Finite Elements in Analysis and Design 47 (2011) 906–912

Contents lists available at ScienceDirect

Finite Elements in Analysis and Design

journal homepage: www.elsevier.com/locate/finel

Fluid–structure coupling using lattice-Boltzmann and fixed-grid FEM

Manuel Garcia *, Jorge Gutierrez, Nestor Rueda

Applied Mechanics Research Group, EAFIT University, Medellin, Colombia

ARTICLE INFO

Article history:
Received 30 November 2010
Accepted 27 December 2010
Available online 12 April 2011

ABSTRACT

This paper presents a method for the fluid–structure interaction by a hybrid approach that uses lattice-Boltzmann method (LBM) for the fluid dynamics analysis and fixed-grid FEM (FGFEM) for the structural analysis. The method is implemented in a high performance platform using GPUs to provide a high level of interactivity with the simulation. The solution uses the same Cartesian grid for both solvers.

The coupling between both methods is accomplished by mapping the macroscopic pressure, velocity or momentum values from the LBM simulation into the corresponding nodes of the FGFEM structural problem. In spite of being based on a Cartesian grid, both solvers take into account the effect of curve boundaries. Also the effect of a moving boundary is considered in the fluid simulations. The examples presented in this paper show that the accuracy of the solution is as the same level of the finite volume method of the finite element method. On the other hand, the performance of the parallel implementation of the proposed method is of the order that allows real-time visualisation of the computing values for two-dimensional problems.

1. Introduction

The fluid–structure interaction (FSI) problems involve the coupling of the computational fluid dynamics (CFD) and computational structural dynamics (CSD) fields. FSI can also be divided into two categories: class one problems that have two different physical domains (one for the fluid and another one for the solid) with a common boundary that defines the interface and class two problems in which there is not a defined boundary between the fluid and solid domains, such as seepage through porous media.

There are two types of approximations for this type of FSI simulations. The *monolithic approach* where the equations that govern the fluid and the solid are solved simultaneously in the same solver [1,2], and the *partitioned approach* where the equations that govern the fluid and the solid are solved separately, with two independent solvers, connected by a coupling algorithm [3,4].

In the partitioned approach each domain uses a different type of discretisation. Two different approximations have been used for fluid–structure interaction (FSI) problems that considered big deformations of the solid. The fluid can be described as (i) a moving mesh (which is solved through a Lagrangian Eulerian formulation ALE) or as (ii) a fixed Cartesian grid (FG). In both cases, the geometry of the moving boundary needs to be explicitly described through an interface. This geometric movement for the high deformations is traduced in complications for the fluid solvers that are based on meshes (such as the finite element method or finite volume method), because certain mesh quality parameters as the aspect ratio and minimum size need to be kept to ensure that the solution is accurate. Also, in some cases mesh refinement may be required, which complicates even more the task of solving the fluid properly. A comparison of some variations of the fictitious domain method and an ALE method is presented in van Loon et al. [5].

When a FG is used (case ii), these restrictions do not applied and the fluid–structure boundary can move freely on the fixed grid. The difficulty lies in the solid domain where the structural analysis needs to be made and in the interface conditions between the structure and the fluid. In this case there is no grid that follows the shape of the body and therefore it becomes more difficult to solve the important boundary layer in viscous flows. However, local adaptive refinement has been used to solve this problem [6].

Gamnitzer and Wall [7] propose an approach that uses a Quimera model in which a deformable fluid mesh that is attached to the structure (and it uses a ALE formulation) is superimposed on a background fixed grid for the fluid. This approach can be seen as a combination of cases (i) and (ii). The interaction is accomplished by a coupling of Dirichlet–Newman or Dirichlet–Robin boundary conditions, transferring forces from the background fixed grid to the moving mesh and then to the solid and the velocities from the structural boundary to de moving mesh and then to the fixed grid.

Another interesting approach using the extended finite element method (XFEM) for the fluid has been developed by Gerstenberger and Wall [8], Legay et al. [9] and Walhorn et al. [10]. An interface mesh is introduced in both solvers by mortar

* Corresponding author.
E-mail address: mgarcia@eafit.edu.co (M. Garcia).

0168-874X/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.finel.2011.03.010

methods [11]. In order to accurately represent the interface, the fluid domain elements are enriched by the step functions to deal with the jumps that are produced in the pressure and velocity fields when the structure advance on the fluid. Both approaches are compared in Gerstenberger and Wall [8] and Wall et al. [12].

The immersed boundary method developed by Peskin [13] involves Eulerian and Lagrangian variables linked by a Dirac delta function. The spatial discretisation is based on a fixed grid for Eulerian functions and on a curvilinear mesh for Lagrangian variables.

Kwon [14] used the lattice Boltzmann method (LBM) and FEM for a two-dimensional fluid with flexible beams. Geller et al. [15] used lattice Boltzmann with quadtree-type grids for fluid–structure interaction problems. More recently, Kollmannsberger et al. [16] used a combination of a structural solver that incorporates high order spatial resolution through p-FEM and lattice Boltzmann method (LBM) for fluids. The method was tested for two-dimensional cases with good results.

