Noname manuscript No.
(will be inserted by the editor)

Numerical methods for efficient fluid-structure interaction simulations of paragliders

Tom Lolies

the date of receipt and acceptance should be inserted later

Abstract After specifying the general context and needs of the paragliding industry, a new model to compute paraglider cloth dynamics is presented. Applications in research and development are then shown. Results are very promising: using simulations is drastically changing the way paragliders are being developed. Then the firsts steps towards the use of Immersed Boundaries with Lattice Boltzmann method for the highly coupled, transient high fidelity fluid-structure interaction simulation of Paragliders are presented. Despite strong bias due to under-resolved boundary layers, numerical results of the cloth deformations are in an acceptable agreement with wind tunnel measurements conducted in a previous study on a small and simple parachute geometry.

Keywords Fluid-Structure interaction · Lattice-Boltzmann Methods · Mass-Spring cloth simulation

1 Introduction

Paragliders are manned lightweight aircraft. They are non-motorized: pilots rely on thermals (hot air currents flowing up) to gain altitude, and then glide until the next thermal to go as far as possible. The pilot is connected with bridle lines to the wing (also called canopy), made of cloth, and pressurized thanks to air intakes on the leading edge (see figure 3). Thus, paragliders are not rigid and can be folded to fit into a backpack. This is also why they are prone to collapse in turbulent air, see figure 2. Main performance indicators of a paraglider are glide ratio and sink rate, and the designer who aims to optimise its performance will strive to maximise the former and minimise the latter, while keeping a safe and reliable behaviour in case of collapse.

Bruce Goldsmith Design gmbh - French R&D Offices,
ISAE-Supaero - Departement for Aerodynamics, Energy and Propulsion E-mail: tom.lolies@isae-supaero.fr

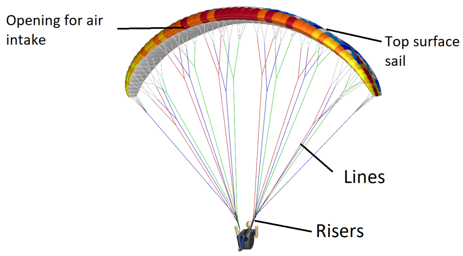

Fig. 1 Components of a Paraglider

Fig. 2 Test pilot performing a collapse for certification

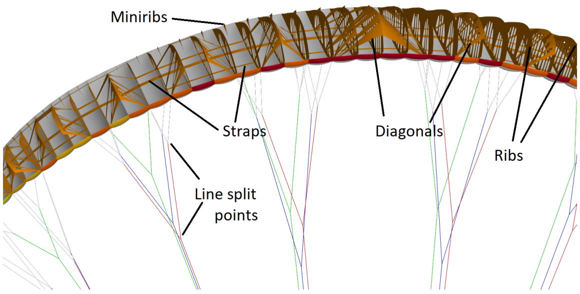

Fig. 3 Internal structure example

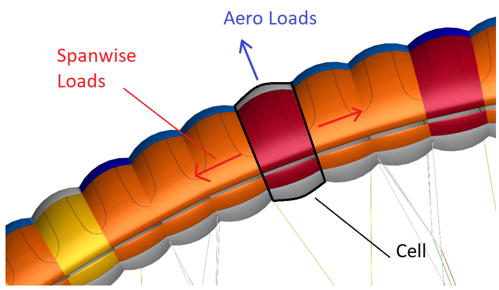

Fig. 4 A Billowed cell.

A cell is the area delimited by two ribs. Diagonals and straps (see figure 3) are used in order to decrease the number of line attachment points (also called tabs) on the canopy. There are no stiff elements in the span-wise direction. Instead, in order to maintain the wing shape in flight, paragliders rely on horizontal aerodynamic loads that result from the curvature of the wing, which is also called arc. Figure 4 illustrates how a well designed paraglider must have the correct top surface and lower surface panel shape in order not to stray away from the original design once inflated. The idea of lengthening panels to balance aerodynamic pressure forces with span-wise tensions is called panel shaping or billow shaping.

