}(\mathbf{r}, t) - f_i(\mathbf{r}, t)), \tag{4}$$

where $\mathbf{c}_i$ is the $i$th link, $f_i(\mathbf{r}, t)$ is the density of particles moving in the $\mathbf{c}_i$-direction, $\tau$ is the BGK relaxation parameter, and $f_i^0(\mathbf{r}, t)$ is the equilibrium distribution function towards which the particle populations are relaxed. A common choice for $f_i^0(\mathbf{r}, t)$ is [13]

$$f_i^0 = t_i\rho\left(1 + \frac{1}{c_s^2}\,(\mathbf{c}_i\cdot\mathbf{v}) + \frac{1}{2c_s^4}\,(\mathbf{c}_i\cdot\mathbf{v})^2 - \frac{1}{2c_s^2}\,v^2\right), \tag{5}$$

where $t_i$ is a weight factor depending on the length of the vector $\mathbf{c}_i$, $c_s$ is the speed of sound, $\rho$ is the density and $\mathbf{v}$ is the velocity. The density and the velocity are obtained from moments of the discrete velocity distribution $f_i(\mathbf{r}, t)$,

$$\rho(\mathbf{r}, t) = \sum_{i=0}^{N} f_i(\mathbf{r}, t) \quad \text{and} \quad \mathbf{v}(\mathbf{r}, t) = \frac{\sum_{i=0}^{N} f_i(\mathbf{r}, t)\mathbf{c}_i}{\rho(\mathbf{r}, t)}, \tag{6}$$

where $N$ is the number of links per lattice point. In the $D_3Q_{19}$ model, the weight factors are equal to $\frac{1}{3}$, $\frac{1}{18}$ and $\frac{1}{36}$ for the rest particle, nearest neighboring and diagonal neighboring links respectively. The fluid pressure, $p(\mathbf{r}, t)$, is given by the relation

$$p(\mathbf{r}, t) = c_s^2(\rho(\mathbf{r}, t) - \bar{\rho}) = c_s^2\Delta\rho, \tag{7}$$

where $\bar{\rho}$ is the mean density of the fluid. Note that in the LBM, the pressure is generated automatically by the spatial density fluctuations.

The models presented here yield the correct hydrodynamic behavior for an incompressible fluid in the limit of low Mach and Knudsen numbers [13]. The kinematic viscosity of the simulated fluid $v$ and the speed of sound $c_s$ in lattice units, are $v = (\tau - \frac{1}{2})/3$ and $c_s = \sqrt{\frac{1}{3}}$ [13].

Briefly stated, in the lattice-Boltzmann algorithm, the flow field is computed by evolving particle densities in time constrained to local conservation of mass and momentum. The hydrodynamic fields, such as the density, velocity and the pressure, are computed from the particle densities as described above. At each time step, the particle densities are propagated to a neighboring lattice point along the link on which they reside. After this the particle densities are relaxed to an equilibrium distribution function, which in fact is Maxwellian in the low Mach number limit. The kinematic viscosity of the fluid is controlled by the relaxation rate.

It is obvious that the FEM and the LBM are two quite different numerical approaches. The FEM is based on approximations of flow equations that are governed by basic physical conservation laws on the *macroscopic* scale, whereas the LBM is based on evolution rules that obey the same conservation laws on a *mesoscopic* scale. In the LBM, the physical evolution rules are discrete while in the FEM methods, the discretization is performed on the level of the

Table I. The fluid flow properties and the SMRX dimensions adopted from van Dijck's experiments [3] and used to validate the simulations

| | |
|---|---|
| Fluid flow properties | |
| Newtonian viscosity | 1.46 Pa s |
| Fluid density | 1053 kg m$^{-3}$ |
| Flow rate | 0–250 L h$^{-1}$ |
| Reactor dimensions | |
| Height/width | 8.5 cm |
| Length | 17.0 cm (one SMRX) |
| | 25.5 cm (two SMRX) |
| Mixer dimensions | |
| Height/width/length | 8.5 cm |
| Number of tubes per SMRX | 21 |

macroscopic flow equations. The LBM can be viewed as a minimal model for the Navier–Stokes equations instead of a full molecular dynamics approach. Indeed, fluid flow is mainly determined by the *collective* behavior of many molecules and not really by the detailed molecular interactions.

