ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ALMA MATER STUDIORUM - UNIVERSITÁ DI BOLOGNA
SCUOLA DI DOTTORATO IN INGEGNERIA CIVILE ED ARCHITETTURA

Dottorato di Ricerca in Ingegneria Strutturale ed Idraulica

Tesi di dottorato

Fluid-structure interaction by a coupled lattice Boltzmann-finite element approach

Settore Scientifico-Disciplinare ICAR-08/B2

Tutor:
Prof. Francesco Ubertini

Dottorando:
Alessandro De Rosis

Coordinatore del Dottorato:
Prof. Erasmo Viola

Anno Accademico 2012-2013 — Ciclo XXV

a Francesco,
per essere stato mio amico

Contents

Ringraziamenti

Agnosco veteris vestigia flammae...

Il primo ringraziamento, doveroso da un lato, voluto dall'altro, va al Professor Francesco Ubertini, per avere investito tempo, pazienza e fatica in me, senza farmelo mai pesare, per aver suscitato in me l'interesse verso la tematica dell'interazione fluido-struttura, per la comprensione dimostrata in molteplici occasioni, ma, soprattutto, per non aver mai dubitato. Ringrazio, inoltre, il Professor Stefano Ubertini e il Dottor Giacomo Falcucci, per avermi introdotto all'universo parallelo del lattice Boltzmann, per le giornate (e serate) passate a districarci tra le boundary conditions e per avermi trattato, spesso e volentieri, fraternamente. Tengo molto a ringraziare il Direttore dell'Istituto per le Applicazioni del Calcolo, Dottor Sauro Succi, per quanto ho imparato leggendo il suo libro e le sue pubblicazioni, nonchè per avermi suscitato l'interesse nei confronti della dinamica cooperativa.Un ringraziamento va anche al Dottor Stefano de Miranda, per l'amicizia dimostratami nel corso di questo triennio passato tra una convenzione e l'altra. Un pensiero non può che correre verso il gruppo di lavoro del LABMEC dell'Università della Calabria: i Professori Aristodemo, Casciaro e Garcea, il Dottor Leonardo Leonetti, il Dottor Antonio Madeo e il Dottor Giuseppe Zagari, per essere stati i primi a suscitare in me l'interesse per la Scienza delle Costruzioni, per la Meccanica Computazionale e per il mondo della programmazione in ambiente C++. Ringrazo, infine, il Professor Erasmo Viola, sui libri del quale mi sono formato, e che è stato a dir poco un piacere conoscere.

Sommario

In questa tesi viene proposta una strategia per modellare il comportamento dei fluidi e la loro interazione con corpi deformabili. Un approccio classico a questo tipo di problemi si basa sulla soluzione delle equazioni di Navier-Stokes, che governano il moto del fluido a scala macroscopica. Qui, in contrasto a questo classico approccio, il dominio fluido è modellato utilizzando il metodo lattice Boltzmann, analizzando conseguentemente la dinamica del fluido da un punto di vista prettamente mesoscopico. È stato dimostrato che la soluzione fornita da tale metodo equivale a risolvere le equazioni di Navier-Stokes per un flusso incomprimibile con un ordine di accuratezza pari al secondo. Il metodo lattice Boltzmann è stato preferito all'approccio classico per diversi motivi, come il minor costo computazionale e la maggiore modularità nello sviluppo di un software numerico in ambiente C++. L'efficacia dell'implementazione è stata valutata rispetto a vari problemi di riferimento che coinvolgono corpi rigidi investiti da un fluido.
La struttura è modellata con elementi finiti di trave snella a comportamento meccanico elastico lineare. Il regime di grandi spostamenti è descritto utilizzando la formulazione corotazionale. Le equazioni del moto sono risolte utilizzando il metodo Time Discontinuous Galerkin. La scelta di questo schema di integrazione nel tempo rispetto agli algoritmi standard alla Newmark è motivato dalle migliori proprietà in termini di stabilità, convergenza e accuratezza.
Vengono quindi utilizzate due diverse metodologie di soluzione, una per il dominio fluido e l'altra per la parte strutturale, che coinvolgono equazioni, formulazioni e processi risolutivi distinti. Questi due metodi hanno bisogno

di comunicare e trasferire informazioni l'uno all'altro, come sforzi, velocità e spostamenti. Al fine di garantire uno scambio continuo, efficace e reciproco di informazioni, una strategia di accoppiamento, composta da tre differenti algoritmi, è stata sviluppata. In particolare, l'efficacia dei tre algoritmi è mostrata in termini di energia dell'interfaccia prodotta artificialmente dal parziale soddisfacimento delle condizioni di compatibilità e di equilibrio all'interfaccia fluido-struttura.
Il presente approccio accoppiato viene utilizzato per risolvere diversi problemi di interazione fluido-struttura, come travi a mensola immerse in un fluido viscoso, l'impatto degli scafi delle navi sulla superficie libera marina, il flusso sanguigno in una parete vascolare deformabile e anche il moto di due ali che simulano il decollo di una farfalla. I buoni risultati ottenuti in ciascuna applicazione evidenzia l'efficacia della metodologia proposta e del software sviluppato in ambiente C++ nell'approcciarsi a qualsiasi problema bidimensionale di interazione fluido-struttura.
Alcuni risultati presenti nella tesi sono stati oggetti di partecipazioni a congressi e pubblicati su riviste internazionali [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

Abstract

In this thesis, a strategy to model the behavior of fluids and their interaction with deformable bodies is proposed. Usually, a classical approach to such kind of problems is based on the solution of the Navier-Stokes equations, governing the fluid at a macroscopic scale. Here, in opposition to the classical approach, the fluid domain is modeled by using the lattice Boltzmann method, thus analyzing the fluid dynamics by a mesoscopic point of view. It has been proved that the solution provided by this method is equivalent to solve the Navier-Stokes equations for an incompressible flow with a second-order accuracy. The lattice Boltzmann method has been preferred to the classical approach for several reasons, such as its extreme simplicity, the lower computational cost involved and the great attitude to be implemented in a C++ numerical software. The effectiveness of the implementation is tested against several benchmark problems involving rigid bodies invested by a fluid.
Slender elastic structures idealized through beam finite elements are used. Large displacements are accounted for by using the corotational formulation. Structural dynamics is computed by using the Time Discontinuous Galerkin method. The choice of this time integration scheme over standard Newmark's algorithms is motivated by the better properties in terms of stability, convergence and accuracy.
Therefore, two different solution procedures are used, one for the fluid domain and the other for the structural part, respectively. These two solvers need to communicate and to transfer each other several information, i.e. stresses, velocities, displacements. In order to guarantee a continuous, ef-

fective, and mutual exchange of information, a coupling strategy, consisting of three different algorithms, has been developed and numerically tested. In particular, the effectiveness of the three algorithms is shown in terms of interface energy artificially produced by the approximate fulfilling of compatibility and equilibrium conditions at the fluid-structure interface.
The proposed coupled approach is used in order to solve different fluid-structure interaction problems, i.e. cantilever beams immersed in a viscous fluid, the impact of the hull of the ship on the marine free-surface, blood flow in a deformable vessels, and even flapping wings simulating the take-off of a butterfly. The good results achieved in each application highlight the effectiveness of the proposed methodology and of the C++ developed software to successfully approach several two-dimensional fluid-structure interaction problems.
Some results reported in the present thesis have been published in conference proceedings and international journals, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

Chapter 1

Introduction

Sommario

La fluido-dinamica computazionale è un ambito di ricerca che si prefigge lo scopo di usare strumenti di calcolo, metodi e algoritmi numerici per risolvere problemi che coinvolgono fluidi e l'eventuale interazione di questi con solidi. La fluido-dinamica computazionale puó essere studiata e analizzata a vari livelli di osservazione. Un primo livello è detto macroscopico e si concretizza nel risolvere con classiche tecniche ai volumi o agli elementi finiti le equazioni di Navier-Stokes. Un secondo approccio impiega il metodo lattice Boltzmann ed è basato sull'equazione del trasporto di Boltzmann con il suo particolare punto di vista mesoscopico. Il terzo livello consiste nel ricondurre un problema di fluido-dinamica a uno di dinamica molecolare. In questa tesi l'attenzione è stata focalizzata sul metodo lattice Boltzmann allo scopo di risolvere problemi di interazione fluido-struttura. La struttura, intesa come solido deformabile, viene modellata tramite un approccio agli elementi finiti, usando elementi di trave piana geometricamente nonlineare. Il regime di grandi spostamenti viene tenuto in conto tramite la formulazione corotazionale. L'integrazione dell'equazione del moto della struttura viene effettuata tramite il metodo Time Discontinuous Galerkin.

Trattandosi di due metodologie risolutive diverse, una per il dominio fluido e l'altra per il dominio solido, ognuna delle quali possiede un proprio background teorico e diverse modalità implementative, uno dei principali contributi della tesi è stato lo sviluppo di un algoritmo di accoppiamento tale da garantire un continuo, efficace, accurato e mutuo scambio di informazioni tra le due. Il lavoro si è quindi concretizzato nello sviluppo di un codice in ambiente C++ la cui efficacia e la cui robustezza emergono dalla disamina di varie applicazioni.

1.1 The state of the art

Computational fluid dynamics (CFD) is a branch of the fluid dynamics aiming to solve problems involving fluids and their interaction with solids by using numerical algorithms and models. A reliable prediction of the interaction between fluids and structures is extremely important in several industrial, technological, biological and environmental processes. This great interest promoted a huge research effort in the last two decades. CFD can be observed and consequently analyzed through three different levels of view. The first one, also know as *classical* approach to CFD, consists in the solution of the Navier-Stokes equations [14], governing the fluid at a macroscopic scale. A second level employes the lattice Boltzmann (LB) method [15]; it is based on Boltzmann's kinetic equation [16], with its characteristic *mesoscopic* point of view, and not on Navier-Stokes *continuum* assumption. The lowest level approaches the problem from a microscopic molecular point view, thus the macroscopic behavior arises from the dynamics of a system of molecules.
Concerning the classical approach, the solution of the Navier-Stokes equations in a closed form is possible only in some special cases. In any practical applications, these equations are solved by using finite element (FE) or finite volume techniques. Even if such approach is widespread, it presents several drawbacks. First, each classical CFD solver inevitably needs to treat the convective term, involving a huge computational effort, since it is non-local and non-linear. In some cases this term is neglected, that is, for example, the potential flow theory. As it is known, such theory governs the macroscopic behavior of a fluid under the assumptions of irrotational flow and inviscid fluid, representing an ideal condition, usually far from the practical applications. In addition, CFD solvers for the incompressible Navier-Stokes equations need to solve the Poisson equation for the pressure, which is another expensive operation. Moreover, if the interaction with solid bodies is studied, moving meshes are needed, thus requiring, again, high computational costs. These drawbacks suggest to use an alternative approach for the solution of a CFD problem.

1.2 The proposed approach

In this thesis, CFD is employed in order to solve two-dimensional fluid-structure interaction problems and the attention is focused on the mesoscopic level and, consequently, on the LB method [17, 18, 19, 20]. The LB method was successfully employed for several complex fluid dynamics problems, such as large-eddy simulations of turbulent flows [21], multiphase flows [22, 23, 24, 25, 26], non-Newtonian blood flow [27, 28], and even for ferrofluids [29], among the others. Complex boundary geometries were successfully simulated by means of the interpolation scheme described in [30, 31, 32] and recently the LB method has been also used to solve fluid-structure interaction problems (FSI), [33, 34, 7, 35, 10]. As well known, two different approaches, typical of any coupled problem, can be used to tackle fluid-structure interaction: monolithic approach [36] and partitioned approach [37]. Here, the partitioned approach is adopted, since it is the most suited for practical problems, [38]. The basic strategy is to treat separately the fluid and the structural domains and to properly discretize each of them, in order to adopt numerical methods developed and optimized for both computational fluid dynamics and computational structural dynamics. To meet the continuity conditions on the common fluid-structure boundary, an effective procedure should be devised to couple the two solvers.
Structure dynamics is modelled by means of the Finite Element (FE) method. Specifically, linear elastic slender structures idealized by beam finite elements are considered. Large displacements are accounted for by adopting the corotational formulation [39, 40] and time integration is performed by using higher order methods [41]. In particular, the Time Discontinuous Galerkin (TDG) method, implemented according to the two-stage algorithm recently proposed in [42], is adopted. This choice is motivated by its good accuracy and stability properties, that are expected to play an important role in the performance of the whole procedure.
Two different solvers, one for the fluid domain and the other for the solid one, are used. Each one differs from the other in the theoretical background, in the discretization method and, consequently, in the solution procedure. Thus, in order to solve a FSI problem, the two solvers need to communi-

cate. To this purpose, a proper coupling strategy has been implemented. In this thesis, three different coupling algorithms for FSI are considered and their properties in terms of both accuracy and stability are numerically investigated. The same time discretization is adopted for both fluid and structure. The fluid is solved on a fixed grid (lattice) and the structure can move/deform upon the grid, similarly to immersed boundaries [43], with no restriction on matching between fluid and structure nodes (non-boundary-fitted method). This is one of the main advantages of this approach, since no moving meshes are needed that usually require high computational times. In order to account for the exact position of a solid not aligned with the grid, two different curved boundary conditions are used. First, the curved boundary condition [32, 44] represents the so-called interpolated bounce-back scheme. It has been shown that is second-order accurate. Upon structure motion, a simple refill procedure is used to initialize new activated fluid nodes. Secondly, the Immersed Boundary (IB) method [45] is adopted. This method possesses a lower accuracy compared to the previous boundary condition, even if it is very flexible, easy to be implemented and to account for complex geometries. In addition, no refill procedure are needed. To compute the forces acting on the structure, the stress tensor at lattice sites is computed by the simple and effective procedure discussed in [46]. The first coupling algorithm considered, called FELBA Explicit, is very simple and can be classified as standard staggered algorithm. The second coupling algorithm, called FELBA, is similar to the previous one apart from the introduction of a structure predictor, based on the idea proposed in [47, 48]. The third one, called FELBA Implicit [10], is obtained by the previous algorithm by iterating within each time step until a convergence criterion on interface conditions is met. It can be classified as strongly-coupled partitioned algorithm. In the presence of very light structures, Aitken's under-relaxation is used to prevent from potential instability due to added-mass effects [38, 49, 50, 51].
The LB and FE methods, together with proper boundary and initial conditions, and the coupling strategies have been implemented, leading to the development of a C++ code. In order to develop a performant computational tool, special attention was paid to the high performance computing.

Several numerical libraries have been used in order to develop a code able to exploit hardware resources as much as possible. Such code has been preliminarly tested on benchmark problems involving rigid bodies in order to assess the effectiveness of the LB implementation. Then, the interaction with solid bodies has been investigated in several different contexts, showing the ability of the present FSI approach to be applied to several CFD problems.

1.3 The outline of the thesis

The present thesis is organized as follows. In Chapter 2, the LB method is presented, from the continuum background to the discrete implementation. Moreover, the attention is focused on the fluid boundary conditions. Different methods based on Eulerian and Lagrangian descriptions of the immersed solid are presented and compared each other. In Chapter 3, details on the FE structure solver are given. The kinematics of a two-node slender beam element is achieved by using the corotational formulation, whereas the integration of the equation of the solid motion is performed by using the non-linear TDG scheme. In Chapter 4, the coupling algorithms are discussed, focusing on the implementation and on the fulfilling on the interface continuity conditions, both in terms of compatibility and equilibrium. Moreover, the added-mass effect is investigated and a strategy to prevent instabilities is shown. Finally, some numerical tools are presented. The impact of the high performance computing on the factorization of the effective matrix of the TDG scheme is discussed, showing the effect of these tools in the performance of the FSI algorithm. In Chapter 5, in order to test the robustness of the developed software, first benchmark problems are investigated, concerning rigid obstacle invested by an incoming flow; findings in terms of drag coefficients and velocity profiles in several sections are compared to literature values. The accuracy of the interpolated bounce-back scheme and of the IB method is evaluated by computing the aerodynamics coefficients of rigid bodies. Then, the behavior of deformable

beams clamped to rigid boundaries immersed in a viscous fluid is investigated. Chapter 6 contains several applications of the present approach to different FSI problems. An industrial/defense application is investigated: the impact of the hull of a ship on the marine surface, known as hull slamming. Moreover, two applications from biomechanics are considered: the blood flow in a deformable vessel and the lift generation in a flapping wing simulating the vertical take off of a butterfly. The achieved good results show the capability of the present coupled approach to solve very different FSI problems. In Chapter 7, some concluding remarks on the proposed strategy are discussed. Some additional details are given in Appendices A and B.

Chapter 2

Fluid modeling

Sommario

Al continuo l'equazione di Boltzmann descrive l'evoluzione nello spazio e nel tempo di una funzione di distribuzione di particelle le quali si muovono e collidono l'un l'altra. Dall'evoluzione di questa funzione è possibile ricavare delle quantitá di interesse macroscopico. Per esempio, dalla carica e dal calore trasportati dalla funzione di distribuzione delle particelle è possibile risalire alla conducibilitá termina ed elettrica di un mezzo continuo. Nell'equazione di Boltzmann compare un termine detto operatore di collisione che regola il comportamento delle particelle nel momento in cui collidono. A rigore, esso possiede un'espressione differenziale molto complicata e, in virtú di ció, se ne adotta un modello semplificato ideato nel 1954, dove l'operatore di collisione viene espresso in termini di rilassamento a un equilibrio locale. L'equazione di Boltzmann viene discretizzata e risolta su una griglia cartesiana. Se il numero di Mach del modello numerico é sufficientemente piccolo, allora é possibile dimostrare che il metodo lattice Boltzmann approssima la soluzione delle equazioni di Navier-Stokes per flusso incomprimibile con un'accuratezza al secondo ordine. Dettagli sulla formulazione, sul metodo e sulle condizioni al contorno verranno discussi.

2.1 The kinetic theory and Boltzmann's equation

The kinetic theory of gases describes a gas as a large number of small particles (atoms or molecules) in constant random motion, continuously colliding each other. Kinetic theory explains macroscopic properties of gases, such as pressure, temperature, and volume, by considering their molecular composition and motion. For example, this theory is used to compute the electric or heat capacity of a continuous medium by tracking the electric charge or the heat that particles carry with them. The cornerstone of the kinetic theory is Boltzmann's equation.
Consider a set of N molecules moving in a box of volume V. The probability distribution function $f(\boldsymbol{x}, \mathbf{v}, t)$ to find a particle in the position $\boldsymbol{x}$ at time t moving with velocity $\mathbf{v}$ is predicted by Boltzmann's equation [16]. Boltzmann's kinetic equation reads as follows:

$$\dot{f}(\boldsymbol{x}, \mathbf{v}, t) + \mathbf{v}(\boldsymbol{x}, t) \cdot \nabla f(\boldsymbol{x}, \mathbf{v}, t) = Q(\boldsymbol{x}, \mathbf{v}, t), \tag{2.1}$$

where the superimposed dot indicates differentiation with respect to time, the left-hand side represents the molecular streaming motion and the operator Q represents the effect of intermolecular collision. Assuming a dilute gas of point-like massless molecules interacting via a short-range potential, intermolecular interactions can be described in terms of localized binary collisions, with molecules spending most of their life span on independent free trajectories, i.e. no correlations between particles entering a collision. In order to facilitate numerical and analytical solutions of Boltzmann's equation, the complicated non-linear integral collision operator is replaced by a simpler expression. In particular, the Bhatnagar-Gross-Krook model (BGK) is adopted [52], where Q is expressed as a relaxation to a local equilibrium, that is,

$$Q = \frac{f - f^{eq}}{\tau}. \tag{2.2}$$

Mathematically, the local equilibrium is defined as a local distribution function f^{eq}, such that gains and losses are in exact balance so that the collision term is annihilated, $Q(f^{eq}, f^{eq}) = 0$. The relaxation parameter τ is a typical time-scale associated with collision relaxation to the local equilibrium.

The major advantage of the BGK model is that the non-linearity is purely local, while the non-locality is linear.

2.2 The lattice Boltzmann method

Equation 2.1 is discretized in space and time through a finite difference scheme upon a square grid. The particle distribution function can moves only along several imposed directions, known as lattice speed directions $\mathbf{c}_j$. In particular, the so called D2Q9 lattice model is used, that is a standard 9-speed (the index j spans these directions) and two-dimensional lattice grid (see Figure 2.1), [53]. In this formulation the lattice vectors $\mathbf{c}_j$ are chosen in such a way that the spatial discretizaion Δx and the time step Δt are both equal to 1, leading to a simple implementation. The discretized lattice BGK equation reads as follows:

$$f_j(\boldsymbol{x} + \Delta t\, \mathbf{c}_j, t + \Delta t) = f_j(\boldsymbol{x}, t) + \frac{1}{\tau}\left[f_j^{eq}(\boldsymbol{x}, t) - f_j(\boldsymbol{x}, t)\right], \tag{2.3}$$

where $f_j = f_j(\boldsymbol{x}, t)$ and f_j^{eq} refer to the j-th particle distribution function. It is worth to notice that Equation (2.3) is linear and explicit. The local equilibrium is given by a second-order expansion in the Mach number of the local Maxwell distribution:

$$f_j^{eq} = w_j \rho \left[1 + \frac{\mathbf{c}_j \cdot \mathbf{v}}{c_s^2} + \frac{(\mathbf{c}_j \cdot \mathbf{v})^2}{2c_s^4} - \frac{\|\mathbf{v}\|^2}{2c_s^2}\right], \tag{2.4}$$

where w_j is a set of weights depending on the lattice model ($w_0 = 4/9$, $w_1 = w_2 = w_3 = w_4 = 1/9$, $w_5 = w_6 = w_7 = w_8 = 1/36$) , $c_s^2 = 1/3$ is the lattice sound speed, ρ and $\mathbf{v}$ are the macroscopic fluid density and velocity at the position $\boldsymbol{x}$, respectively, defined as discussed in the following. Notice that the relaxation parameter is strictly related to the viscosity ν, that is,

$$\nu = \left(\tau - \frac{1}{2}\right) c_s^2. \tag{2.5}$$

The relaxation parameter τ expresses the time that the particle distribution function needs to relax to the local equilibrium. The lower the parameter,

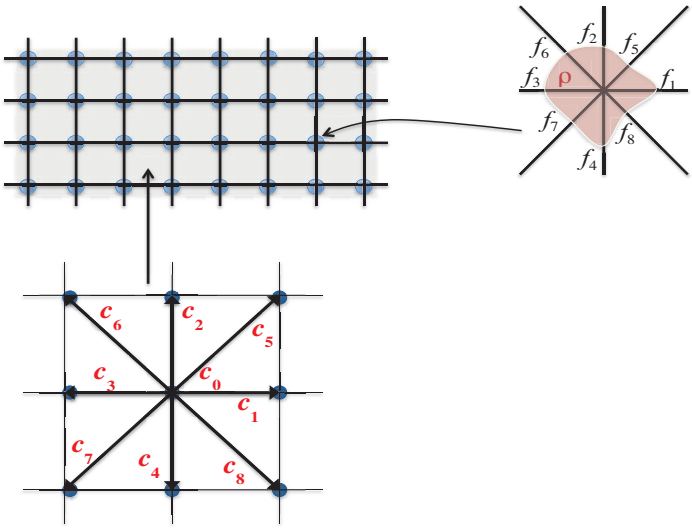


Figure 2.1: D2Q9 particle speed model.

the faster the relaxation to the equilibrium is, thus corresponding to a less viscous fluid. The C++ routine which computes f_j^{eq} is reported in Listing 2.1.
The stress tensor can be determined as a function of the particle distribution:

$$\boldsymbol{\sigma} = -p\mathbf{I} - \left(1 - \frac{1}{2\tau}\right)\boldsymbol{\Pi}^{(1)}, \tag{2.6}$$

with

$$p = \rho\, c_s^2, \tag{2.7}$$

$$\boldsymbol{\Pi}^{(1)} = \sum_j f_j^{(1)} \mathbf{c}_j \otimes \mathbf{c}_j, \tag{2.8}$$

where $\boldsymbol{\Pi}^{(1)}$ is computed with the non-equilibrium part of the particle distribution function $f_j^{(1)} = f_j - f_j^{eq}$ and $\mathbf{I}$ is the identity tensor. The discretized Equation (2.3) is conveniently solved by two steps:

Listing 2.1: Particle equilibrium distribution function.

```
void equili()
{
    double A, B, C;
    for(int i=0; i<nx+2; i++)
        for(int j=0; j<ny+2; j++)
        {
            C = −1.5∗(u(i,j)∗u(i,j)+v(i,j)∗v(i,j));
            for(int k=0; k<np; k++)
            {
                A = u(i,j)∗cx(k)+v(i,j)∗cy(k);
                B = 4.5∗A∗A;
                feq[k][i][j] = w(k)∗rho(i,j)∗(1.+3.∗A+B+C);
            }
        }
}
```

- collision

$$f_j^{out}(\boldsymbol{x},t) = f_j^{in}(\boldsymbol{x},t) - \frac{1}{\tau}[f_j^{in}(\boldsymbol{x},t) - f_j^{eq}(\boldsymbol{x},t)], \tag{2.9}$$

- streaming

$$f_j^{in}(\boldsymbol{x} + \Delta t\, \mathbf{c}_j, t + \Delta t) = f_j^{out}(\boldsymbol{x},t). \tag{2.10}$$

Listing 2.2 shows the routines performing Equation (2.9) and (2.10).
Once Equation (2.3) is solved, macroscopic fluid density and velocity fields are computed as:

$$\rho = \sum_j f_j, \tag{2.11}$$

$$\mathbf{v} = \frac{\sum_j f_j \mathbf{c}_j}{\rho}. \tag{2.12}$$

This numerical model can be derived not only from continuum Boltzmann's equation. In particular, the LB method can be viewed as a continuous version of several microscopic models known as cellular automata and lattice gas cellular automata [15, 54].

Listing 2.2: Collision and streaming steps.

```
void collision()
{
    for(int k=0; k<np;k++)
        for(int i=1; i<=nx; i++)
            for(int j=1; j<=ny; j++)
                f1[k][i][j] = f1[k][i][j]*(1.0-omega) + omega*feq[k][i][j];
}
//-----------------------------------------------
void streaming()
{
    int h, l;
    for(int k=0; k<np; k++)
    {
        h = (int)cx(k);
        l = (int)cy(k);
        for(int i=1; i<=nx; i++)
            for(int j=1; j<=ny; j++)
                f2[k][i][j] = f1[k][i-h][j-l];
        }
}
```

2.3 Chapman-Enskog expansion

In this section, it will be shown that the LB method is suitable for fluid flow simulations since it recovers the Navier-Stokes equations for an incompressible flow. First, a perturbative expansion of the particle distribution function is performed, i.e.

$$f = f^0 + \epsilon f^1, \tag{2.13}$$

where superscript 0 denotes local equilibrium, while superscript 1 departure from this equilibrium. The Chapman-Enskog analysis consists of a double expansion in the smallness parameter ϵ of both dependent $f(\boldsymbol{x}, t)$ and independent $(\boldsymbol{x}, t)$ variables. The small perturbation parameter ϵ is the Knudsen number $\mathcal{K}n$, that is a dimensionless number equal to the ratio between the molecular mean free path length and a representative physical length scale. The Knudsen number is helpful to identify when statistical mechanics or the continuum mechanics formulation of fluid dynamics

should be used: if $\mathcal{K}n \geq 1$, the mean free path of a molecule is comparable to the length scale of the problem, and the continuum assumption of fluid mechanics is no longer a good approximation. In this case statistical methods must be used. Notice that the Knudsen number can be derived as the ratio between the Mach number $\mathcal{M}a$ and the Reynolds number $\mathcal{R}e$.
The expansion of space-time variables is the center of a multiscale method, whose idea is to represent space and time in terms of a hierarchy of scales, such that each variable is $O(1)$ at its relevant scale. By denoting with $\boldsymbol{x}_1$ and t_1 the linear sound wave regime and with t_2 the long-term dynamics, the multiscale representation reads as follows:

$$\boldsymbol{x} = \epsilon^{-1}\boldsymbol{x}_1, \qquad t = \epsilon^{-1}t_1 + \epsilon^{-2}t_2, \tag{2.14}$$

leading to the differential operators

$$\partial_x = \epsilon\partial_{x1}, \tag{2.15}$$

$$\partial_t = \epsilon\partial_{t1} + \epsilon^2\partial_{t2}. \tag{2.16}$$

Chapman-Enskog assumptions require that

$$f_j^{(0)} = f_j^{eq}, \tag{2.17}$$

$$\rho = \sum_j f_j = \sum_j f_j^{(0)}, \tag{2.18}$$

$$\rho\mathbf{v} = \sum_j \mathbf{c}_j f_j = \sum_j \mathbf{c}_j f_j^{(0)}, \tag{2.19}$$

$$\sum_j f_j^{(k)} = 0, \quad \text{if} \quad k \geq 1, \tag{2.20}$$

$$\sum_j \mathbf{c}_j f_j^{(k)} = 0, \quad \text{if} \quad k \geq 1. \tag{2.21}$$

The streaming operator can be written in terms of second-order space derivatives, that is,

$$\epsilon\partial_{t1} + \epsilon^2\partial_{t2} + \epsilon v_\alpha\partial_{x1\alpha} + \frac{1}{2}\epsilon^2 v_\alpha v_\beta\partial_{x1\alpha}\partial_{x1\beta}, \tag{2.22}$$

where v_α and v_α denote the Cartesian components of the velocity vector $\mathbf{v}$. The same Chapman-Enskog procedure can be applied to the collision operator,

$$Q \sim \epsilon C' f^1, \tag{2.23}$$

where the symbol $(\prime)$ denotes the differentiation with respect to f. Notice that in the above relation, the zeroth-order term $C[f^0]$ is neglected, since $f^0 = f^{eq}$, thus annihilating this term.
These expansions are very important, since macroscopic equations at different scales emerge naturally from the mathematical requirement that the operatorial coefficients if the expansions must be identically zero term-by-term. From mass and momentum conservations, it is possible to obtain:

$$\epsilon \hat{M}_1 + \epsilon^2 \hat{M}_2 = 0, \tag{2.24}$$
$$\epsilon \hat{J}_1 + \epsilon^2 \hat{J}_2 = 0, \tag{2.25}$$

respectively. At order ϵ, it is possible to assess

$$\partial_{t1}\rho + \partial_{\alpha 1} J_\alpha = 0, \tag{2.26}$$
$$\partial_{t1} J_\alpha + \partial_{\beta 1} \int v_\alpha v_\beta f^0 \, \mathrm{d}\mathbf{v} = 0, \tag{2.27}$$

being $J_\alpha = \rho v_\alpha$. Notice that in the above relations the velocity space integral can be solved analytically leading to $\rho v_\alpha v_\beta + \rho T \delta_{\alpha\beta}$ with T the fluid temperature, thus recovering the Euler equations for an inviscid fluid. The symbol $\delta_{\alpha\beta}$ denotes the Kronecker delta function.
The second-order equations involve the equilibrium and non-equilibrium levels of the multiscale expansion and, after some algebra, lead to the Navier-Stokes equations

$$\partial_t \rho + \partial_\alpha J_\alpha = 0, \tag{2.28}$$
$$\partial_t J_\alpha + \partial_\beta \left(\rho v_\alpha v_\beta + \rho T \delta_{\alpha\beta}\right) = \partial_\beta \tau_{\alpha\beta}. \tag{2.29}$$

By imposing $\boldsymbol{T}_{\alpha\beta} = \rho v_\alpha v_\beta + \rho T \delta_{\alpha\beta}$, it can be shown that the tensors $\boldsymbol{T}_{\alpha\beta}$ and $\tau_{\alpha\beta}$ represent the equilibrium and non-equilibrium components of the

momentum flux tensor, respectively,

$$\boldsymbol{T}_{\alpha\beta} = m \int f^{eq} v_\alpha v_\beta \, \mathrm{d}\mathbf{v}, \tag{2.30}$$

$$\tau_{\alpha\beta} = m \int (f - f^{eq}) \, v_\alpha v_\beta \, \mathrm{d}\mathbf{v}, \tag{2.31}$$

being m the mass. Notice that the tensor $\boldsymbol{T}_{\alpha\beta}$ is related to the non-dissipative Newtonian dynamics, whereas the tensor $\tau_{\alpha\beta}$ represents dissipative effects associated with the relaxation to the local equilibrium.