A glider's arc plays a key role in determining span-wise tensions and thus the panel shaping distribution. However, this depends on many factors. For instance, line angles affect span-wise loads and are therefore crucial to determine billow shape and handle constraints within the canopy. Up to now the chord-wise and span-wise distribution for panel shaping were chosen empirically based on the designer's experience. However small changes affect tremendously the behavior of the canopy and a precise optimization requires a long and expansive design process. The goal of this project is to develop new simulation tools to be able to tackle these issues with fewer prototypes and test flights.

2 Structural Modeling

For the structural modelling of the cloth, a simple yet efficient approach is widely studied in the literature: Mass-Spring Systems. They are known for their low computational cost and ability to represent buckling and other complex non-linear behavior. Mass-spring models are in general used to obtain a realistic rendering of cloth deformations in video games or animated movies. However, Shi et al [1] demonstrated that these models can also be applied to simulate round-parachute inflation. Physical properties of the fabric (Young modulus, Poisson ratio) were validated in their code. This article showed that mass-spring systems have a great potential for paraglider cloth simulations. X.Provot [2] presented a general framework to implement a model for cloth dynamics. The mesh is arranged as a network of structural, bending and shear springs as illustrated in figure 5. The order of magnitude of the natural period of the system is given by:

$$T_0 \approx 2\pi\sqrt{\frac{m}{K}}$$

With m being an order of magnitude for the mass and K a stiffness factor. Consequently, when the cloth is relatively rigid with very small elastic deformations (which is the case for paragliders), K is very high and the time step for numerical integration has to be very low to ensure numerical stability. This can be compensated for with artificial stiffness such as Provot's deformation constraints to simulate cloth with low stiffness parameters while avoiding unrealistic deformations in high stress regions. However, even if such technique is good to achieve realistic motion of the fabric it is not acceptable for a fine representation of the cloth physics, especially when one wants to analyze the stress distribution within the sail of a glider for instance. Thus explicit time integration methods, in spite of their low computational costs, will require low time steps to avoid instabilities which will lead to expansive simulation times. It is important to bear in mind that explicit integration methods are generally used for transient simulations, where energy must be conserved at all times to ensure accuracy. This is why they are not the most efficient when it comes to reaching a steady state as fast as possible.

A more cost-efficient alternative, implicit integration, which allows very large time steps and shows excellent stability properties, is presented by Baraff and Witkin [3]. However such method requires solving a large dense system. Given the number of points in the mesh, between $5x10^5$ and $1.5x10^6$ nodes, storing the matrix in memory can be a problem without massive supercomputers. A solution is to reformulate Euler's implicit scheme as an optimization problem, as presented by Martin et al [4] and Liu [5]. Their results are promising: they achieved stable simulations of very stiff fabric. One drawback of implicit integration would be the important damping introduced by this very dissipating scheme. However, when the objective is to reach a steady state as fast as possible this is not a problem.

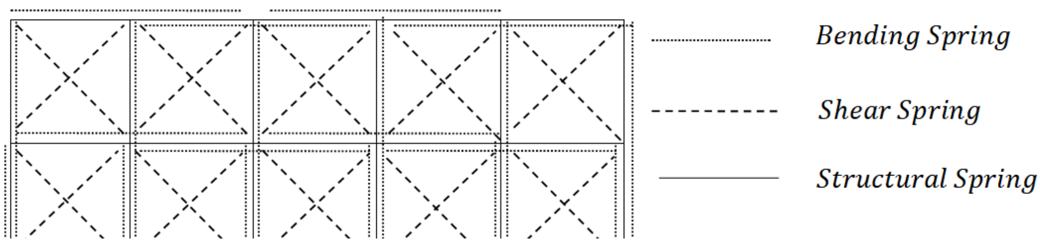

Fig. 5 Arrangement of strings for the structural mesh. There are three different types of springs for which stiffness has to be calibrated according to the fabric properties.