## 3. 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 and van Dierendonck [3] for the pressure drop at different flow rates are used to validate both numerical methods. Therefore, the simulation parameters correspond to those used in these experiments (Table I). The rectangular tabular reactor consists of an inlet section, followed by one or two mixer elements and an outlet section.

Concerning the FEM simulations, the first step was to generate a satisfactory mesh of the SMRX geometry. Generating a body-fitted mesh for such a complex geometry still remains quite a challenge [2]. Using the mesh generator included in the commercial I-DEAS package of SDR-C, we succeeded in creating two adequate meshes made of roughly 35000 and 45000 tetrahedral elements (Figure 2). However, even with this powerful software, some problems occurred at certain intersections between the tubes. To overcome these problems, four tubes with the same diameter as the mixer tubes were placed perpendicular to the flow direction at the intersections. It can be argued that they have only minor influence on the flow since they 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 are satisfying the so-called Brezzi–Babuska condition [23], a theoretical compatibility condition that ensures reliable computations, especially for the pressure. The $P_1^+ - P_0$ linear element is an eight nodal point 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 point 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

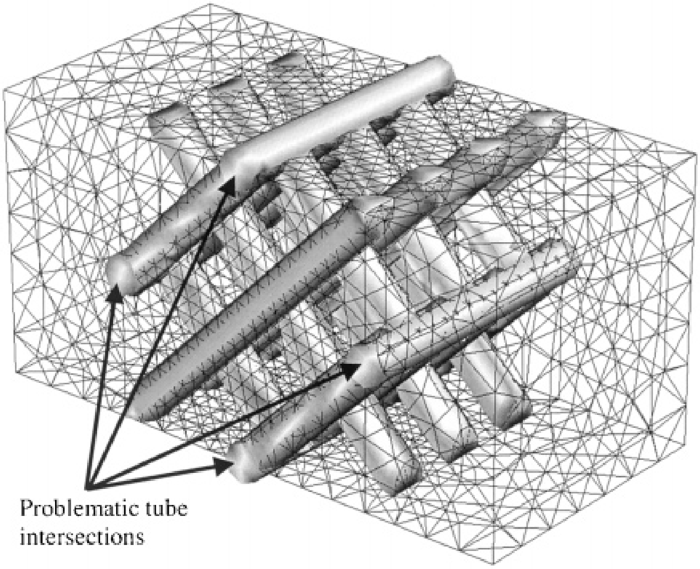

Figure 2. Finite element mesh for the SMRX reactor (35000 elements). The need for the problematic tubes is described in the text.

of Bertrand *et al.* [24] for more details. The flow simulations were performed on a RISC6000 77 MHz node of an IBM 9076 SP2 with 512 MB of nodal memory using POLY3D™. The boundary conditions used are summarized in Table II. The memory space usage was 129, 480 and 165 MB respectively for the 35000 element $P_1^+ - P_0$ and $P_2^+ - P_1$ meshes and for the 45000 element $P_1^+ - P_0$ mesh. The computational time was 40, 190 and 57 CPU min respectively.

In the 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, otherwise it is marked as a fluid point. To obtain a satisfactory discretization of the SMRX element we have used lattices of dimension 112 × 56 × 56 and 224 × 112 × 112 grid points, based on a tube radius discretization of four and eight grid points respectively. Compared with the FEM, the grid generation is much easier for the 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 the same 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

Table II. Boundary conditions used in both methods

| Boundary | FEM | LBM |
|---|---|---|
| Inlet | $v_x$ using a quasi-parabolic velocity distribution given by a series approximation [38]; $v_y = v_z = 0$ | Periodic |
| Outlet | $-p + \rho(\partial v_x/\partial x) = 0$; $v_y = v_z = 0$ | Periodic |
| Solid walls | $v_x = v_y = v_z = 0$ | Bounce-back rule |

$v_x$, $v_y$ and $v_z$ are the components of the velocity vector and the flow is in the $x$-direction.

application of LBM schemes based on non-uniform lattices is still an important research topic [25–27].