2.4 Dimensionless formulation

In order to simulate a real phenomenon through the LB method, it has to be converted in the lattice world, where $\Delta x = \Delta t = 1$; therefore, the real variables of the physical problem have to be transferred in lattice dimensionless units. First, the real units have to be converted in dimensionless units, by selecting a set of dimensionless parameters which govern the problem itself. The Mach number, $\mathcal{M}a = \frac{V}{c_s}$, and the Reynolds number, $\mathcal{R}e = \frac{V\,D}{\nu}$, usually represent two typical parameters of a problem involving a viscous fluid. Notice that V and D are the characteristic velocity and length of the problem, respectively. Many other parameters may exist, depending on the problem itself. For example, if gravity is considered, the Froude number, $\mathcal{F}r = \frac{V}{\sqrt{g\,L}}$, have to be taken into account. Moreover, for biomedical engineering involving the pulsatile blood flow in a vessel with radius R, a very popular dimensionless group is represented by the Womersley number, $\alpha = R\sqrt{\frac{\omega}{\nu}}$, being ω the frequency of the pulsatile flow.

2.5 Fluid boundary conditions

In this section, two schemes to account for the position of a solid body in the grid are discussed. In particular, these methods can successfully handle obstacle not aligned with the lattice nodes. First, the so called interpolated bounce back rule is discussed, which consists in computing the

particle distribution function bouncing back from a solid off-grid node to a fluid one through an interpolation of the known on-grid f_j. Notice that this boundary condition is strictly related to the Eulerian nature of the LB method, as it will be discussed in the following. Secondly, the Immersed Boundary method is presented. It consists in the verification of the zero-velocity condition on a mesh depicting the immersed solid by only knowing its position. Thus, the Immersed Boundary condition is characterized by a Lagrangian point of view.

Interpolated bounce-back

The half-way bounce-back scheme is used to implement the boundary conditions, [55, 56]. Moreover, the procedure proposed in [32] is used to account for the exact position of an obstacle not aligned to the grid. The post-collision density function $\tilde{f}_j$, which bounces-back from an immersed boundary node $\boldsymbol{x}_b$ located inside the obstacle, is computed by means of the interpolation scheme proposed in [30]:

$$\tilde{f}_j(\boldsymbol{x}_b, t) = 2w_j\rho(\boldsymbol{x}_b, t)\frac{\mathbf{c}_j \cdot \mathbf{v}(\boldsymbol{x}_w, t)}{c_s^2} - (1-\chi)\tilde{f}_j(\boldsymbol{x}_f, t) - \chi f_j^*(\boldsymbol{x}_b, t), \quad (2.32)$$

where $\boldsymbol{x}_w$ denotes the exact position of a wall node in the lattice background and $\boldsymbol{x}_f$ represents fluid nodes surrounding the obstacle (see Figure 2.2). In Equation (2.32), $f_j^*(\boldsymbol{x}_b, t)$ represents a fictitious equilibrium density function given by

$$f_j^*(\boldsymbol{x}_b, t) = w_j\rho(\boldsymbol{x}_f, t)\left[1 + \frac{\mathbf{c}_j \cdot \mathbf{v}_*}{c_s^2} + \frac{[\mathbf{c}_j \cdot \mathbf{v}(\boldsymbol{x}_f, t)]^2}{2c_s^4} - \frac{\|\mathbf{v}(\boldsymbol{x}_f, t)\|^2}{2c_s^2}\right] \quad (2.33)$$

and quantities χ and $\mathbf{v}_*$ are determined as follows:

$$\begin{cases} \mathbf{v}_* = \dfrac{2\delta - 3}{2\delta}\mathbf{v}(\boldsymbol{x}_f, t) + \dfrac{3}{2\delta}\mathbf{v}(\boldsymbol{x}_w, t), & \chi = \dfrac{2\delta - 1}{\tau + 0.5} \quad \text{if} \quad \delta \geq 1/2, \\ \mathbf{v}_* = \mathbf{v}(\boldsymbol{x}_f - \Delta t\, \mathbf{c}_j, t), \quad \chi = \dfrac{2\delta - 1}{\tau - 2} \quad \text{if} \quad \delta < 1/2, \end{cases} \quad (2.34)$$

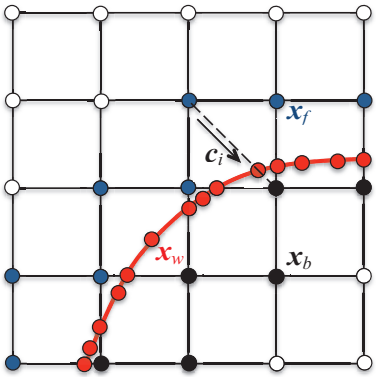


Figure 2.2: Curved boundary condition.

where $\delta = ||\boldsymbol{x}_f - \boldsymbol{x}_w||/||\boldsymbol{x}_f - \boldsymbol{x}_b||$. Equation (2.32) has been implemented for a cylindrical obstacle as reported in Listing 2.3.
A simpler interpolated bounce-back condition has been proposed in [44] and is reported in Listing 2.4, where the off-grid particle distribution function $\tilde{f}_{\bar{j}}$ which bounces-back from a solid node $\boldsymbol{x}_b$ to a fluid one $\boldsymbol{x}_f$ is obtained performing a quadratic interpolation of the populations located at the neighbour nodes, that is,

$$\begin{aligned}\tilde{f}_{\bar{j}}(\boldsymbol{x}_f, t) = \delta(1+2\delta) f_j(\boldsymbol{x}_f + \mathbf{c}_j, t) + (1-4\delta^2) f_j(\boldsymbol{x}_f, t) + \\ - \delta(1-2\delta) f_j(\boldsymbol{x}_f - \mathbf{c}_j, t) + 3 w_j [\mathbf{c}_j \cdot \mathbf{v}(\boldsymbol{x}_w, t)] \quad \text{if} \quad \delta < 1/2,\end{aligned} \tag{2.35}$$

$$\begin{aligned}\tilde{f}_{\bar{j}}(\boldsymbol{x}_f, t) = \frac{1}{\delta(1+2\delta)} f_j(\boldsymbol{x}_f + \mathbf{c}_j, t) + \frac{2\delta - 1}{\delta} f_{\bar{j}}(\boldsymbol{x}_f - \mathbf{c}_j, t) + \\ - \frac{2\delta - 1}{2\delta + 1} f_{\bar{j}}(\boldsymbol{x}_f - 2\mathbf{c}_j, t) + \frac{3 w_j}{\delta(1+2\delta)} [\mathbf{c}_j \cdot \mathbf{v}(\boldsymbol{x}_w, t)] \quad \text{if} \quad \delta \geq 1/2,\end{aligned} \tag{2.36}$$

being $\mathbf{c}_{\bar{j}} = -\mathbf{c}_j$. Aiming at using the fluid solver for fluid-structure interaction, a proper procedure is needed to tackle moving boundaries due to structure deformation. In particular, because of the fixed nature of the grid employed for the fluid computation, new fluid nodes can be activated, as a consequence of structure deformation. Thus, the newly-activated nodes must be properly initialized, in terms of particle distribution function: to

Listing 2.3: Boundary condition in [32] for a cylindrical obstacle.

```
void mei_bc()
{
    int A, B, O, AO, BO, C;
    double ubf, vbf, fstar_fil, deltax, deltay, delta,
           chi, R, U, V, U1, V1, UV1, UV2, U3;
    for(int k=1; k<np; k++)
        for(double i=xc-R-1; i<=xc+R+1; i++)
            for(double j=yc-R-1; j<=yc+R+1; j++)
                if(flag[k][i][j] == 10)
                                {
                    A = (int)cx[k];
                    B = (int)cy[k];
                    O = opp[k];
                    AO = (int)cx[opp[k]];
                    BO = (int)cy[opp[k]];
                    R = rho[i][j];
                    U = u[i][j];
                    V = v[i][j];
                    U1 = U*cx[opp[k]];
                    V1 = V*cy[opp[k]];
                    UV1 = U1+V1;
                    UV2 = U*U+V*V;
                    deltax = fabs(xfluid[k][i][j]-xwall[k][i][j]);
                    deltax /= fabs(xfluid[k][i][j]-xbound[k][i+AO][j+BO]);
                    deltay = fabs(yfluid[k][i][j]-ywall[k][i][j]);
                    deltay /= fabs(yfluid[k][i][j]-ybound[k][i+AO][j+BO]);
                    delta = sqrt(deltax*deltax+deltay*deltay);
                    if(delta >= 0.5)
                    {
                        chi = (2.*delta-1.)/(tau+0.5);
                        ubf = (2.*delta-3.)/(2.*delta)*U;
                        vbf = (2.*delta-3.)/(2.*delta)*V;
                    }
                    else
                    {
                        chi = (2.*delta-1.)/(tau-2.);
                        ubf = u[i+A][j+B];
                        vbf = v[i+A][j+B];
                    }
                    U3 = ubf*cx[opp[k]]+vbf*cy[opp[k]];
                    fstar_fil = w[O]*R*(1.+3.*U3+4.5*UV1*UV1-1.5*UV2);
                    f_fil[k][i][j] = (1.-chi)*f[O][i][j]+chi*fstar_fil;
                }
}
```

Listing 2.4: Boundary condition in [44] for a cylindrical obstacle.

```
void lall_bc
{
    for(int k=1; k<np; k++)
        for(double i=xc-R-1; i<=xc+R+1; i++)
            for(double j=yc-R-1; j<=yc+R+1; j++)
                if(flag[k][i][j] == 10)
                {
                    A = opp[k];
                    B = (int)cx[A];
                    C = (int)cy[A];
                    deltax = fabs(xfluid[k][i][j]-xwall[k][i][j]);
                    deltax /= fabs(xfluid[k][i][j]-xbound[k][i+AO][j+BO]);
                     deltay = fabs(yfluid[k][i][j]-ywall[k][i][j]);
                     deltay /= fabs(yfluid[k][i][j]-ybound[k][i+AO][j+BO]);
                    delta = sqrt(deltax*deltax+deltay*deltay);
                    if(delta < 0.5)
                        f_fil[k][i][j] = delta*(1.+2.*delta)*f[A][i+B][j+C]
                                + (1.-4.*delta*delta)*f[A][i][j]
                                - delta*(1.-2.*delta)*f[A][i-B][j-C];
                    else
                        f_fil[k][i][j] = 1./(delta*(2.*delta+1.))*f[A][i+B][j+C]
                                + (2.*delta-1.)/delta*f[k][i-B][j-C]
                                - (2.*delta-1.)/(2.*delta+1.)*f[k][i-2*B][j-2*C];
                }
}
```

assign a proper initial value, a very simple refill procedure has been adopted. In particular, a linear distribution of f_j is assumed close to the new lattice node and a linear interpolation of the values at the surrounding nodes placed externally to the solid is adopted. On the other hand, upon structure deformation some nodes may become inactive. In this case, the associated values are disregarded, leading to a loss of mass. Notice that this refill procedure does not satisfy the mass conservation law, since it introduces new nodes, and consequently mass, in the system each time a new fluid node arises. This issue suggests to use a solid boundary condition which does not involves any refill procedure.

The Immersed Boundary method

The idea to compute the particle distribution function in an off-grid position has been proved to be very effective [30, 31, 32]. On the other hand, such approach is affected by several drawbacks. It lies in the Eulerian lattice grid, thus it is difficult to handle and implement complex geometries that cannot be described by a predefined expression. In particular, for the generic obstacle, the boundary staircase nodes $\boldsymbol{x}_b$ must be detected; then, the exact position of the wall node $\boldsymbol{x}_w$ is computed by intersecting each lattice direction with the equation/expression of the solid surface. As it can be immediately realized, handling a geometry becomes computationally hard, especially upon solid deformations. In order to avoid this issue, the Immersed Boundary (IB) method proposed by Peskin [57, 45] can be used, following the approach in [58, 43, 59]. As show in Figure 2.3, the solid is represented by a Lagrangian mesh for the boundaries, generally non-stationary and unstructured. Therefore, two different coordinate systems are used: an Eulerian grid for the lattice BGK equation and a Lagrangian mesh for the immersed boundary, i.e. a vector containing the nodal coordinates. These two systems communicate through interpolations rules which satisfy the no-slip condition for the velocity and the momentum conservation, i.e. Newton's law. The boundary influences the fluid domain via an external forcing term, thus there is no direct boundary condition for the fluid acting on the populations as in the interpolated bounce-back scheme.

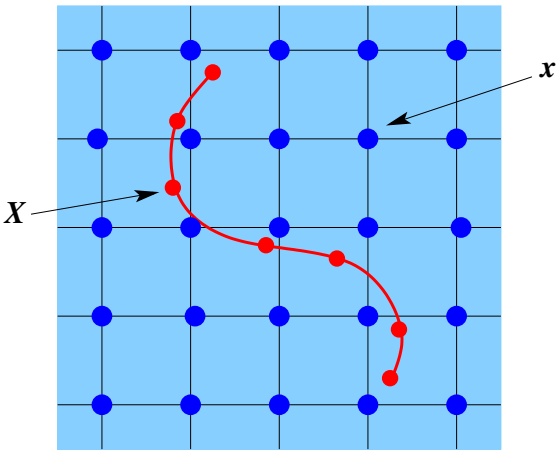


Figure 2.3: A Lagrangian solid mesh immersed in the Eulerian lattice fluid domain.

This means that populations can travel through the boundary without seeing it, but the macroscopic behavior is affected by its presence. Moreover, the fluid fills the entire domain, even inside the boundary region, thus no refill procedure is necessary and the consequent issue related to the mass conservation law is disregarded. Denoting with $\boldsymbol{X}_k$ and $\boldsymbol{V}_k$ the position and the velocities, respectively, of the k-th Lagrangian boundary node and with $\boldsymbol{T}_k$ the force acting on the same node, the discretized IB method equations read as follows:

$$\boldsymbol{V}_k = \sum_i \mathbf{v}(\boldsymbol{x}_i)\, W(\boldsymbol{X}_k - \boldsymbol{x}_i), \tag{2.37}$$

$$\boldsymbol{t}(\boldsymbol{x}_i) = \sum_k \boldsymbol{T}_k\, W(\boldsymbol{X}_k - \boldsymbol{x}_i), \tag{2.38}$$

where $\boldsymbol{t}(\boldsymbol{x}_i)$ is the fluid force density at the Eulerian point $\boldsymbol{x}_i$ which is added to the right-hand side of the collision step equation. Notice that capital letters indicate Lagrangian variables. The interpolation kernel W is chosen to be short-ranged with a finite cut-off length in order to reduce the

computational effort. Moreover, momentum and angular momentum have to be identical when evaluated either in the Eulerian or the Lagrangian frame. It is convenient to factorize the kernel as $W(x_1, x_2) = w(x_1) \cdot w(x_2)$, being ($x_1$,$x_2$) the Eulerian vector basis, and

$$w(x_i) = \begin{cases} \frac{1}{8}\left(3 - 2|x_i| + \sqrt{1 + 4|x_i| - 4x_i^2}\right) & \text{for} \quad 0 \leq |x_i| \leq 1, \\ \frac{1}{8}\left(5 - 2|x_i| - \sqrt{-7 + 12|x_i| - 4x_i^2}\right) & \text{for} \quad 1 \leq |x_i| \leq 2, \\ 0 & \text{otherwise,} \end{cases} \tag{2.39}$$

with $i = 1, 2$. The one-dimensional kernel is sketched in Figure 2.4. The

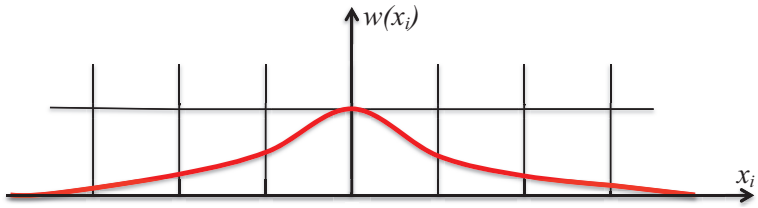


Figure 2.4: Sketch of the one-dimensional kernel $w(x_i)$.

present immersed boundary condition is shown in Listing 2.5.

2.6 LB method vs Navier-Stokes equations

The main features of the LB method versus the classical approach can be summarized as follows:

- classical CFD solvers inevitably need to treat the non-linear convective term, $\mathbf{v} \cdot \nabla \mathbf{v}$; the LB method totally avoids the non-linear convective term, because the convection becomes simple advection, that is a uniform data shift;
- CFD solvers for the incompressible Navier-Stokes equations need to solve the Poisson equation for the pressure; in the LB method, the

Listing 2.5: Immersed boundary method for a cylindrical obstacle.

```
int IB(int mn)
{
    double rx, ry, ds, eps; int l = 0;
    mtl1 Ux(N,0.), Uy(N,0.), ustarx(N,0.), ustary(N,0.), g1x(N,0.), g1y(N,0.);
    for(int k=0; k<N; k++){
        Ux(k) = 0.; Uy(k) = 0.;
        for(int i=min(X)-5; i<max(X)+5; i++)
            for(int j=min(Y)-5; j<max(Y)+5; j++){
                rx = (double)i-X(k); ry = (double)j-Y(k);
                ustarx(k) += u(i,j)*wIB(rx)*wIB(ry); ustary(k) += v(i,j)*wIB(rx)*wIB(ry);
            }
    }
    mtl2 ul(nx+2,ny+2), vl(nx+2,ny+2);
    set_to_zero(ul); set_to_zero(vl); g1x = Ux-ustarx; g1y = Uy-ustary;
    do
    {
        l++;
        set_to_zero(gx); set_to_zero(gy);
        ds = 2.*M_PI*(max(X)-min(X))/2./((double)N);
        for(int i=min(X)-5; i<max(X)+5; i++)
            for(int j=min(Y)-5; j<max(Y)+5; j++)
                for(int h=0; h<N; h++){
                    rx = (double)i-X(h); ry = (double)j-Y(h);
                    gx(i,j) += g1x(h)*wIB(rx)*wIB(ry)*ds;
                    gy(i,j) += g1y(h)*wIB(rx)*wIB(ry)*ds;
                }
        ul = u+gx; vl = v+gy;
        mtl1 ulx(N,0.), uly(N,0.), U(2*N,0.), ul(2*N,0.), provv(2*N,1.);
        for(int k=0; k<N; k++)
            for(int i=min(X)-5; i<max(X)+5; i++)
                for(int j=min(Y)-5; j<max(Y)+5; j++){
                    rx = (double)i-X(k); ry = (double)j-Y(k);
                    ulx(k) += ul(i,j)*wIB(rx)*wIB(ry);
                    uly(k) += vl(i,j)*wIB(rx)*wIB(ry);
                }
        g1x += Ux-ulx; g1y += Uy-uly;
        for(int i=0; i<N; i++){
            U(i) = Ux(i); U(N+i) = Uy(i);
            ul(i) = ulx(i); ul(N+i) = uly(i);
        }
        eps = norma(U,ul,2*N)/norma(U,provv,2*N);
        provv=0; U=0; ul=0; ulx=0; uly=0;
    }while(eps>0.0001);
    g1x=0; g1y=0; ul=0; vl=0; ustarx=0; ustary=0; Ux=0; Uy=0;
    return l;
}
```

pressure field is simply and directly related to the density one by means of an equation of state, i.e. $p = \rho\, c_s^2$ (Equation 2.7);

- since Boltzmann's equation is kinetic-based, the physics associated with the molecular level interaction can be incorporated more easily in the LB method. Hence, the LB method can be fruitfully applied to micro-scale fluid flow problems;

- in the LB method, the Courant-Friedrichs-Lewy number is proportional to $\Delta t/\Delta x$; the grid CFL number is equal to 1 based on the lattice units of $\Delta t = \Delta x = 1$. Consequently, the time dependent LB method is inefficient for solving steady-state problems;

- the spatial discretization in the LB method is dictated by the discretization of the particle velocity space. This coupling between discretized velocity space and configuration space leads to regular square grids. This is a limitation of the LB method, especially for aerodynamic applications where both the far field boundary condition and the near wall boundary layer need to be carefully implemented.

Chapter 3

Structure modeling

Sommario

Il solido è modellato tramite un approccio agli elementi finiti. In particolare, viene utilizzato un modello di trave piana snella in grandi spostamenti a comportamento meccanico elastico lineare. Il regime di grandi spostamenti è valutato tramite l'approccio corotazionale, la cui formulazione viene discussa. L'equazione del moto del solido deformabile è risolta tramite il metodo Time Discontinuous Galerkin, i cui dettagli sullo schema di integrazione vengono esposti. È stato scelto questo algoritmo di integrazione rispetto ai classici schemi alla Newmark o metodo α per le migliori proprietà in termini di convergenza, stabilità e accuratezza. Nell'Appendice A verranno mostrati i dettagli del modello di Eulero-Bernoulli di trave piana a comportamento lineare per geometria e per materiale.

3.1 Geometrically nonlinear beam finite elements

Linear elastic plane structures idealized by beam finite elements are considered. Large displacements are accounted for through a corotational formulation [39, 40]. Making reference to Figure 3.1, such approach splits the reference configuration into two configurations: the initial configuration C_0, which is fixed during the overall analysis, and the corotated configuration C_R, which is different for each element and evolves together with the individual element. In this way the element rigid body motion is separated by the purely deformational one, that is measured with respect to the corotated configuration. A shear undeformable linear elastic straight beam

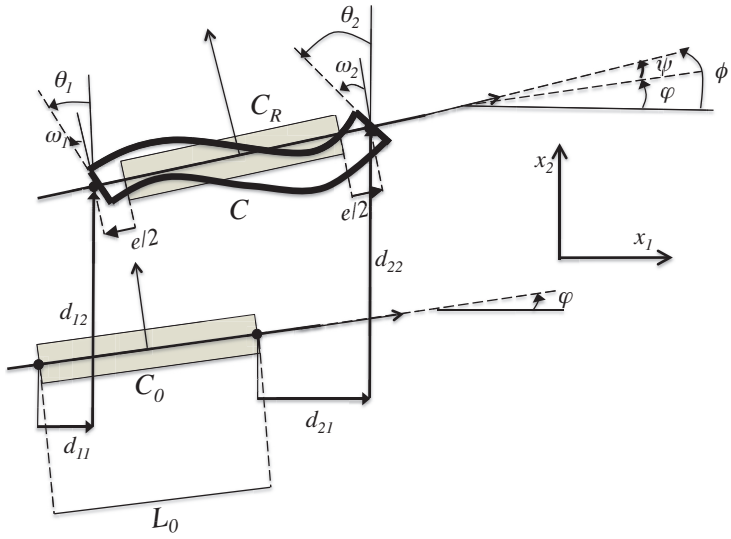


Figure 3.1: Corotational kinematics: initial (C_0) and corotated (C_R) configurations.

element with two nodes is considered. The corotational configuration is selected so that the longitudinal axis passes through the current position of the end nodes. According to Figure 3.1, the orientation of the beam longitudinal axis is φ in the initial configuration and $\phi = \varphi + \psi$ in the corotated one. The beam properties are: elastic modulus E, cross section area

A_0, inertia moment about the neutral axis I_0 and initial length L_0. The element kinematics is described by the vector of the nodal displacements

$$\boldsymbol{u}^{(e)} = [d_{11}\, d_{12}\, \theta_1\, d_{21}\, d_{22}\, \theta_2]^{\mathrm{T}}\,. \tag{3.1}$$

The assumed deformation parameters are the axial elongation e and the node rotations ω_1 and ω_2, measured with respect to C_R,

$$\mathbf{e} = [e\, \omega_1\, \omega_2]^{\mathrm{T}}\,. \tag{3.2}$$

The resultant compatibility equations are:

$$\begin{aligned} e &= (d_{21} - d_{11})c_\phi + (d_{22} - d_{12})s_\phi + L_0(1 - c_\phi), && (3.3)\\ \omega_1 &= \theta_1 - \psi, && (3.4)\\ \omega_2 &= \theta_2 - \psi, && (3.5) \end{aligned}$$

being $c_\phi = \cos\phi$ and $s_\phi = \sin\phi$.
The first derivatives of the deformation variables with respect to the nodal parameters can be written as follows:

$$\nabla_{\boldsymbol{u}}\mathbf{e} = \begin{bmatrix} -c_\phi & -s_\phi & 0 & c_\phi & s_\phi & 0 \\ -s_\phi/L & c_\phi/L & 1 & s_\phi/L & -c_\phi/L & 0 \\ -s_\phi/L & c_\phi/L & 0 & s_\phi/L & -c_\phi/L & 1 \end{bmatrix}. \tag{3.6}$$

The second derivatives of the deformation variables are given by

$$\nabla^2_{\boldsymbol{u}} e = \frac{1}{L}\begin{bmatrix} s_\phi^2 & -s_\phi c_\phi & 0 & -s_\phi^2 & s_\phi c_\phi & 0 \\ -s_\phi c_\phi & c_\phi^2 & 0 & s_\phi c_\phi & -c_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -s_\phi^2 & s_\phi c_\phi & 0 & s_\phi^2 & -s_\phi c_\phi & 0 \\ s_\phi c_\phi & -c_\phi^2 & 0 & -s_\phi c_\phi & c_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.7}$$

for the axial deformation, while for the node rotations ω_1 and ω_2:

$$\nabla^2_{\boldsymbol{u}}\omega_1 = \nabla^2_{\boldsymbol{u}}\omega_2 = \frac{1}{L^2}\begin{bmatrix} -2s_\phi c_\phi & c_\phi^2 - s_\phi^2 & 0 & 2s_\phi c_\phi & s_\phi^2 - c_\phi^2 & 0 \\ c_\phi^2 - s_\phi^2 & 2s_\phi c_\phi & 0 & s_\phi^2 - c_\phi^2 & -2s_\phi c_\phi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2s_\phi c_\phi & s_\phi^2 - c_\phi^2 & 0 & -2s_\phi c_\phi & c_\phi^2 - s_\phi^2 & 0 \\ s_\phi^2 - c_\phi^2 & -2s_\phi c_\phi & 0 & c_\phi^2 - s_\phi^2 & 2s_\phi c_\phi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.8}$$