Other methods included the particle simulation method [17,18] and the particle element method [19,20], have been recently developed to discretise the continuum into particles that move under the action of external forces. In particular, Zhao et al. [21] simulated magma intrusion problems using both solid and fluid particles and considered the interaction between the viscous fluid (i.e. magma) and the solid (i.e. rock) in particle models. The main advantage of using the particle simulation method is that the fluid particles are allowed to move through the cracks between the solid particles so that both the moving boundary of the fluid and the propagation of cracks in the solid can be simulated more realistically.

This article presents an approach using case (ii) to solve the fluid in problems with deformations of the solid. The fluid solver uses the lattice Boltzmann method (LBM) and the structural solver uses the finite element method with fixed grid (FEM-FG).

Unlike traditional finite element method FEM, fixed-grid method FEM-FG does not use meshes that fits to the object shape, instead it uses a Cartesian mesh and the object boundary is gotten from interpolation techniques or implicit representation using level functions (level set). The method is particularly attractive when the boundary of the object moves because there is no need to update the mesh. It was initially developed for fluid, but then it was adapted for elasticity problems by Garcia and Steven [22] and has been applied to a variety of optimisation problems [23–25].

By using fixed grid to solve this problem it is intended that the grid is the same for the different domains and what is updated are their borders. The forces information is calculated from the flow analysis and it is transmitted directly to the fluid domain causing a deformation in the solid. By using fixed grid is not necessary to update the grid geometry speeding the interaction analysis.

In the work presented here, the whole domain is subdivided by a fixed-grid and the elements are identified as fluid, solid or fluid-solid elements. The boundary movement continuously updates the state of the elements in the grid. The coupling is accomplished by transferring forces, induced by the fluid, to the solid solver that causes the solid to deform and move. The movement of the solid adds a momentum to the fluid in the direction of its movement. The system is updated in a time marching scheme.

2. Fixed-grid FEM

Fixed grid (FG) methodology was first introduced by Garcia and Steven [22] as an engine for numerical estimation of two-dimensional elasticity problems. The advantages of using FG are simplicity and speed at a permissible level of accuracy [26]. Two-dimensional FG has proven effective in approximating the strain and stress field with low requirements of time and computational resources.

FG consists of dividing the bounding box of the topology of an object into a set of equally sized cubic elements. Elements are assessed to be inside (I), outside (O) or neither inside nor outside (NIO) of the object.

There are different approaches for the approximation of the NIO elements. One of them is by assigning different material properties to the inside and outside medium transforming the problem into a multi-material elasticity problem. As a result of the subdivision NIO elements have non-continuous properties. They can be approximated in different ways which range from simple setting of NIO elements as I or O, depending on how much of the element is inside the domain, to complex non-continuous domain integration. If homogeneously averaged material properties are used to approximate the NIO element, the element stiffness matrix can be computed as a factor of a standard stiffness matrix thus reducing the computational cost of creating the global stiffness matrix.

An additional advantage of FG is found when accomplishing re-analysis, since there is no need to recompute the whole stiffness matrix when the geometry changes. This feature is particularly useful in FSI problems because of the continuous deformation of the solid domain.

A novel technique based on the use of Heaviside function to approximate the stiffness of the NIO elements is used in this work. A Heaviside function is a function that takes the value of one if a point where it is evaluated is inside the solid domain and zero if the point is outside:

$$H(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad \boldsymbol{x} \in \Omega, \\ 0 & \text{if} \quad \boldsymbol{x} \notin \Omega\,. \end{cases}$$

In this way, the stiffness matrix for an element on the boundary (NIO) will be given by

$$[K]^{NIO} = \int_{NIO} H(\boldsymbol{x})[B^T][C][B]\, d\Omega. \tag{1}$$

As a result the accuracy of the method is improved.

3. Lattice Boltzmann method

From the statistical mechanics point of view, the distribution function $f(\boldsymbol{x},\boldsymbol{p},t)\, d\boldsymbol{x}\, d\boldsymbol{p}$ gives the probability of finding a particular molecule with a given position and momentum. The distribution function at a time $t+dt$ can be expressed in terms of the known position and momenta $(\boldsymbol{x},\boldsymbol{p})$ as

$$f(\boldsymbol{x}+d\boldsymbol{x},\boldsymbol{p}+d\boldsymbol{p},t+dt)\, d\boldsymbol{x}d\boldsymbol{p} = f(\boldsymbol{x},\boldsymbol{p},t)\, d\boldsymbol{x}d\boldsymbol{p} + [\Gamma^+ - \Gamma^-]\, d\boldsymbol{x}\, d\boldsymbol{p}\, dt, \tag{2}$$

where $[\Gamma^+ - \Gamma^-]\, d\boldsymbol{x}\, d\boldsymbol{p}\, dt$ is the contribution due to the collision of the particles. Using the first order terms of a Taylor series expansion of the left hand side of Eq. (2) give the Boltzmann equation:

$$\boldsymbol{u} \cdot \nabla_x f + \boldsymbol{F} \cdot \nabla_p f + \frac{\partial f}{\partial t} = (\Gamma^+ - \Gamma^-)\, dt, \tag{3}$$

where ∇_x is $(\partial/\partial x_\alpha, \partial/\partial x_\beta, \ldots)$, ∇_p is $(\partial/\partial p_\alpha, \partial/\partial p_\beta, \ldots)$, and $\boldsymbol{F}$ is the net force (internal plus external). Although this equation was obtained using an ideal gas as reference it also can be derived for an arbitrary chemical component [27]. Eq. (3) is a non-linear integral differential equation and it is particularly complicated to solve. With the lattice Boltzmann methods (LBM) an approximation to Eq. (2) is found instead of trying to solve (3). This is made from the particle perspective. Notice that right hand side of Eq. (2) contains two terms the first one define the notion of *stream* and the second the notion of *collide*. These notions are central to the lattice Boltzmann method.