The boundary conditions used in the LBM simulations are summarized in Table II. We see that the flow boundaries and especially the no-slip boundaries can be implemented quite easily due to the particle based approach. The inlet 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 inlet 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. [28]). The solid walls have been modeled by the bounce-back boundary condition; particles that reach the wall are reflected in the opposite direction. The bounce-back rule generates a no-slip boundary, which is located somewhere between the solid and the adjacent fluid nodes [29,30]. 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 flow simulations were performed on a 32 node Parsytec CC parallel machine with 128 MB of memory per node (133 MHz PowerPC 604). The total memory space usage was 50 and 400 MB for the $112 \times 56 \times 56$ and $224 \times 112 \times 112$ lattice respectively. The computational time on one node of the parallel machine was 110 CPU min and 55 CPU hours (estimated, because this simulation could not be executed on a single node of the machine due to memory constraints) respectively. However, the real runs were performed on 16 nodes of the parallel machine and then the computation time was 10 CPU min (parallel efficiency of 0.7 [16]) and 210 CPU min (parallel efficiency close to 1) for the two grid resolutions. The computations of the FEM and the LBM have been performed on different machines. However, the computational times required by the coarse grid LBM simulations on one processor of the Parsytec CC machine were similar to the computational time required by the LBM on one processor of the IBM 9076 SP2 machine. Timing measurements show that the turn-around time on one node of the SP2 machine was around 10% lower than that on one node of the Parsytec CC machine.

From all these results, it is evident that the sequential computation time of the fine grid LBM simulations is quite high. The reason for this 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 the LBM, a complete time-dependent flow is simulated in contrast to the FEM, where the time-independent Navier–Stokes equations are solved given some initial flow field.

Plate 1(a)–(f) shows the norm of the velocity at three slices along the reactor ((a), (c) and (e) are the results of the FEM, while (b), (d) and (f) are the LBM results). We clearly see that qualitatively there is a good agreement for the local velocities obtained by the two methods. We have to underline the fact that the viewer 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, but 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 3. The reason for multiplying by the void factor is to avoid small fluctuations due to the four 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

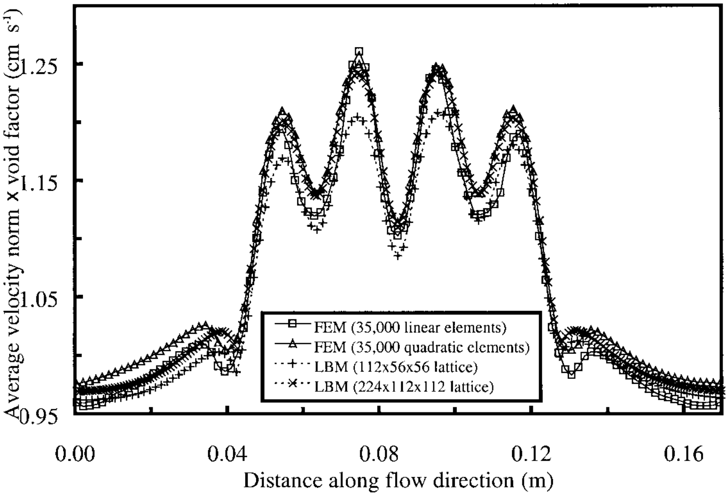

Figure 3. Average velocity norm times void factor along the reactor (flow rate is 250 L h$^{-1}$).

(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 the FEM, the average and maximum difference between the velocity field of the two meshes is approximately 1.7% and 4.2% respectively. For the 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, is depicted in Figure 4. 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