The deformation energy of the beam element can be written as

$$U = U_a + U_b + U_g, \tag{3.9}$$

where U_a, U_b and U_g are associated to axial deformation, bending deformation and initial-stress geometric effects, respectively. The following expressions for the three terms are assumed:

$$U_a = N^0 e + \frac{1}{2}\frac{EA_0}{L_0}e^2, \tag{3.10}$$

$$U_b = M_2^0\omega_2 - M_1^0\omega_1 + \frac{1}{2}\frac{EI_0}{L_0}\begin{bmatrix}\omega_1 \\ \omega_2\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}4 & 2 \\ 2 & 4\end{bmatrix}\begin{bmatrix}\omega_1 \\ \omega_2\end{bmatrix}, \tag{3.11}$$

$$U_g = \frac{1}{2}\frac{N^0 L_0}{30}\begin{bmatrix}\omega_1 \\ \omega_2\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}4 & -1 \\ -1 & 4\end{bmatrix}\begin{bmatrix}\omega_1 \\ \omega_2\end{bmatrix}, \tag{3.12}$$

being N^0, M_1^0 and M_2^0 the stress resultants in the initial configuration, as shown in Figure 3.2. As usual, shear forces can be recovered by equilibrium. Notice that, due to the assumed kinematics, axial and shear forces are constant along the beam element, while bending moment varies linearly. Hence, the internal force vector $\boldsymbol{S}^{(e)}$ and the tangent stiffness matrix $\hat{\boldsymbol{K}}_t^{(e)}$ for the element are obtained based on standard arguments by differentiation with respect to the element degrees of freedom. With some algebra, the

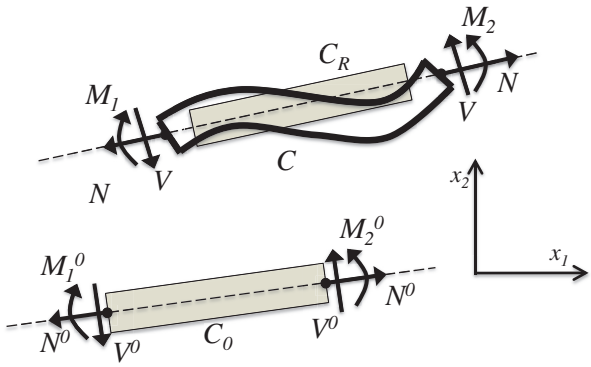


Figure 3.2: Corotational kinematics: stress resultants.

vector of the internal forces can be written as

$$\boldsymbol{S}^{(e)} = \begin{bmatrix} -Nc_\phi + Vs_\phi - 3as_\phi(\omega_1+\omega_2)/L \\ -Ns_\phi - Vc_\phi + 3ac_\phi(\omega_1+\omega_2)/L \\ -M_1 + a(4\omega_1 - \omega_2) \\ Nc_\phi - Vs_\phi + 3as_\phi(\omega_1+\omega_2)/L \\ Ns_\phi + Vc_\phi - 3ac_\phi(\omega_1+\omega_2)/L \\ -M_2 + a(4\omega_2 - \omega_1) \end{bmatrix}, \tag{3.13}$$

where L is the length of the element in such configuration, and $a = \frac{N^0 L_0}{30}$. The stress resultants in the current configuration N, V, M_1 and M_2 are defined as

$$N = N^0 + \frac{EA}{L_0} e, \tag{3.14}$$

$$M_1 = M_1^0 - \frac{2EJ}{L_0}(2\omega_1 + \omega_2), \tag{3.15}$$

$$M_2 = M_2^0 - \frac{2EJ}{L_0}(\omega_1 + 2\omega_2), \tag{3.16}$$

$$V = \frac{M_1 - M_2}{L} = V^0 \frac{L_0}{L} + \frac{2EJ}{LL_0}(\omega_1 - \omega_2). \tag{3.17}$$

The corresponding expression for the tangent stiffness matrix reads as follows:

$$\hat{\boldsymbol{K}}_t^{(e)} = \boldsymbol{K}_M^{(e)} + \boldsymbol{K}_G^{(e)}, \tag{3.18}$$

being $\boldsymbol{K}_M^{(e)}$ the classical stiffness matrix of the Euler-Bernoulli beam theory (see Appendix A) and $\boldsymbol{K}_G^{(e)}$ the geometric stiffness matrix that can be expressed as

$$\boldsymbol{K}_G^{(e)} = \boldsymbol{K}_{GN}^{(e)} + \boldsymbol{K}_{GV}^{(e)}, \tag{3.19}$$

where $\boldsymbol{K}_{GN}^{(e)}$ is the geometric stiffness for an Hermitian beam element under axial force

$$\boldsymbol{K}_{GN}^{(e)} = \frac{N^0 L_0}{30} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 36 & 3L & 0 & -36 & 3L \\ 0 & 3L & 4L^2 & 0 & -3L & -L^2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -36 & -3L & 0 & 36 & -3L \\ 0 & 3L & -L^2 & 0 & -3L & 4L^2 \end{bmatrix} \tag{3.20}$$

and $\boldsymbol{K}_{GV}^{(e)}$ takes into account for varying bending moment along the beam axis:

$$\boldsymbol{K}_{GV}^{(e)} = \frac{V}{L} \begin{bmatrix} \sin 2\phi & -\cos 2\phi & 0 & -\sin 2\phi & \cos 2\phi & 0 \\ -\cos 2\phi & -\sin 2\phi & 0 & \cos 2\phi & \sin 2\phi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin 2\phi & \cos 2\phi & 0 & \sin 2\phi & -\cos 2\phi & 0 \\ \cos 2\phi & \sin 2\phi & 0 & -\cos 2\phi & -\sin 2\phi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.21}$$

The forces exerted by the fluid on the beam element are computed in the corotated configuration by means of the stress tensor given in Equation (2.6). The corresponding equivalent nodal loads are determined using the standard procedure. Notice that, since the loads depend on the current configuration, a load contribution should be added to the tangent stiffness

matrix. However, this contribution is neglected in the spirit of a modified Newton-Raphson strategy. The inertia forces are accounted for by using the classical mass matrix of the two-node Euler-Bernoulli beam element. Finally, the equation of motion of the entire structure is obtained by standard finite element assembly procedures.

3.2 Time integration scheme

The equation of motion for the structure, discretized using the finite element method, reads as follows:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{S}(\boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{u}), \tag{3.22}$$

together with proper initial conditions

$$\boldsymbol{u}(0) = \boldsymbol{u}_0, \tag{3.23}$$
$$\dot{\boldsymbol{u}}(0) = \dot{\boldsymbol{u}}_0, \tag{3.24}$$

where $\boldsymbol{u}$ is the vector of the nodal displacements, $\boldsymbol{M}$ and $\boldsymbol{C}$ are the mass and damping matrices, respectively, $\boldsymbol{S}(\boldsymbol{u})$ is the internal force vector and $\boldsymbol{F}(\boldsymbol{u})$ is the vector of the nodal forces exerted by the fluid. For simplicity, no external loads are considered. The Time Discontinuous Galerkin method is applied to integrate Equation (3.22). Piecewise linear time interpolations are selected for both nodal displacements and nodal velocities, considered as independent variables [60], and depicted in Figure 3.3. The resultant algorithm is one step and the predicted values for displacements and velocities at the end of the current time step are given by:

$$\boldsymbol{v}_{i+1}^{-} = \Delta t \bar{\boldsymbol{u}}_1, \tag{3.25}$$
$$\boldsymbol{u}_{i+1}^{-} = \boldsymbol{u}_i^{-} + (1-\mu)\bar{\boldsymbol{u}}_0 + \mu\bar{\boldsymbol{u}}_1, \tag{3.26}$$

where $\boldsymbol{u}_i^{-}$ is the displacement at the end of the previous time step, $\bar{\boldsymbol{u}}_k$ are the vectors that govern the displacement time interpolation and $\mu = 1 - \sqrt{2}/2$. Notice that the superscript $-$ is used to emphasize the discontinuous nature

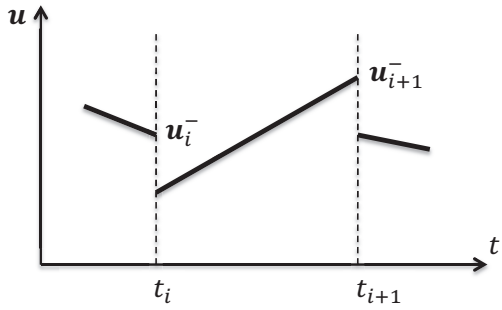


Figure 3.3: Piecewise linear approximation.

of the method. In order to advance in time, the non-linear algebraic equations, providing the time evolution of the structure, are solved by means of a Newton-Raphson recursive strategy. In particular, the two-stage iterative scheme proposed in [42] is used:

$$\boldsymbol{K}^{\star(l)}\boldsymbol{\Delta}\bar{\boldsymbol{u}}_0^{(l+1)} = \frac{1}{\mu}\left(\boldsymbol{F}_0^{\star(l)} - \boldsymbol{P}_0^{(l)} - \boldsymbol{M}\tilde{\boldsymbol{a}}_0^{(l+1)}\right), \tag{3.27}$$

$$\begin{aligned}\boldsymbol{K}^{\star(l)}\boldsymbol{\Delta}\bar{\boldsymbol{u}}_1^{(l+1)} &= \frac{1}{\mu}\Big(\boldsymbol{F}_1^{\star(l)} - \boldsymbol{P}_1^{(l)} - \boldsymbol{M}\tilde{\boldsymbol{a}}_1^{(l+1)} + \\ &- \frac{1-\mu}{\mu}\hat{\boldsymbol{K}}_t^{(l)}\boldsymbol{\Delta}\bar{\boldsymbol{u}}_0^{(l+1)}\Big),\end{aligned} \tag{3.28}$$

where (l) refers to a quantity evaluated at iteration l and vectors $\tilde{\boldsymbol{a}}_k^{(l+1)}$ are computed as

$$\tilde{\boldsymbol{a}}_0^{(l+1)} = \frac{1}{\mu\Delta t}\left[\frac{1}{\Delta t}\bar{\boldsymbol{u}}_0^{(l)} - \boldsymbol{v}_i^- - \beta\left(\bar{\boldsymbol{v}}_0^{(l)} - \bar{\boldsymbol{v}}_1^{(l)}\right)\right], \tag{3.29}$$

$$\tilde{\boldsymbol{a}}_1^{(l+1)} = \frac{1}{\mu\Delta t}\left[\frac{1}{\Delta t}\bar{\boldsymbol{u}}_1^{(l)} - \boldsymbol{v}_i^- - (1-\mu)\bar{\boldsymbol{v}}_0^{(l+1)}\right], \tag{3.30}$$

being $\beta = \sqrt{2} - 4/3$, $\bar{\boldsymbol{v}}_j$ the vectors that govern the velocity time interpolation and $\boldsymbol{v}_i^-$ the velocity at the end of the previous time step. Velocities

are updated by

$$\bar{\boldsymbol{v}}_j^{(l+1)} = \Delta t\, \tilde{\boldsymbol{a}}_j^{(l+1)} + \frac{1}{\mu \Delta t} \bar{\boldsymbol{u}}_j^{(l)}, \quad j = 0, 1 \tag{3.31}$$

and the effective stiffness matrix is defined as

$$\boldsymbol{K}^{\star(l)} = \frac{1}{\mu^2 \Delta t^2} \boldsymbol{M} + \hat{\boldsymbol{K}}_t^{(l)}, \tag{3.32}$$

being $\hat{\boldsymbol{K}}_t^{(l)}$ the tangent stiffness matrix at iteration l. Indeed, a modified tangent stiffness matrix is adopted, since material and geometric contributions are actually computed, but the load contribution is disregarded, as discussed in the previous section. Notice that all the quantities with the superscript (l) are known from the previous iteration and $(\tilde{\ })$ refers to predictors. Denoting with $(\bullet)_0$ and $(\bullet)_{2/3}$ the values of $(\bullet)$ at $t = t_i$ and $t = t_i + (2/3)\Delta t$, respectively, vectors $\boldsymbol{F}_k^\star$ and $\boldsymbol{P}_k$ are calculated as

$$\boldsymbol{F}_0^\star = \frac{1}{4}\left[\left(3\sqrt{2} - 2\right) \boldsymbol{F}(\boldsymbol{u}_0) + 3\left(2 - \sqrt{2}\right) \boldsymbol{F}(\boldsymbol{u}_{2/3})\right], \tag{3.33}$$

$$\boldsymbol{F}_1^\star = \frac{1}{2}\left[3\boldsymbol{F}(\boldsymbol{u}_{2/3}) - \boldsymbol{F}(\boldsymbol{u}_0)\right], \tag{3.34}$$

$$\boldsymbol{P}_0 = \frac{1}{4}\left[\left(3\sqrt{2} - 2\right) \boldsymbol{S}(\boldsymbol{u}_0) + 3\left(2 - \sqrt{2}\right) \boldsymbol{S}(\boldsymbol{u}_{2/3})\right], \tag{3.35}$$

$$\boldsymbol{P}_1 = \frac{1}{2}\left[3\boldsymbol{S}(\boldsymbol{u}_{2/3}) - \boldsymbol{S}(\boldsymbol{u}_0)\right], \tag{3.36}$$

where $\boldsymbol{u}_0$ and $\boldsymbol{u}_{2/3}$ are given by

$$\boldsymbol{u}_0 = \boldsymbol{u}_i^- + \frac{\sqrt{2}}{6}(\bar{\boldsymbol{u}}_0 - \bar{\boldsymbol{u}}_1), \tag{3.37}$$

$$\boldsymbol{u}_{2/3} = \boldsymbol{u}_i^- + \frac{7\sqrt{2}}{18}\bar{\boldsymbol{u}}_0 + \frac{12 - 7\sqrt{2}}{18}\bar{\boldsymbol{u}}_1. \tag{3.38}$$

At the beginning of each time step, the initial values for $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$ are set to zero, which implies to select the approximate solution at the end of the previous time interval as the starting value.

The proposed scheme is clearly based on two implicit correctors for each iteration, which are performed using the same iteration matrix $\boldsymbol{K}^{\star(l)}$. This effective matrix has the same form of standard algorithms and retains the same structure as $\boldsymbol{M}$ and $\hat{\boldsymbol{K}}_t$. To reduce the computational cost, the effective matrix is formed and factorized only once in a time step.
The outlined algorithm has proved to rapidly converge to the target solution, while preserving the target stability and dissipation properties at each iteration. Therefore, iterative corrections are performed only to improve accuracy, but in the linear regime no more than two iterations are needed to obtain the optimal accuracy order. In fact, second-order accuracy is achieved after the first iteration and third order accuracy (that is the order of accuracy of the method) at the second one. It can be demonstrated that the first iteration leads to a Nørsett algorithm [61]. In addition, reliable error estimates are readily available [62]. The very good stability properties of the outlined scheme in non-linear dynamics are discussed in [63].
An efficient time discontinuous Galerkin procedure for linear regimes has been developed in [60] and employed by the author in [7].

Chapter 4

Coupling strategy and numerical tools

Sommario

Nei capitoli precedenti sono state presentate due diverse metodologie. Da un lato, il metodo lattice Boltzmann per il fluido viene risolto su una griglia cartesiana a maglia quadrata. Dall'altro lato, il solido è modellato agli elementi finti e la sua soluzione è figlia di una metodologia diversa da quella usata per il dominio fluido. Pertanto, risolvere un problema di interazione fluido-struttura consiste, tra l'altro, nello sviluppare una strategia di accoppiamento tra i due solutori che garantisca un continuo ed efficace scambio di informazioni tra loro. In particolare, tre diversi algoritmi di accoppiamento sono stati sviluppati e le proprietà di ognuno valutate numericamente. La definizione di un'energia di interfaccia è stata funzionale al computo dell'errore introdotto da questo approccio partizionato sul soddisfacimento delle condizioni di equilibrio e compatibilità all'interfaccia tra fluido e solido.
Grande attenzione è stata rivolta a sviluppare un software di calcolo prestazionale. In questo capitolo, infatti, verrà mostrato come il miglioramento

delle prestazioni del software numerico sia possibile tramite l'impiego di librerie di algebra lineare (i.e. suite di routines) in grado di attingere risorse dall'architettura dell'hardware nello svolgimento di operazioni quali prodotti tra matrici, risoluzione di sistemi lineari. In particolare, ne verrà mostrata l'efficacia rispetto alla risoluzione di un sistema lineare poichè questa operazione, eseguita per risolvere l'equazione del moto del solido, rappresenta un collo di bottiglia nel software sviluppato.

4.1 Fluid-structure interaction

A reliable prediction of the interaction between fluids and structures is extremely important in several industrial, technological, biological and environmental processes. This great interest promoted a huge research effort in the last two decades. As well known, two different approaches, typical of any coupled problem, can be used to tackle fluid-structure interaction (FSI): monolithic approach and partitioned approach. Here, the partitioned approach is adopted, since it is the most suited for practical problems, [38]. The basic strategy is to treat separately the fluid and the structural domains and to properly discretize each of them, in order to adopt numerical methods developed and optimized for both computational fluid dynamics and computational structural dynamics.
In Chapter 2 and Chapter 3, two different solvers have been discussed. On the fluid side, the LB method, based on continuum Boltzmann's kinetic equation, is adopted and the discretized lattice BGK equation is solved on a certain grid. On the other hand, the structure is solved in the framework of the FE method, based on partial differential equation, mesh and solution procedure which are different from the LB ones. These two solvers need to communicate, i.e. an efficient, effective, continuous, and mutual transfer of informations between the two involved solvers is due. Such information consists of static variables (stresses) and kinematic variables (displacements and velocities). Regarding this aim, one of the main goals of the present thesis has been the development of a coupling algorithm able to guarantee this exchange of informations.
To meet the continuity conditions on the common fluid-structure boundary, an effective procedure should be devised to couple the two solvers. On the fluid-structure interface Γ, compatibility condition should be met, i.e. the equivalence of the velocities computed by both fluid and solid sides:

$$\boldsymbol{v}_i^{(f)} - \boldsymbol{v}_i^{(s)} = \mathbf{0}. \tag{4.1}$$

In addition, the sum of tractions computed on the interface considered as fluid boundary, $\boldsymbol{t}_i^{(f)}$, and as structure boundary, $\boldsymbol{t}_i^{(s)}$, should be zero:

$$\boldsymbol{t}_i^{(f)} + \boldsymbol{t}_i^{(s)} = \boldsymbol{0}. \tag{4.2}$$

Generally, these conditions are not exactly fulfilled by partitioned solution procedures. The error in terms of interface energy artificially produced within the typical time step is given by [34]:

$$J = \int_{t_i}^{t_{i+1}} \int_{\Gamma} \left(\boldsymbol{t}^{(f)} \cdot \boldsymbol{v}^{(f)} + \boldsymbol{t}^{(s)} \cdot \boldsymbol{v}^{(s)} \right) \mathrm{d}\xi \, \mathrm{d}t. \tag{4.3}$$

Ideally, the interface energy should be zero. In the following section, some numerical tests are carried out to evaluate the interface energy for each coupling algorithm: J, in fact, is expected to reveal the effectiveness of each algorithm in transferring forces and displacements. For simplicity, J is computed as

$$J = \frac{\Delta t}{2} \int_{\Gamma} \left(\boldsymbol{t}_{i+1}^{(f)} \cdot \boldsymbol{v}_{i+1}^{(f)} + \boldsymbol{t}_{i+1}^{(s)} \cdot \boldsymbol{v}_{i+1}^{(s)} + \boldsymbol{t}_i^{(f)} \cdot \boldsymbol{v}_i^{(f)} + \boldsymbol{t}_i^{(s)} \cdot \boldsymbol{v}_i^{(s)} \right) \mathrm{d}\xi, \tag{4.4}$$

where subscript i refers to a quantity evaluated at $t = t_i$.

4.2 Coupling algorithms

Three different coupling algorithms are discussed and numerically tested. As discussed in the previous chapters, the same time step is used for both fluid and structure solvers. The first algorithm is sketched in Figure 4.1. It is explicit and corresponds to a standard staggered approach. To advance in time, five actions are sequentially performed. Firstly, collision step is performed while keeping the structure frozen. Secondly, forces before collision, $\boldsymbol{F}(\boldsymbol{u}_0)$, and after collision are transferred to the structure solver. Forces after collision are assumed as a good approximation of forces at half time step, $\boldsymbol{F}(\boldsymbol{u}_{1/2})$, and $\boldsymbol{F}(\boldsymbol{u}_{2/3})$ are determined by linear interpolation:

$$\boldsymbol{F}(\boldsymbol{u}_{2/3}) = \frac{4}{3}\boldsymbol{F}(\boldsymbol{u}_{1/2}) - \frac{1}{3}\boldsymbol{F}(\boldsymbol{u}_0). \tag{4.5}$$

Thirdly, structure is moved. Fourthly, the fluid boundary is updated based on the new configuration of the structure. Fifthly, streaming is done while keeping the structure frozen. Then, the next time step is performed.

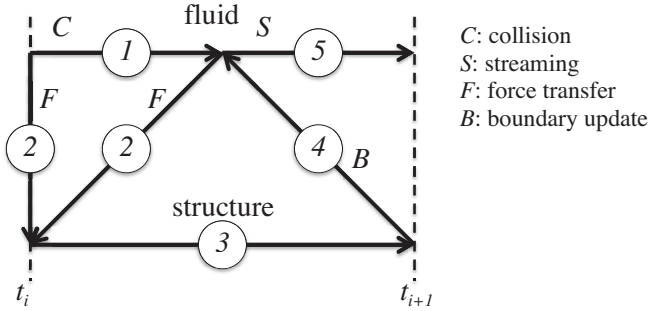


Figure 4.1: FELBA explicit scheme.

The second algorithm, sketched in Figure 4.2, is called simply FELBA: it is still explicit but it is characterized by the adoption of a structure predictor. As the previous algorithm, FELBA is characterized by five steps, too. Firstly, the configuration of the structure at the end of the current time step is predicted as follows, [47, 48]:

$$\tilde{\boldsymbol{u}}_1 = \boldsymbol{u}_i^- + \alpha_0 \Delta t \boldsymbol{v}_i^- + \alpha_1 \Delta t \boldsymbol{\Delta v}_{i-1}, \tag{4.6}$$

being $\boldsymbol{\Delta v}_{i-1}$ the structure velocity change within the previous time step. A second-order accurate prediction is expected by choosing $\alpha_0 = 1$ and $\alpha_1 = 0.5$. Secondly, the fluid boundary is updated according to the predicted configuration of the structure $\tilde{\boldsymbol{u}}_1$. Thirdly, fluid is solved by performing both collision and streaming. Fourthly, forces computed before action two, $\boldsymbol{F}(\boldsymbol{u}_0)$, and after action three, $\boldsymbol{F}(\tilde{\boldsymbol{u}}_1)$, are transferred to the structure solver. Vector $\boldsymbol{F}(\boldsymbol{u}_{2/3})$ is taken by a linear interpolation of the two:

$$\boldsymbol{F}(\boldsymbol{u}_{2/3}) = \frac{2}{3}\boldsymbol{F}(\tilde{\boldsymbol{u}}_1) + \frac{1}{3}\boldsymbol{F}(\boldsymbol{u}_0). \tag{4.7}$$

Fifthly, structure is moved. Then, as for the previous version, no iteration is done and the next time step is performed.

The last algorithm, called FELBA implicit, is similar to the previous one apart from the fact that some iterations are performed before advancing in time. Its flow chart is given in Table 4.1. The configuration of the structure obtained by action five is used as predictor value for the next iteration. Iterations stop when a convergence criterion on the final configuration of the structure is met. This third algorithm corresponds to the strongly-coupled partitioned approach and is expected to be the most accurate, although a higher computational cost is involved.

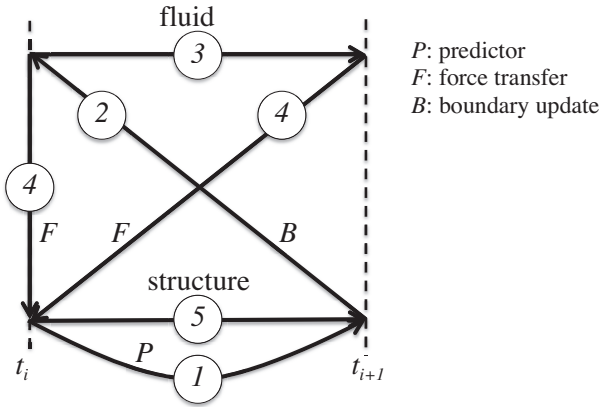


Figure 4.2: FELBA scheme.

4.3 Added-mass effect

In order to prevent from poor convergence rate in the case of very light structures with large Reynolds numbers, due to the well-known added mass effect [38, 49, 50, 51], Aitken's under-relaxation scheme is used. The so called added-mass is the inertia added to a fluid system when an accelerating or decelerating body moves a volume of surrounding fluid. For simplicity this can be modeled as some volume of fluid moving with the object, though in reality all the fluid will be accelerated. The dimensionless added-mass coefficient is the added mass divided by the displaced fluid

Table 4.1: Flow chart of the implicit algorithm.

1 - the final configuration of the structure is predicted:
$\tilde{\boldsymbol{u}}_1 = \boldsymbol{u}_i^- + \alpha_0 \Delta t \boldsymbol{v}_i^- + \alpha_1 \Delta t \boldsymbol{\Delta v}_{i-1}$;
2 - with the updated fluid boundary the LB steps are performed:
- collision $f_j^{out}(\boldsymbol{x}, t) = f_j^{in}(\boldsymbol{x}, t) - \frac{1}{\tau}[f_j^{in}(\boldsymbol{x}, t) - f_j^{eq}(\boldsymbol{x}, t)]$;
- streaming $f_j^{in}(\boldsymbol{x} + \Delta t\, \mathbf{c}_j, t + \Delta t) = f_j^{out}(\boldsymbol{x}, t)$;
- boundary conditions;
3 - the finite element solution is computed:
- effective stiffness matrix: $\boldsymbol{K}^{\star(l)} = \frac{1}{\mu^2 \Delta t^2}\boldsymbol{M} + \hat{\boldsymbol{K}}_t^{(l)}$;
- external forces and structural response vectors;
- $\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{S}(\boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{u})$;
4 - check the convergence criterion.

mass, i.e. divided by the fluid density times the volume of the body. In general, the added-mass is a second-order tensor, relating the fluid acceleration vector to the resulting force vector on the body.
Due to this effect, instability may arise if very deformable structures are modeled. To avoid this issue, Aitken's under-relaxation procedure is used. This methodology modifies the finite element solution, called non-under-relaxed solution, by means of a relaxation coefficient. By indicating with $\hat{\boldsymbol{u}}_i^{(l)}$ the (non-under-relaxed) finite element displacements at the l-th iteration, the under-relaxed displacements used to update the fluid boundary for the next fluid iteration are defined as

$$\tilde{\boldsymbol{u}}_i^{(l)} = \tilde{\boldsymbol{u}}_i^{(l-1)} - \gamma^{(l)} \Delta \tilde{\boldsymbol{u}}_i^{(l)}, \tag{4.8}$$

where

$$\Delta \tilde{\boldsymbol{u}}_i^{(l)} = \hat{\boldsymbol{u}}_i^{(l)} - \tilde{\boldsymbol{u}}_i^{(l-1)}, \tag{4.9}$$

l is the iteration index within the current time step and the relaxation coefficient is $\gamma^{(l)} = 1 - \omega^{(l)}$, with

$$\omega^{(l)} = \omega^{(l-1)} + (\omega^{(l)} - 1)\frac{(\Delta \tilde{\boldsymbol{u}}_i^{(l-1)} - \Delta \tilde{\boldsymbol{u}}_i^{(l)})^{\mathrm{T}} \Delta \tilde{\boldsymbol{u}}_i^{(l)}}{\|\Delta \tilde{\boldsymbol{u}}_i^{(l-1)} - \Delta \tilde{\boldsymbol{u}}_i^{(l)}\|^2}. \tag{4.10}$$

Within the typical time step, such iterative procedures stops when a convergence criterion is satisfied, that is,

$$\frac{\omega^{(l)} - \omega^{(l-1)}}{\omega^{(l-1)}} \leq 10^{-4}. \tag{4.11}$$

The effectiveness of the proposed scheme is tested in Chapter 5.

4.4 High Performance Computing

Here, several computational tools are tested in order to assess the ability to improve the performance of the developed numerical software. Such tools are known as linear algebra libraries (i.e. a set of routines performing matrix-vector computations, the solution of linear systems and eigenvalue problems) and are optimized to fully exploit hardware resources, that is expected to play a crucial role in the overall performance of the numerical software.
One supposes to perform the dot product of two vectors of size n: the complexity of the problem, in terms of the number of elementary operations to be computed, amounts to $2n$. One would think that on the same machine this operation, carried out by tools using the identical algorithm, will be executed showing the identical CPU time, but this is not true. In contrast to a naive implementation of an algorithm, these libraries are able to exploit the technology of the computer and then draw resources on the particular computer architecture, thus saving the duration of the analysis. Therefore, it is necessary to discuss the technology of modern microprocessors equipping computers.
With the advance of technology, it was possible to incorporate a larger number of transistors in a chip, thus several operations can be performed parallely by reducing the execution time. The pipeline is the maximum data parallelization of the work of a microprocessor. A pipelined CPU is composed of five specialized stages, each able to perform one of the elementary operations described above. The CPU works as an assembly line and then each stage should carry out only a specific task. At the same time,

each unit develops parallel different subsequent instructions. The progress in the technology results in the creation of microprocessors faster than the main memory; thus, the access to the memory represents a real bottleneck in the overall performance of an application. In 1970, Moore [64] suggested that the CPU speed would grow very fast, while the RAM speed would be almost constant in the next thirty years. In particular, Moore stated that processors speed would be doubled every eighteen months, while memory speed every seven years. Consequently, an application can waste a lot of time waiting for data, so involving a negative impact on the overall performance, and in addition preventing the exploitation of the high speed of the CPU by applications. A solution is to insert the cache, a small high-speed memory between the processor and main memory; therefore, the application takes advantage of loading data from the cache rather from the main memory, [65]. This solution is more efficient if data fit into the cache or data must be reused multiple times after they have been loaded into the cache. In addition to cache, other features can improve the performance, as multi-core and SSE/AVX, which are not explored by a naive implementation of an algorithm.
In the present context, the continuous differential problem becomes a discretized algebraic one described by matrices and vectors, simple to handle numerically by using a class of libraries able to solve matrix-vector operations and linear systems.