The lattice Boltzmann models simplify Boltzmann's original concept by reducing the number of possible particle spatial positions and momenta from a continuum to a few possible

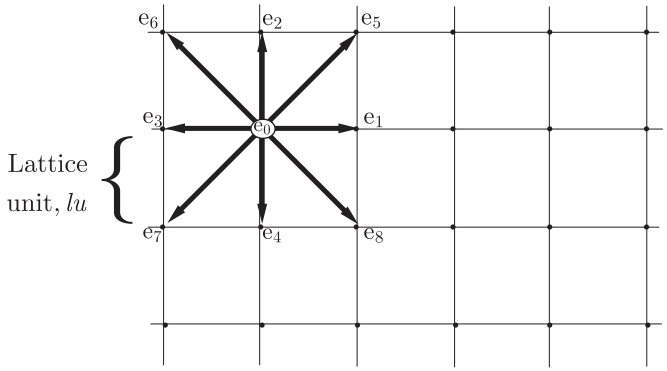

Fig. 1. LBM Cartesian grid and vectors of the D2Q9 lattice.

results, also the time is discretised into *time steps*. In the D2Q9 model used here particle positions are confined to the nodes on a lattice. Variations of momenta are reduced to nine directions and three magnitudes, mass is reduced to a single particle mass. Fig. 1 shows the Cartesian lattice and velocities e_i where $i=0,\ldots,8$, e_0 represents particles at rest.

In order to comply with the isotropy condition, the D2Q9 model must have the following velocities magnitudes e_i and the corresponding weights ω_i

$$e_i=\begin{cases}0 & \text{for } i=0,\\ 1 & \text{for } i=1,2,3,4,\\ \sqrt{2} & \text{for } i=5,6,7,8,\end{cases}\quad \omega_i=\begin{cases}4/9 & \text{for } i=0,\\ 1/9 & \text{for } i=1,2,3,4,\\ 1/36 & \text{for } i=5,6,7,8.\end{cases}$$

3.1. The BGK model

The BGK (Bhatnagar–Gross–Krook) model of the collision operator was proposed by Qian et al. [28] as a simplified collision operator similar to the one proposed for the Boltzmann equation by Bhatnagar et al. [29]. Incorporating this collision operator into Eq. (2) the lattice Boltzmann BGK model is obtained

$$f_i(\boldsymbol{x}+\boldsymbol{e}_i,t+1)=\left(1-\frac{1}{\tau}\right)f_i(\boldsymbol{x},t)+\frac{1}{\tau}f_i^{eq}(\boldsymbol{x},t), \tag{4}$$

where τ is a free parameter known as the *relaxation time* and f_i^{eq} is the equilibrium distribution function of particles. The equilibrium distribution is used to simulate the collision between fluid particles. It can be derived from the Maxwell–Boltzmann velocity distribution from statistical mechanics as [30]

$$f_i^{eq}(\boldsymbol{x})=\omega_i\rho(\boldsymbol{x})\left[1+3\frac{\boldsymbol{e}_i\cdot\boldsymbol{u}}{c^2}+\frac{9(\boldsymbol{e}_i\cdot\boldsymbol{u})^2}{2c^4}-\frac{3\boldsymbol{u}^2}{2c^2}\right], \tag{5}$$

in which $\rho(\boldsymbol{x})$ is the fluid density and c is the lattice speed of the sound. This equilibrium distribution preserves mass and momentum [31]. It is valid for low Reynolds numbers and can be modified to simulate different fluids like plasmas, non-Newtonian fluids, and other physical situations like high Reynolds numbers.

3.2. Macroscopic variables

According to *statistical mechanics* rules [27] the density and velocity can be defined as

$$\rho=\sum_{i=0}^{8}f_i,\quad u=\frac{1}{\rho}\sum_{i=0}^{8}f_ie_i. \tag{6}$$

Numerical viscosity, which is not related to the physical viscosity, but to the relaxation parameter, is given by [27]:

$$v_{lb}=\frac{1}{3}\left(\tau-\frac{1}{2}\right). \tag{7}$$

These equations relate discrete microscopic velocities to a continuum macroscopic velocities that represent the fluid's motion.

3.3. Boundary conditions

The lattice-Boltzmann algorithm consists of two basic steps streaming and collision. In the streaming step direction-specific densities are moved to the nearest cell in the direction they are pointing. Then the equilibrium functions are computed for each node in the grid and the functions are updated for all internal nodes. In case the node lies inside a solid body, the corresponding boundary conditions must be applied. The standard LBM includes bounce-back for the solid walls, Von Newman for the flux and Dirichlet for the pressure. From all the boundary conditions, special treatment is given in this paper to curved and moving boundaries, and force computation at solid boundaries.