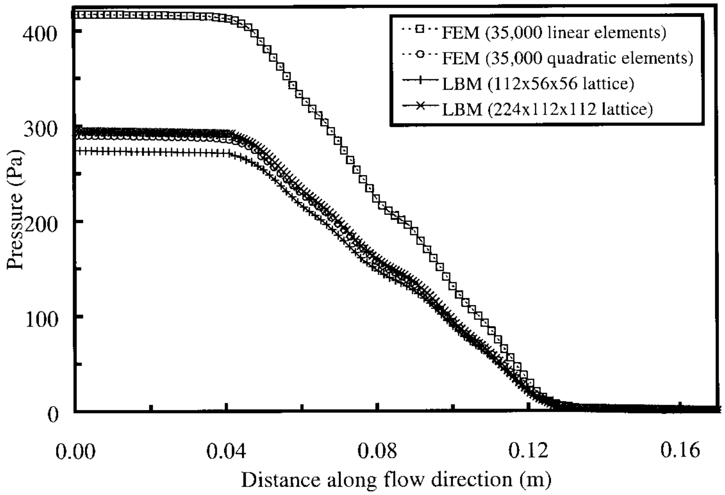

Figure 4. Pressure along the reactor (flow rate 250 L h$^{-1}$).

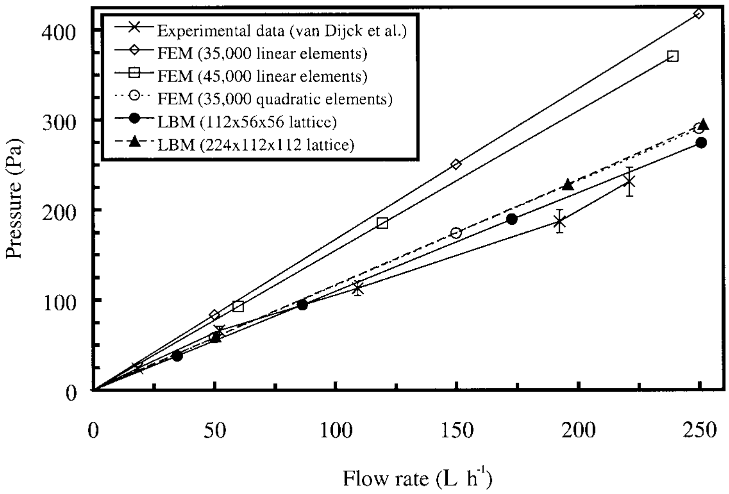

Figure 5. Pressure drop vs. flow rate.

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 the $P_2^+ - P_1$ FEM simulations and the LBM simulations on the fine grid. The difference in pressure drop is around 1.6%, and the small fluctuations in the mean pressure along the SMRX element zone are also similar.

The pressure drop as a function of the flow rate is shown in Figure 5. Here we have included the results obtained by the two methods on the different grids and meshes. We see that for the FEM there is a big 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 $P_1^+ - P_0$ are not that accurate. An, approximate, 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 and van Dierendonck [3]. The systematic error in the experimental data is approximately 7%. 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%, which is acceptable for such a complex geometry. 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 [3]. Furthermore, more detailed experimental measurements (at least more data points) are required in order to judge the actual cause of the slight disagreement between simulations and experiments.

We have also simulated fluid flow in a reactor consisting of two SMRX elements with the LBM. The geometry of this reactor is shown in Figure 6. In this set-up the second SMRX element is rotated 90°. The results on the different grids are shown in Figure 7. We clearly see that, in this case also, there is a good agreement between simulation and experiment. Here we found a deviation in the order of 15% between simulation and experiments, which can be considered already quite accurate for such a complex geometry and considering experimental

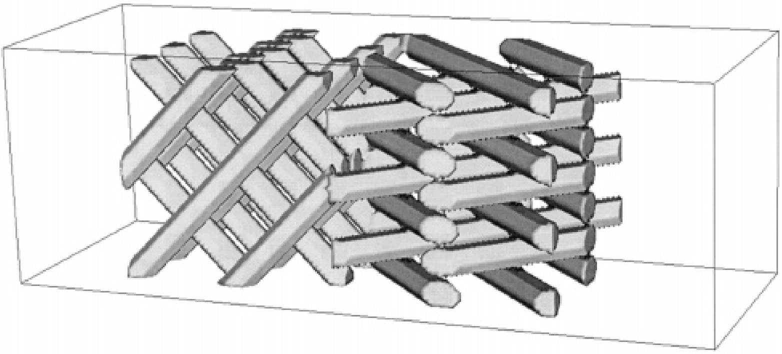

Figure 6. LBM discretization of two SMRX reactors (tube radius of four lattice points).

uncertainties already mentioned. This case could not be simulated using the FEM due to memory requirements. For the LBM, the total amount of memory usage depends linearly on the number of mixer elements, and thus 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 clearly that the FEM has more difficulties in predicting the right pressure drop. With the FEM, a not-fine-enough 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, although, strictly speaking, it would have been interesting to check what would be the precision achieved by a $P_1^+ - P_0$ simulation using the same amount of memory as that used by the $P_2^+ - P_1$ simulation.