4.5 Some results

A set of computational tools is tested in order to assess their performances on solving a linear system. Banded and sparse storage techniques are used. The performance profile is estimated for by using the procedure in [66]. Let $\mathbf{S}$ be a set of solvers whose performances have to be estimated on a set of problem $\mathbf{T}$ by monitoring one or more informations. A parameter $s_{ij} \geq 0$ is related to the solver $i \in \mathbf{S}$ when it is applied to the problem $j \in \mathbf{T}$. The lower is s_{ij}, the better we can consider the solver i. For all the set $\mathbf{T}$, the performance of the solver i is compared to the best solver in $\mathbf{S}$. Let

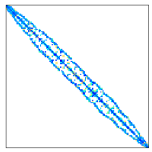
$\hat{s}_j = min\{s_{ij}; i \in \mathbf{S}\}$. Then, for $\alpha \geq 1$ and for each $i \in \mathbf{S}$ it is possible to define

$$k(s_{ij}, \hat{s_j}, \alpha) = \begin{cases} 1, \text{ if } s_{ij} \leq \alpha \hat{s_j} \\ 0, \text{ otherwise} \end{cases} . \tag{4.12}$$

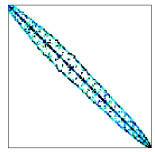
The performance profile of the solver i is given by the expression

$$p_i(\alpha) = \frac{\sum_{j \in \mathbf{T}} k(s_{ij}, \hat{s_j}, \alpha)}{|\mathbf{T}|} \quad \text{with} \quad \alpha \geq 1. \tag{4.13}$$

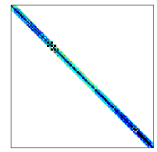
The quantity $p_i(1)$ gives the fraction of examples for which the solver i is the most efficient, in terms of the s_{ij} parameter. This parameter is identified with the involved CPU time.
The solution of the linear system $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ is tested, where $\mathbf{A}$ is a matrix of coefficients to be decomposed, $\mathbf{x}$ is the vector of the unknowns and the right-hand side vector $\mathbf{b}$ is known, [67, 68]. The coefficient matrix given by the finite element procedure, $\mathbf{A}$, has a banded structure. Therefore several ad-hoc solvers are used. The *ACML* and *MKL* routines are two variants of the same algorithm: the former is optimized for AMD processors, the latter for Intel ones. In addition, a naive implementation of the Cholesky-Crout decomposition of a banded matrix, which is not optimized with respect to hardware resources, has been implemented. Moreover, the following solvers for sparse matrices are used: *CHOLMOD*, *SPOOLES*, and *PARDISO*. To assess the efficiency of linear algebra libraries in improving the performance of the code, first it is crucial to understand the employed mathematical algorithms in the solution process, thus giving an idea of the number of elementary operations involved in the calculation, [69]. Details on the Cholesky-Crout decomposition are given in Appendix B. Table 4.2 presents test matrices from the University of Florida sparse matrix collection `http://www.cise.ufl.edu/research/sparse/matrices/` and are sketched in Figure 4.3. Performance profiles are depicted in Figure 4.4, showing that sparse solvers perform better than banded ones. In particular *PARDISO* is the best choice, since it is designed for Intel machine and compiler, as in this case. For the same reason, among the banded solvers the *MKL* library exhibits a more performant behavior. In addition, the naive implementation is the worst choice, since it doesn't exploit



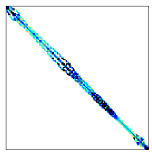
(a) bcsstk14



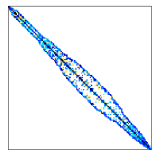
(b) bcsstk15



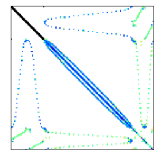
(c) bcsstk16



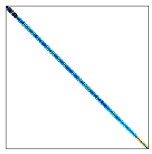
(d) bcsstk17



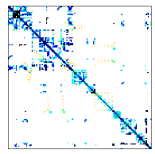
(e) bcsstk18



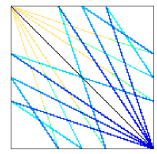
(f) bcsstk24



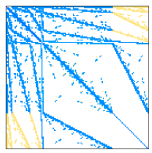
(g) bcsstk25



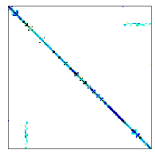
(h) ct20stif



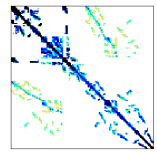
(i) cvxbqp1



(j) Dubcova2



(k) pwtk



(l) smt

Figure 4.3: Sketch of test matrices from `http://math.nist.gov/MatrixMarket/`.

Table 4.2: Linear system solution: characteristics of test matrices.

file	Dimension	Non-zeros	Band	Band/Dimension [%]
bcsstk14	1806	63454	162	8.97
bcsstk15	3948	117816	438	11.09
bcsstk16	4884	290378	141	2.89
bcsstk17	10974	428650	522	4.76
bcsstk18	11948	149090	1244	10.41
bcsstk24	3562	159910	3334	93.60
bcsstk25	15439	2522241	293	1.90
ct20stif	52329	2600295	52329	100
cvxbqp1	50000	349968	46440	92.88
pwtk	217918	11524432	9693200	84.11
smt	25710	3749582	19594	76.22

hardware resources. Consequently, analyzes implementing the naive algorithm involve CPU times which are 10-250 times longer than the analyzes using the optimized libraries. Since the solution of the linear system is the

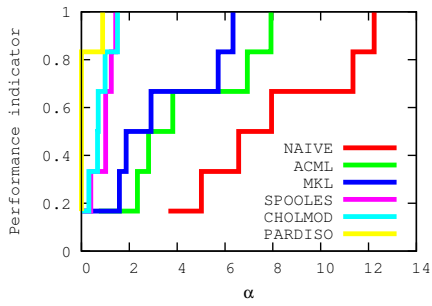


Figure 4.4: Linear system solution: performance profiles of band and sparse solvers (α in log2 scale).

operation involving the highest computational cost in the proposed FSI solution strategy, two tests are carried out on the decomposition of stiffness matrices generated by a finite element analysis: a simply supported beam and a double cantilever beam characterized by different meshes. Regarding these two beams, Table 4.3 and Table 4.4 show the CPU time expressed

in seconds. The profiles in Figure 4.5 depict the better performance of

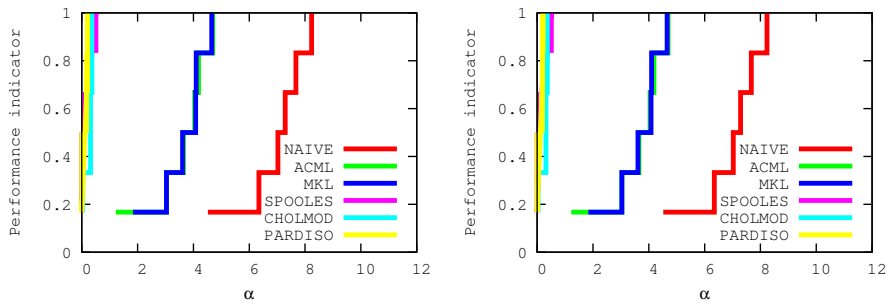
Table 4.3: Beam 1: CPU time [s] for different meshes.

Mesh	**Chol**	**Pard**	**Spool**	**Acml**	**Mkl**	**Naive**	**Dim**	**band/Dim [%]**
100-15	0.72	1.02	0.82	1.67	2.56	16.52	14600	2.06
40-25	0.45	0.48	0.44	3.62	3.61	35.72	9560	5.03
40-30	0.62	0.48	0.55	6.00	5.87	62.57	11390	5.01
40-40	0.93	0.75	0.87	13.64	12.80	152.80	15050	4.99
20-30	0.22	0.20	0.17	2.81	2.89	26.46	5790	9.86
20-40	0.31	0.25	0.25	6.50	6.27	74.60	7650	9.81

Table 4.4: Beam 2: CPU time [s] for different meshes. NaN stands for an involved CPU time larger than 1800 s.

Mesh	**Chol**	**Pard**	**Spool**	**Acml**	**Mkl**	**Naive**	**Dim**	**band/Dim [%]**
40-25	0.71	0.65	0.72	21.85	21.11	260.13	15826	5.95
40-30	0.95	0.78	0.90	37.81	34.98	1262.85	18876	5.95
40-40	1.41	1.20	1.39	89.60	77.59	NaN	24976	5.94
20-30	0.27	0.31	0.29	12.79	15.88	230.57	9596	11.70
20-40	0.40	0.39	0.40	37.51	37.22	549.23	12696	11.68

sparse solvers over banded ones. The naive implementation of the Cholesky decomposition is confirmed to be the worst choice. It has been shown that the new generation of numerical libraries is able to increase significantly the performance of the present code if compared to a naive implementation of the algorithm which does not fully exploit the computer architecture. In some cases, the speed-up was orders of magnitude larger. Such wide difference is due to the manner in which, through appropriate implementations, the CPU accesses the data it needs. Since the linear algebra has a decisive role, the idea to use optimized numerical libraries to reduce the duration of analyzes arose.
Usually, the matrix to be decomposed is real, symmetric, positive definite and shows a band pattern. This is common in the academic examples, but complex geometries tend to impede band patterns. Solvers developed for band and sparse storage were employed and results show the most performant tool is *PARDISO*, since it is developed for a computer equipped with



(a) Beam 1. (b) Beam 2.

Figure 4.5: Performance profiles (α in log2 scale).

an Intel processor. Thus, it can be stated that the adoption of such sparse solver can be taken into consideration even in cases in which the matrix has a band structure, since it demonstrates superior performance than solvers designed ad-hoc for a band storage. In addition, the *MKL* library is the best choice among the band solvers, if Intel machine and processor are used.

Chapter 5

Verification & validation

Sommario

In questo capitolo l'affidabilità della metodologia proposta viene verificata risolvendo alcuni casi studio tipici della letteratura del settore. Dapprima, vengono studiati corpi rigidi investiti dal fluido, allo scopo di valutare su questi i coefficienti aerodinamici di drag e di lift attestanti la corretta valutazione del tensore degli sforzi sia su punti allineati con la griglia che genericamente posizionati. Si passa quindi a studiare corpi il cui moto è consentito da molle elastiche, mostrando l'affidabilità dell'algoritmo di accoppiamento esplicito. Successivamente, l'influenza di diverse condizioni al contorno di parete viene verificata nella valutazione del tensore degli sforzi su corpi rigidi. Infine, si considerano i solidi come corpi deformabili. L'attenzione verte su due mensole incastrate a corpi o contorni rigidi e investite dal fluido. Lo scopo di queste due applicazioni è quello di mettere in risalto l'efficacia delle diverse strategie di accoppiamento, nonchè delle metodologie di imposizione delle condizioni al contorno e dello schema di integrazione dell'equazione del moto. L'effetto di massa aggiunta viene valutato rispetto a un caso studio, mostrando l'efficacia dello schema di Aitken nello stabilizzare la procedura.

5.1 Fixed rigid bodies

In the following testes, rigid fixed bodies are considered. For the square cylinder benchmark, no curved boundary conditions are needed, since it is aligned with the lattice grid. On the other hand, if solids not aligned with the grid are involved in the computation, the second-order accurate condition in [32] is used.

Square cylinder

Herein, a square cylinder is invested by a viscous fluid, [46]. As depicted in Figure 5.1, a parabolic velocity profile is present at the inlet section, whose top value is equal to 0.03. Outflow conditions are used at the outlet (i.e. zero-gradient of velocity), while at both top and bottom wall the no-slip boundary condition is implemented. The edge of the square cylinder has length $D = 10$, the length of the channel is $L = 30D$ and the height is $H = 8D$. The distance between the inlet section and the left edge of the square cylinder is set to $L/3$. All the above quantities are shown in lattice units. Various values of the Reynolds number, $\mathcal{R}e = \frac{UD}{\nu}$, are achieved

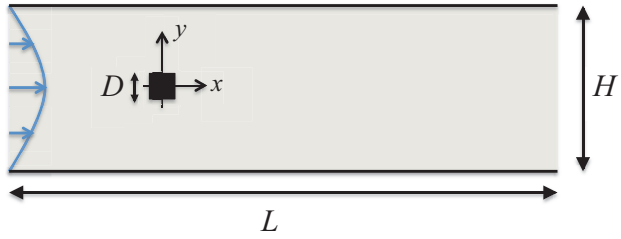
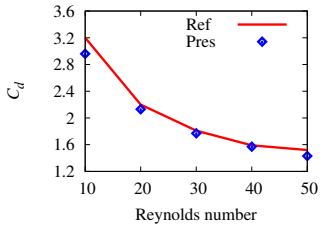


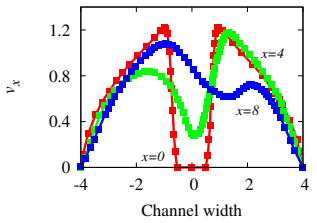
Figure 5.1: Sketch of problem definition.

by modifying the relaxation parameter τ, being U the peak value of the inlet velocity profile and ν the fluid viscosity. Figure 5.2(a) plots the drag coefficient C_d against the Reynolds number, showing a perfect agreement between benchmark and present values. In Figure 5.2(b), velocity profiles, normalized with respect to the peak value of the inlet velocity profile, in three different vertical sections are presented: one is located at the middle of

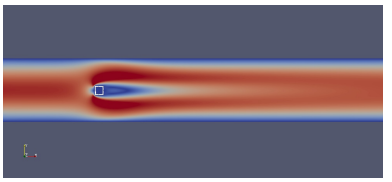
the obstacle ($x = 0$); a second is quite near the right edge of the obstacle, specifically $4D$ far from ($x = 4$); the last is $8D$ far form the right edge ($x = 8$). Continuous lines represent benchmark curves, while dots show present results. A slight difference can be identified between present and reference values.



(a) Drag coefficient vs Reynolds number.



(b) Normalized horizontal velocity profiles.



(c) Velocity field.

Figure 5.2: Square cylinder benchmark.

Circular cylinder

A cylindrical cylinder is immersed in a viscous fluid, [32]. At the inlet and outlet sections, boundary conditions are identical to the previous case. Free-slip boundary conditions are enforced at bottom and top walls. Referring to Figure 5.3, the cylinder has diameter $D = 16.66$, the grid is composed of 350×220 lattice sites and the distance between the inlet section and the center of the cylinder is 117.
The variation of the drag coefficient with the Reynolds number is depicted in Figure 5.4(a), while Figure 5.4(b) shows the velocity profile in the ver-

tical section from the top point of the cylinder to the top wall normalized with respect to the peak value of the inlet velocity profile. Both figures demonstrate the effectiveness of the numerical procedure.

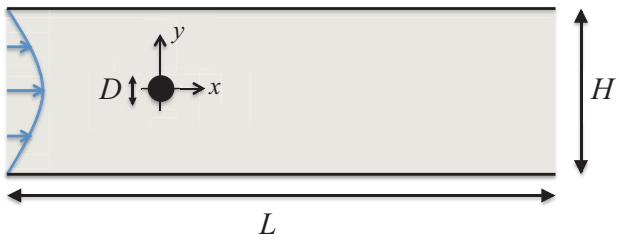
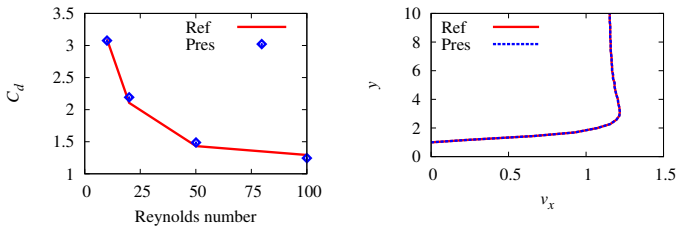
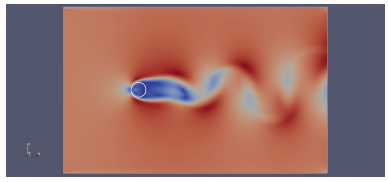


Figure 5.3: Sketch of the problem definition.



(a) Drag coefficient vs Reynolds number.

(b) Normalize upstream velocity profile.

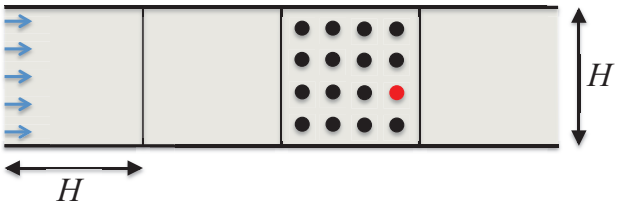


(c) Velocity field.

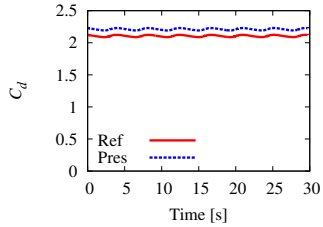
Figure 5.4: Circular cylinder benchmark.

Arrays of cylinders

Here, the test in [70] is proposed. The fluid domain consists of four square blocks, horizontally aligned, see Figure 5.5(a).



(a) Sketch of the problem definition.



(b) Time history of the drag coefficient.

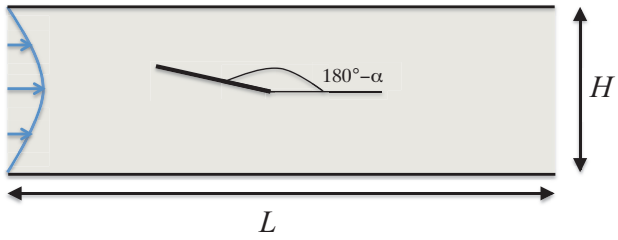
Figure 5.5: Arrays of cylinders.

In the second block, a set of four arrays of cylinders, each composed of four cylinders, is invested by a viscous fluid and the flow is characterized by a Reynolds number equal to 200. Each block has dimension $H = 370$. Moreover, the diameter of the cylinders and the distance between the nearest points of any pair of cylinders are both equal to $H/8$. In addition, topmost points of the first horizontal row are far $H/16$ from the wall (the same for the bottom wall). In the inlet section, a constant velocity profile $U = 0.1$ is prescribed; free-slip boundary conditions at bottom and top walls and outflow conditions at the outlet complete the definition of the problem. All the previous quantities are expressed in lattice units. The drag force acting on the red cylinder is tracked. Figure 5.5(b) depicts a slight mismatch (5%) between present findings and benchmark results in [70] imputable to the

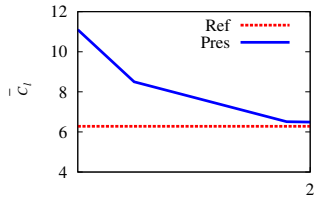
fact no grid refinements are used. Notice that a low computational cost is involved in the solution process if compared to the previous test case.

Airfoil

Under the assumptions of inviscid fluid (i.e. Reynolds number $\rightarrow \infty$), incompressible and irrotational flow, a potential theory can be used to study the aerodynamics of a bar, [71]. In particular, for a bar of length b and deadrise angle α, as depicted in Figure 5.6(a), the lift coefficient normalized respect to b and α is equal to 2π. Here, a 1 meter long bar is represented by 6500 lattice nodes and the deadrise angle is $\alpha = 3°$. In addition, a velocity profile characterized by a top value V=10 m/s (0.1 in lattice units) is prescribed at the inlet section; outflow boundary conditions are implemented at the outlet and free-slip conditions characterize top and bottom walls.



(a) Sketch of the problem definition.



(b) Normalized lift coefficient vs Reynolds number.

Figure 5.6: Airfoil.

The grid dimensions are 10500×941 and the distance between the inlet section and the leftmost point of the bar is equal to 2000. As known, the LB method is a viscous fluid solver. Therefore, in order to achieve a value of $\bar{C}_l = 2\pi$, conditions characterized by high Reynolds number have been simulated. In addition, the viscous component of the stress tensor has been neglected when computing forces on the bar. Figure 5.6(b) shows that the higher the Reynolds number, the closer the result of the simulation to the reference value is. The relaxation parameter τ is varied to achieve different values of the Reynolds number.

5.2 Moving rigid bodies

The simulated test case is a uniform flow over an oscillating cylinder that can move only in the direction normal to the inlet velocity. The geometry of the problem is displayed in Figure 5.7.

Figure 5.7: Oscillating cylinder: definition of the problem.

The spring is linear, with stiffness $k = 5.79$ N/m and damping factor $c = 0.325$ g/s. The mass and the diameter of the cylinder are $m = 2.979$ g and $D = 0.16$ cm. The fluid is water with viscosity $\mu = 0.01$ g/(cm s) and density $\rho = 1$ g/cm^3. Different far-field fluid velocities are considered, such that the Reynolds number, given by $\mathcal{R}e = \rho U D/\mu$, varies between 90 and 130. In all numerical experiments, the fluid is started from rest conditions. At the inlet, a given mainstream speed U is prescribed, while at the outlet, zero pressure boundary conditions are imposed. At the upper and lower boundaries, zero-velocity conditions are applied. Zero-velocity conditions

are also applied at the moving surface of the cylinder.
Figure 5.8(a) shows the vortex shedding frequency f_V as a function of the Reynolds number. The frequency f_V is evaluated from the evolution of the lift coefficient and shown in Figure 5.8(a). The quantity f_N is the natural frequency of the rigid body as a mass-spring system ($f_N = \sqrt{k/m}$). Benchmark values are taken from the vortex shedding frequencies associated with the stationary cylinder, as suggested in [72, 73]. Figure 5.8(b) shows the relative amplitude Y/D as a function of the Reynolds number, where Y is the displacement along vertical direction induced by the lift force. The figure clearly shows the onset of a symmetry-breaking instability in the range $95 < \mathcal{R}e < 110$, in close agreement with the experimental data obtained in [74]. At the low Reynolds number end of the lock-in region, the oscillations start with a sudden jump-up. As the Reynolds number increases, the amplitude of the oscillations smoothly decreases so that the jump-down at the upper end of the lock-in region is less significant. Figures 5.8(a) and 5.8(b) show a good agreement between our results and literature data. In Figures

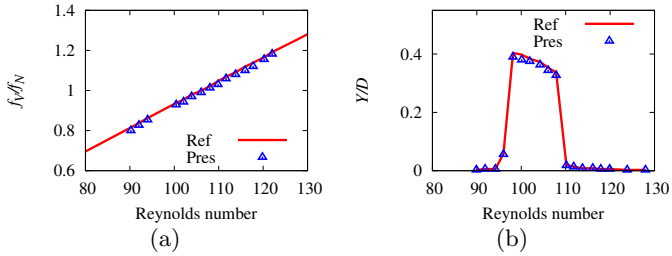


Figure 5.8: Oscillating cylinder: a) frequencies vs Reynolds number. The solid line denotes the vortex shedding frequency associated with a stationary cylinder, [72, 73]; b) amplitudes of the vertical motion vs the Reynolds number. The figure clearly displays the onset of a symmetry-breaking instability in the range $95 < \mathcal{R}e < 110$, in close agreement with experimental results (solid line), [74].

5.9, the time evolution of the relative amplitude Y/D is plotted at $\mathcal{R}e = 90$ and $\mathcal{R}e = 120$. In a standard PC (Intel I7-920, frequency clock 2.6GHz, 8 cores, 8 MB cache L3, 8 GB RAM), the method takes about 0.05 CPU seconds per time step, for a LB grid of 350×220 lattice sites, slightly higher

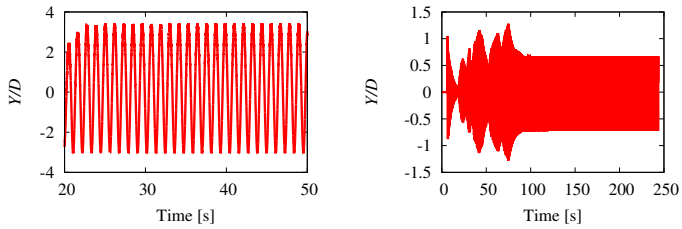
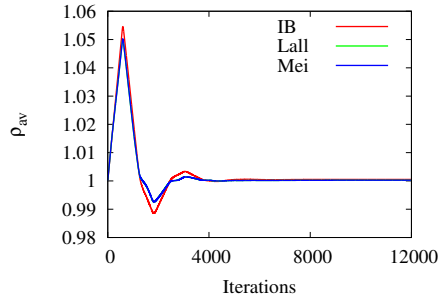


Figure 5.9: Amplitude of the oscillations: (a) $\mathcal{R}e = 90$ and (b) $\mathcal{R}e = 120$.

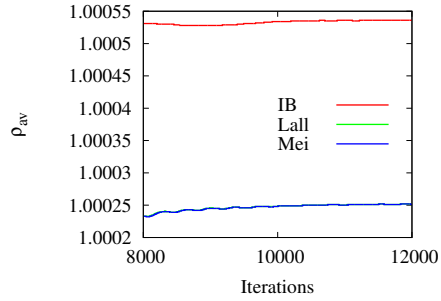
(about 10%) than the computational time needed for the flow over a fixed cylinder. The computational cost is mainly due the fluid-solid boundary condition. The computational cost due to the fluid solver is negligible, due to the fact that the rigid body motion is described by a single degree of freedom.

5.3 Boundary conditions

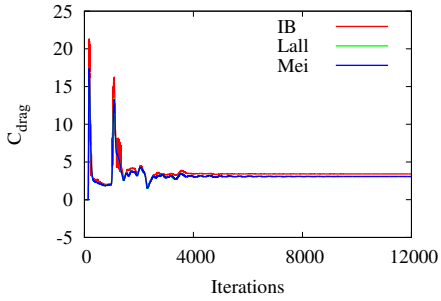
In this section, the above discussed boundary conditions are tested in order to highlight their different behavior. In particular, these conditions are used to simulate a cylinder obstacle invested by a viscous fluid for a given Mach number, $\mathcal{M}a = 0.0289$, and different values of the Reynolds number, $\mathcal{R}e = 10$ and $\mathcal{R}e = 50$. At the inlet section, a parabolic velocity profile is inserted, while at the outlet fixed density and $\frac{\partial \mathbf{v}}{\partial n} = 0$ are set. At bottom and top walls, the orthogonal velocity component is null, while on the cylinder surface the no-slip condition is imposed. The center of the cylinder is located at $(Xc, Yc) = (100, 110)$ with a diameter $D = 20$, and the grid consists of 350×220 lattice nodes.
The time-history of the average density, together with the time-histories of the drag and lift coefficients are depicted in Figures 5.10 and 5.11, for $\mathcal{R}e = 10$ and $\mathcal{R}e = 50$, respectively. As it is possible to observe, a general agreement between the three boundary conditions is met. In particular, the Immersed boundary method predicts larger values of average density,



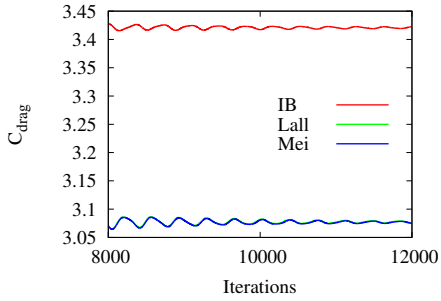
(a) Average density vs iterations. $\mathcal{R}e = 10$.



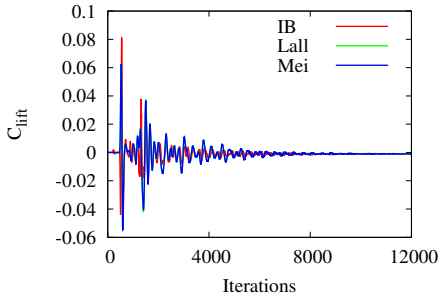
(b) Average density vs iterations: zoom. $\mathcal{R}e = 10$.



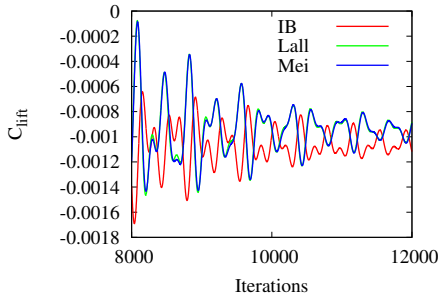
(c) Drag coefficient vs iterations. $\mathcal{R}e = 10$.



(d) Drag coefficient vs iterations: zoom. $\mathcal{R}e = 10$.

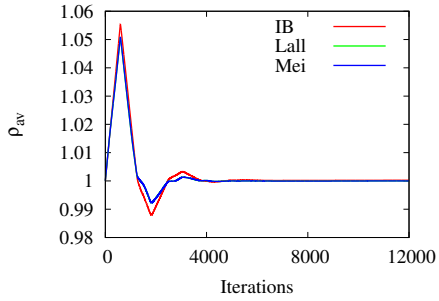


(e) Lift coefficient vs iterations. $\mathcal{R}e = 10$.

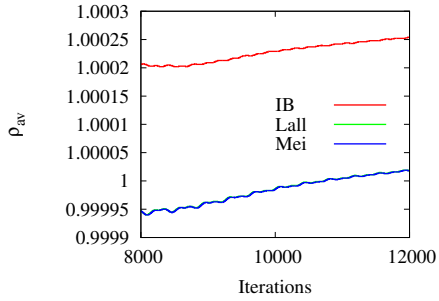


(f) Lift coefficient vs iterations: zoom. $\mathcal{R}e = 10$.

