Interactive simulation for easy decision-making in fluid dynamics

Mengchen Wang[1,2], Nicolas Férey[1], Frédéric Magoulès[2] and Patrick Bourdot[1]

[1]VENISE Team, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, Université Paris-Saclay, Orsay, France
[2]Mathematics in Interaction with Computer Science, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France

Abstract

A conventional study of fluid simulation involves different stages including conception, simulation, visualization, and analysis tasks. It is, therefore, necessary to switch between different software and interactive contexts which implies costly data manipulation and increases the time needed for decision making. Our interactive simulation approach was designed to shorten this loop, allowing users to visualize and steer a simulation in progress without waiting for the end of the simulation. The methodology allows the users to control, start, pause, or stop a simulation in progress, to change global physical parameters, to interact with its 3D environment by editing boundary conditions such as walls or obstacles. This approach is made possible by using a methodology such as the Lattice Boltzmann Method (LBM) to achieve interactive time while remaining physically relevant. In this work, we present our platform dedicated to interactive fluid simulation based on LBM. The contribution of our interactive simulation approach to decision making will be evaluated in a study based on a simple but realistic use case.

CCS Concepts

• Computing methodologies → Interactive simulation; Simulation environments;

1. Introduction

Interactive simulations have been proved to be useful to solve many scientific questions. For instance, *Fold'It* [KDC*11] helps users to solve the structure of an HIV protein by using an interactive molecular simulation approach. We believe that interactive simulations applied to fluid mechanics can be an interesting tool to support decision making, problem solving and optimization in this field, especially when humans are needed in the loop.

Using Computational Fluid Dynamics (CFD) requires a large set of expertise and tools to design and perform a simulation, then to visualize, explore and analyze the simulation results. Different applications are used sequentially, including 3D modeling software, simulation and visualization software. This process is time-consuming because it involves a lot of technical and unproductive tasks such as data processing and wrapping that require different tools. Moreover, to get accurate relevant physical results, realistic 3D fluid simulations need expansive computational resources. Therefore, it is crucial to detect early errors on the simulation model or the parameters used before the end of the computation. Consequently, it is necessary to regularly extract and analyze simulation snapshots. By taking advantage of advances in calculation, algorithms, and simulation methodology, an interactive simulation approach was proposed to address these issues, especially to reduce the time lost during this complex process.

There are many works dedicated to interactive data visualization or real-time data visualization in the CFD domain. *ViSTA FlowLib* [SGvR*03] uses haptic rendering techniques to give a better understanding of the unsteady fluid flows data. Another work [VMN*09] provides and evaluates multimodal feedback such as sonification during interaction with fluid simulation, especially to address visual overload. Stam J. has pioneered the development of the interactive simulation using an unconditionally stable model [Sta99]. Until today, most applications based on interactive simulation approach use particle-based methods such as Translating Eulerian Grids [CTG10] or Smoothed Particle Hydrodynamics (SPH) [MCG03]. These methods target extreme performance, high level rendering, and stability, but at the expense of the physical relevance, which is required by decision-making, physics education or research. Using more relevant methods requires computing centers that are often isolated from the visualization resources which are usually undersized. Even if a lot of technical results were published to support the interactive fluid simulation approach, this methodology remains rarely used. Moreover, there are only a few studies that deal with the problem of the usefulness of interactive simulations in terms of performance and user experience. We propose in this paper a work-in-progress to address this issue by designing an interactive fluid simulation platform based on *Unity 3D* and evaluating the benefit of the interactive simulation approach on decision making from fluid simulation on a simple but realistic use case.

2. A platform coupling Unity 3D with classical simulation tools

To provide interactive fluid simulation features, the visualization tool needs to be graphically synchronized with the initial model

used as an input to the simulation software. Usually, the user input includes physical properties of the fluid (e.g., speed, pressure, temperature, fluid type), and boundary conditions (e.g., wall, floor, obstacles). During the simulation, the visualization application needs to get the state of the model sent at each time-step from the simulation to update the rendering. To take into account interactive modification on the fluid or its boundary conditions, the interactive visualization tool sends to the simulation the modified fluid states or boundary conditions. According to the rendering and simulation frame rates, it is possible to tune the frequency of data exchange. To realize the communication between two systems, we chose to use a network protocol.