Let's consider a spring between two connected nodes P_i and P_j of the mesh (see figure 6). Each node is subject to an external force (for instance because of aerodynamic pressure forces). Regarding internal forces, they are given by Hooke's law:

$$\mathbf{f}_{ij} = k_{ij}(d_{ij} - deq_{ij})\frac{\mathbf{P_iP_j}}{d_{ij}}$$

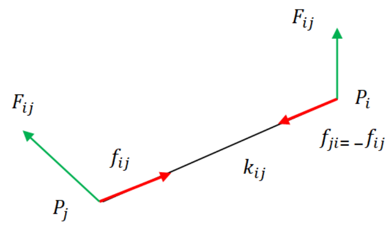

Fig. 6 Mesh nodes are connected by springs answering Hooke's law.

where d_{ij} is the length of the spring at a time t and deq_{ij} is the initial rest length of the spring. Initial constrains can be applied at the beginning of the simulation by changing these values.
Internal forces can be re-written as:

$$\mathbf{f}_{ij} = k_{ij}\mathbf{P_i P_j}(1 - \frac{deq_{ij}}{d_{ij}})$$

We can then apply the law of dynamics. For each point i in the mesh:

$$\sum_{j=1}^{Nbi} \mathbf{f}_{ij} + \mathbf{F}_i - \beta \frac{d\mathbf{Pi}}{dt} = m_i \frac{d^2\mathbf{Pi}}{dt^2}$$

Where Nbi is the number of nodes connected with i. β is a numerical damping factor. Then using the above expression:

$$m_i \frac{d^2\mathbf{Pi}}{dt^2} = \sum_{j=1}^{Nbi} k_{ij}\mathbf{P_i P_j}(1 - \frac{deq_{ij}}{d_{ij}}) + \mathbf{F}_i - \beta \frac{d\mathbf{Pi}}{dt}$$

We can reformulate this equation as follows:

$$m_i \frac{d^2\mathbf{Pi}}{dt^2} = \sum_{j=1}^{N} A_{ij}\mathbf{P_j} + \mathbf{F}_i - \beta \frac{d\mathbf{Pi}}{dt} \tag{1}$$

where if $i \neq j$ and i and j are connected nodes:

$$A_{ij} = k_{ij}(1 - \frac{deq_{ij}}{d_{ij}})$$

and

$$A_{ii} = -\sum_{l=1}^{Nbi} k_{il}(1 - \frac{deq_{il}}{d_{il}})$$

and $A_{ij} = 0$ if i and j are not connected by a spring. The matrix A is of size $N \times N$ with N being the number of nodes in the mesh. Despite its large size, the matrix is very sparse, because a given node is only connected to a small number of neighbours.

To update and realize operations on the matrix A the C++ library *Eigen* was widely used. It features as well OpenMP parallel compatibility to significantly reduce simulation costs. equation 1 can be divided in three uncoupled equations:

$$M * a_X = A * X + \beta * V_X + F_X \tag{2}$$

With $X = x, y$ or z. Only the numerical integration of x coordinate equation will be detailed. For the two others, it is strictly analogous. M is the (diagonal) mass matrix, x, v_x, a_x contains the x-coordinates for the position, speed and acceleration of all points respectively. Baraff and Witkin [3] presented the implicit Euler integration as follows:

$$x_{n+1} - 2x_n + x_{n-1} = dt^2 M^{-1}(Ax_{n+1} + \beta\frac{x_{n+1} - x_n}{dt} + F_x) \tag{3}$$

Equation 3 can then be reformulated according to Liu's [5] notations, denoting the unknown state $x := x_{n+1}$, the previous known states $y := 2x_n - x_{n-1}$, and the right hand side $f(x) := Ax_{n+1} + \beta\frac{x_{n+1}-x_n}{dt} + F_x$.

$$M(x - y) = dt^2 f(x) \tag{4}$$

Solving 4 implies minimizing the function g below:

$$g(x) = \frac{1}{2}(x - y)^T M(x - y) + dt^2 E(x) \tag{5}$$

Where $\nabla g = 0$ is exactly 4. E has the dimension of an energy $\nabla E(x) = f(x)$. One of the great advantage of this form is that there is no need to calculate the Hessian matrix which is dense and represents very high memory cost. This formulation is called Variational or Optimization Euler implicit.