Moreover, in the case tested, the LBM uses roughly ten times less memory than the FEM to reach a similar accuracy, since the solution given by the LBM coarse grid already reaches a satisfactory precision. Although the simulations were not executed on the same computer, it appears also that the LBM coarse grid simulation requires roughly the same computational

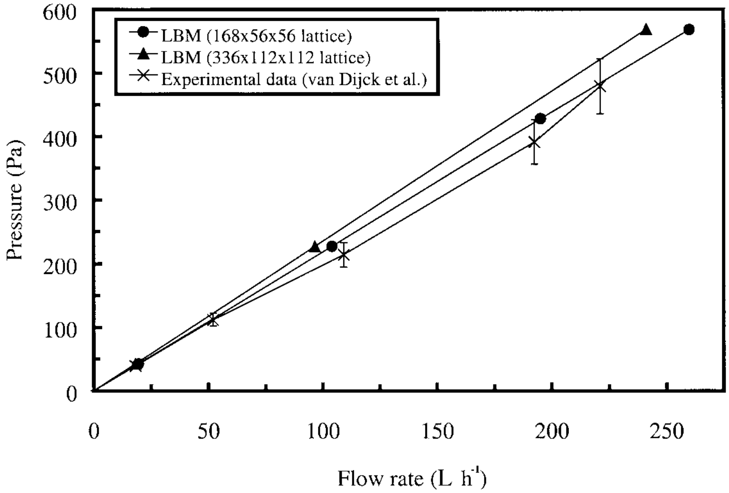

Figure 7. Pressure drop vs. flow rate for a two SMRX element reactor (flow rate is 250 L h$^{-1}$).

Table III. A methodological comparison between the FEM and the LBM

| | FEM | LBM |
|---|---|---|
| Code implementation | Long | Easy |
| Memory usage (Number of ‘elements’/MB) | $P_1^+ - P_0$: 274<br>$P_2^+ - P_1$: 74 | 6800 |
| CPU time ( = K · (number of ‘elements’)$^X$) | $X = 1.4$ | $X = 1.7$ |
| Parallelization | Not trivial | Inherent locality |
| Local mesh refinement | Available | In development |
| Transient flow | Available | Inherent |
| Heat transfer | Available | Research topic |
| Mass transfer | Available | Available |
| Multi-phase | Not as straightforward as in the LBM | Available |
| Non-Newtonian/elastic fluid rheology | Available | Recent development |
| Turbulence | Partially successful | Partially successful |

In the case of the LBM ‘elements’ means grid points.

time (on a sequential machine) compared with the FEM fine mesh simulation. However, the computational time required by the LBM fine grid simulation shows a drastic increase compared with the coarse grid simulation. Therefore, the cost-effectiveness of the LBM is partly due to its implementation on a parallel machine.

## 4. METHODOLOGICAL COMPARISON

In the previous section, we have demonstrated that the 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 of simulating more complicated flow problems by the two methods. Also, more practical aspects, like, for example, code development and memory usage, will be considered. Table III summarizes the different aspects.