2.1. A dedicated network protocol and API for interactive fluid simulation

To be able to integrate any simulation tools daily used by experts, and to ensure the quality of the simulation results, we have developed an API called *CFDriver* [WFBM19] embedding a network protocol to perform the data communication between the simulation tool and visualization one. As a cross-platform API, it works with different programming languages and on different operating systems. It allows an expert to connect our interactive simulation platform to any simulation tools with only a few lines of code. This API provides a distributed architecture to enable the use of computational centers for large fluid simulations. This network protocol allows us to receive the simulation results and also sent the updates from the users to the simulation server during the simulation.

2.2. Unity 3D as a user interface for interactive simulation

For the interactive visualization tools, we decided to use *Unity 3D* for our platform. This software was initially designed for game programming but is a good compromise between speed of development and rendering performance. We provide classical camera manipulation tools for rotating, panning, and zooming. For the user's input, we chose to use ray casting to select and change the state of each voxel corresponding to a part of the simulation grid. We use the same cursor to perform multiple tasks such as setting a wall, filling a voxel with water, or emptying a voxel. Besides, it is possible to stop, start or restart, pause or continue the simulation after updating its content such as boundary conditions, grid cell states.

2.3. Lattice Boltzmann as simulation approach

We chose to use Lattice Boltzmann Method (LBM) for the simulation tool because it is a good compromise between performance and physical relevance. New solvers such as adaptive relaxation method can simulate more and more conditions like fluid-solid two way coupling simulation [LCD*20]. The LBM has become very popular for CFD simulations as it can provide fully parallel algorithms [Yeo06]. Since macroscopic variables are translated into particle distribution functions [NCGB95], the simulation is less sensitive to parameter changes during the simulation. Furthermore, this method uses a regular grid which makes the user's input easier to process. Therefore, we can change the parameters cell by cell during the simulation. This method can also provide high performance to achieve interactive rendering times. We use Palabos [LMK*20] as our LBM software coupled with Unity 3D.

3. A user study to compare interactive vs. classical simulations

The goal of this experiment is to compare the performance between the interactive simulation condition and the non-interactive one. We aimed to measure the added value of interactive simulation in terms of user experience and performance for decision making in a simple water dam design scenario.

3.1. A water dam design as a use case

In this scenario, a wall is fixed on the left to stop a water flow coming from the right. In front of the wall, there are different kinds of obstacles to slow down the water wave. Each scene has a certain type of obstacle and different water quantity, corresponding to a different optimal solution in terms of wall height. These scenes allow us to test our hypotheses on different levels of difficulty. The task of the participants was to find the minimum height of the wall to stop the water wave, and optimal solutions were computed by offline simulations performed before the experiment.

During the simulation, the system detects if the water overflows the wall or the wall is built too high for the water wave. In this case, the participant has to perform another attempt. Once the system detects that the correct height has been achieved after fluid stabilization, the next scene is loaded (see Fig.1).

In the interactive simulation condition, boundary height can be edited during a simulation in progress to perform the targeted task. In the non-interactive condition, the simulation must be restarted from the initial conditions when modifications are performed.

The simulation grid is composed of $30 \times 30 \times 72$ cells (86400 cells in total), simulating a box whose size is of $1m \times 1m \times 3.2m$. The water surface is estimated and rendered using the water fraction method for each voxel/cell of the grid.
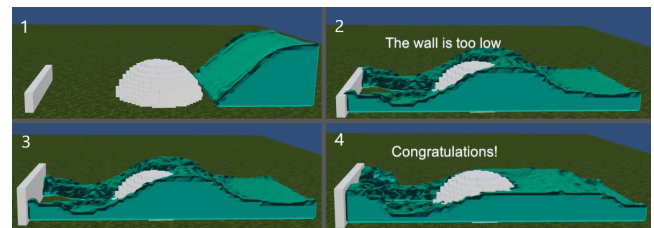