3.3.1. Moving boundary

Moving an object inside the fluid is prone to break the continuity of the model if it is not done properly. This problem is solved by (i) computing a new direction-specific density each time the boundary is moved and (ii) computing a new momentum added by the solid to the fluid in the direction it moves [32]. The new direction-specific densities are given by

$$f_{\bar{i}}=f_i-2\rho\frac{w_i(\boldsymbol{c}_i\cdot\boldsymbol{u})}{c_s^2}, \tag{8}$$

where $f_{\bar{i}}$ are the bounced direction-specific densities f_i with $e_i=-e_i$, $\boldsymbol{u}$ is the velocity of the moving obstacle inside the fluid and c_s is the speed of the sound in the medium, that for the model we are using is $1/\sqrt{3}$ [33].

3.3.2. Curved boundary conditions

The procedure to incorporate curved boundaries to the lattice Boltzmann method is based on the bounce back scheme and interpolations as proposed by Bouzidi et al. [34]. If q is the distance between the fluid node x_f and the boundary x_w, see Fig. 2, then the interpolation will depend on $q<1/2$ or $q\geq 1/2$. Notice that the interpolations have to be calculated before streaming and bounce-back collision and after streaming and bounce-back collision, respectively. This leads to the following interpolation formulas:

$$f_{\bar{i}}(x_f,t)=\begin{cases}q(1+2q)\hat{f}_i(x_f,t)+(1-4q^2)\hat{f}_i(x_{ff},t)-q(q-2q)\\ \quad\hat{f}_i(x_{fff},t) & \text{if } q<1/2,\\ \dfrac{1}{q(2q+1)}\hat{f}_i(x_f,t)+\dfrac{2q-1}{1}\hat{f}_i(x_{ff},t)\\ \quad-\dfrac{2q-1}{2q+1}\hat{f}_i(x_{fff},t) & \text{if } q\geq 1/2,\end{cases} \tag{9}$$

where $\hat{f}_i$ is the post-collision distribution and x_f, x_{ff} and x_{fff} are the fluid nodes used in the interpolation to find the value of f at the boundary point x_w.

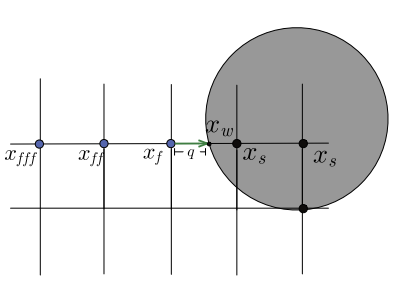

Fig. 2. Curved boundaries in LBM.

3.3.3. Force evaluation

The force is calculated using the *momentum exchange algorithm* (*MAE*) as proposed by Caiazzo [35]. MAE allows to evaluate the force at a solid boundary using directly the variables of LBM. It considers the momentum transferred to the solid from each boundary fluid node, along a link joining it to a solid node given by the difference between the distributions moving in opposite directions as seen in Fig. 3 and according to the equation below:

$$F_i = e_i \hat{f}_i(x_s, dt) + e_i f_i(x_f, dt+1), \tag{10}$$

which in fact corresponds to $\boldsymbol{F} = \Delta \boldsymbol{p}/\Delta t$. Note that it is needed to sum over all links i to get the total *force* excerpted by the node x_f.

3.4. Boundary coupling

The coupling of the solvers is made via boundary conditions. As both solvers use a fixed Cartesian grid only one mesh is needed. In this way fluid and solid nodes perfectly fit within the same grid, as seen in Fig. 4 thus avoiding the need of interpolation to match fluid nodes to solid nodes. The way to transfer fluid variables from the fluid domain to the solid obstacle is via the force exerted on the links between a fluid node and a solid node.

Due to the microscopic nature of the LBM, a unit conversion is necessary to scale the problem from LB units to physical units used by the FG-FEM solver. This is accomplished with similarity analysis by setting the drag coefficient equal in both systems [15]. This is

$$F_{ph} = F_{lb} \frac{p_{ph} v_{ph}^2 L_{lb}}{\rho_{lb} v_{lb}^2 L_{ph}}, \tag{11}$$

where *ph* stands for the physical system used by the structural solver, *lb* means lattice units and L is the reference length. After the units have been scaled the structural solver returns the displacements suffered by the object due to the force of the fluid. This geometry is then updated and send back to the fluid solver. The process should be repeated until the system reaches steady state or until some user time definition.

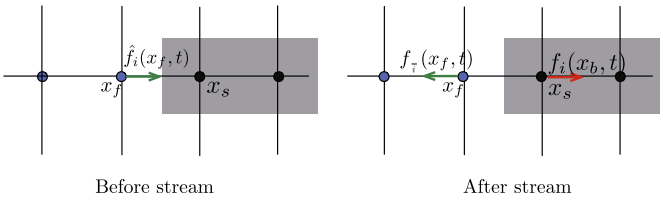

Fig. 3. Momentum exchange.

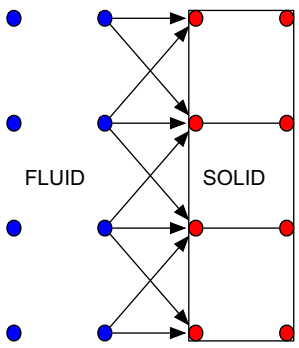

Fig. 4. Fluid and solid nodes.

4. Algorithm summary

A summary of the algorithm implemented in this work is presented in Algorithm 1. The main loop of the algorithm considers the fluid and solid computation for each time step. Notice that in the functions stream and collide the same operation is applied to each lattice node and therefore it is fully data parallel and convenient for a GPU implementation. Once the force is computed for each time step it is transferred to the FGFEM solver where the deformation is computed and the geometry is updated in the fluid domain.