Finally the optimization problem can be summarized as follows:

$$min_x g(x) = (x - y)^T M(x - y) - dt^2(x^T Ax + x^T F_x - \beta x^T v_x) \tag{6}$$

Knowing:

$$v_x = \frac{x - x_n}{dt} \tag{7}$$

and:

$$\nabla g(x) = M(x - y) - dt^2(Ax + F_x + \beta v_x) \tag{8}$$

The crux of the matter is now to choose an adequate technique to solve the optimization problem given by equation 6. Martin et al or Liu suggest Newton classical gradient-based descent technique. However, there are several drawbacks: first, the Hessian matrix has to be computed, which can cause memory issues, and in some cases it led to stability problems.

Therefore, Hessian-based methods should be avoided in our case. After some bibliographical research, the most appropriate method seemed the Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). Indeed, it only uses an approximation of the inverse of the Hessian matrix, with efficient linear memory usage. The optimization-L-BFGS numerical integration shows excellent stability properties and converges fast.

3 Application to Paragliders

3.1 Geometry management

Specific surface meshing program has been developed to generate surface discretization based on CAD shape. In this current model local spring stiffness are defined as global constants which can be correct according to V.Baudet et al. [6], H.Delingette et al. [7] if the size of mesh quadrangles remains constant in the entire structure. Thus, special effort has been carried out to minimize mesh distortions. Figure 7 shows how a wing section is meshed. The mesh is coarsened for visualization purposes. A close up view of the opening is given in figure 8. In the simulation cases presented in section III. the internal flow is not resolved, replaced by an arbitrary value for the internal pressure, and the opening is not visible on the mesh.

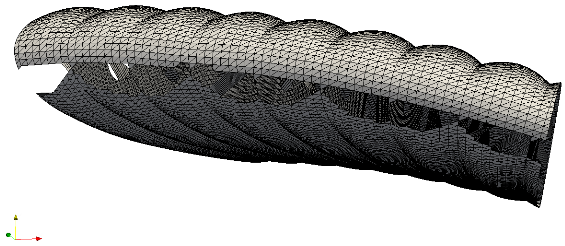

Fig. 7 Mesh for a wing section.

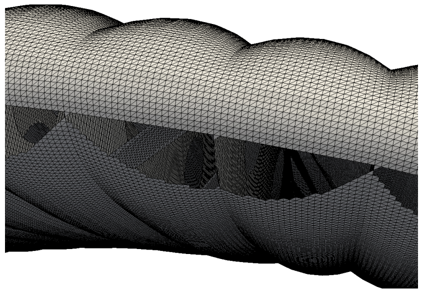

Fig. 8 Close up view of the opening

3.2 First applications

Fig. 9 Leading edge view.

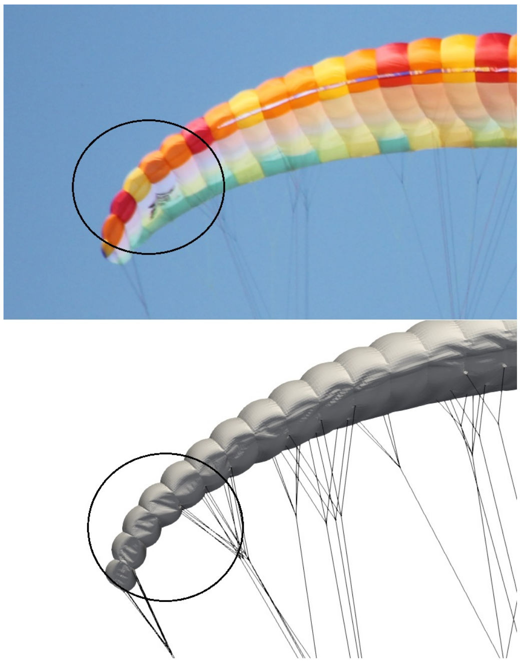

Fig. 10 Sail support issue (inside the black circle).