Figure 1: *Interactive simulation of water dam conception: 1 - The simulation starts, 2 - The system detects that water is higher than the wall, 3 - User modifies the wall height, 4 - User finds the optimal height.*

3.2. Hypothesis

We expected that the interactive simulation would have better user performance regarding task completion time. We also assumed that the task was easier in the interactive simulation condition. Indeed, this condition allows users to make more attempts with different setups to achieve the task, reducing the time to make a decision, and shortening the loop between simulation, visualization and analysis.

- H1: Less time is required to find the correct wall height in the interactive simulation condition.

- H2: Interactive simulation allows users to spend less time to observe and analyze the scene and result.
- H3: Overall workload is lower with interactive simulation
- H4: Interactive simulation allows users to perform more attempts to explore larger solution space.

3.3. Experimental process

Because of the Covid-19 sanitary condition, an experiment in the lab was not permitted. Therefore, before the experiment, an email containing an information notice and a consent form was sent to the subjects. Once they signed the document, they were contacted by phone and invited to connect to our computer remotely via *Teamviewer*. Then, they had to perform a 5-minute familiarisation task with a training scene to familiarize themselves with interactive features and goals. After this training stage, the first session of the experiment started with each mode of simulation (interactive mode or non-interactive condition). After 3 scenes with the same mode of simulation, they were invited to fill the NASA TLX questionnaire [Har06]. Then the subject entered the second session: another training part with the other mode of simulation started followed by 3 other scenes of the experiment. After the second part, the experiment ended with the same questionnaire for the second session.

Participants: 24 participants (13 males and 11 females) aged from 17 to 36 ($\mu = 27$, $\sigma = 4.83$)
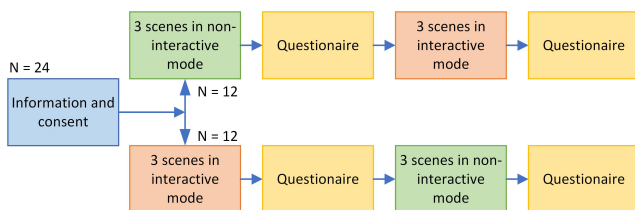

Figure 2: *24 participants were divided in two groups. 12 of them started with interactive simulation then finished with non-interactive simulation. The other 12 started with non-interactive simulation then finished with interactive simulation. The Latin-square design was constructed for counterbalancing the condition and scene.*

Data collection: For each session of the experiment, we logged all the user activities (changing viewpoint, interaction events, editing boundary condition, etc.), the results received from simulation, and completion time. From this raw data, the following measures were extracted:

- Task completion time (TCT): The duration for the user to complete the task. The measurement started when the scene was loaded until the user found the optimal height of the wall.
- Number of failures: The number of failures before achieving the task for each scene.
- Observation time: The average observation time between each interactive attempt including camera manipulation.

3.4. Results

The results presented in this section were considered statistically significant when $p < 0.05$. All the analyses were performed using R. In the bar plots, error bars show the standard deviation (SD).

Quantitative results: We found the data was not all normally distributed in a Shapiro test. In the following part of this section, we performed Wilcoxon Signed Rank test. We registered 144 trials: 2 conditions × 3 scenes × 24 participants.

For TCT, the Wilcoxon Signed Rank test showed that the task was significantly quicker ($p < 0.0001$) to achieve with interactive simulation (Mean = 230.4, SD = 50.0) than with non-interactive simulation (Mean = 141.625, SD = 23.4).

For the number of failures, the Wilcoxon Signed Rank test showed that the participants tend to try more times ($p = 0.0004$) with interactive simulation (Mean = 7.58, SD = 1.32) than with non-interactive simulation (Mean = 9.79, SD = 2.41).