Algorithm 1. Lattice Boltzmann FSI algorithm.

while (TRUE) **do**
Calculate macroscopic variables: $\rightarrow \rho = \sum_{i=0}^{8} f_i$ and $u = \frac{1}{\rho}\sum_{i=0}^{8} f_i e_i$
Stream: $\rightarrow f(\boldsymbol{x}+\boldsymbol{e}, t+1) - f(\boldsymbol{x}, t) = \Omega(\boldsymbol{x}, t)$
Move boundary: $\rightarrow f_{\bar{i}} = f_i - 2\rho w_i (\boldsymbol{c}_i \cdot \boldsymbol{v})/c_s^2$
Apply boundary conditions
Collide: $\rightarrow f_i^{eq}(x) = \omega_i \rho(x)\left[1 + \frac{3(e_i \cdot u)}{c^2} + \frac{9(e_i \cdot u)^2}{2c^4} - \frac{3u^2}{2c^2}\right]$
Calculate Force: $\rightarrow F_i = e_i \hat{f}_i(x_s, dt) + e_i f_i(x_f, dt+1)$
Transfer force to structural solver
end while

5. GPU implementation

The two main reasons to use a GPU in CFD are speed and price. Mapping the lattice Boltzmann method to GPU hardware is very straightforward, and it has been a key point of the popularisation to the method. The lattice Boltzmann method applies the same operation to each node independently, so it does not need to implement a complex communication scheme, which makes the method very parallel. Fig. 5 summarise the algorithm. This is a "perfect" match for the *single instruction multiple data* (SIMD) paradigm that is used to program GPUs, and the reason why a GPU seems like a logical architecture to implement the method.

LBM and FG-FEA were implemented using CUDA [36,37]. A CUDA kernel is used for each mayor function of the method, this way it is possible to launch a thread for each node of the grid. Also, in order to increase speed the direction-specific densities were stored in textures for the streaming step as seen in Fig. 6.

While in the LBM CUDA was used for the stream and collide functions, in the FG-FEM structural solver was used to solve the linear system of equations. That was implemented using CULA, a GPU-accelerated linear algebra library that utilises CUDA. It contains a LAPACK interface comprised mathematical routines from the industry standard for computational linear algebra.

6. Examples

6.1. Full rigid body FSI

The first case consists of a circular rigid body moving on a fluid channel. Initially the fluid is at rest and the solid cylinder moves across it at a speed of 2 cm/s. The size of the domain is 2 m long by 1 m height and the radius of the cylinder is 7 cm. The aim of the experiment was to compare the perturbation of the fluid with the case when the object is still and the fluid moves in the opposite direction. The domain was discretised with a mesh of 200×400 elements and 80 000 nodes and it was solved using a NVIDIA Geforce 8800-GT GPU. The comparison was made against

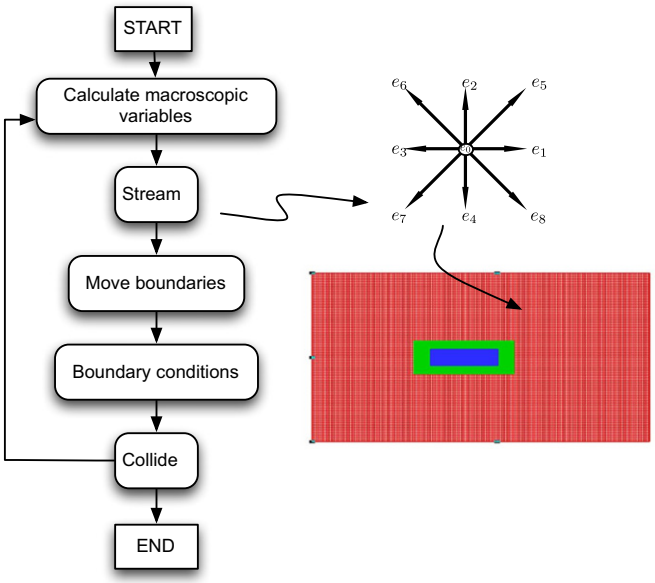

Fig. 5. LBM runs the same algorithm over each node of grid.

FEM using Comsol Multiphysics [38] with a mesh of 6800 triangular elements.

Fig. 7 shows two different time steps of the LBM simulation. The object is moving from right to left at a speed of 2 cm/s. The steady state is obtained after some time iterations and is shown in Fig. 7(b). The velocity field obtained by FEM is shown in Fig. 8. This was a steady state simulation with a static frame of reference. It can be appreciated that the velocity field of the moving cylinder at a time when it has reached a steady state, as shown in Fig. 7(b), is similar to the velocity field of the case of a static object with a fluid flowing at 2 cm/s. The experiment should give the same qualitative results. The difference is that in case one, the frame of reference is fixed on the ground and the cylinder moves across the fluid. In the second case the frame of reference is moving with the cylinder and therefore it looks still while the fluid moves around the cylinder. The LBM implementation took approximately 3 s to arrive to a steady state. The FEM steady state solver took 19 s to converge within 0.0001 for the velocity residual. A comparison of the velocity field among both methods gives an average error of 6%. However, at points near the cylinder the difference was found 47%. This is a subject of further study.