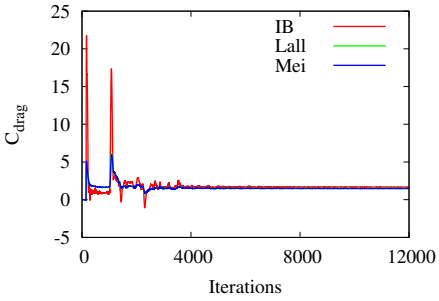
Figure 5.10: Time histories of average density, drag coefficient and lift coefficient at $\mathcal{R}e = 10$.



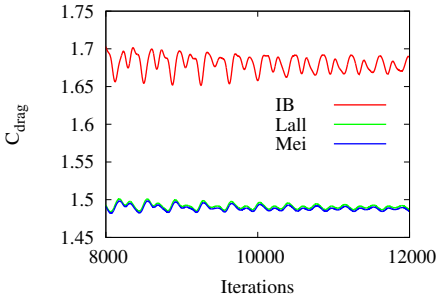
(a) Average density vs iterations. $\mathcal{R}e = 50$.



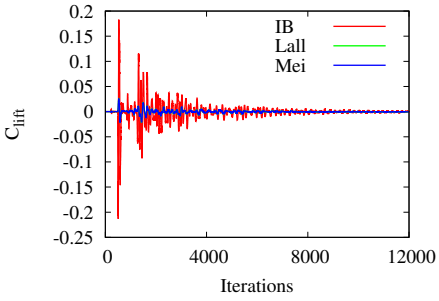
(b) Average density vs iterations: zoom. $\mathcal{R}e = 50$.



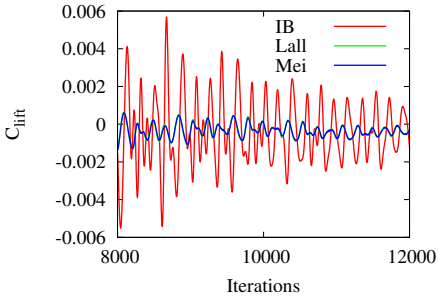
(c) Drag coefficient vs iterations. $\mathcal{R}e = 50$.



(d) Drag coefficient vs iterations: zoom. $\mathcal{R}e = 50$.



(e) Lift coefficient vs iterations. $\mathcal{R}e = 50$.



(f) Lift coefficient vs iterations: zoom. $\mathcal{R}e = 50$.

Figure 5.11: Time histories of average density, drag coefficient and lift coefficient at $\mathcal{R}e = 50$.

drag and lift coefficient, than the other two strategies. This slight mismatch suggests that, for a given grid resolution, the interpolated bounce-back condition is more accurate. On the other hand, the IB implementation is very flexible, since the only input consists of the set of Lagrangian boundary coordinates. Thus, the IB method is more general and can be used in situations involving very complex geometries. In order to use the condition in [32, 44] to handle complex geometries, where *complex* stands for geometries that cannot be easily identified by a mathematical function, this operation becomes computationally hard, since a lot of Eulerian informations, i.e. $\boldsymbol{x}_f$, $\boldsymbol{x}_w$, $\boldsymbol{x}_b$, δ, must be computed and given as input.
In a second test, the obstacle possesses square cross section. The lattice grid consists of 500 nodes in horizontal direction and 80 in vertical direction (IB1). The length of the edge of the square cylinder is 10 lattice units and the distance between the leftmost edge and the inlet section is 125 nodes. Inflow and outflow conditions are identical to the previous case, while at top and bottom walls the no-slip condition is adopted. In this case, it is useless to use the boundary conditions in [32, 44], since the obstacle is aligned to the grid and Equations (2.33) and (2.35) reduce to the standard bounce-back and no interpolated scheme is needed. The comparison between the standard bounce-back on link and the IB method is shown in Table 5.1. In particular, for a given Reynolds number, the drag coefficient is compared to results in [46]. The IB method is also used on two more refined grid: the first consists of 1000×160 lattice nodes (IB2), while the second has dimensions 1500×240 (IB3). Values which are closer to the reference ones are shown. Even in this case, the better performance of the bounce-back scheme are substantially confirmed.

Table 5.1: Drag coefficient vs Reynolds number: reference values from [46] (Ref), bounce-back on link (BB), and immersed boundary method on increasingly refined grid (IB1, IB2, IB3).

$\mathcal{R}e$	Ref	BB	IB1	IB2	IB3
10	3.2	2.96	3.88	3.58	3.47
20	2.2	2.13	2.74	2.51	2.48
30	1.81	1.77	2.29	2.07	2.05
40	1.59	1.57	2.01	1.85	1.83
50	1.52	1.43	1.88	1.69	1.67

5.4 Deformable bodies

Cantilever beam under constant moment

A test is carried out in order to validate the corotational formulation employed to take in account for large displacements. A constant moment

$$M = \frac{2\pi EI}{L} \tag{5.1}$$

is applied to the free tip of a cantilever beam (see Figure 5.12) of length $L = 10^3$, elastic modulus $E = 10^5$, Poisson ratio equal to zero, cross sectional area $A = 1$, and inertia moment $I = 1/12$. The beam is discretized by using 10 finite elements. Figure 5.13 show the deformed configurations of

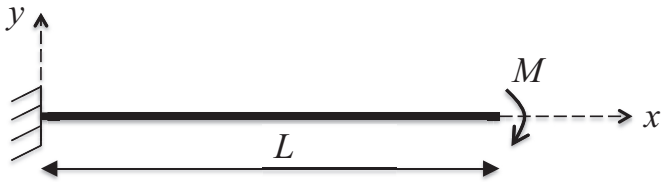


Figure 5.12: Cantilever beam under constant moment.

the cantilever beam due to different values of the dimensionless constant moment

$$M^\star = \frac{ML}{2\pi EI}. \tag{5.2}$$

As it is possible to observe, large displacements are successfully captured.

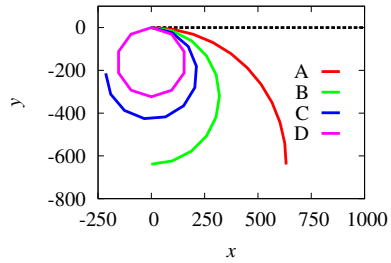


Figure 5.13: Deformed configurations of a cantilever beam under constant moment: A) $M^\star = 0.25$, B) $M^\star = 0.50$, C) $M^\star = 0.75$, D) $M^\star = 1.00$. The dashed line represents the undeformed configuration.

Deformable flag clamped to a rigid fixed square cylinder

Consider a viscous fluid that flows past a rigid fixed square cylinder, as depicted in Figure 5.14, [75]. The channel height is $H = 12D$, while longitudinal dimensions L_l and L_r are set to $5.5D$ and $14D$, respectively, D being the length of the edge of the square cylinder. A deformable flag is clamped to the square cylinder and its length is $L = 4D$. At the inlet, a constant velocity profile $V = 51.3\,\mathrm{m/s}$ is given, while at the outlet outflow boundary conditions are imposed. At the fluid-solid interface the boundary condition in [32] is used. The flow is characterized by a Reynolds number $\mathcal{R}e = \frac{VD}{\nu} = 333$. Wall free-slip conditions at bottom and top walls complete the fluid definition. The cantilever possesses Young modulus $E = 2.5 \cdot 10^6\,\mathrm{N/m^2}$, cross sectional area $A_0 = 0.06 \times 0.06\,\mathrm{m^2}$ and density $\rho_s = 0.1\,\mathrm{kg/m^3}$. In Figure 5.15, the time history of the vertical displacement of the end node is depicted together with the solution computed in [75]. The present solution is obtained in lattice dimensionless units with a rectangular grid, spanning 1734×1068 lattice sites, with an inlet velocity profile $V = 0.05$, viscosity set to $\nu = 0.0133$ and characteristic length D equal to 89. The LB simulation possesses the same Reynolds number of the physical problem. The characteristic velocity, the characteristic length and the viscosity are shown in Table 5.2 in both physical and lattice units.

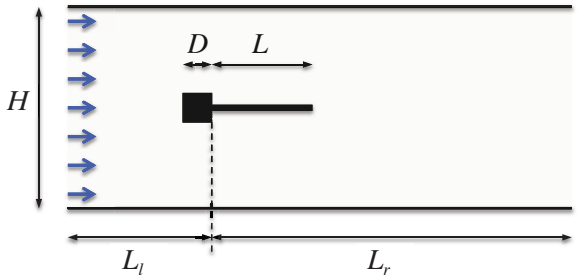


Figure 5.14: Sketch of problem definition.

The cantilever beam is discretized with 356 elements. As it can be observed

Table 5.2: Physical and lattice units corresponding to $\mathcal{R}e = 333$.

	Physical	Lattice
Velocity	$51.3, \mathrm{m}^2/\mathrm{s}$	0.05
Length	1 m	89
Viscosity	$0.154\,\mathrm{m}^2/\mathrm{s}$	0.0133

in Figure 5.15, a close agreement between the reference solution (Ref) and present numerical results is obtained (Pres). In Figure 5.16, the velocity

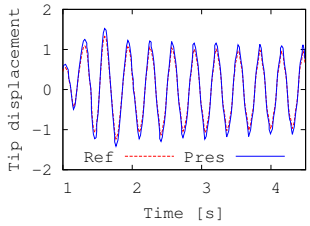


Figure 5.15: Time history of the tip vertical displacement [m].

filed is depicted. The convergence properties of the three proposed coupling strategies, discussed in Chapter 4, are assessed in Figures 5.17 for a Reynolds number equal to 10. In the following, the explicit algorithm is indicated by $EXPL$, the explicit predictor-based algorithm by $FELBA$,

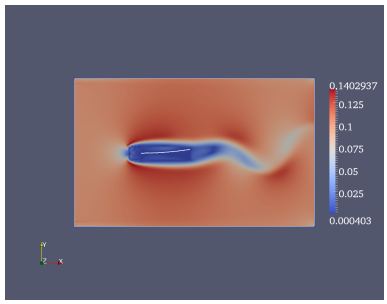


Figure 5.16: Evolution of the velocity field. The vortex shedding, characteristic of the fluid flow conditions of the simulation ($\mathcal{R}e = 333$), is clearly visible. The movie is available in the pdf version.

and the implicit one by $IMPL$. Uniform refinements of grid and beam discretization are adopted. The reference solution is obtained using a very fine grid with dimensions 2598×1600. In the first figure, the percentage error at $t = 1\,\mathrm{s}$ in the drag force computation can be observed against the number of lattice sites in a log-log scale. The red curve corresponds to the fixed flag case and can be considered as an upper bound on convergence rate, because it is not affected by any error due to coupling, but it is affected only by the error due to the fluid solver. In the second figure, the convergence of the tip displacement is shown. Both the figures highlight that all the algorithms exhibit good convergence properties and that the implicit coupling algorithm is characterized by an almost optimal convergence rate, that is equal to the one experienced in the case of rigid beam. As expected, the lower computational cost involved by the explicit strategy is paid by a slower convergence rate. Finally, the predictor is proved to actually improve the convergence properties also in the explicit version. The computational cost of the explicit version is almost the same of the fluid solution with a rigid structure. Thus, no appreciable increase of the computational effort is experienced with respect to the fluid solver alone. Using the predictor involves a very slight increase of the computational time (about 1%). On the contrary, the computational effort is almost double using the implicit algorithm with a convergence tolerance equal to 10^{-4}, since the average

number of iterations per time step is 2.2. In Figures 5.17(c) and 5.17(d), the effectiveness of the TDG integration scheme for the structural dynamics is compared to the backward Euler scheme and to the trapezoidal rule. In both cases, independently from the coupling strategy, the TDG performs better than the other two schemes in terms of both accuracy and convergence.

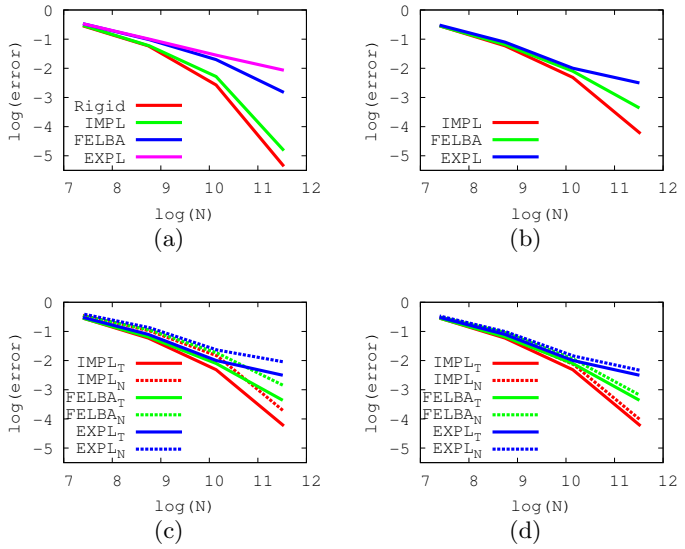


Figure 5.17: Convergence properties of the three proposed coupling algorithms: (a) drag, (b) tip displacement, (c) drag using TDG and backward difference (dashed line), (d) drag using TDG and trapezoidal rule (dashed line). The number of lattice sites is denoted by N.

The error introduced by the coupling algorithms at the fluid-structure interface is investigated in Figure 5.18(a). Specifically, the figure shows the dimensionless interface energy computed from the fluid side against that computed from the structure side. As assessed in Chapter 4, these two quantities should be equal if the interface conditions are exactly satisfied. Thus, the predicted values at the various instants of the simulations should

lie upon the black line. The larger the deviation from this ideal behavior, the larger the associated error is. As it can be observed, all the coupling algorithms exhibit an almost linear relation, but the error tends to increase passing from implicit to explicit version. Also in this case, a remarkable improvement is experienced by adopting the predictor. Figures 5.18(b) and 5.18(c) show the same dimensionless interface energies but predicted by using the trapezoidal and backward schemes instead of the TDG one. Very similar results are obtained apart from the amplitude. It should be noted that the maximum values are 10% larger using the trapezoidal scheme and 50% larger using the backward one.

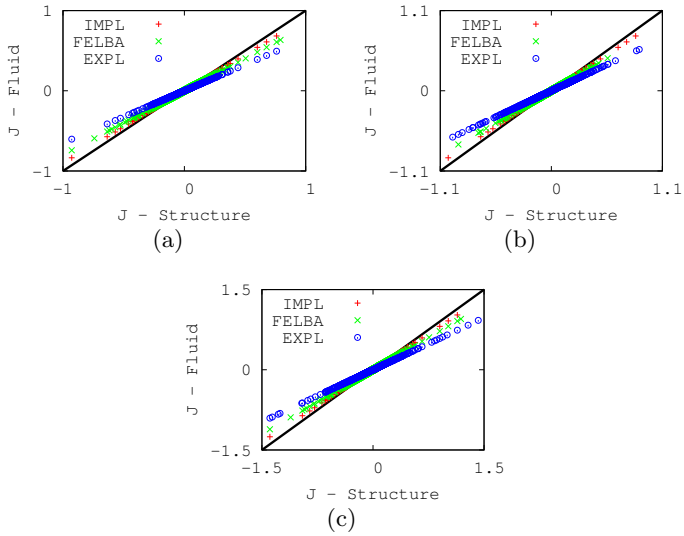


Figure 5.18: Interface energy for the three coupling algorithms using: (a) TDG scheme, (b) trapezoidal scheme, (c) backward difference scheme.

Backward facing step

A viscous fluid passing over a backward facing step impacts a flexible cantilever, as in [76]. Figure 5.19 depicts the problem setting, where lengths are

expressed in meters. It is characterized by an inlet constant velocity profile equal to 1 m/s, outflow boundary condition at the outlet and free-slip conditions elsewhere. The flexible cantilever is treated as a curved boundary condition immersed into the lattice background by means of the interpolation scheme in [32]. Moreover, fluid and solid densities are set to $1000\,\mathrm{kg/m^3}$ and $7800\,\mathrm{kg/m^3}$ and the cross sectional area is $0.3 \times 0.3\,\mathrm{m^2}$. The Reynolds number is set equal to 10 and the Young modulus to $10^5\,\mathrm{N/m^2}$. The grid has dimensions 6400×600. In Figure 5.20 the velocity field, normalized to

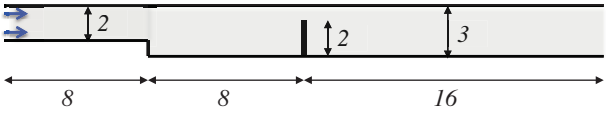
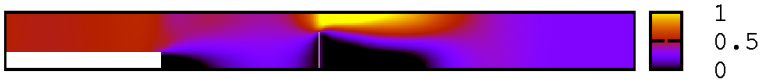
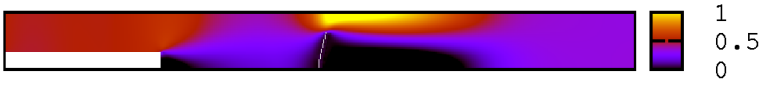


Figure 5.19: Sketch of the problem definition. Lengths are expressed in meters.

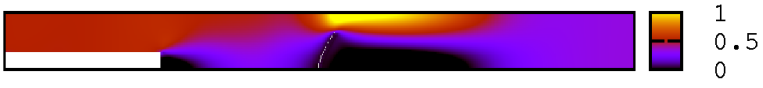
the peak value (2.6 m/s), is reported for three different time instants. The implicit version of the coupling algorithm and the TDG time integration scheme are employed. As in the previous case, a convergence analysis on



(a) Velocity, $t = 8.50\,\mathrm{s}$.



(b) Velocity, $t = 11.03\,\mathrm{s}$.



(c) Velocity, $t = 11.92\,\mathrm{s}$.

Figure 5.20: Velocity magnitude field at three different instants normalized with respect to the top value (2.6 m/s).

the drag force at $t = 20\,\mathrm{s}$ acting on the cantilever has been carried out by uniformly refining the fluid grid and the beam discretization, as shown

in Figure 5.21. The results obtained in the previous test are substantially confirmed in terms of convergence properties. Also in this case, the predictor is proved to be very effective. The computational cost of the explicit algorithm is confirmed to be comparable to that involved by the sole fluid solver in the case of a rigid obstacle. No significant increase of the computational cost is experienced using the structure predictor. As expected, the computational effort increases with the implicit algorithm by a factor 1.8, using a convergence tolerance equal to 10^{-4}. In fact, the average number of iterations per time step is 1.8. Then, the fluid-structure interface energy

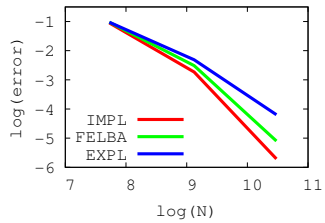


Figure 5.21: Drag convergence for the three coupling algorithms. The number of lattice sites is denoted by N.

is investigated for different Reynolds numbers: $\mathcal{R}e = 20$ and $\mathcal{R}e = 60$, and different cantilever Young moduli: $E = 10^3\,\mathrm{N/m^2}$, $E = 10^5\,\mathrm{N/m^2}$ and $E = 10^7\,\mathrm{N/m^2}$. The Reynolds number is defined as $\mathcal{R}e = \frac{VD}{\nu}$, where the characteristic inlet velocity is set to $V = 0.01$ and the characteristic length, the one of the cantilever, is $D = 400$, both in lattice units. Different Reynolds numbers are achieved by varying the viscosity ν. For the sake of brevity, only the results predicted by the implicit coupling algorithm are reported. Both TDG and trapezoidal schemes are used for structural dynamics. As Figure 5.22 highlights, the coupling algorithm seems to be robust and very accurate. The TDG scheme is confirmed to be more accurate than the trapezoidal scheme, as it can be realized by considering the axes ranges on the graphs in Figure 5.22. Indeed, the superior performance of TDG affects also stability properties. This is clearly shown by Figure 5.23 that refers to the case of $\mathcal{R}e = 120$ and $E = 10^7\,\mathrm{N/m^2}$, with the grid

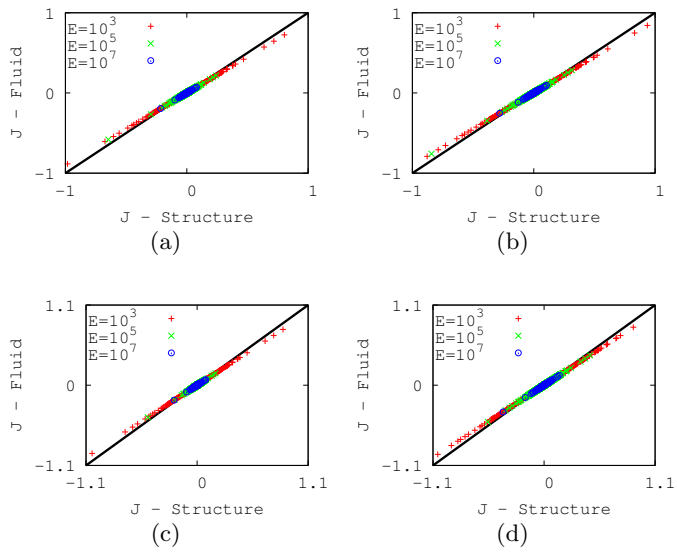


Figure 5.22: Interface energy of the implicit coupling algorithm for different Reynolds numbers, cantilever Young moduli and time integration schemes for structural dynamics: (a) $\mathcal{R}e = 20$ and TDG scheme, (b) $\mathcal{R}e = 60$ and TDG scheme, (c) $\mathcal{R}e = 20$ and trapezoidal scheme, (d) $\mathcal{R}e = 60$ and trapezoidal scheme.

dimensions previously defined. For these values, the algorithm with the trapezoidal method as structural time integration scheme becomes unstable and fails. On the contrary, in the same conditions the algorithm with the TDG is still stable and successfully compute the solution. Summing up,

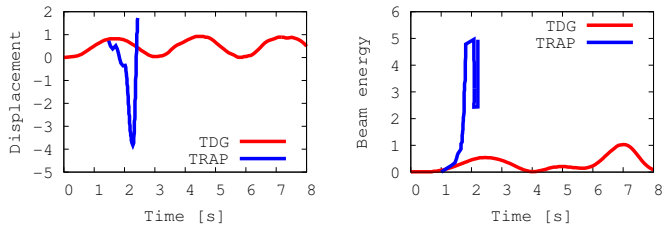


Figure 5.23: Implicit coupling algorithm with TDG and trapezoidal scheme in the case of $\mathcal{R}e = 120$ and $E = 10^7\,\mathrm{N/m^2}$: (a) normalized tip displacement, (b) normalized beam total energy.

the implicit algorithm is the most accurate and exhibits a nearly optimal convergence rate, that is the same convergence rate experienced in the case of the fluid with a rigid structure. However, it involves the highest computational cost. In general, three iterations are sufficient and the overall computational cost is three times that involved by the explicit algorithm. The explicit algorithm enhanced with structure predictor offers a very good convergence rate with a computational cost that is comparable to the one involved by the sole fluid solver in the case of rigid structure. All the numerical tests confirm the superior properties of TDG scheme with respect to standard integration schemes for structural dynamics, in terms of both accuracy and stability.

5.5 Added-mass effect

The added-mass effect is tested against the problem sketched in Figure 5.24 and defined in [77]. At inlet west section an incoming pressure is imposed, while at the outlet east section an outflow boundary condition is set. On the

bottom wall, free-slip boundary condition is used. On the top wall, a slender elastic beam is present, whose configuration can be modified by the fluid flow. At the fluid-solid interface, no-slip condition is imposed. For a given

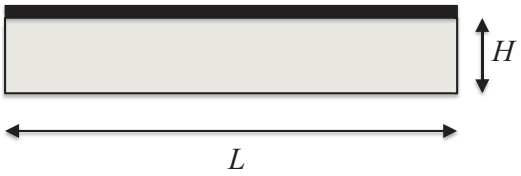


Figure 5.24: Sketch of the problem definition.

Reynolds number, the ratio r_ρ between the fluid density, ρ_f, and the solid density, ρ_s, is varied and the corresponding number of iterations is depicted in Figure 5.25. As expected, the lower r_ρ is, the more iterations need

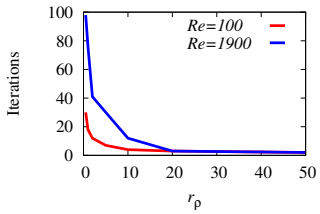


Figure 5.25: Number of iterations against r_ρ for $\mathcal{R}e = 100$ and $\mathcal{R}e = 1900$.

in order to achieve convergence, since the structure deformability grows. Notice that, if the Reynolds number increases, for a given r_ρ, the number of iterations increases too. In fact, higher values of the viscosity play an important role in the coupling algorithm, stabilizing the overall procedure. In Table 5.3 the number of iterations of the implicit coupling algorithm are reported for $\mathcal{R}e = 100$ and $\mathcal{R}e = 1900$ and for different values of r_ρ. Moreover, the number of iterations in the standard FELBA implicit algorithm, B, is compared to the usage of Aitken's scheme in the same coupling algorithm, A. Notice that, for high values of r_ρ, the number of iterations is slightly lower in B. On the other hand, Aitken's scheme is

Table 5.3: Number of iterations needed to achieve convergence for r_ρ ranging from 0.5 to 50 and two different Reynolds numbers, $\mathcal{R}e = 100$ and $\mathcal{R}e = 1900$. Simulations using Aitken's scheme correspond to A, otherwise B. The symbol $\cdot$ indicates that no convergence has been achieved.

r_ρ	$\mathcal{R}e = 100$		$\mathcal{R}e = 1900$	
	A	B	A	B
50	2	1	2	1
20	3	2	3	3
10	4	4	12	12
5	7	7	30	30
2	12	12	41	$\cdot$
1	18	18	76	$\cdot$
0.5	30	$\cdot$	98	$\cdot$

required to achieve convergence if low values r_ρ are used.

Chapter 6

Some applications

Sommario

In questo capitolo, viene proposta una serie di applicazioni volutamente in ambiti diversi. Dapprima, è stato studiato un fenomeno noto come hull slamming, che consiste nell'impatto degli scafi delle imbarcazioni contro la superficie marina. Questo fenomeno è caratterizzato da brevisssime durate ed elevati e localizzati picchi di concentrazione di sforzo. Una seconda applicazione riguarda il flusso sanguigno pulsante in un'arteria. La deformabilità del vaso sanguigno è stata considerata tramite una relazione costitutiva che tenesse conto del comportamento anisotropo, viscoealstico e irrigidente della stessa. Infine, un'affascinante applicazione riguarda la simulazione del decollo verticale di una farfalla, le cui ali sono simulate come due travi incernierate e immerse in un fluido. In particolare, è stata indagata l'incidenza della forza di gravità, della massa e della deformabilità delle ali sul librarsi in volo. La scelta di applicazioni afferenti a tematiche diverse con il comune denominatore dell'interazione fluido-struttura è stata orientata volutamente a indagare situazioni e fenomenologie molto diverse, allo scopo di dimostare la versatilità e l'affidabilità della metodologia proposta.

6.1 Hull slamming

The proposed method is tested against the simplified 2D hull slamming problem schematically depicted in Figure 6.1 along with the cartesian coordinates x and y. Specifically, the impact of a wedge composed of two straight beams clamped at the deadrise angle β on the free surface of a placid fluid is studied. A simplified and effective free-surface model, schematically represented in Figure 6.2, is employed here. The air-water interface is regarded as a boundary condition by assuming that the row of nodes that lays on the free-surface (red) is traction-free. Furthermore, air flow is neglected, thus, solving the LB only in the liquid nodes. Therefore, the stress tensor is computed only on the immersed portions of the wedge.

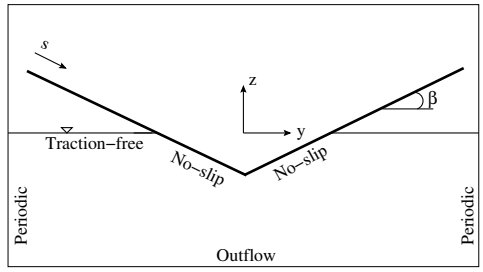


Figure 6.1: Sketch of problem geometry.

Results are compared with analytical and experimental findings spanning a broad set of relevant scenarios. All comparisons refer to a mass density of $1000\,\mathrm{kg/m^3}$. All simulations are conducted on structures whose dimension is on the order of 1 m. The fluid domain is discretized with a spatial resolution of $\Delta x = 2 \times 10^{-5}\,\mathrm{m}$ and it consists of a grid of $2,800$ lattice nodes in the vertical y-direction and $\left(5\,000 + \frac{D}{\Delta x}\right)$ nodes in the horizontal x-direction, where D is a typical characteristic length for the hull corresponding to the largest values attained by the hull wet surface during each analysis. Therefore, the computational domain has a width which is on the order of three times the maximum wet surface area of the wedge to

minimize the boundary effects. On the lateral sides of the domain, periodic boundary conditions are used; on the bottom, outflow is imposed; and on the wedge-water interface, no-slip conditions are enforced. The wedge geometry is symmetric with respect to the y-axis. Constitutive parameters and boundary conditions are varied across simulations to compare with results from the literature. The fluid is always initially at rest, that is, the density is equal to ρ_f and the speed is zero.