The structural model has been loosely coupled with steady low fidelity Computational Fluid Dynamics method. In these simulations only the risers are imposed Dirichlet conditions. the rest of the structure is completely free. Results are post-processed once the glider converges towards its equilibrium position. A first application is presented in figure 9, the cloth surface is colored with mechanical stresses distribution.

During the development of this entry level school glider the use of simulation lead to an excellent control of sail deformations. However only visual partial validation of simulation results has been performed at the moment. Another example is shown in figure 10 were a sail support issue already known in an existing rejected prototype has been observed in simulation as well. Then figure 11 shows how fluid-structure interaction simulations can be used to identify undesired stress concentrations. In this case, there are on the lower surface high stress regions. Such design problem can be anticipated with simulation to avoid very expansive late stage design issue like a load test failure.

Using simulations changes significantly the way paragliders are developed. Indeed, previous empirical processes consisted in finding a balance between profile, arc, twist, sweep and panel shaping with several prototypes. The approach now is to choose the geometrical features of the wing to achieve a desired level of glide performance and handling qualities. Then using the simulation program designers can calculate the panel shaping distribution to ensure that the design shape is respected when the glider is flown. So the same optimization can be performed with fewer prototypes.

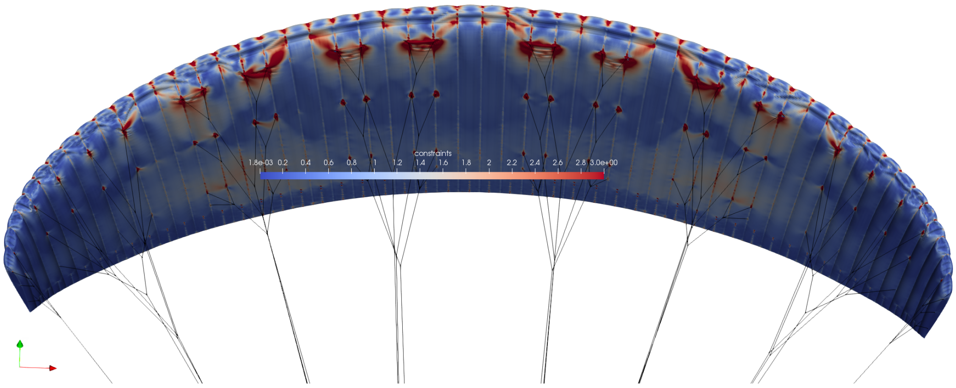

Fig. 11 Visualisation of undesired stress concentrations on the lower surface.

4 Lattice Boltzmann approach

The development of Fluid-Structure interaction methods specifically dedicated to paragliders has not been detailed in the literature. However, Ram-air and round parachutes are more thoroughly studied. N. Fogell et al. [8] present a loosely coupled approach to simulate a single parachute cell with a Reynolds-Average Navier Stokes fluid simulation coupled to a finite-element model. However extension to full 3D wing is not presented because of the high computational cost it would represent. B.Perin et al. [9] showed that with ALE (Arbitrary Lagrangian Eulerian) method coupled to Finite Element structural model interesting results regarding cloth deformations can be obtained despite a very coarse aerodynamic modeling.

These studies show that a crucial point is to manage fluid re-meshing when the cloth is moving (an other example is given in [10]. Thus a new approach is considered to achieve high-fidelity strongly coupled fluid-structure interaction simulation without complex dynamic fluid mesh adaptation around the cloth's structure: the immersed boundary method. This boundary treatment strategy is widely used and studied in the literature with a Lattice Boltzmann approach (IB-LBM), for instance in [11]. LBM are particularly interesting for high performance computing and efficient management of Cartesian grids.