Although the frame of reference in COMSOL is static, there are many implementations of the finite element method for

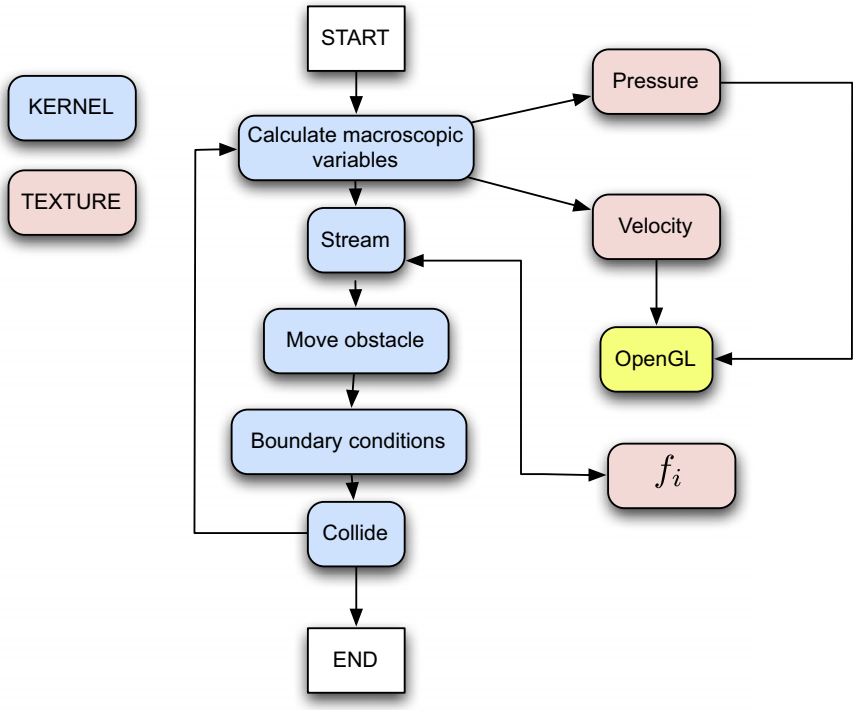

Fig. 6. CUDA implementation.

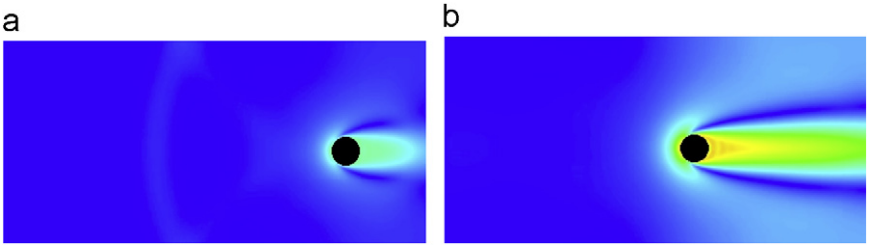

Fig. 7. Different velocity fields for the fluid when the object is moving on an initially still fluid. (a) Velocity profile at time t. (b) Velocity profile at time $t+x\delta t$.

fluid–structure interaction for rigid and deformable solids that use a moving frame of reference. Tezduyar [39] has been used stabilised formulations to simulate problems like parachute behaviour. The performance of both implementations is of the same order of magnitude and it is lower than the performance of the parallel LB implementation.

6.2. Elastic object FSI

The elastic object FSI test consists of a channel, with solid walls on the top and bottom, and a fluid that flows from left to right with Reynolds number of 130. An elastic beam is located at the centre of the bottom wall and deforms as it interact with the passing flow. The mesh consists of a rectangular grid of 400×100 elements for the fluid domain and 4×20 elements for the solid domain as shown in Fig. 9. Performance tests were made with an Intel i7 8 GB DDR2 of RAM. The GPU tests were made on the same machine running on a NVIDIA Geforce 8800 GT.

The numerical results of the FG-LBM were compared to a similar case setup in ANSYS Workbench which uses finite volume method for fluid dynamics and FEM for the solid part (Fig. 10). The Fluid field can be appreciated for both cases in Fig. 9. It can be observed the close similarity of the fields. On the solid domain, the maximum deformation of the beam differs in 5% when the displacement by the two methods is compared. The FG implementation is based on linear elastic formulation giving valid results for small displacements. However, the force exerted by the fluid to the structure is not large enough to produce large deformations in the solid. Therefore, the small displacements model assumed is valid in this case. Future work is aimed to consider a large displacement formulation in order to accurately predict the deformations of the solid within the fluid domain.

Due to the low dimension of the mesh in the structural domain of this example the GPU computation of the structural deformation took longer time than the CPU computation. This is due to the extremely expensive CPU–GPU memory transfers and the small systems of equations that had to be solved. Therefore, in this example it used a mixed GPU–CPU solver, GPU for the fluid domain and CPU for the structural analysis. On the other hand, FG-FEM has a restriction on the minimum number of nodes in order to produce accurate results that have to be taken into account when solving complex structures. In those cases it could be necessary to use a higher density mesh for the structural domain.

The performance of the FG-LBM with the mixed GPU–CPU solver was approximately 60 time iterations per second, which was enough to display the solution in real-time. The length of each time iteration was 0.001 s so each simulation second is equivalent to 0.3 s of the physical time.