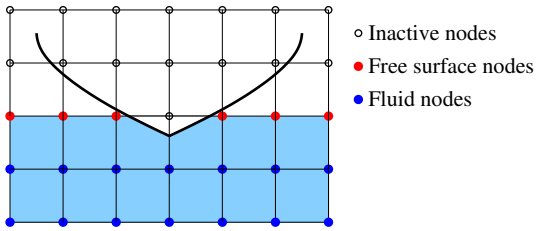


Figure 6.2: Sketch of the fluid-structure interaction model. The hull entering the body of water is immersed in the lattice, free surface (red nodes) is treated as the fluid top boundary and air flow is neglected (inactive nodes).

Comparison with analytical solutions

Here, numerical results from our approach are compared with analytical solutions [78, 79, 80, 81]. Specifically, comparison is carried out with the classical results from [78, 81] for rigid wedges impacting ideal fluids. Therein, under the hypotheses of inviscid, incompressible, and irrotational flow, the fluid is described through a velocity potential which satisfies the Laplace equation. Furthermore, no-penetration conditions are assumed at the hull-fluid interface, see Figure 6.2. Numerical results are also compared with the results of [79], based on the approach proposed in [78] to investigate the effect of compliance, and with those of [80], which account for compressibility through a tractable model of a compliant hull. Since the employed numerical model refers to a viscous fluid, modeling parameters are set in order

to guarantee a mesoscopic flow regime comparable with the inviscid-flow model. Specifically, all simulations are performed with a Reynolds number, $\mathcal{R}e = \frac{VD}{\nu}$, larger than 3 000, where V is a characteristic penetration speed for the wedge. This minimizes viscous effects consistently with the spatial resolution.

Rigid wedge at a constant penetration velocity

The impact of a rigid wedge on a free surface with a uniform constant vertical downward speed V is studied. In this case the position of the wedge is described at each time step, that is, $\boldsymbol{u}(t)$ is a priori assigned and Equation (3.22) is discarded. At the beginning of the analysis, the lowest point of the wedge is considered to touch the free surface. The total hydrodynamic load acting on the wedge and the pressure distribution along the wedge for different values of β and V are found.
According to [78], the vertical component of the total force acting on the wedge can be approximated as

$$F(t) = \rho_f \left[\frac{\pi}{4} a^2(t) \ddot{h}(t) + \frac{\pi^2 a(t) \dot{h}^2(t)}{4 \tan \beta} \right]. \tag{6.1}$$

Here, $h(t)$ is the time-dependent penetration depth of the hull, which, in this case, is $h(t) = V\ t$, and $a(t)$ is the semi-length of the wetted hull, which based on Wagner's theory is defined as

$$a_{_W}(t) = \frac{\pi}{2 \tan \beta} h(t) \tag{6.2}$$

and based on Von Karman's theory [81] as

$$a_{_{VK}}(t) = \frac{h(t)}{\sin \beta}. \tag{6.3}$$

In the present simulations the semi-length of the wetted hull corresponds to $a_{_{VK}}(t)$.
Simulations are carried out at four deadrise angles, $\beta = 5°, 10°, 15°, 25°$ and

three different wedge speeds, $V = 5\,\mathrm{m/s}, 7\,\mathrm{m/s}, 10\,\mathrm{m/s}$. The corresponding LB simulations are performed by using $\Delta t = 4 \times 10^{-7}\,\mathrm{s}$, $\Delta t = 2.86 \times 10^{-7}\,\mathrm{s}$, and $\Delta t = 2 \times 10^{-7}\,\mathrm{s}$, respectively. The Reynolds number is set to $\mathcal{R}e = 3.0 \times 10^3$, where D is set to 1 m and corresponds to the final value of $a(t)$ used in the simulation, as in [79].
Figure 6.3 shows simulation results compared to both Wagner's and Von Karman's solution [78]. Numerical findings are reported as a function of $a(t)$, as defined in Equation (6.2), in terms of the known penetration depth $h(t)$. In general, the force exerted by the fluid on the structure increases with time as the wedge penetrates the water column at the selected speed. The linear dependence on time of $F(t)$ stemming from the potential flow solution in Equation (6.1) is confirmed by our numerical results. As expected, $F(t)$ tends to linearly increase with $a(t)$ whereas it decreases as β increases. In general, the force exerted by the fluid on the structure increases with time as the wedge penetrates the water column at the selected speed. The linear dependence on time of $F(t)$ stemming from the potential flow solution is confirmed by numerical results. As expected, $F(t)$ tends to linearly increase with $a(t)$ whereas it decreases as β increases. For the same wetted surface area, numerical results from the proposed approach are bounded above by theoretical predictions from Wagner theory and below by Von Karman theory [3]. The evolution of the velocity field is depicted in Figure 6.4. The accuracy of the proposed methodology is assessed by computing the relative difference in time of the predicted value of $F(a)$ with respect to Equation (6.1), as shown in Table 6.1. Here and henceforth, for the sequence of numerical data $\{a_k\}_{k=1}^{M}$ such difference is computed as

$$\Delta_{\mathrm{rel}} = \frac{\sqrt{\sum_{k=1}^{M} |\bar{a}_k - a_k|^2}}{\sqrt{\sum_{k=1}^{M} |\bar{a}_k|^2}}, \tag{6.4}$$

where superimposed bar indicates benchmark values and M is the length of the sequence. Notably, when assessing the accuracy of $F(a)$, M refers to

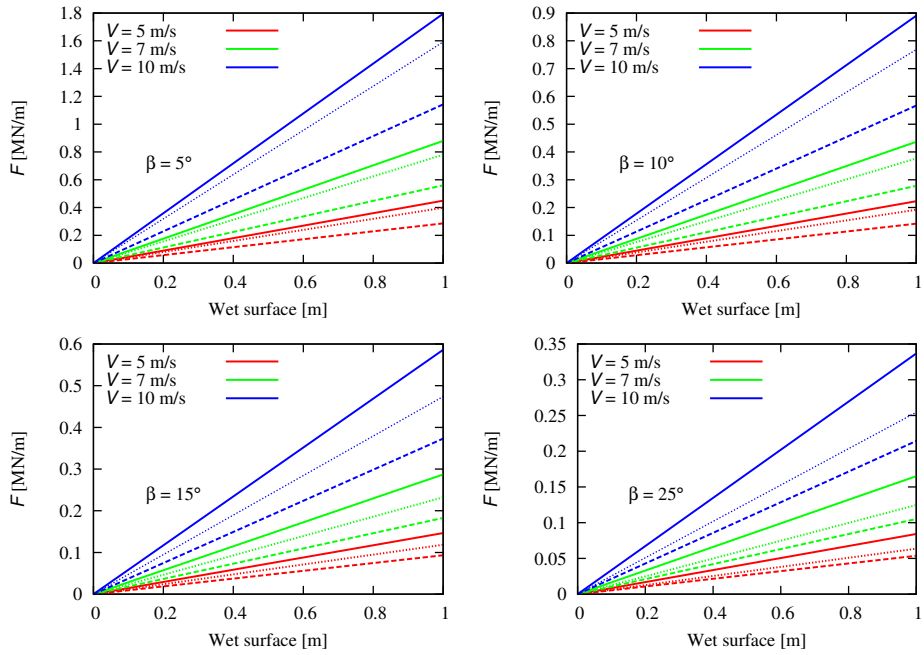


Figure 6.3: Total hydrodynamic load vs wet semi-length $a(t)$ for different values of dead-rise angle: $\beta = 5°$, $\beta = 10°$, $\beta = 15°$, $\beta = 25°$. Comparison between Wagner's solution (continuous line), see Equation (6.1), Von Karman solution (dashed line) [81], and present simulation (dotted line).

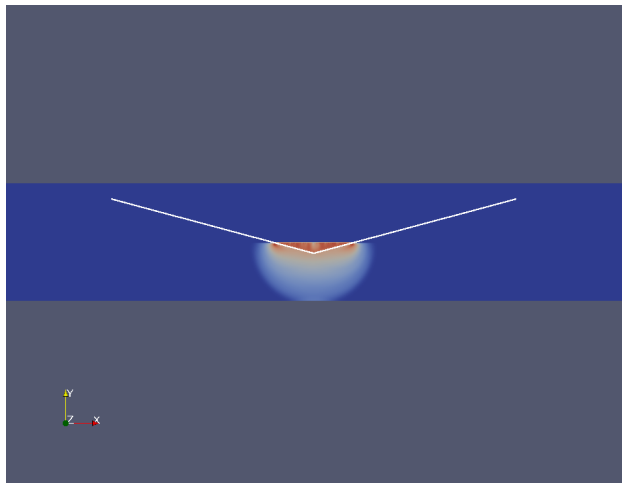


Figure 6.4: Evolution of the fluid velocity as the wedge penetrates. The movie is available in the pdf version.

the number of time steps in the whole simulation. The values of Δ_{rel} are reported in Table 6.1, where the relative difference computed by detracting the viscous shear and normal stresses from the computation of the overall force resultant are also presented. It is noted that the numerical results are close to the analytical ones when the deadrise angle is small, as expected from the assumptions of Wagner's theory. In particular, for a deadrise angle $\beta = 5°$, the relative error between Wagner's results and the present simulations at a given wetted surface is about 10%. As the deadrise angle increases, the effect of viscosity becomes important and differences between the numerical solution and the potential flow theory are substantial.
It is noted that the numerical results are close to analytical predictions from Wagner's theory when the deadrise angle is small, as expected from the assumptions the model. In particular, for a deadrise angle $\beta = 5°$, the relative error between Wagner's results and the present simulations at a given wetted surface is about 10%. As the deadrise angle increases, differences between the numerical solution and Wagner's theory are more

Table 6.1: Relative difference in the total hydrodynamic load between the present approach with/without accounting for viscous stress in the computation and literature results, [78, 81], computed for $V = 10\,\mathrm{m/s}$.

β	Δ_{rel}	Δ_{rel} without viscosity
5°	0.1138	0.1210
10°	0.1378	0.1459
15°	0.1922	0.2001
25°	0.2450	0.2598

substantial.

To dissect the role of viscosity and compressibility on the predictions of the proposed approach, variations of $\mathcal{R}e$ and $\mathcal{M}a$ are investigated. Specifically, Figure 6.5(a) displays the force per unit depth on the semi-wedge for a fixed value of the Mach number $\mathcal{M}a = 0.0173$ as the Reynolds number is varied. Results reported therein confirm that for $\mathcal{R}e \geq 3000$, viscosity is a secondary parameter in the slamming impact. In the same vein, the effect of compressibility is shown in Figure 6.5(b) where the slamming coefficient, $C_s = \frac{2F}{\rho v^2 D}$, is plotted versus the Mach number. The solution for a certain wetted surface deviates from the analytical one as $\mathcal{M}a$ increases, in agreement with the findings of Carcaterra and his coworkers, [80]. Further comparison between the proposed numerical approach and Wagner's theory [78] is garnered by studying the normal distributed load acting on the wet part of the hull. In case of potential flow theory, this term corresponds to the hydrodynamic pressure, whose expression according to [78] is given by:

$$p_s(y,t) = \rho_f \left[\ddot{h}(t)\sqrt{a^2(t) - y^2} + \frac{\pi \dot{h}^2(t)}{2\tan\beta} \frac{a(t)}{\sqrt{a^2(t) - y^2}} \right], \qquad (6.5)$$

where $p_s(y,t)$ is the pressure on the wedge as a function of y and t. Given the sharp changes of the pressure profiles in wave slamming problems, these comparisons are very important to validate the proposed methodology. Figure 6.6 shows the pressure distribution predicted by the Wagner solution and the normal stress from the proposed method for $\beta = 5°, 10°$, 15°, and 25°, $V = 5\,\mathrm{m/s}$, $7\,\mathrm{m/s}$, and $10\,\mathrm{m/s}$, and $t = 1\,\mathrm{ms}$, $2\,\mathrm{ms}$, and $3\,\mathrm{ms}$. In general,

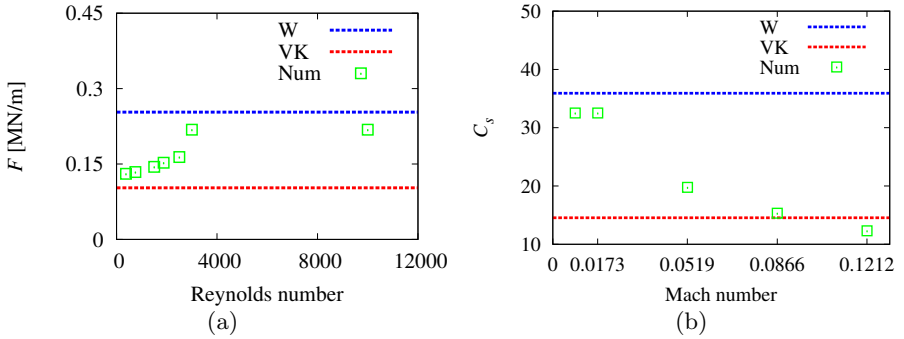


Figure 6.5: The role of viscosity and compressibility: a) total hydrodynamic load versus Reynolds number for Von Karman (VK) and Wagner (W) theories and the proposed approach (Num) for a rigid wedge impacting the fluid surface at a constant speed with $\beta = 5°$ and $V = 10\,\mathrm{m/s}$, $\mathcal{M}a = 0.0173$ and $h = 5 \times 10^{-3}\,\mathrm{m}$; b) slamming coefficient, C_s as a function of wedge Mach number for $\beta = 5°$.

the normal stress shows a similar behavior for all wetted surfaces, deadrise angles, and velocities, wherein slow variations with respect to the horizontal coordinate in the vicinity of the wedge vertex are followed by steep changes as approaching the free surface. The difference between the analytical and numerical solutions is larger at high deadrise angles, consistently with the prediction of the force resultant as discussed above. Accuracy in estimating normal stress is assessed by computing the relative difference with respect to the analytical pressure, as defined in Equation (6.4), at a prescribed wetted surface and at the vertex point of the wedge. Results summarized in Table 6.2 show that Δ_{rel} is always below 10%, with a higher deviation for $\beta = 15°$ and $\beta = 25°$.

Rigid beams connected by a rotational spring with a constant penetration velocity

In this test, the wedge impacting the free surface is modeled as two rigid beams connected by a rotational elastic spring of constant k. Each beam has mass m and e is the distance between the center of mass of the wedge

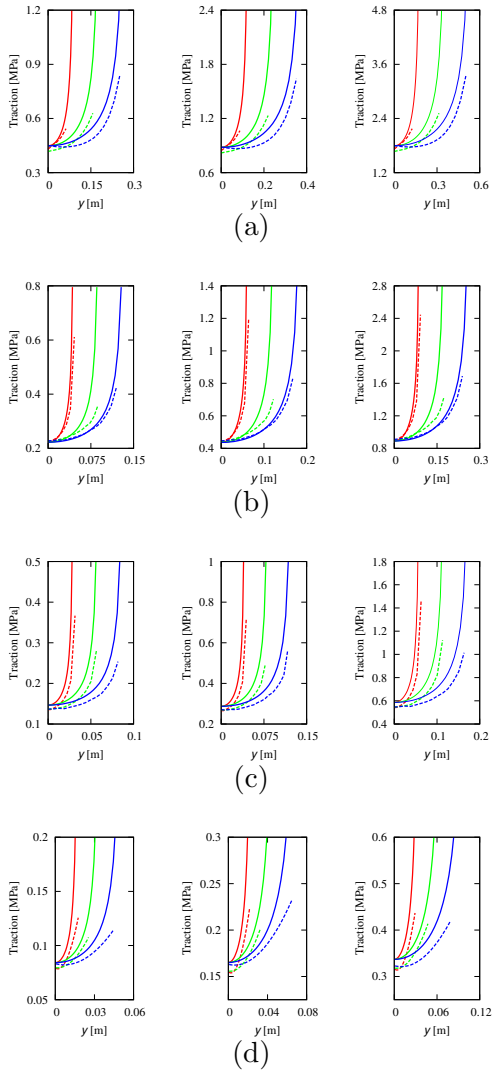


Figure 6.6: Hydrodynamic normal stress computed using the proposed approach (points) and the analytical solution (lines) for: (a) $\beta = 5°$; (b) $\beta = 10°$; (c) $\beta = 15°$; and (d) $\beta = 25°$; three different velocities are used: $V = 5\,\mathrm{m/s}$ (red), $7\,\mathrm{m/s}$ (green), and $10\,\mathrm{m/s}$ (blue). For each case, the pressure is computed for three different values of time: 1 ms, 2 ms, and 3 ms.

Table 6.2: Relative difference in the pressure at the vertex point of the wedge between the present and Wagner's results.

β	t	Δ
5°	1 ms	0.0761
	2 ms	0.0997
	3 ms	0.0303
10°	1 ms	0.0400
	2 ms	0.0200
	3 ms	0.0198
15°	1 ms	0.0920
	2 ms	0.0719
	3 ms	0.0701
25°	1 ms	0.1049
	2 ms	0.0774
	3 ms	0.0779

and the vertex. The numerical results predicted by the present approach are compared with the analytical solution from [80], obtained by neglecting surface tension, air trapping, viscosity, and gravity and by considering a potential flow with compressible effects and a constant and finite sound speed c. The vertex of the wedge moves downwards at a constant speed V. Differently from the previous test case, the wedge and fluid dynamics are coupled as the structure is modeled as a one degree of freedom elastic system in terms of the deadrise angle $\beta(t)$. The beams are assumed to be initially at rest at the deadrise angle $\beta(0)$ prior to the impact with the water surface. Relevant physical parameters are: $m = 1\,\text{kg}$, $e = 1\,\text{m}$, $c = 1480\,\text{m/s}$, and $V = 74\,\text{m/s}$. The corresponding Mach number is equal to 0.05. The simulation is performed with $\mathcal{R}e = 10^5$ by taking $D = 2\,\text{m}$ which corresponds to $2e$, that is, the initial wet surface. In addition, Δt is set to $1.89 \times 10^{-7}\,\text{s}$. Figure 6.7 displays the dimensionless elastic moment, defined as:

$$Q_e(t) = \sqrt{2}\frac{M_w}{M_0}\beta(t) \tag{6.6}$$

for different parameter values, where $M_w = \frac{e}{c}\sqrt{\frac{k}{me^2}}$ is the dimensionless oscillatory speed and $M_0 = \frac{V}{c}$ is the dimensionless entry velocity. Four

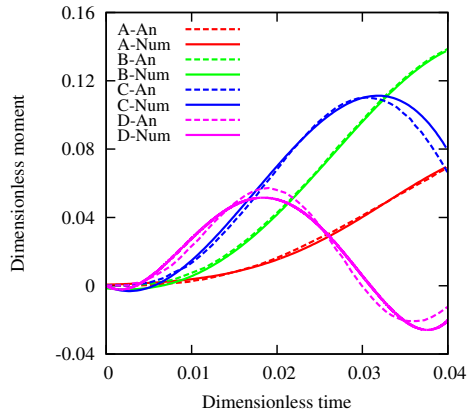


Figure 6.7: Time evolution of the dimensionless elastic moment for different values of M_w computed using the proposed approach (Num) and results from [80] (An). Here, curve A: $M_w = 1$, curve B: $M_w = 2.5$, curve C: $M_w = 5$; and curve D: $M_w = 10$.

curves are generated for the values $M_w = 1$, 2.5, 5, and 10, $M_0 = 0.05$, and an initial deadrise angle of 1°. In Figure 6.7, time is normalized with respect to the oscillation period. The proposed approach is in good agreement with the analytical solution, as it captures very well the rapid increase in the hydrodynamic moment experienced by the wedge. A closer comparison between the findings from the two approaches is obtained by computing the relative difference in the elastic moment as displayed in Table 6.3. Results show a good agreement between the analytical solution and the numerical one, as the relative maximum difference is at most 14.99%.

Table 6.3: Relative difference in the elastic moment between the proposed method and the analytical solution in [80].

M_w	Δ_{rel}
1	0.0244
2.5	0.0261
5	0.1063
10	0.1499

Deformable wedge with constant penetration velocity

The predictions of the proposed method are compared with analytical results from [79], where the effect of wedge compliance is taken into consideration. The beams are rigidly clamped with a deadrise angle $\beta = 10°$ at the vertex and their length is $L = 0.5\,\mathrm{m}$. The two extreme nodes of the wedge moves at a constant vertical downward velocity $V = 10\,\mathrm{m/s}$ and, prior to the impact, each beam rigidly translates along the z-axis at the velocity V. The beam bending stiffness per unit width is $\frac{Eh^3}{12} = 1.458 \times 10^6\,\mathrm{Nm}$ and the mass per unit surface is $\rho_s h = 270\,\mathrm{kg/m^2}$.
The time step used is $\Delta t = 2.0 \times 10^{-7}\,\mathrm{s}$ and the simulation ends when the 98% of the length of the beam is under the water surface consistently, with [79]. The LB simulation is performed with $\mathcal{R}e = 1.5 \times 10^3$ by using $D = 1\,\mathrm{m}$, which corresponds to approximately the maximum wet surface $2L$.
Figure 6.8 compares the analytical solution with the numerically computed time history of the mid-span deflection, in order to demonstrate the accuracy of the present methodology. The relative difference between the two solutions is approximatively about 14% at a given wetted surface.

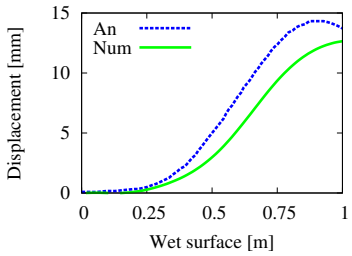


Figure 6.8: Comparison between the solution in [79] (An) and the numerically computed time histories of the mid-span deflection (Num).

Comparison with experimental results

Here, the proposed methodology is validated against experimental data on both rigid and flexible wedges, comparing to experimental results on rigid wedges from [82] and flexible wedges from [83]. All comparisons refer to water with a kinematic viscosity of $\nu = 1.11 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$ and a mass density of $1000\,\mathrm{kg/m}^3$. In the first test (rigid case), a wedge-shaped mass is dropped from a given height and impacts the free surface under the effect of gravity. Differently from the case of a rigid wedge impacting the free surface with a constant velocity, the position of the wedge is a priori unknown and is obtained by solving the equations of motion. For comparison with experimental results from [82], a rigid wedge of 72.5 kg with a deadrise angle $\beta = 25°$ falling from a height of 0.192 m is considered. The time step is set to $\Delta t = 3.44 \times 10^{-5}\,\mathrm{s}$. In Figure 6.9, the penetration depth and the drop velocity are plotted versus time. The relative difference between ex-

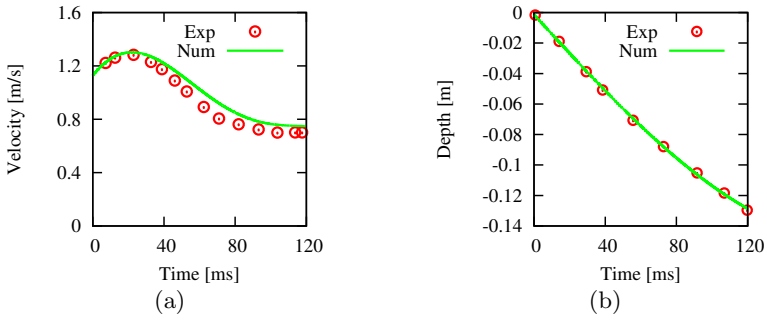


Figure 6.9: Comparison between experimental data (Exp) in [82] and the proposed approach (Num) for: (a) penetration depth and (b) and drop velocity.

perimental and numerical results is only 7.9% for the drop velocity. These results substantiate that the proposed coupling algorithm is able to accurately capture the physics of the impact.
In the second test, a steel plate impacts the free surface with a constant velocity, as if it were a wedge with null deadrise angle [83]. The plate

is released at a varying height so that different impact velocities V are obtained. The beam has length $L = 0.50\,\mathrm{m}$, width of $0.10\,\mathrm{m}$, thickness $h = 0.008\,\mathrm{m}$, mass per unit surface $\rho_s h = 62\,\mathrm{kg/m^2}$, and bending stiffness $Eh^3/12 = 8960\,\mathrm{Nm}$. In the experiments, the axial strain ε is measured at $h_a = 0.004\,\mathrm{m}$ from the neutral axis at the plate's center and the nondimensional strain and penetration velocity are defined as follows

$$\varepsilon_d = \frac{\varepsilon}{h_a V}\sqrt{\frac{Eh^3/12}{\rho_s L}}, \qquad V_d = V\sqrt{\frac{\rho_s L^3}{Eh^3/12}}. \tag{6.7}$$

Numerical simulations are carried out for speeds V ranging from $2.2\,\mathrm{m/s}$ to $6.2\,\mathrm{m/s}$. The time step used is $\Delta t = 1.88 \times 10^{-5}\,\mathrm{s}$ and 2 500 elements are used to discretize the beam which is assumed to be free at its ends. Moreover, the strain at h_a is inferred from the curvature using the standard beam theory. As shown in Figure 6.10, the maximum strain ε_d obtained using the proposed approach is close to the experimental results in the whole observation range.

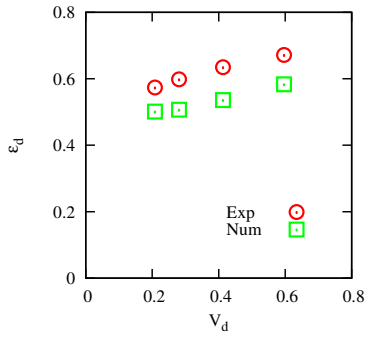


Figure 6.10: Comparison of computed (Num) and experimental (Exp) dimensionless strain as a function of the dimensionless impact velocity.

6.2 Blood flow

Here, the proposed approach in the explicit version is used to solve the problem of a non-Newtonian fluid in a channel with the strategy depicted in Figure 6.11. In particular, in order to simulate a blood flow in an arterial wall, an appropriate rheological model is used for both fluid and structure. Differently from the previous cases, here the structure deformation is analytically predicted. Given the stress tensor, the internal pressure acting on the wall is determined and, consequently, the inner radius is computed according to an analytical approach. In addition, the stress is used to correct the relaxation time, in order to account for the variation of the viscosity in the channel. According to such values, the position of the wall in the lattice background is updated and, finally, the lattice Boltzmann solution is performed. Then, the solution is advanced in time. For simplicity, a-

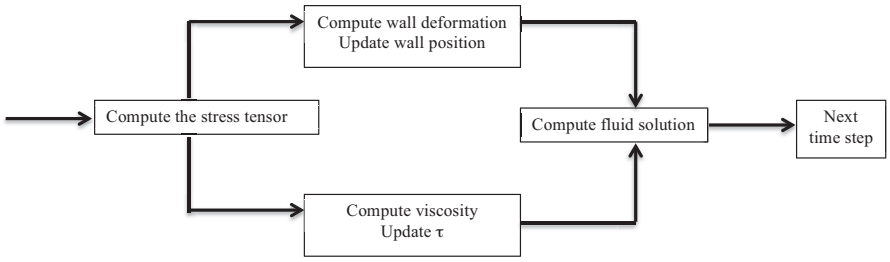


Figure 6.11: Sketch of the solution strategy.

xialsymmetric conditions for both the fluid and the wall are assumed; thus, it results in a two-dimensional problem. Moreover, the tangential stress is neglected in order to predict the deformation of the artery. In the following the non-Newtonian feature of the blood is discussed, together with the LB implementation. In addition, details about the mechanical properties of the artery are given. Finally, results involving rigid and deformable vessels are presented.