The FEM is in use in many different commercial computational fluid dynamics (CFD) packages like POLY3D™, POLYFLOW™ or FIDAP™. This in contrast to the LBM, which is a rather new method and it is still in its development phase. The only commercial code at the moment is POWERFLOW™ developed by the EXA Corporation. However, the implementation of an LBM code is quite straightforward, whereas the implementation of an 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 ten times lower than that of the FEM. In Table III, 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’ 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 that 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 and 1.7 for LBM. Notice that in the LBM, a decrease of the lattice spacing by a factor of 2 results in eight times more lattice points. The lattice spacing is thus proportional to $N^{1/3}$ ($N$ is the number of lattice points) and the number of time steps to reach a steady state depends quadratically on the lattice spacing [Kandhai *et al.*, 'Implementation aspects of 3D Lattice-BGK: boundaries, accuracy and a new fast relaxation method', submitted]. Therefore, the total computational time is proportional to $N^{2/3}N = N^{1.7}$, 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 proposed in [Kandhai *et al.*, 'Implementation aspects of 3D lattice-BGK: boundaries, accuracy and a new fast relaxation method', submitted] may be used to accelerate the convergence of the LBM simulations.

Moreover, the inherent locality of the update rules in the LBM makes efficient parallelization straightforward [16], whereas parallelization of the FEM codes may be more complicated, especially when implicit methods are used.

Concerning the mesh or grid generation for complex geometry, it is by far much easier to generate a grid for the LBM than to generate a mesh for the FEM, since the FEM was required solely until recent body-fitting of the geometry. However, a new method developed by Bertrand *et al.* [31], called the virtual finite element method (VFEM), which belongs to the class of Lagrange multiplier-based fictitious domain method (FDM) (Glowinski *et al.* [32]), allows the inner part of the geometry to be imposed fictitiously using kinematic constraints introduced into the mathematical formulation by means of Lagrange multipliers. Then, as for the LBM, 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. However, in return for the gain in terms of mesh generation, a much longer computational time was observed to reach equivalent convergence criteria in preliminary simulations, see Figure 8. The VFEM method allowed us to simulate our test case without the introduction of the four 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 three-dimensional FEM meshing, only the use of tetrahedral elements allows a

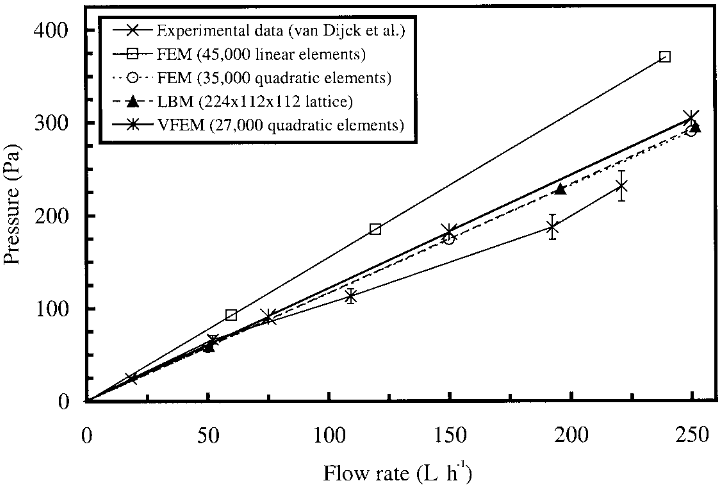

Figure 8. Preliminary comparison between the VFEM, the conventional FEM, the LBM and the experimental data (memory usage VFEM: 413 MB).

suitable mesh for geometries like our test case be created. Fortunately, mesh refinement techniques are fully available for the FEM, which support mesh refinement only where needed, whereas it is still a research topic for the LBM. Recently, some schemes based on non-uniform grids for the LBM have been proposed [26,27].

Simulation of transient flows is a fully available feature for both methods and it is even inherent in the LBM. Heat transfer, mass transfer and non-Newtonian/elastic fluid rheology are other features that have already been studied for many years by using FEMs, but are quite recent developments in the LBM [11,33]. Extensions of both models to deal with turbulence have been studied for several test cases (see, e.g. [34]). It is, however, important to notice that the complex phenomena that is inherent to turbulence makes both methods only partially successful. However, simulation of multi-phase flow and suspension flow using the FEM is not as straightforward as in the 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.* [35] and Ladd [36,37]. 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.

## 5. CONCLUSION

Our results based on the flow in an SMRX reactor show good agreement between the FEM simulations, the LBM simulations and experimental measurements. This suggests that the LBM is an accurate method for flow through complex geometries compared with well-established methods like the FEM. It appears that the LBM is less memory consuming and uses computational times comparable with 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/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 trivial in the coming years. In the near future we will apply the LBM to study mixing of two fluids in the SMRX reactor.