Table 1 shows the elapsed time to reach a steady state by both solvers. ANSYS Workbench solution uses a steady state method while FG-LBM is a transient method. The time reported for ANSYS is the time to reach a tolerance of 1×10^{-4}. ANSYS Workbench solutions were sequential and parallel using two cores. The solution times presented as a reference and do not constitute a fare comparison as the solution in ANSYS was not executed in parallel. Nevertheless, it can be observed a speed up of 222 times with one core and 78 times with two cores.

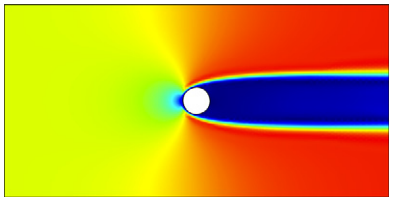

Fig. 8. Velocity field when the cylinder is still and the fluid moves around the cylinder. Results obtained with FEA [38].

Fig. 9. Fluid and structure domains. (a) Fluid domain. (b) Solid domain.

7. Conclusions

This paper presented a lattice Boltzmann/fixed-grid FEA parallel GPU implementation for the fluid–structure interaction problems with low Reynolds and small displacement fields in two dimensions. The coupling is made via boundary conditions and considering the effect of moving boundaries in the fluid. In spite of the Cartesian nature of the grid curve boundaries was considered in both the lattice Boltzmann and the fixed-grid FEA methods.

Two examples were presented. In the first example a moving cylinder in a still fluid was compared against a static cylinder in a moving fluid. The results show agreement between both cases with a difference of 5% on average in the velocity fields. However, at points over the cylinder there was a higher difference which is a subject of further study. In the second example an elastic beam interacting with a moving fluid was simulated and the results compared against a finite volume/finite element method. The difference among solutions was found of the order of 10^{-3} and therefore it can be concluded that they are equivalent.

It can be observed a great benefit in computational time when using the GPU implementation of the method. The solution time of each time step was smaller than 1/30 of a second thus allowing a real-time visualisation of the simulation for the cases tested.

Care must be taken as if the size of the structural model is not large enough then there is not advantage on using the GPU as the

Table 1
Speed in seconds achieved by the GPU and CPU solvers.

Solver	Time
ANSYS Workbench (1 core)	369
ANSYS Workbench (2 cores)	130
FG-LBM GPU–CPU	1.66

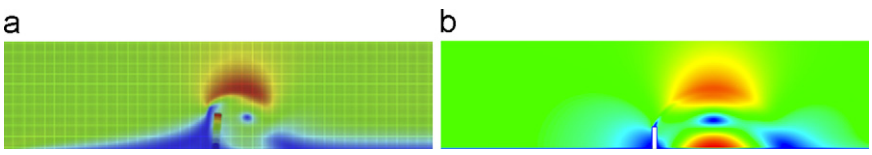

Fig. 10. Comparison of LBM/FG-FEM with ANSYS Workbench. (a) Velocity field FG-LBM. (b) Velocity field workbench.

CPU/GPU communication time can be greater than the solution time. Future work includes non-linear solid deformation that considers the effect of large deformations, variable mesh resolution for fluid and solid structures, and a three-dimensional implementation of the method.

References

[1] M. Heil, An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems, Computer Methods in Applied Mechanics and Engineering 193 (2004) 1–23.

[2] K.-J. Bathe, H. Zhang, Finite element developments for general fluid flows with structural interactions, International Journal for Numerical Methods in Engineering 60 (2004) 213–232.

[3] H.G. Matthies, J. Steindorf, Partitioned strong coupling algorithms for fluid–structure interaction, Computers and Structures 81 (8–11) (2003) 805–812 ISSN 0045-7949, 10.1016/S0045-7949(02)00409-1 K.J Bathe 60th Anniversary Issue.

[4] H.G. Matthies, R. Niekamp, J. Steindorf, Algorithms for strong coupling procedures, Computer Methods in Applied Mechanics and Engineering 195 (17–18) (2006) 2028–2049 ISSN 0045-7825, 10.1016/j.cma.2004.11.032, fluid–structure interaction.

[5] R. van Loon, P.D. Anderson, F.N. van de Vosse, S.J. Sherwin, Comparison of various fluid–structure interaction methods for deformable bodies, Computers and Structures 85 (2007) 833–843.

[6] P.G. Hu, L. Xue, S. Mao, R. Kamakoti, H. Zhao, N. Dittakavi, Z. Wang, Q. Li, K. Ni, M. Brenner, Material point method applied to fluid–structure interaction (FSI)/aeroelasticity problems, In: AIAA (Ed.), 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010.

[7] P. Gamnitzer, W. Wall, An ale-chimera method for large deformation fluid structure interaction, in: P.J. Wesseling, E. Onate (Eds.), Proceedings of the European Conference on Computational Fluid Dynamics, ECCOMAS CFD, The Netherlands, TU Delft, 2006, pp. 1–14.

[8] A. Gerstenberger, W. Wall, An extended finite element method/Lagrange multiplier based approach for fluid structure interaction, Computer Methods in Applied Mechanics and Engineering 197 (2008) 1699–1714.

[9] A. Legay, J. Chessa, T. Belytschko, An Eulerian Lagrangian method for fluid structure interaction based on level sets, Computer Methods in Applied Mechanics and Engineering 195 (2006) 2070–2087.

[10] E. Walhorn, A. Kölke, B. Hübner, D. Dinkler, Fluid structure coupling within a monolithic model involving free surface flows, Computers and Structures 83 (2005) 2100–2111.