For the observation time, the Wilcoxon Signed Rank test showed that the participants spent more time ($p = 0.002$) with non-interactive simulation (Mean = 8.26, SD = 3.89) than interactive simulation (Mean = 5.66, SD = 3.21).

Subjective questionnaire: For the NASA-TLX questionnaire, the ratings are averaged to calculate the overall workload. We ran the Wilcoxon Signed Rank test for the overall workload, but we did not found a significant difference ($p = 0.346$). While the Wilcoxon Signed Rank test showed that participants made less effort ($p = 0.02$) with interactive simulation (Mean = 3.71, SD = 1.83) than non-interactive simulation (Mean = 4.38, SD = 1.93).
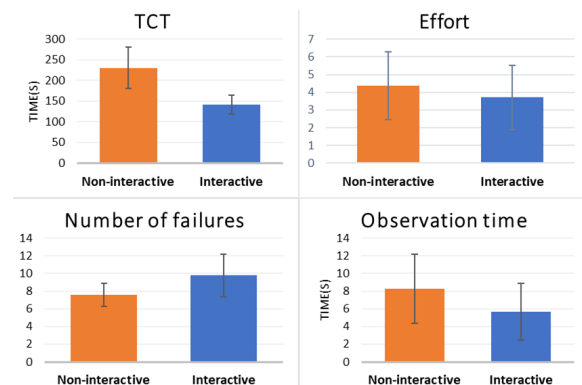

Figure 3: *Mean TCT (top-left), effort (top-right), number of failure (bottom-left) and observation time (bottom-right). Error bars show standard deviation.*

4. Discussion

From the results, we can first highlight that the participants had 39% better times completion performances when in the interactive simulation condition (H1). The interactive simulation allows the user to modify the wall height more frequently without restarting the simulation. It is much more efficient for interactive simulation to show the result with the modified wall, as it skipped most parts of the water simulation.

We also found that users made more attempts (H4) and spent less time observing the scene (H2) to achieve the task in the interactive simulation. We interpret this as that the interactive simulation costs less time to make more trials. In non-interactive mode, the participants tried to analyze the result after having the result of the first simulation as they would need to wait longer time to get the results of the modified simulation. Interactive mode let users to make more attempts in limited time to achieve the goal or get a better solution.

On the contrary to our expectations, we did not find significant differences in the overall workload, which rejects H3. From the discussion with the participants, we observed that as the participants were not experts in fluid dynamics, they had difficulties in predicting the movement of the water. We also assumed that the time needed to wait for the end of the simulation in the non-interactive condition might decrease the whole cognitive load. In the meanwhile, we did find that there were significant differences in effort. We interpret this as that since users tended to make more attempts to achieve the task, they did not make efforts to try to understand the simulation. Furthermore, as participants spent less time to achieve the task, it appeared much less difficult to them in performing it.

5. Conclusion and perspectives

In this work we present a platform that couples fluid simulation tools with *Unity 3D* to realize the visualization and interaction with a fluid simulation. We address the complex issue of modifying boundary conditions during a simulation in progress by using the LBM. We conducted a first user study to evaluate the added value of interactive simulations compared to the conventional approach in a water dam design scenario. The first results show that users have better performance in achieving the task with an interactive simulation in task completion time. As interactive simulation is less time consuming, more attempts of simulation can replace some work of analyzing the simulation results, which also makes users feel easier to achieve the task. In the meanwhile, the interactive simulation does not reduce the overall users' workload in achieving the task.

Our platform coupling *Unity 3D* and classical simulation tools are presently mainly used to conduct user studies to evaluate the added value of interactive fluid simulation and to compare different contexts such as classical and immersive environments, augmented reality context, or collaborative ones. We plan in short term to conduct further works to confirm the benefit of interactive simulation to larger fluid simulations, including other interaction contexts such as immersive and augmented reality to support collaboration. We will especially test if using ecological movement around the fluid phenomena using tracking and if collaboration features increase the performance and the quality of the decision-making process, and decrease the cognitive load.