### ACKNOWLEDGMENTS

This work was partly carried out within the Massive Parallel Computing (MPR) project 'Many Particle Systems' funded by the Dutch foundation for basic research. Also, the authors would like to thank Robert Belleman for his help on the data visualization and Francois Bertrand, Antti Koponen and Jaap Kaandorp for many useful discussions and suggestions on the manuscript.

## REFERENCES

1. M. Mutsakis, F.A. Streiff and G. Schneider, 'Advances in static mixing technology', *Chem. Eng. Prog.*, **82**, 42–48 (1986).
2. E.S. Mickaily-Huber, F. Bertrand, P. Tanguy, T. Meyer, A. Renken, F.S. Rys and M. Wehrli, 'Numerical simulations of mixing in an SMRX static mixer', *Chem. Eng. J.*, **63**, 117–126 (1996).
3. L.A.J. van Dijck and L.L. van Dierendonck, 'Sulzer mixer reactor: experimental verification of the computer simulations of the hydrodynamical behavior', *Project Report*, Ecole Polytechnique Fédérale de Lausanne, 1992.
4. J. Gyenis and T. Blickle, 'Simulation of mixing during nonsteady state particle flow in static mixer tubes', *Acta Chim. Hung.*, **129**, 647–659 (1992).
5. A. Bakker and R. LaRoche, 'Flow and mixing with Kenics static mixers', *Cray Channels*, **15**, 25–28 (1993).
6. F. Bertrand, P.A. Tanguy and F. Thibault, 'A numerical study of the residence time distribution in static mixing', *Int. Chem. Eng. Symp. Ser.*, **136**, 163–170 (1994).
7. P.A. Tanguy, F. Bertrand, R. Lacroix, L. van Dijck, T. Meyer and A. Renken, 'Finite element flow simulations in an SMRX static mixer', *Engineering Foundation Conferences: Mixing XIV*, Santa Barbara, CA, 1996.
8. T. Avalosse and M.J. Crochet, 'Finite element simulation of mixing: three-dimensional flow through a Kenics mixer', *AIChE J.*, **43**, 588–597 (1997).
9. R. Benzi, S. Succi and M. Vergassola, 'The lattice-Boltzmann equation—theory and applications', *Phys. Rep.*, **222**, 145–197 (1992).
10. S. Chen, Z. Wang, X. Shan and G. Doolen, 'Lattice-Boltzmann computational fluid dynamics in three dimensions', *J. Stat. Phys.*, **68**, 379–400 (1992).
11. S. Chen and G.D. Doolen, 'Lattice-Boltzmann method for fluid flows', *Annu. Rev. Fluid Mech.*, **30**, 329–364 (1998).
12. D.H. Rothman and S. Zaleski, *Lattice Gas Cellular Automata*, Cambridge University Press, Cambridge, 1997.
13. Y.H. Qian, D. d'Humieres and P. Lallemand, 'Lattice-BGK models for Navier–Stokes equation', *Europhys. Lett.*, **17**, 479–484 (1992).
14. J.A. Kaandorp, C. Lowe, D. Frenkel and P.M.A. Sloot, 'The effect of nutrient diffusion and flow on coral morphology', *Phys. Rev. Lett.*, **77**, 2328–2331 (1996).
15. A. Koponen, D. Kandhai, E. Hellén, M. Alava, A. Hoekstra, M. Kataja, K. Niskanen, P. Sloot and J. Timonen, 'Permeability of three-dimensional random fibre webs', *Phys. Rev. Lett.*, **80**, 716–719 (1998).
16. D. Kandhai, A. Koponen, A. Hoekstra, M. Kataja, J. Timonen and P.M.A. Sloot, 'Lattice-Boltzmann hydrodynamics on parallel systems', *Comput. Phys. Commun.*, **111**, 14–26 (1998).
17. C. Cuvelier, A. Segal and A.A. van Steenhoven, *Finite Element Methods and Navier–Stokes Equations*, Reidel Publishing, Dordrecht, 1986.