[11] Y. Maday, C. Mavriplis, A.T. Patera, Nonconforming mortar element methods: application to spectral discretizations, In: Domain Decomposition Methods (Los Angeles, CA), SIAM, Philadelphia, PA, 1989, pp. 392–418.

[12] W. Wall, P. Gammnitzer, A. Gerstenberger, Fluid structure interaction approaches on fixed grids based on two different domain decomposition ideas, International Journal of Computational Fluid Dynamics 22 (2008) 411–427.

[13] C. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 1–39.

[14] Y.W. Kwon, Development of coupling technique for IBM and FEM for FSI application, Engineering Computations 23 (8) (2006) 860–875.

[15] S. Geller, J. Tolke, M. Krafczyk, Lattice Boltzmann methods on quadree-type grids for fluid structure interaction, fluid structure interaction, modelling, simulation and optimisation, in: Lecture Notes in Computational Science and Engineering, vol. 53, 2006.

[16] S. Kollmannsberger, S. Geller, A. Düster, J. Tölke, C. Sorger, M. Krafczyk, E. Rank, Fixed-grid fluid structure interaction in two dimensions based on a partitioned lattice Boltzmann and p-FEM approach, International Journal for Numerical Methods in Engineering 79 (2009) 817–845.

[17] C. Zhao, T. Nishiyama, A. Murakami, Numerical modelling of spontaneous crack generation in brittle materials using the particle simulation method, Engineering Computations 23 (2006) 566–584.

[18] C. Zhao, B.E. Hobbs, A. Ord, P.A. Robert, P. Hornby, S. Peng, Phenomenological modeling of crack generation in brittle crustal rocks using the particle simulation method, Journal of Structural Geology 29 (2007) 1034–1048.

[19] E. Oñate, S.R. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method: an overview, International Journal of Computational Methods 1 (2) (2004) 267–307.

[20] S. Idelsohn, E. Onate, F. Del Pin, N. Calvo, Fluid structure interaction using the particle finite element method, Computer Methods in Applied Mechanics and Engineering 195 (17–18) (2006) 2100–2123.

[21] C. Zhao, B.E. Hobbs, A. Ord, S. Peng, Particle simulation of spontaneous crack generation associated with the laccolithic type of magma intrusion processes, International Journal for Numerical Methods in Engineering 75 (2008) 1172–1193.

[22] M.J. Garcia, G.P. Steven, Fixed grid finite elements in elasticity problems, Engineering Computations 16 (2) (1999) 145–164.

[23] H.A. Kim, M.J. Garcia, G. Steven, O. Querin, Y. Xie, Fixed grid finite element analysis in evolutionary structural optimisation, Engineering Computations 17 (4) (2000) 427–439.

[24] Y. Liu, F. Jin, Q. Li, S. Zhou, A fixed-grid bidirectional evolutionary structural optimization method and its applications in tunnelling engineering, International Journal for Numerical Methods in Engineering 73 (2008) 1788–1810.

[25] M.J. Garcia, P. Boulanger, M. Henao, Structural optimization of as-built parts using reverse engineering and evolution strategies, Structural and Multidisciplinary Optimization 35 (2008) 541–550.

[26] M. Garcia, G. Steven, Displacement error for fixed grid finite FEA elasticity problems, in: III Congreso Colombiano en Elementos Finitos y Modelación Matemática, Medellín, Colombia, 1999.

[27] M. Sukop, D. Thorne, Lattice Boltzmann Modeling. An Introduction for Geoscientists and Engineers, Springer, 2005.

[28] Y.H. Qian, D. d'Humières, P. Lallemand, Lattice BGK models for Navier–Stokes equation, Europhysics Letters 17 (1992) 479–484. doi:10.1209/0295-5075/17/6/001.

[29] P. Bhatnagar, E. Gross, M. Krook, A model for collisional processes in gases i: small amplitude processes in charged and neutral one-component system, Physical Review 94 (1954) 511.

[30] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annual Review of Fluid Mechanics 30 (1998) 329–364.

[31] E. Magnus Viggen, The lattice Boltzmann method with applications in acoustics, Master's Thesis, NTNU, 2009.

[32] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part i. Theoretical foundation, Journal of Fluid Mechanics 271 (1994) 285–309.

[33] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford University Press, 2001.

[34] M'hamed Bouzidi, Mouaouia Firdaouss, Pierre Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Phys. Fluids 13, 3452 (2001), doi:10.1063/1.1399290 (8 pages) pp. 452–459.

[35] M. Junk, A. Caiazzo, Boundary Forces in Lattice Boltzmann: Analysis of Momentum Exchange Algorithm, Elsevier Science, 2006.

[36] NVIDIA, Nvidia CUDA programming guide, 〈http://www.nvidia.com/cuda〉, 2 2010.

[37] J. Nickolls, I. Buck, M. Garland, K. Skadron, Scalable parallel programming with cuda, Queue 6 (2008) 40–53 ISSN 1542-7730.

[38] COMSOL, Comsol Multiphysics User Guide version 4.0 a, COMSOL Inc., June 2010.

[39] T.E. Tezduyar, Finite element methods for fluid dynamics with moving boundaries and interfaces, in: Encyclopedia of Computational Mechanics Fluids, vol. 3, John Wiley & Sons, 2004.