Lattice Boltzmann theory is presented by Chen and Doolen [12]. Lattice Boltzmann models do not directly simulate the evolution of the flow velocity. Instead, they calculate the particle distribution function $f_i(\mathbf{x}, t)$ with the velocity c_i at every point $\mathbf{x}$ and time t thanks to a microscopic description of the fluid. Navier-Stokes equations for incompressible fluid flows can be recovered from the LB numerical scheme [13]. The fluid is modelled by identical particles whose velocities are restricted to a finite set of vectors. Here, the 3D Lattice with 19 vectors D3Q19 is considered. Macroscopic variables on each node are defined as moments of the particle populations:

$$\rho = \sum_{i=0}^{q-1} f_i$$

$$\rho \mathbf{u} = \sum_{i=0}^{q-1} c_i f_i$$

A Lattice Boltzmann iteration takes the system to a time t towards a time $t+1$ in two steps. First, a collision operator Ω is evaluated on each node, to take the distribution function to its post-collision state:

$$f'_i(\mathbf{x}, t) = \Omega_i(f_0(\mathbf{x}, t), ..., f_{q-1}(\mathbf{x}, t))$$

This is followed by a streaming step, which takes the post-collision variables to a neighbor node determined by the corresponding lattice vector:

$$f_i(\mathbf{x} + c_i, t+1) = f'_i(\mathbf{x}, t)$$

Immersed Boundary (IB) conditions in Palabos, the open source LBM library used for this project, are implemented as proposed by T. Inamuro [14]. IB are particularly adapted to cloth simulation as there is no need to explicitly formulate bounce back conditions and destroy or create lattice nodes to adapt the fluid mesh to the fabric geometry. Lets consider a parachute surface composed of N points.
We will now use Lattice units where: $\Delta t = 1$ and $\Delta x = 1$.
Let $\mathbf{X}_k(t)$ and $\mathbf{U}_k(t)$, $k = 0..N$ be the points of the moving boundary and their velocities respectively.

Then the temporal fluid velocities $\mathbf{u}(\mathbf{X}_k, t)$ are interpolated on the Lagrangian surface points as follows:

$$\mathbf{u}(\mathbf{X}_k, t) = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x}, t) W(\mathbf{x} - \mathbf{X}_k)$$

Where $\sum_x$ indicates the sum of all the Eulerian grid points. W is a weight function for the fluid-solid velocity projection, with $W(a, b, c) = w(a)w(b)w(c)$. Here, a, b and c are the three components of the vector $\mathbf{x} - \mathbf{X}_k$, and:

$$w(r) = \begin{cases} \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}), \text{ if } |r| \leq 1, \\ \frac{1}{8}(5 - 2|r| + \sqrt{-7 + 12|r| - 4r^2}), \text{ if } 1 \leq |r| \leq 2, \\ 0 \text{ otherwise} \end{cases}$$