18. R.B. Bird, W.E. Stewart and E.N. Lightfoot, *Transport Phenomena*, Wiley, New York, 1960.
19. U. Frish, B. Hasslacher and Y. Pomeau, 'Lattice-gas automata for the Navier–Stokes equation', *Phys. Rev. Lett.*, **56**, 1505–1508 (1986).
20. G. McNamara and G. Zanetti, 'Use of the Boltzmann equation to simulate lattice-gas automata', *Phys. Rev. Lett.*, **61**, 2332–2335 (1988).
21. F.J. Higuera and J. Jimenez, 'Boltzmann approach to lattice gas simulations', *Europhys. Lett.*, **9**, 663–668 (1989).
22. F.J. Higuera, S. Succi and R. Benzi, 'Lattice gas dynamics with enhanced collision', *Europhys. Lett.*, **9**, 345–349 (1989).
23. M. Fortin, 'Old and new finite elements for incompressible flows', *Int. J. Numer. Methods Fluids*, **1**, 347–364, (1981).
24. F. Bertrand, M. Gadbois and P.A. Tanguy, 'Tetrahedral elements for fluid flow problems', *Int. J. Numer. Methods Eng.*, **33**, 1251–1267 (1992).
25. F. Nannelli and S. Succi, 'The lattice-Boltzmann equation on irregular lattices', *J. Stat. Phys.*, **68**, 401–407 (1992).
26. N. Cao, S. Chen, S. Jin and D. Martinez, 'Physical symmetry and lattice symmetry in the lattice-Boltzmann method', *Phys. Rev. E*, **55**, 21–24 (1997).
27. X. He, L.S. Luo and M. Dembo, 'Some progress in lattice-Boltzmann methods. Part I. Nonuniform mesh grids', *J. Comput. Phys.*, **129**, 357–363 (1996).
28. R.S. Maier, R.S. Bernard and D.W. Grunau, 'Boundary conditions for the lattice-Boltzmann method', *Phys. Fluids*, **8**, 1788–1801 (1996).
29. M.A. Gallivan, D.R. Noble, J.G. Georgiadis and R.O. Buckius, 'An evaluation of the bounce-back boundary condition for lattice Boltzmann simulations', *Int. J. Numer. Methods Fluids*, **25**, 249–263 (1997).
30. D. Kandhai *et al.*, 'Implementation aspects of 3D lattice-BGK: boundaries, accuracy and a new fast relaxation method', *J. Comput. Physics*, **150**, 402–501 (1999).
31. F. Bertrand, P.A. Tanguy and F. Thibault, 'A three-dimensional fictitious domain method for incompressible fluid flow problems', *Int. J. Numer. Methods Fluids*, **25**, 1–18 (1997).
32. R. Glowinski, T. Pan and J. Périaux, 'A fictitious domain method for external incompressible viscous flow problems modelled by Navier–Stokes equations', *Comput. Methods Appl. Eng.*, **112**, 133–148 (1994).
33. Y.H. Qian and Y.F. Deng, 'A lattice BGK model for viscoelastic media', *Phys. Rev. Lett.*, **79**, 2742–2745 (1997).
34. S. Hou, J. Sterling, S. Chen and G.D. Doolen, 'A lattice-Boltzmann subgrid model for high Reynolds number flows', *Fields Inst. Commun.*, **6**, 151–166 (1996).

35. D. Grunau, S. Chen and K. Eggert, 'A lattice-Boltzmann model for multiphase flows', *Phys. Fluids A*, **5**, 2557–2562 (1993).
36. A.J.C. Ladd, 'Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. A theoretical foundation', *J. Fluid. Mech.*, **271**, 285–309 (1994).
37. A.J.C. Ladd, 'Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results', *J. Fluid. Mech.*, **271**, 311–339 (1994).
38. F.M. White, *Viscous Fluid Flow*, McGraw-Hill, New York, 1974.

Plate 1. 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 the FEM [(a), (c) and (e)] and the LBM [(b), (d) and (f)] simulations (finest meshes and flow rate is 250 L h$^{-1}$).