Non-Newtonian fluid model

The stress tensor in the fluid can be expressed as

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\dot{\boldsymbol{\varepsilon}}, \tag{6.8}$$

where μ is the viscosity and the strain rate tensor is given by

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^{\mathrm{T}}\right). \tag{6.9}$$

It can be computed at lattice scale as

$$\dot{\boldsymbol{\varepsilon}} = -\frac{\boldsymbol{\Pi}}{2\rho c_s^2 \tau}, \tag{6.10}$$

with $\boldsymbol{\Pi}$ given by Equation (2.8).
The most common non-Newtonian models are the power-law model and Casson's model. A power-law fluid is governed by the relation:

$$\mu = k(\dot{\gamma})^{n-1}, \tag{6.11}$$

where μ is an apparent effective viscosity depending on the shear strain rate $\dot{\gamma}$, k is a consistency index (the larger k, the larger the viscosity is) and n is the dimensionless flow behavior index. If $n = 1$, a Newtonian fluid is recovered. The shear strain rate is defined as

$$\dot{\gamma} = 2\left\|\dot{\boldsymbol{\varepsilon}}\right\| = 2\sqrt{\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}}. \tag{6.12}$$

Casson's rheological model [84] is based on the following assumptions: small shear rate, i.e. $\dot{\gamma} \leq 10s^{-1}$, and hematocrit less than 40%. The resultant model is:

$$\sqrt{\eta|\dot{\gamma}|} = \begin{cases} \sqrt{\sigma} - \sqrt{\sigma_y} & \text{if } \sigma > \sigma_y, \\ 0 & \text{otherwise,} \end{cases} \tag{6.13}$$

where η is a constant viscosity, σ_y is a limit stress known as yield stress, and σ is defined as

$$\sigma = \left\|\dot{\boldsymbol{\sigma}}^{D}\right\| = \sqrt{\boldsymbol{\sigma}^{D} : \boldsymbol{\sigma}^{D}}, \tag{6.14}$$

being $\boldsymbol{\sigma}^D$ the deviatoric stress tensor. If $\sigma < \sigma_y$, no shear strain is observed; moreover, if $\sigma \gg \sigma_y$, a typical Newtonian behavior is recovered. In fact, at high shear rate the above model reduces to $\sigma = \eta|\dot{\gamma}|$. This rheological model is used to model the non-Newtonian behavior of the blood. Casson's equation can be written as

$$\sigma = \mu|\dot{\gamma}| \qquad \text{where} \qquad \mu = \mu(|\dot{\gamma}|). \tag{6.15}$$

Thus, Equation (6.13) becomes

$$\mu = \begin{cases} \dfrac{\left(\sqrt{\sigma_y} + \sqrt{\eta|\dot{\gamma}|}\right)^2}{|\dot{\gamma}|} & \text{if } \sigma > \sigma_y, \\ \infty & \text{otherwise.} \end{cases} \tag{6.16}$$

To introduce the above behavior in the LB framework, the model proposed in [27] is used. In particular, the relaxation parameter τ is locally adjusted at each time step to properly account for the dependence of the effective viscosity upon the shear strain rate. In fact, based on Equation (6.9) it is given by

$$\dot{\gamma} = \frac{\hat{\sigma}}{\tau c_s^2 \rho}, \tag{6.17}$$

with $\hat{\sigma} = |\boldsymbol{\Pi}| = \sqrt{\boldsymbol{\Pi} : \boldsymbol{\Pi}}$. Using Equation (6.13), in the case $\sigma > \sigma_y$, Equation (6.15) can be rewritten as

$$|\dot{\gamma}| = \frac{\sigma_y}{\left(\sqrt{\mu} - \sqrt{\eta}\right)^2}. \tag{6.18}$$

Equating $|\dot{\gamma}|$ in Equations (6.17) and (6.18) leads to the following consistency condition

$$\sqrt{\frac{\mu}{\eta}} = \frac{1}{1-\zeta}\left[1 + \sqrt{\zeta[1 + \frac{\rho c_s^2}{2\eta}(1-\zeta)]}\right], \tag{6.19}$$

where $\zeta = \sigma_y/\hat{\sigma}$. Thus, at each time step the stress tensor is computed and solving the above equation the effective viscosity is determined. Then, the

value of the relaxation parameter τ is updated according to

$$\tau = \left(\frac{\mu}{\rho c_s^2} + \frac{1}{2} \right). \tag{6.20}$$

The bound values of the relaxation parameter are chosen in order to satisfy the stability condition of the LB solution. On this regard, notice that Equation (6.16) assesses that, if the shear stress is less than the yield threshold, the viscosity μ goes to infinity which is impossible to achieve numerically in the LB method. Following the approach in [27], thus, a maximum value of $\tau = 5$ is numerically forced, representing an artificially allowed threshold. On the other hand, the parameters are scaled in the lattice framework such that at the boundaries $\tau = 1$ corresponds to a viscosity equal to η.

Model of an arterial wall

An arterial wall can be depicted as a multilayer cylinder where it is possible to identify three concentric layers: the *intima*, *media*, and *adventitia* tunics. The intima is a simple layer directly interacting with the blood and its mechanical role can be neglected. The media is usually the thickest layer of the wall and has different structure in different parts of the vascular system. In large arteries closest to the heart, the media consists of elastic plates alternated by connective tissue, collagen fiber and smooth muscle fibers. One of the major roles is to accumulate blood during systole and to expel during diastole, thereby damping the pulsation generated by the heart. In smaller arteries, the media shows smooth muscle cells and a small part of connective tissue, collagen and elastin immersed in a matrix of proteoglycans. Muscle cells are wound in a variable pitch propeller and arranged in a layered structure. The adventitia has a thickness generally less than that of the media and is composed of collagen fibers hatched longitudinally and by a reduced percentage of elastin fibers.
In general, biological materials show an increasing stiffness when the stress grows up, exhibiting a non-linear elastic behavior. Moreover, the vascular wall possesses an architectural organization which is spatially oriented according to privileged directions, thus responding to evolutionary needs to

adapt the wall to applied stresses. As a result, the stress-strain relation shows different characteristics depending on the direction of the imposed stress (radial, longitudinal or circumferential). Therefore, the vascular wall has an anisotropic mechanical behavior. In addition to the non-linear and anisotropic features, such tissues have a time-dependent mechanical behavior, since the response is mediated by a transient time which leads progressively to a new state of equilibrium. In particular, such time dependency implies an energy dissipation (due to internal friction), retarding the response of a tissue to an applied stress. Such property is defined as viscoelasticity.
Following the approach in [85], each tunic is modeled as a fiber-reinforced material and fibers are oriented as the principal stress directions of the collagen. The deformation energy is assumed to depend on the isotropic stress invariant I_1 and on the anisotropic stress invariants I_4 and I_6; thus, two different energetic components can be identified:

$$\Psi(I_1, I_4, I_6) = \Psi_i(I_1) + \Psi_a(I_4, I_6), \tag{6.21}$$

being Ψ_i the strictly isotropic part and Ψ_a the anisotropic portion. In particular, the only second term of the right hand side is related to the principal directions $\mathbf{n}_1$ and $\mathbf{n}_2$, corresponding to the orientation of collagen fibers. The three invariants involved are defined as $I_1 = tr(\mathbf{C})$, $I_4 = \mathbf{C} : \mathbf{n}_1 \otimes \mathbf{n}_1$ and $I_6 = \mathbf{C} : \mathbf{n}_2 \otimes \mathbf{n}_2$, where $\mathbf{C}$ is the right Cauchy-Green tensor. Experimental findings in [86] show an high extensibility for low pressure values, while an hardening non-linear behavior arises when the pressure increases. Therefore, it is possible to use a neo-Hookean model for the isotropic part and exponential functions for the anisotropic one:

$$\Psi_i(I_1) = \frac{q_1}{2}\left(I_1 - 3\right), \tag{6.22}$$

$$\Psi_a(I_4, I_6) = \frac{q_2}{2q_3} \sum_{a=4,6} \left\{ \exp\left[q_3 \left(I_a - 1\right)^2 \right] - 1 \right\}. \tag{6.23}$$

Herein, q_1, q_2 and q_3 are three material parameters. In the following, the artery is modeled as a two-layers thick wall, as depicted in Figure 6.12.

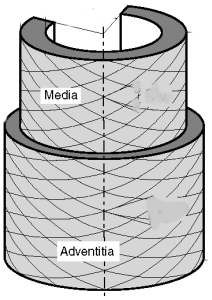


Figure 6.12: Sketch of an artery wall model modified from [85].

As previously stated, for simplicity the artery behavior is assumed to be axisymmetrical. Moreover, the deformation is assumed to be isochoric and the dependence upon the artery axis is neglected. In this way the relationship between the internal pressure and the local deformation can be computed in a closed form and used in the coupling strategy.
Three configurations are considered. First, the configuration Γ_{res} corresponds to a load-free configuration in which, if a cut in radial direction is exerted, the artery tends to open and to assume a stress-free configuration Γ_0, that is a reference undeformed and fixed configuration. The configuration Γ_0 is defined in the cylindrical polar coordinate system (R, Θ, Z):

$$R_i \leq R \leq R_o, \quad 0 \leq \Theta \leq 2\pi - \vartheta, \quad 0 \leq Z \leq L, \tag{6.24}$$

being R_i, R_o, L, and ϑ the inner and outer radii, the length and the opening angle of the vessel. The isochoric deformation χ transforms the stress-free configuration into the current one, Γ_p, described by the cylindrical coordinate system (r, θ, z):

$$r_i \leq r \leq r_o, \quad 0 \leq \theta \leq 2\pi, \quad 0 \leq z \leq l, \tag{6.25}$$

being r_i, r_o, and l the deformed inner and outer radii and the deformed length. Imposing the deformation to be isochoric leads to

$$r(R) = \sqrt{\frac{R^2 - R_i^2}{k\lambda_z} + r_i^2} \quad \theta = k\Theta, \tag{6.26}$$

where λ_z is the axial stretch that is assumed to be constant. Besides λ_z, the principal components of the deformation are:

$$\lambda_r(R) = \frac{\partial r}{\partial R} = \frac{R}{rk\lambda_z}, \tag{6.27}$$

$$\lambda_\theta(R) = \frac{r}{R}\frac{\partial \theta}{\partial \Theta} = \frac{kr}{R}, \tag{6.28}$$

with $k = 2\pi/(2\pi - \vartheta)$. Notice that the problem is strictly one-dimensional. As it can be immediately verified, the volume is preserved, i.e. $\lambda_z\lambda_r\lambda_\theta = 1$. Upon integration of the equilibrium condition

$$\frac{\mathrm{d}\sigma_r}{\mathrm{d}r} + \frac{\sigma_r - \sigma_\theta}{r} = 0, \tag{6.29}$$

together with the boundary conditions $\sigma_r = -p_i$ at $r = r_i$ and $\sigma_r = 0$ at $r = r_o$, the following equation is obtained

$$p_i = \int_{r_i}^{r_o} (\sigma_r - \sigma_\theta)\frac{\mathrm{d}r}{r}, \tag{6.30}$$

where the radial and circumferential components of the stress tensor are given by

$$\sigma_r = \frac{\partial \Psi}{\partial \lambda_r}, \tag{6.31}$$

$$\sigma_\theta = \frac{\partial \Psi}{\partial \lambda_\theta}, \tag{6.32}$$

respectively. Substituting the above relations into Equation (6.30) yields to an equation in the only unknown r_i. Indeed, the integral should be numerically perfomed, using for example the Gaussian rule. Thus, for a given λ_z, it is possible to obtain r_i in terms of p_i:

$$r_i = r_i(p_i). \tag{6.33}$$

This relation is used to predict structure deformation in the coupling strategy. Given the geometrical and mechanical parameters in Table 6.4, the

Table 6.4: Material and geometrical data from [85]. The inner radius is $R_i = 0.71\,\text{mm}$.

	Media	**Adventizia**
q_1	3.0000 kPa	0.3000 kPa
q_2	2.3632 kPa	0.5620 kPa
q_3	0.8393	0.7112
$R_i - R_o$	0.26 mm	0.13 mm
β	29°	62°

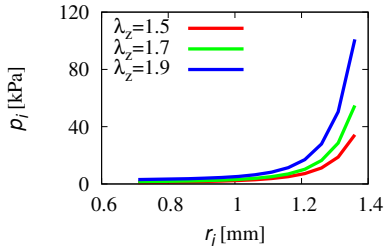


Figure 6.13: Internal pressure vs inner radius.

internal pressure is plotted against the inner radius in Figure 6.13 for different vaules of the imposed axial deformation λ_z. It is shown that the higher the pressure, the wider the cross section of the artery is; this kind of mechanical behavior is due to the presence of the collagen: the artery becomes stiffer due to fiber stretch as the pressure increases.

Some results

First, results are compared to [27], where a Casson's flow develops in a cylindrical rigid channel of width D. Figure 6.14 depicts the problem definition. At the inlet section, a velocity profile is given, while at the outlet section the outflow boundary condition is imposed. At top and bottom walls no-slip boundary condition is enforced. The inlet velocity profile is

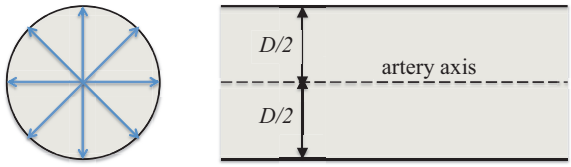


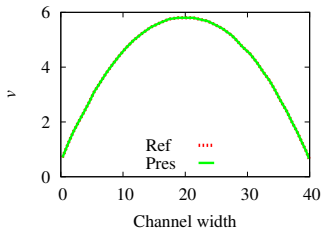
Figure 6.14: Sketch of the problem definition.

given by the expression:

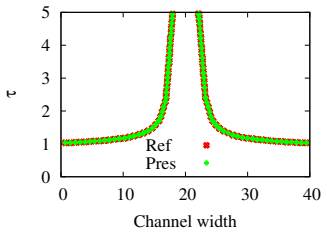
$$v(r) = -\frac{\mathrm{d}p}{4\eta \mathrm{d}x} \begin{cases} D^2 - r^2 + \frac{8}{3}\sqrt{r_c}\sqrt[3]{(D^2 - r^2)} + 2r_c(D - r), & \text{if} \quad r \geq r_c \\ D^2 - \frac{8}{3}\sqrt{r_c}\sqrt[3]{D^2} + 2r_c D - \frac{1}{3}r_c^2, & \text{if} \quad r \leq r_c, \end{cases} \tag{6.34}$$

being $\mathrm{d}p/\mathrm{d}x$ the pressure gradient. The characteristic critical radius r_c is defined so that no shear stress is observed when $r < r_c$.
Here, the fluid domain consists of 41 and 128 lattice nodes in vertical and horizontal direction, respectively; the threshold stress is $\sigma_y = 10^{-6}$, the critical radius is $r_c = 0.01 \cdot 41$. In addition, the flow is driven by a force equal to 0.0001. All these quantities are given in lattice units. In Figure 6.15(a) the velocity profile is plotted against the channel width in the central cross section, since the solution is uniform and steady-state. Then, the variation of the relaxation parameter τ in the central cross section is depicted in 6.15(b). Finally, Figure 6.15(c) presents the effective viscosity for different values of the shear stress normalized with respect to the yield threshold. As it can be observed, very close agreement between present and benchmark values in [27] is obtained. In the following, the influence of deformability is investigated. Equations (6.22) and (6.23) are different for the media and the adventitia tunics, since coefficients q_1, q_2 and q_3 depend on material properties. Thus, six parameters govern the model. Unit vectors in cylindrical components are defined as

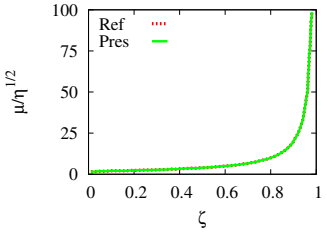
$$\mathbf{n}_1 = \begin{bmatrix} 0 \\ \cos\beta \\ \sin\beta \end{bmatrix}, \qquad \mathbf{n}_2 = \begin{bmatrix} 0 \\ \cos\beta \\ -\sin\beta \end{bmatrix}, \tag{6.35}$$



(a) Velocity in a cross section.



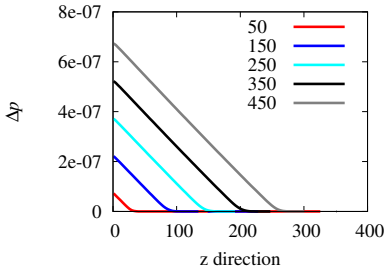
(b) Relaxation time in a cross section.



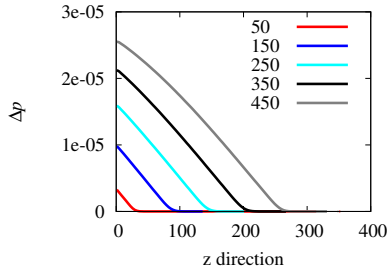
(c) Effective viscosity vs normalized shear stress.

Figure 6.15: Rigid vessel: comparison between benchmark (Ref) and present (Pres) values.

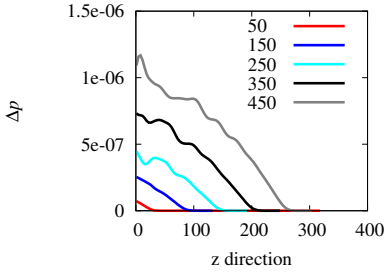
where the first component refers to the artery axis. Details about coefficients and geometry are given in Table 6.4. Moreover, the value λ_z is set to 1.5. The computational domain consists of a grid of dimensions 352×100. An incoming pulsatile overpressure is applied at the inlet section and it is characterized by a frequency of 250 pulses/minute. The corresponding Womersley number is $\alpha = R_{im}\sqrt{\frac{\omega}{\nu}} = 0.9318$, being $\omega = 26.1799$ and $\nu = 3.8 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$. For an imposed Reynolds number $\mathcal{R}e = 2R_{im}V/\nu$, the corresponding velocity is computed; then, by using the Poiseuille law, the overpressure is computed and, consequently, applied at the inlet section. For different values of the Reynolds number, the overpressure and velocity profiles along the artery axis are depicted in Figures 6.16 and 6.17, respectively. Moreover, the evolution of the velocity field is reported in Figure 6.18. As it is possible to observe, the deformability takes an important role in the behavior of a pulsatile blood flow. In particular, when a deformable vessel is considered, the velocity and pressure profiles computed at the artery axis are quite different from the rigid case if $\mathcal{R}e = 374$ is considered.



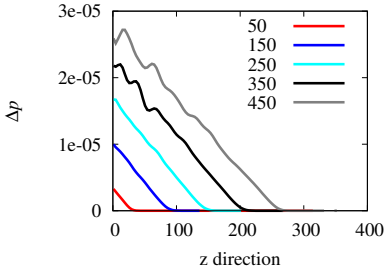
(a) Rigid vessel, $\mathcal{R}e = 374$.



(b) Rigid vessel, $\mathcal{R}e = 526$.

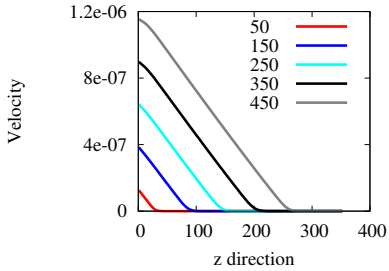


(c) Deformable vessel, $\mathcal{R}e = 374$.

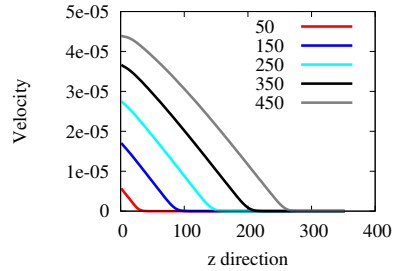


(d) Deformable vessel, $\mathcal{R}e = 526$.

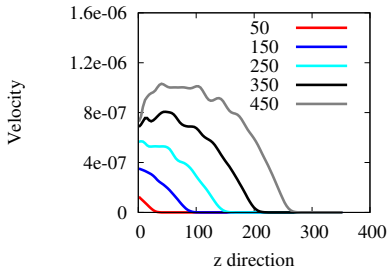
Figure 6.16: Dimensionless pressure profiles in the central horizontal section for different values of the Reynolds number at different time steps. Rigid and deformable vessels are compared.



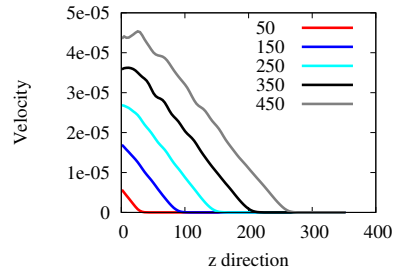
(a) Rigid vessel, $\mathcal{R}e = 374$.



(b) Rigid vessel, $\mathcal{R}e = 526$.



(c) Deformable vessel, $\mathcal{R}e = 374$.



(d) Deformable vessel, $\mathcal{R}e = 526$.

Figure 6.17: Dimensionless velocity profiles in the central horizontal section for different values of the Reynolds number at different time steps. Rigid and deformable vessels are conmpared.

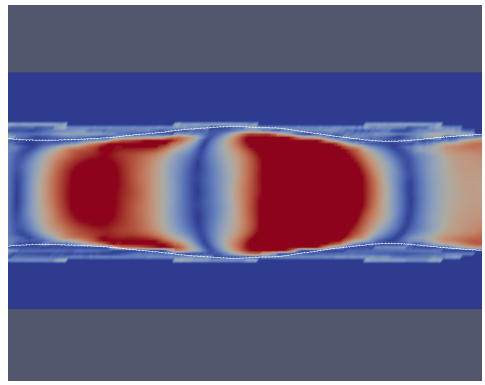


Figure 6.18: Velocity field. The movie is available in the pdf version.

6.3 Flapping wings

A two-dimensional symmetric flapping wing is immersed in a viscous, incompressible fluid characterized by density ρ and viscosity ν [12]. Wings can travel only in vertical direction. Two different scenarios are investigated: rigid and flexible wings. The wings are represented by two beams with length L connected to a hinge where the mass of the whole set is concentrated, as shown in Figure 6.19. At the time t, the angular position $\theta(t)$

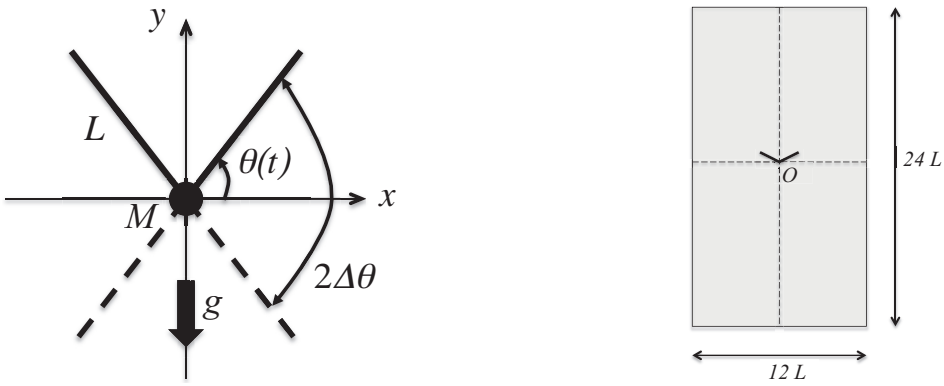


Figure 6.19: Sketch of the problem definition.

is given by

$$\theta(t) = \Delta\theta \cos(2\pi t/T), \tag{6.36}$$

where $\Delta\theta$ is the amplitude, T is the period of the harmonic oscillation, and the time-averaged tip velocity is defined as:

$$v_{tip} = 4L\Delta\theta/T. \tag{6.37}$$

According to [87], the following parameters corresponding to a butterfly are used: wing mass 3.5×10^{-6} kg, body mass 5.0×10^{-5} kg, hinge-wing distance 5.0×10^{-3} m, wing length 3.0×10^{-2} m two-dimensional air density $\rho_0 = 7.0 \times 10^{-3}$ kg/m^2. Notice that the total mass is equal to 5.7×10^{-5} kg and the total length of the wing is $L = 3.5 \times 10^{-2}$ m.

The dimensionless parameters of the problem are: the Reynolds number $\mathcal{R}e = \frac{v_{tip}L}{\nu}$, the dimensionless mass m, the dimensionless bending stiffness EJ and the Froude number, $\mathcal{F}r = 1/g$, where g is the gravity acceleration. The mass is normalized by $\rho_0 L^2$ and the bending stiffness by $\rho_0 L^3 v_{tip}^2$. Moreover, in the following the wings position is normalized by L. In order to avoid high velocities in the flow field due to the motion of the wings, the grid dimensions are chosen in such a way that the maximum tangential velocity is less than $c_s/3$. In particular, for $\mathcal{R}e = 40$ the grid consists of 1440×720 lattice nodes and the wings are modeled by using 60 elements, while at $\mathcal{R}e = 200$ the dimensions are 1920×960 and 80 beam elements are used.
First, the effect of the Reynolds number is discussed with rigid wings. As it is possible to observe in Figure 6.20, a very close agreement between the present solution and the results obtained in [59] is achieved. Notice that at $\mathcal{R}e = 200$ the wings successfully go upward, while at $\mathcal{R}e = 40$ they tend to oscillate about a fixed position. The take-off is depicted in Figure 6.21. Then, the effect of wings flexibility is investigated. Figure 6.22

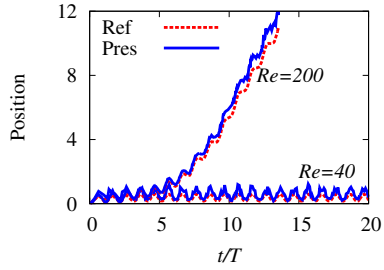
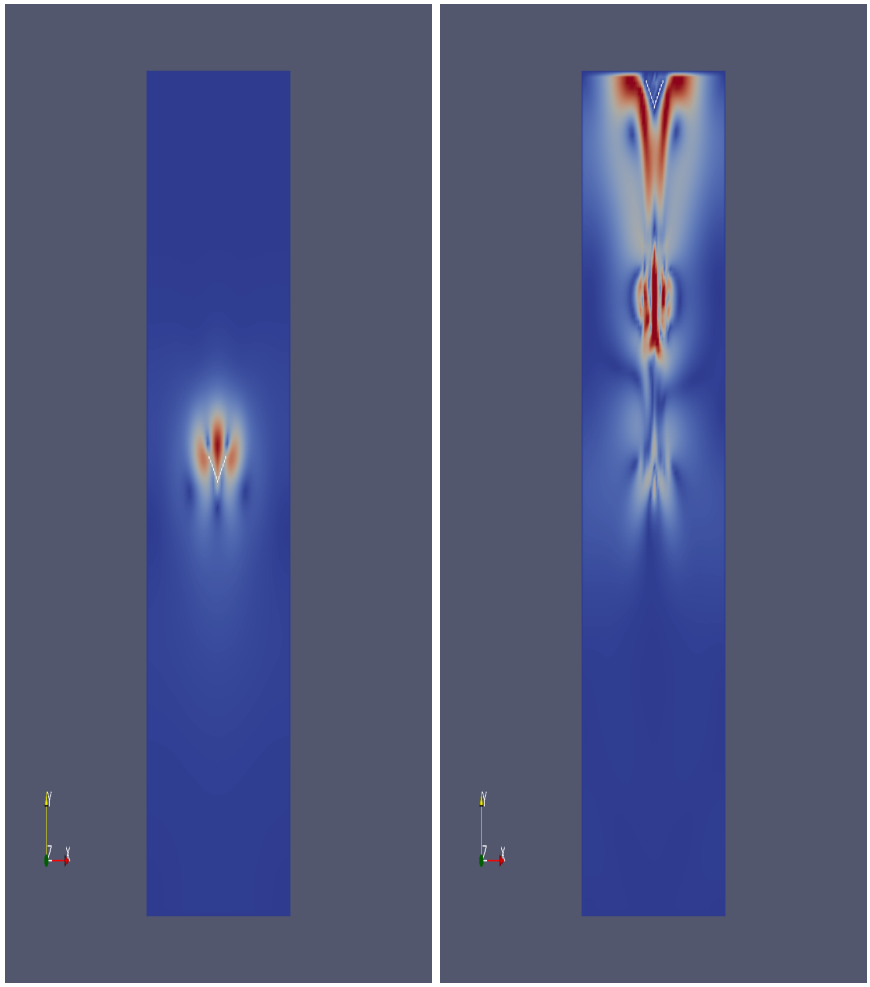


Figure 6.20: Effect of the Reynolds number.

shows the dimensionless trajectory for different values of the dimensionless bending stiffness EJ. Gravity is neglected and the dimensionless mass is $m = 9.05$. Various amplitudes are considered: $\Delta\theta = 15°$, $\Delta\theta = 30°$, $\Delta\theta = 46.8°$. As it is possible to observe, the larger the amplitude, the faster the take-off is. Moreover, the bending stiffness plays an important role, since the bird goes upward faster for large values of EJ. For a certain



(a) $\mathcal{R}e = 40$. (b) $\mathcal{R}e = 200$.

Figure 6.21: Velocity field during the evolution of the trajectory. The movie is available in the PDF version.

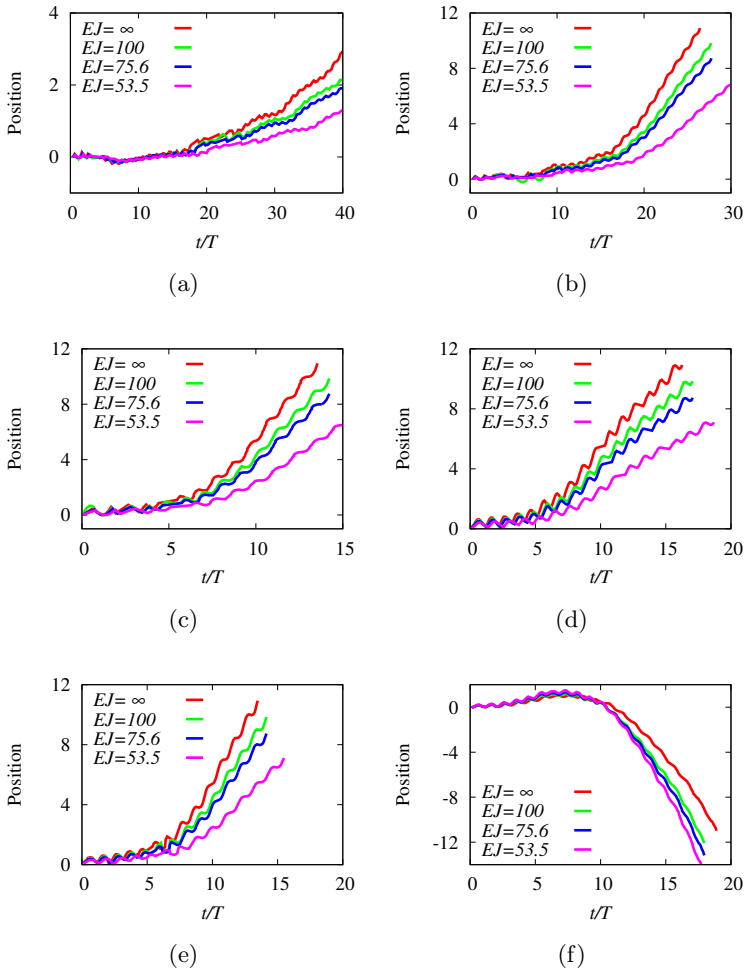


Figure 6.22: Influence of the dimensionless bending stiffness on the dimensionless position y/L of the center of wings for different amplitudes and masses: a)$\Delta\theta = 15°$, b) $\Delta\theta = 30°$, c) $\Delta\theta = 46.8°$ with $m = 9.05$, and d) $m = 4.53$, e) $m = 9.05$, f) $m = 36.2$ with $\Delta\theta = 46.8°$.

flapping amplitude, $\Delta\theta = 46.8°$, three different values of the dimensionless mass are used: $m = 4.53$, $m = 9.05$ and $m = 36.2$. Also in this case, gravity is neglected and the influence of the mass is shown. In particular, at $m = 4.53$ and $m = 9.05$ the wings go successfully upward, whereas at $m = 36.2$ the trajectory moves in downward direction. Finally, the effect of the gravity is investigated by varying the Froude number in Figure 6.23. The dimensionless mass is $m = 9.05$ and the maximum amplitude is set to $\Delta\theta = 45°$. As expected, when the gravitational force increases, the take off becomes arduous.