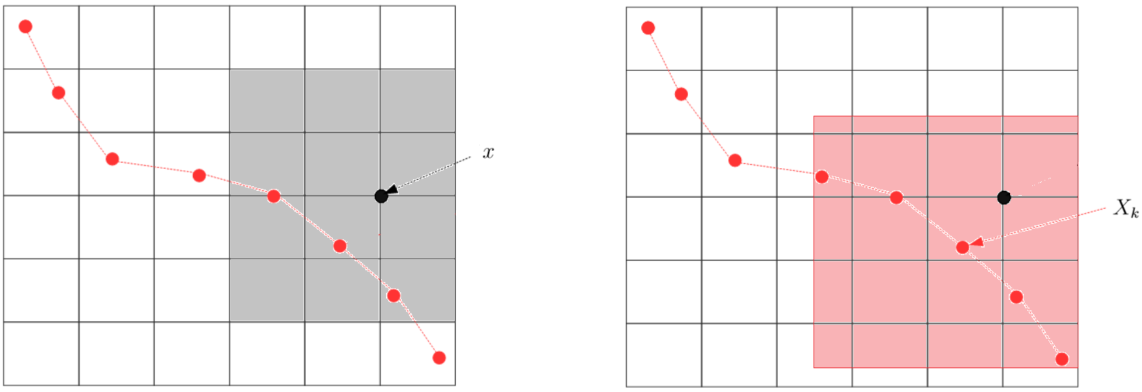

Fig. 12 Area where a given lattice node x influences the structural mesh through velocity projection (Left). Area where a structural node influences lattice velocities (Right)

Thus the coupling between a fluid and a structural node is effective only if the distance between the two is smaller than $2\Delta x = 2$ in lattice units. This projection is also performed to force structural velocities onto the fluid mesh in a similar way. This is illustrated in figure 12

To test and validate the numerical method a small and simple ram-parachute model has been simulated. This geometry had been previously tested in wind tunnel facilities. Figure 13 shows the visual comparison between simulation results and wind tunnel pictures. During the test, using a 3D digital image correlation technique described in [9], a measurement of the lower surface shape was conducted. Simulation results are confronted to these measures in figure 14 and also to LS-Dyna ALE computations presented in the same article.

There is a significant offset between experimental results and both simulations on the right side of the canopy. According to [9], this is because the real Parachute is not symmetric, contrary to the geometry processed during the simulations. Currently, the simulation has been conducted with uniform Cartesian grids and without any wall model. However, given the Reynolds numbers at hand (from $5x10^5$ to $2.5x10^6$), boundary layers on paragliders are in most cases fully turbulent. This is why without any turbulence model and adequate wall grid refinement this method is not applicable to real paraglider flight configurations. However, in the case of this simple parachute structure, the flow in the experiment is almost fully separated which is easy to simulate even with under-resolved boundary layer treatment. Thus results presented above are in a quite good agreement with wind tunnel measurement data.

5 Conclusion

A cost-efficient cloth model based on a mass-spring system with implicit time integration has been presented. To avoid extensive memory costs, at each time step the integration is reformulated as an optimization problem which is solved with L-BFGS algorithm. Then few industrial applications on paragliders are presented, showing that there is a promising future for numerical approaches in the paragliding industry. However these results are not finely validated with experimental measurements yet. Then, first simulations using an Immersed Boundary Lattice-Boltzmann method are presented. Cloth deformations are in good agreement with wind tunnel measures for a simple parachute geometry. High fidelity computations of flow physics on paragliders will however require extensive development to propose suitable boundary layer treatment. This

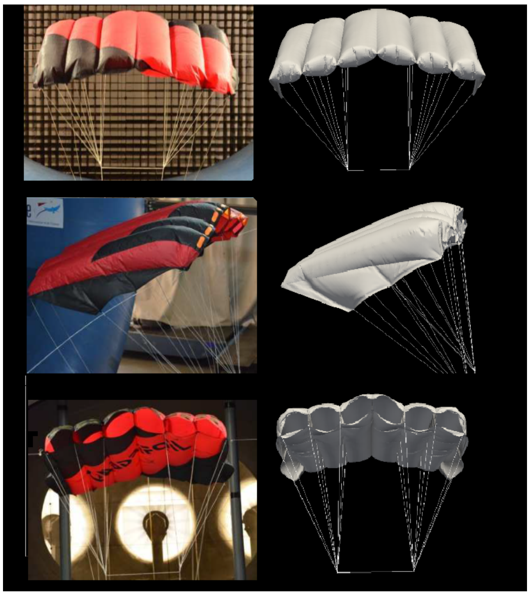

Fig. 13 Visual comparison with pictures from wind tunnel tests.