This platform also lays the first bricks for a serious game fluid simulation application. Serious games are used for non-entertainment purposes and are applied on different domains [BEJL07], such as education, oil and gas application, health care, etc. Serious games enable learners to adapt learning to their cognitive needs and to increase intrinsic motivation of student [Mal81]. *Fold'It* is one of the most famous examples. In our case, our goals are to design scenarios and to work on gameplays to allow students

to get a better understanding of the fluid physics phenomena. Inspired by *Fold'It*, we will use our platform as a serious game to illustrate fluid mechanics concepts through appropriate scenarios designed with experts, and to know if serious game and citizen science approaches could solve practical fluid mechanical problems.

References

[BEJL07] BACKLUND P., ENGSTROM H., JOHANNESSON M., LEBRAM M.: Games and traffic safety-an experimental study in a game-based simulation environment. In *2007 11th International Conference Information Visualization (IV'07)* (2007), IEEE, pp. 908–916. 4

[CTG10] COHEN J. M., TARIQ S., GREEN S.: Interactive fluid-particle simulation using translating eulerian grids. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), ACM, pp. 15–22. 1

[Har06] HART S. G.: Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting* (2006), vol. 50, Sage publications Sage CA: Los Angeles, CA, pp. 904–908. 3

[KDC*11] KHATIB F., DIMAIO F., COOPER S., KAZMIERCZYK M., GILSKI M., KRZYWDA S., ZABRANSKA H., PICHOVA I., THOMPSON J., POPOVIĆ Z., JASKOLSKI M., BAKER D., GROUP F. C., GROUP F. V. C.: Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature Structural & Molecular Biology 18*, 10 (2011), 1175–1177. 1

[LCD*20] LI W., CHEN Y., DESBRUN M., ZHENG C., LIU X.: Fast and scalable turbulent flow simulation with two-way coupling. *ACM Trans. Graph. 39*, 4 (July 2020). doi:10.1145/3386569.3392400. 2

[LMK*20] LATT J., MALASPINAS O., KONTAXAKIS D., PARMIGIANI A., LAGRAVA D., BROGI F., BELGACEM M. B., THORIMBERT Y., LECLAIRE S., LI S., ET AL.: Palabos: Parallel lattice boltzmann solver. *Computers & Mathematics with Applications* (2020). 2

[Mal81] MALONE T. W.: Toward a theory of intrinsically motivating instruction. *Cognitive science 5*, 4 (1981), 333–369. 4

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 154–159. 1

[NCGB95] NOBLE D. R., CHEN S., GEORGIADIS J. G., BUCKIUS R. O.: A consistent hydrodynamic boundary condition for the lattice boltzmann method. *Physics of Fluids 7*, 1 (1995), 203–209. 2

[SGvR*03] SCHIRSKI M., GERNDT A., VAN REIMERSDAHL T., KUHLEN T., ADOMEIT P., LANG O., PISCHINGER S., BISCHOF C.: Vista flowlib - framework for interactive visualization and exploration of unsteady flows in virtual environments. In *Proceedings of the Workshop on Virtual Environments 2003* (New York, NY, USA, 2003), EGVE '03, Association for Computing Machinery, p. 77–85. 1

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., p. 121–128. doi:10.1145/311535.311548. 1

[VMN*09] VÉZIEN J. M., MÉNÉLAS B., NELSON J., PICINALI L., BOURDOT P., AMMI M., KATZ B. F., BURKHARDT J. M., PASTUR L., LUSSEYRAN F.: Multisensory vr exploration for computer fluid dynamics in the corsaire project. *Virtual Real. 13*, 4 (2009), 257–271. 1

[WFBM19] WANG M., FEREY N., BOURDOT P., MAGOULÈS F.: Interactive 3d fluid simulation: steering the simulation in progress using lattice boltzmann method. In *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)* (2019), IEEE, pp. 72–75. 2

[Yeo06] YEOMANS J.: Mesoscale simulations: Lattice boltzmann and particle algorithms. *Physica A: Statistical Mechanics and its Applications 369*, 1 (2006), 159–184. 2