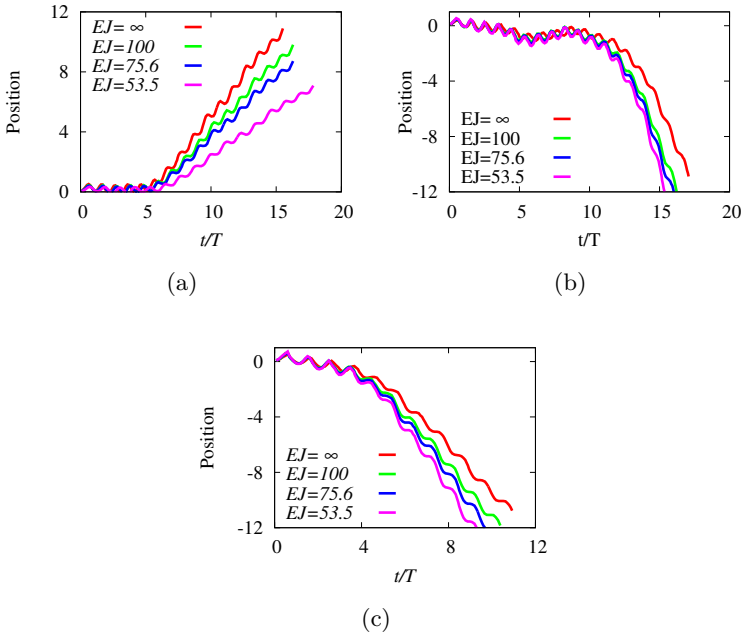


Figure 6.23: Influence of the dimensionless bending stiffness on the dimensionless position y/L of the center of wings for different Froude numbers: a) $\mathcal{F}r = 8$, b) $\mathcal{F}r = 7$, c) $\mathcal{F}r = 6$.

Chapter 7

Concluding remarks

Sommario

In questa tesi, il metodo lattice Boltzmann e il metodo agli elementi finiti sono stati accoppiati allo scopo di risolvere problemi di interazione fluido-struttura. Allo scopo di permettere un efficace, accurato, continuo e mutuo scambio di informazioni tra i due solutori, un'opportuna strategia di accoppiamento è stata implementata. Dopo una fase di validazione svolta con ostacoli rigidi, l'attenzione si è focalizzata sui solidi deformabili. In due test riguardanti il comportamento di mensole deformabili immerse in un fluido viscoso sono state discusse le diverse proprietà di accuratezza e stabilità degli algoritmi di accoppiamento ed è stata confermata la scelta effettuata sullo schema di integrazione dell'equazione del moto del solido. Una volta validato, l'approccio sviluppato è stato applicato in tre diversi contesti. Dapprima, è stato studiato l'impatto degli scafi delle navi sulla superficie marina libera, confrontando i risultati numerici con previsioni analitiche e dati sperimentali. Successivamente, l'approccio si è dimostrato efficace anche nei confronti della simulazione di un flusso non-Newtoniano, come il moto del sangue in una parete vascolare. In particolare, la parete è stata modellata tramite un'opportuna relazione costitutiva tale da tener conto del

comportamento anisotropo e irrigidente della stessa. Infine, l'attenzione è stata rivolta alla simulazione del decollo verticale di una farfalla, studiando l'influenza della deformabilità delle sue ali, della massa e della forza di gravità.
La presente strategia accoppiata è stata volutamente applicata a contesti e situazioni diverse, allo scopo di saggiarne l'affidabilità e la versatilità. I risultati ottenuti mostrano che la strategia è molto promettente come strumento per la simulazione di problemi di interazione fluido-struttura.

In the present thesis, a partitioned solution strategy for fluid-structure interaction has been developed by properly coupling the lattice Boltzmann method and the Finite Element method. Special attention has been devoted to the procedure to enforce fluid boundary conditions, on one side, and to the time integration scheme for structural dynamics, on the other side. A proper coupling strategy has been developed and three algorithms have been numerically tested. After the validation of the solution procedure for rigid fixed and moving bodies, the attention is focused on fluid-structure interaction involving deformable solids. In two tests concerning the behavior of a cantilever beam immersed in a viscous fluid, the main properties of the three algorithms are discussed. All of them have been proved to be robust and effective. The implicit algorithm is the most accurate and exhibits a nearly optimal convergence rate, that is the same convergence rate experienced in the case of the fluid with a rigid structure. However, it involves the highest computational cost. In general, three iterations are sufficient and the overall computational cost is three times that involved by the explicit algorithm. The explicit algorithm enhanced with structure predictor offers a very good convergence rate with a computational cost that is comparable to the one involved by the sole fluid solver in the case of rigid structure. Indeed, an implicit-explicit strategy could be used by coupling the proposed algorithms with an adaptive procedure. All the numerical tests confirm the superior properties of TDG scheme with respect to standard integration schemes for structural dynamics, in terms of both accuracy and stability.

The present approach has been used in three applications. First, the response of a compliant structure as it impacts on the surface of a weakly compressible viscous fluid has been investigated. The free-surface is regarded as a traction-free boundary condition, neglecting gravity and surface tension effects. The proposed method is first compared to analytical findings based on potential flow theory for wedges penetrating the fluid at a constant velocity in the following three scenarios: a rigid wedge, a wedge composed of rigid beams connected by a rotational spring, and a deformable wedge. Validation of the method is obtained by comparison with experimental data found in literature: a rigid wedge-shaped mass falling

from rest and impacting the free surface and a straight steel beam penetrating the free surface at constant velocity. A good agreement is observed in both cases, demonstrating the feasibility of the proposed approach for hull slamming analyzes.
Moreover, the LB method has been proved to be able to model the blood flow in an arterial wall. First, non-Newtonian features have been explored, since blood exhibits such kind of behavior. In particular, Casson's rheological model has been used and the anisotropic, hardening, and viscoelastic mechanical behavior of the arterial wall has been accounted for by using a proper constitutive relation. Values of internal pressure from the LB simulations have been employed to compute the deformed inner radius of the artery. The good results show that the proposed method is able to represent the non-Newtonian behavior of blood and to take in account for arterial wall deformation successfully.
Finally, the behavior of two symmetric flapping wings immersed in a viscous fluid has been investigated. First, assuming the wings to be rigid, the effect of the Reynolds number has been shown. In particular, at $\mathcal{R}e = 200$ the wings move upward from the original position. Then, the assumption of rigid wings has been removed. The effect of the bending stiffness has been shown for different conditions, i.e. various values of amplitude, mass and Froude numbers. Results show that the lower the bending stiffness, the more difficult the take off is. Such behavior is due to the fact that the energy generated by the motion is partially absorbed by the deformation of the wings, which increases if the bending stiffness decreases.
The fluid-structure interaction approach developed in this thesis has been tested against very different applications. The good results achieved lead to affirm that the proposed coupled strategy is very effective and promising for fluid-structure interaction.

Appendix A

Euler-Bernoulli beam finite element

Here, the classical linear elastic Euler-Bernoulli beam theory is discussed. The beam is discretized by using two-node elements, as shown in Figure 7.1, and each possesses three degrees of freedom: two translations, u and v, and a rotation θ. Such beam is characterized by cross-section area A and inertia moment J, length L and mass per unit length ρ. The vector of the nodal displacements for the generic beam element can be written as follows:

$$\boldsymbol{u} = [u_i \quad v_i \quad \theta_i \quad u_j \quad v_j \quad \theta_j]^{\mathrm{T}}, \tag{7.1}$$

being i and j the two nodes of the beam. In order to represent the axial displacement, linear shape functions are adopted. The matrix $\mathbf{N}_u$ of the shape functions is

$$\mathbf{N}_u = \begin{bmatrix} 1 - \dfrac{x}{L} & 0 & 0 & \dfrac{x}{L} & 0 & 0 \end{bmatrix}, \tag{7.2}$$

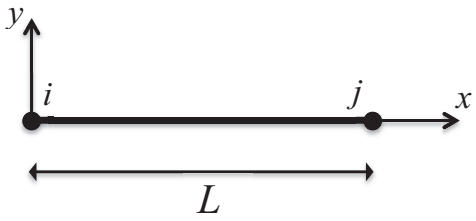


Figure 7.1: Euler-Bernoulli beam element.

being x the abscissa corresponding to the geometric axis of the element. Axial deformation is related to the nodal displacements by means of the equation

$$\varepsilon = \mathbf{B}_u \boldsymbol{u}, \tag{7.3}$$

where the compatibility operator $\mathbf{B}_u$ is defined as

$$\mathbf{B}_u = \begin{bmatrix} -\dfrac{1}{L} & 0 & 0 & \dfrac{1}{L} & 0 & 0 \end{bmatrix}. \tag{7.4}$$

The transversal displacements are represented by using cubic shape functions. In this case, the matrix $\mathbf{N}_v$ of the shape functions is

$$\mathbf{N}_v = \begin{bmatrix} \mathbf{N}_{v1} \\ \mathbf{N}_{v2} \end{bmatrix},$$

where

$$\mathbf{N}_{v1}^{\mathrm{T}} = \begin{bmatrix} 0 \\ 1 - 3\frac{x^2}{L^2} + 2\frac{x^3}{L^3} \\ x - 2\frac{x^2}{L} + \frac{x^3}{L^2} \\ 0 \\ 3\frac{x^2}{L^2} - 2\frac{x^3}{L^2} \\ -\frac{x^2}{L} + \frac{x^3}{L^2} \end{bmatrix}, \qquad \mathbf{N}_{v2}^{\mathrm{T}} = \begin{bmatrix} 0 \\ -6\frac{x}{L^2} + 6\frac{x^2}{L^3} \\ 1 - 4\frac{x}{L} + 3\frac{x^2}{L^2} \\ 0 \\ 6\frac{x}{L^2} - 6\frac{x^2}{L^3} \\ -2\frac{x}{L} + 3\frac{x^2}{L^2} \end{bmatrix}. \tag{7.5}$$

Here, the compatibility operator $\mathbf{B}_v$ is defined as

$$\mathbf{B}_v = \begin{bmatrix} 0 & -\dfrac{6}{L^2} + \dfrac{12x}{L^3} & -\dfrac{4}{L} + \dfrac{6x}{L^2} & 0 & \dfrac{6}{L^2} - \dfrac{12x}{L^3} & -\dfrac{2}{L} + \dfrac{6x}{L^2} \end{bmatrix}. \tag{7.6}$$

Thus, the material stiffness element matrix $\boldsymbol{K}_M^{(e)}$ is defined as

$$\begin{aligned} \boldsymbol{K}_M^{(e)} &= \int_L (\mathbf{B}_u^{\mathrm{T}} EA \mathbf{B}_u + \mathbf{B}_v^{\mathrm{T}} EJ \mathbf{B}_v)\, \mathrm{d}x = \\ &= \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EJ}{L^3} & \frac{6EJ}{L^2} & 0 & -\frac{12EJ}{L^3} & \frac{6EJ}{L^2} \\ 0 & \frac{6EJ}{L^2} & \frac{4EJ}{L} & 0 & -\frac{6EJ}{L^2} & \frac{2EJ}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EJ}{L^3} & -\frac{6EJ}{L^2} & 0 & \frac{12EJ}{L^3} & -\frac{6EJ}{L^2} \\ 0 & \frac{6EJ}{L^2} & \frac{4EJ}{L} & 0 & -\frac{6EJ}{L^2} & \frac{2EJ}{L} \end{bmatrix}. \end{aligned} \tag{7.7}$$

Denoting by $\mathbf{N}$ the whole set of shape functions, the elemental mass matrix $\boldsymbol{M}^{(e)}$ is defined as

$$\begin{aligned}\boldsymbol{M}^{(e)} &= \int_L \mathbf{N}^{\mathrm{T}} \rho A \mathbf{N} \, \mathrm{d}x = \\ &= \rho A L \begin{bmatrix} \frac{1}{3} & 0 & 0 & \frac{1}{6} & 0 & 0 \\ 0 & \frac{13}{35} & \frac{11}{210}L & 0 & \frac{9}{70} & \frac{13}{420}L \\ 0 & \frac{11}{210}L & \frac{1}{105}L^2 & 0 & \frac{13}{420}L & -\frac{1}{140}L^2 \\ \frac{1}{6} & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{9}{70} & \frac{13}{420}L & 0 & \frac{13}{35} & -\frac{11}{210}L \\ 0 & -\frac{13}{420}L & -\frac{1}{140}L^2 & 0 & -\frac{11}{210}L & \frac{1}{105}L^2 \end{bmatrix}. \end{aligned} \tag{7.8}$$

Appendix B

The Cholesky decomposition

The Cholesky decomposition is the factorization of a Hermitian, positive-definite matrix $\mathbf{A}$ into the product of a lower triangular matrix $\mathbf{L}$ and its conjugate transpose $\mathbf{L}^+$:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^+, \qquad \mathbf{A} \in \mathbb{K}^{m\times m}. \tag{7.9}$$

If $\mathbf{A}$ is real and symmetric, the conjugate transpose is equal to the transpose $\mathbf{L}^{\mathrm{T}}$:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{T}}, \qquad \mathbf{A} \in \mathbb{R}^{n\times n}. \tag{7.10}$$

This algorithm starts with

$$\mathbf{A}^{(1)} = \mathbf{A}, \tag{7.11}$$

$$\mathbf{A}^{(i)} = \begin{pmatrix} A_{i,i} & \mathbf{b}_i^* \\ \mathbf{b}_i & \mathbf{B}^{(i)} \end{pmatrix}, \tag{7.12}$$

$$\mathbf{L}_i = \begin{pmatrix} \frac{1}{\sqrt{A_{i,i}}} & 0 \\ -\frac{1}{A_{i,i}}\mathbf{b}_i & \mathbf{I} \end{pmatrix}, \tag{7.13}$$

$$\mathbf{A}^{(i)} = \mathbf{L}_i^{-1} \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{B}^{(i)} - \frac{1}{A_{i,i}}\mathbf{b}_i\mathbf{b}_i^* \end{pmatrix} (\mathbf{L}_i^{-1})^*. \tag{7.14}$$

Then, in the next steps:

$$\mathbf{A}^{(i+1)} = \mathbf{B}^{(i)} - \frac{1}{A_{i,i}}\mathbf{b}_i\mathbf{b}_i^*, \tag{7.15}$$

$$\mathbf{A}^{(i)} = \mathbf{L}_i^{-1} \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{A}^{(i+1)} \end{pmatrix} (\mathbf{L}_i^{-1})^*. \tag{7.16}$$

Iterations stop after n steps when $\mathbf{A}(\mathbf{n}) = 1$. The lower triangular matrix $\mathbf{L}$ is calculated as

$$\mathbf{L} = \mathbf{L}_1 \mathbf{L}_2 \dots \mathbf{L}_n. \tag{7.17}$$

The Cholesky-Crout algorithm starts from the upper left corner of the matrix $\mathbf{L}$ and proceeds to calculate the matrix column by column:

$$L_{i,i} = \sqrt{A_{i,i} - \sum_{k=1}^{i-1} L_{i,k}^2}, \qquad i = 1, ..., m \tag{7.18}$$

$$L_{j,i} = \frac{1}{L_{i,i}} \left(A_{j,i} - \sum_{\iota=1}^{i-1} L_{j,\iota} L_{i,\iota} \right), \qquad j = i+1, ..., m. \tag{7.19}$$

Bibliography

[1] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. Lattice Boltzmann simulations of laminæ interaction with viscous fluids. In *Proceedings of the Sixth M.I.T. Conference*, Boston, USA, June 2011.

[2] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. Lattice Boltzmann study of fluid-structure interaction. In *Proceedings of the 21th DSFD Conference*, Fargo, USA, August 2011.

[3] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. Lattice Boltzmann simulations of cantilevers interactions with viscous fluids. In *Proceedings of the Lattice Boltzmann Workshop*, Oxford, UK, September 2011.

[4] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. Interazione di mensole con fluidi viscosi con il metodo lattice Boltzman. In *Proceedings of the XX AIMETA Conference*, Bologna, Italy, September 2011.

[5] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. La valutazione dello stato tensionale nel metodo lattice Boltzmann: un'efficace rappresentazione dei concorsi curvilinei. In *Proceedings of the XX AIMETA Conference*, Bologna, Italy, September 2011.

[6] A. De Rosis. L'impatto dell'High Performance Computing in meccanica computazionale. In *Proceedings of the XX AIMETA Conference*, Bologna, Italy, September 2011.

[7] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. Lattice Boltzmann analysis of fluid-structure interaction with moving boundaries. *Communications in Computational Physics*, 13(3):823–834, 20123.

[8] A. De Rosis, G. Falcucci, M. Porfiri, S. Ubertini, and F. Ubertini. Hydroelastic analysis of hull slamming coupling lattice Boltzmann and finite element methods. *Submitted*, 2012.

[9] A. De Rosis, M. Porfiri, S. Ubertini, and F. Ubertini. Lattice Boltzmann of hull impacting water. *Proceedings of the DSCC/MOVIC Conference 2012*, 1, 2012.

[10] A. De Rosis and G. Falcucci. A lattice Boltzmann approach to fluid-structure interaction problems. *Proceedings of the 5th WSEAS Conference F-and-B12*, 1:73–77, 2012.

[11] A. De Rosis. The effectiveness of the high performance computing in computational mechanics. *Submitted*, 2012.

[12] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. Modelling fluid-structure interaction in flapping wings through a combined lattice Boltzmann-finite element method. *Proceedings of Stochastic Mechanics 2012*, 3:26–33, 2012.

[13] A. De Rosis, G. Falcucci, S. Ubertini, and F. Ubertini. A coupled lattice boltzmann-finite element approach for two-dimensional fluid-structure interaction. *Submitted*, 2013.

[14] R. Temam. *The Navier-Stokes Equations: Theory and Numerical Analysis*. CHEL Series. American Mathematical Society, 1977.

[15] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Clarendon, Oxford, 2001.

[16] L. Boltzmann. *Lectures on Gas Theory*. Dover, 1995.

[17] C. Cercignani. The Boltzmann equation and its applications. *ZAMM Journal of Applied Mathematics and Mechanics Zeitschrift fúr Angewandte Mathematik und Mechanik*, 67(11):455, 1988.

[18] F.J. Higuera, S. Succi, and R. Benzi. Lattice gas dynamics with enhanced collisions. *Europhysics Letters*, 9(4):345–349, 1989.

[19] R. Benzi, S. Succi, and M. Vergassola. The lattice Boltzmann equation: theory and applications. *Physics Reports*, 222(3):145–197, 1992.

[20] S. Chen and G.D. Doolen. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[21] J.G.M. Eggels. Direct and large-eddy simulation of turbulent fluid flow using the lattice-Boltzmann scheme. *International Journal of Heat and Fluid Flow*, 17(3):307–323, 1996.

[22] G Falcucci, S. Ubertini, C. Biscarini, S. Di Francesco, D. Chiappini, S. Palpacelli, A. De Maio, and S. Succi. Lattice Boltzmann methods for multiphase flow simulations across scales. *Communications in Computational Physics*, 9:269–296, 2011.

[23] G. Falcucci, G. Bella, S. Ubertini, S. Palpacelli, and A. De Maio. Lattice Boltzmann modeling of Diesel spray formation and break-up. *SAE International Journal of Fuels and Lubricants*, 3(1):582–593, 2010.

[24] X. Shan. Analysis and reduction of the spurious current in a class of multiphase lattice Boltzmann models. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, 73(4 Pt 2):047701, 2006.

[25] X. Shan and H. Chen. Simulation of non-ideal gases and liquid-gas phase transitions by lattice Boltzmann equation. *Physical Review E*, 49(4):24, 1994.

[26] A.J. Wagner and J.M. Yeomans. Breakdown of scale invariance in the coarsening of phase-separating binary fluids. *Physical Review Letters*, 80(7):1429–1432, 1998.

[27] R. Ouared and B. Chopard. Lattice Boltzmann simulations of blood flow: Non-Newtonian rheology and clotting processes. *Journal of Statistical Physics*, 121(1-2):209–221, 2005.

[28] M. Krafczyk, M. Cerrolaza, M. Schulz, and E. Rank. Analysis of 3d transient blood flow passing through an artificial aortic valve by lattice-Boltzmann methods. *Journal of Biomechanics*, 31(5):453–462, 1998.

[29] G. Falcucci, S. Succi, and S. Ubertini. Magnetically driven droplet break-up and vaporization: a lattice Boltzmann study. *Journal of statistical mechanics: theory and experiments*, 2010:05010, 2010.

[30] O. Filippova and D. Hänel. Lattice Boltzmann simulation of gas-particle flow in filters. *Computers & Fluids*, 26(7):697–712, 1997.

[31] R. Mei, L.S. Luo, and W. Shyy. An accurate curved boundary treatment in the lattice Boltzmann method. *Journal of Computational Physics*, 155(2):307–330, 1999.

[32] R. Mei, D. Yu, W. Shyy, and L.S. Luo. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review Letters E*, 65(4):041203, 2002.

[33] G. Falcucci, M. Aureli, S. Ubertini, and M. Porfiri. Transverse harmonic oscillations of laminae in viscous fluids: a lattice Boltzmann study. *Philosophical Transactions of the Royal Society - Series A*, 369(1945):2456–2466, 2011.

[34] S. Kollmannsberger, S. Geller, A. Düster, J. Tölke, C. Sorger, M. Krafczyk, and 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(7):817–845, 2009.

[35] J.S. Lee, J.H. Shin, and S.H. Lee. Fluid-structure interaction of a flapping flexible plate in quiescent fluid. *Computers & Fluids*, 57(1):124–137, 2012.

[36] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2087–2104, 2004.

[37] W.A. Wall, S. Genkinger, and E. Ramm. A strong coupling partitioned approach for fluid-structure interaction with free surfaces. *Computers & Fluids*, 36(1):169–183, 2007.

[38] E.H. van Brummelen. Partitioned iterative solution methods for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 65(1-3):3–27, 2011.

[39] C.A. Felippa and B. Haugen. A unified formulation of small-strain corotational finite elements: I. Theory. *Computer Methods in Applied Mechanics and Engineering*, 194(2124):2285–2335, 2005.

[40] G. Garcea, A. Madeo, G. Zagari, and R. Casciaro. Asymptotic post-buckling fem analysis using corotational formulation. *International Journal of Solids and Structures*, 46(2):377–397, 2009.

[41] M. Mancuso and F. Ubertini. A methodology for the generation of low-cost higher-order methods for linear dynamics. *International Journal for Numerical Methods in Engineering*, 56(13):1883–1912, 2003.

[42] M. Mancuso and F. Ubertini. An efficient Time Discontinuous Galerkin procedure for non-linear structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6391–6406, 2006.

[43] T. Inamuro. Lattice Boltzmann methods for moving boundary flows. *Fluid Dynamics Resarch*, 44(4):024001, 2012.

[44] P. Lallemand and L.S. Luo. Lattice boltzmann method for moving boundaries. *Journal of Computational Physics*, 184(2):406–421, 2003.

[45] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11(2):479–517, 2002.

[46] M. Breuer. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume. *International Journal of Heat and Fluid Flow*, 21(2):186–196, 2000.

[47] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3147–3170, 2001.

[48] C. Farhat, K. Vanderzee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):1973–2001, 2006.

[49] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in FSI applications. *Computers & Structures*, 86(23):2224–2234, 2008.

[50] P. Causin, J.F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluidstructure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4506–4527, 2005.

[51] E. H. van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *Journal of Applied Mechanics*, 76(2):021206, 2009.

[52] P.L. Bhatnagar, E.P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94:511–525, 1954.

[53] Y. Qian, D. D'Humières, and P. Lallemand. Lattice BGK models for Navier-Stokes equation. *Europhysics Letters*, 17(6):479–484, 1992.

[54] G.R. McNamara and G. Zanetti. Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20):2332–2335, 1988.

[55] A.J.C. Ladd. Numerical simulation of particulate suspensions via a discretized Boltzmann equation, part 1. Theoretical foundation. *Journal of Fluid Mechanics*, 271:285–310, 1994.

[56] A.J.C. Ladd. Numerical simulation of particular suspensions via discretized Boltzmann equation, part 2. Numerical results. *Journal of Fluid Mechanics*, 271:311–339, 1994.

[57] C.S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[58] Z. Wang, J. Fan, and K. Luo. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow*, 34(3):283–302, 2008.

[59] K. Ota, K. Suzuki, and T. Inamuro. Lift generation by a two-dimensional symmetric flapping wing: immersed boundary-lattice boltzmann simulations. *Fluid Dynamics Research*, 44(4):045504, 2012.

[60] M. Mancuso and F. Ubertini. An efficient integration procedure for linear dynamics based on a Time Discontinuous Galerkin formulation. *Computational Mechanics*, 32(3):154–168, 2003.

[61] M. Mancuso and F. Ubertini. The Nørsett time integration methodology for finite element transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 191(29-30):3297–3327, 2002.

[62] L. Govoni, M. Mancuso, and F. Ubertini. Hierarchical higher-order dissipative methods for transient analysis. *International Journal for Numerical Methods in Engineering*, 67(12):1730–1767, 2006.

[63] S. de Miranda, M. Mancuso, and F. Ubertini. Time discontinuous Galerkin methods with energy decaying correction for non-linear ela-

stodynamics. *International Journal for Numerical Methods in Engineering*, 83(3):323–346, 2010.

[64] E. Mollick. Establishing Moore's law. *IEEE Annals of History of Computing*, 28(3):62–75, 2006.

[65] F. Franek. *Memory as a Programming Concept in C and C++*. Cambridge University Press, 2003.

[66] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[67] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2006.

[68] C. Meszaros. Fast cholesky factorization for interior point methods of linear programming. *Computers & Mathematics with Applications*, 31(45):49–54, 1996.

[69] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, 2006.

[70] S. Geller, M. Krafczyk, J. Tolke, S. Turek, and J. Hron. Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows. *Computers & Fluids*, 35(8-9):888–897, 2006.

[71] Y.C. Fung. *An Introduction to the Theory of Aeroelasticity*. Dover Phoenix Editions, 1955.

[72] A. Roshko. On the development of turbulent wakes from vortex streets. *National Advisory Committee for Aeronautics, NACA*, page 2913, 1953.

[73] W. Dettmer and D. Peric. A computational framework for fluid-rigid body interaction: Finite element formulation and applications.

Computer Methods in Applied Mechanics and Engineering, 195(13-16):1633–1666, 2006.

[74] P. Anagnostopoulos and P.W. Bearman. Response characteristics of a vortex excited cylinder at low Reynolds numbers. *Journal of Fluids and Structures*, 6:39–50, 1992.

[75] E. Walhorn, B. Hübner, and D. Dinkler. Space-time finite elements for fluid-structure interaction. *Proceedings in Applied Mathematics and Mechanics*, 1(1):81–82, 2002.

[76] T. Gratsch and K.J. Bathe. Goal-oriented error estimation in the analysis of fluid flows with structural interactions. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5673–5684, 2006.

[77] E. Burman and M.A. Fernandez. Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility. *Computer Methods in Applied Mechanics and Engineering*, 198(5-8):766–784, 2009.

[78] H. Wagner. Uber stoss und gleitvorgange an der oberache von flussigkeiten. *Zeitschrift für Angewandte Mathematik und Mechanik*, 12(4):192–215, 1932.

[79] Z. Qin and R.C. Batra. Local slamming impact of sandwich composite hulls. *International Journal of Solids and Structures*, 46(10):2011–2035, 2009.

[80] A. Carcaterra and E. Ciappi. Prediction of the compressible stage slamming force on rigid and elastic systems impacting on the water surface. *Nonlinear Dynamics*, 21(2):193–220, 2000.

[81] S. Chuang. Slamming of rigid wedge-shaped bodies with various deadrise angles. *Technical Report*, Report 2268, S-R011 01 01, 1966.

[82] J.R. Shahraki, I. Penesis, G. Thomas, M.R. Davis, and J.R. Whelan. Prediction of slamming behaviour of monohull and multihull forms

using smoothed particle hydrodynamics. In *Proceedings of HSMV: 9th Symposium on High Speed Marine Vehicles*, Naples, Italy, May 2011.

[83] O.M. Faltinsen. Hydroelastic slamming. *Journal of Marine Science and Technology*, 5(2):49–65, 2000.

[84] M.M. Cross. Rheology of non-newtonian fluids: A new flow equation for pseudoplastic systems. *Journal of Colloid Science*, 20(5):417–437, 1965.

[85] G.A. Holzapfel. Computational biomechanics of soft biological tissue. *Encyclopedia of computational mechanics*, 2(34):605–635, 2004.

[86] J. Criscione. Constitutive framework optimized for myocardium and other high-strain, laminar materials with one fiber family. *Journal of the Mechanics and Physics of Solids*, 50(8):1681–1702, 2002.

[87] M. Iima and T. Yanagita. Asymmetric motion of a two-dimensional symmetric flapping model. *Fluid Dynamics Research*, 36(4-6):407–425, 2005.