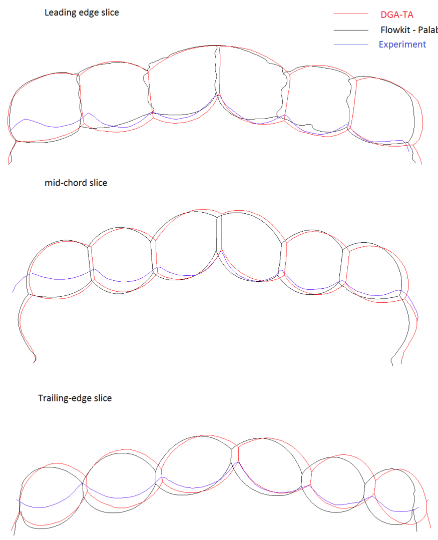

Fig. 14 Comparison between DGA-TA, Palabos and experimental shapes of the Parachute.

is why current research is focusing on grid refinement techniques and turbulent wall models for IB-LBM.

Acknowledgments

The author would like to thank Bruce Goldsmith Design for funding and supporting this project. The gratitude is extended to Nicolas Gourdain, Herv Belloc, Miguel Charlotte and Jonas Latt for supervising the research and providing extensive help, as well to Olivier Caldara for the interesting talks and useful suggestions.

References

1. Q. Shi, D. Reasor, Z. Gao, X. Li, and R. Charles. On theverification and validation of a spring fabric for modeling parachute inflation. *Journal of FluidsandStructures 58 (2015)2039*, 2015.
2. X. Provot. Deformation Constraints in Mass-Spring Model to describe Rigid Cloth Behavior. *INRIA*, 1995.
3. D. Baraff and A. Witkin. Large Step in Cloth Simulation. *SIGGRAPH 98, Orlando, July 19-24, Computer Graphics Proceedings, Annual Conference Series*, 1998.
4. S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross. Example-Based Elastic Materials. *ETH Zurich,Disney Research Zurich*, 2011.
5. T. Liu. Fast Simulation of Mass-Spring Systems. *University of Pennsylvania*, 2013.
6. V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. Integrating tensile parameters in mass-spring system for deformable object simulation. *Rapport de Recherche LIRIS*, 2009.
7. S. Cotin, H. Delingette, and N. Ayache. Efficient linear elastic models of soft tissues for real-time surgery simulation. *RR-3510, INRIA.*, 1998.
8. N. Fogell, S.J. Sherwin, C.J. Cotter, L. Iannucci, R. Palacios, and D.J. Pope. Fluid-structure interaction simulation of the inflated shape of ram-air parachutes. *AIAA Aerodynamic Decelerator Systems (ADS) Conference 25-28 March 2013, Daytona Beach, Florida*, 2013.

9. B. Perin, A. Donnard, P. Bordenave, C. Larrieu, C. Simond, and H. Belloc. Fluid-structure interaction simulation of ram-air parachutes - an application for a kite -. *23rd AIAA Aerodynamics Decelerator Systems Technology Conference*, 2015.
10. H. Altmann. Fluid-structure interaction of ram-air parafoil wings. *23rd AIAA Aerodynamic Decelerator Systems technology conference*, 2015.
11. J. Favier, A. Revell, and A. Pinelli. A lattice boltzmann - immersed boundary method to simulate the fluid interaction with moving and slender flexible objects. *Journal of Computational Physics, Elsevier, 261, pp.145-161.*, 2009.
12. S. Chen and G. D. Doolen. Lattice Boltzmann methods for fluid flows. *Annu. Rev. Fluid Mech. 30:32964*, 1998.
13. J. Latt and B. Chopard. Straight velocity boundaries in the lattice Boltzmann method. *University of Geneva*, 2008.
14. K. Ota, K. Suzuki, and T. Inamuro. Lift generation by a two-dimensional symmetric flapping wing: immersed boundary-lattice Boltzmann simulations. *Fluid Dynamics Research*, 2011.