Interactive and Immersive Simulation using the Continuous Sensitivity Equation Method with Application to Thermal Transport

Agnieszka Kozub, Paul Ellinghaus, Régis Duvigneau, Jean-Christophe Lombardo, Nicolas Dalmasso

HAL Id: inria-00530166
https://hal.inria.fr/inria-00530166

Submitted on 27 Oct 2010

INRIA

Interactive and Immersive Simulation using the Continuous Sensitivity Equation Method with Application to Thermal Transport

A. Kozub — P. Ellinghaus — R. Duvigneau — J.-C. Lombardo — N. Dalmasso

Rapport de recherche

Interactive and Immersive Simulation using the Continuous Sensitivity Equation Method with Application to Thermal Transport

A. Kozub*, P. Ellinghaus* , R. Duvigneau [†] , J.-C. Lombardo [‡] , N. Dalmasso [§]

Abstract: The objective of this work is to develop a computational method that allows the user to visualize the solution of a PDE system (Partial Differential Equations) in virtual reality facilities and interact with the computation at real time. This could be possible only if a fast estimate of the physical solution is constructed.

In the present work, we propose to employ a first-order Taylor series expansion to extrapolate the solution field from a reference solution, as some parameters vary. The continuous sensitivity equation method is used to compute the sensitivity field for the reference solution.

The methodology proposed is applied to a heat transport problem (advection - diffusion) and demonstrated in a virtual reality facility.

Key-words: Sensitivity equation method, virtual reality, thermal transport

* Ecole Polytech'Nice
[†] INRIA Opale Project-Team
[‡] INRIA Dream Team
[§] SimplySim company

Simulation interactive et immersive utilisant la méthode de l'équation des sensibilités continues avec application au transport thermique

Résumé : L'objectif de ce travail est le développement d'une méthode de calcul permettant à un utilisateur de visualiser la solution d'un système d'EDP (Equations aux Dérivées Partielles) dans une salle de réalité virtuelle et d'interagir avec le calcul en temps réel. Ceci ne peut être possible que si une estimation rapide de la solution physique est construite.

Dans ce travail, on propose d'employer une série de Taylor au premier ordre pour extrapoler le champ solution à partir d'une solution de référence lorsque des paramètres varient. La méthode de l'équation des sensibilités continues est utilisée pour calculer le champ de sensibilité pour la solution de référence.

La méthodologie proposée est appliquée à un problème de transport thermique (advection - diffusion) et exposée dans une salle de réalité virtuelle.

Mots-clés : Méthode de l'équation des sensibilités, réalité virtuelle, transport thermique

Contents

Introduction

Simulation and visualization are two tasks in scientific computing that are usually carried out successively. These tasks are possibly performed on different computers: the simulation task is computationally expensive and hence uses high-performance computing facilities. On the contrary, visualization is often considered as a secondary task that can be performed using a simple workstation.

However, there is a growing demand for interactive processes on one side, and for the use of more sophisticated visualization devices on the other side. Indeed, the more and more intensive use of simulation methods in engineering design requires the development of tools that allow the user to modify interactively simulation parameters and visualize changes in the physical solution at real time. Then, engineers can rapidly have an answer to the question: what if .... ? Moreover, it is more and more obvious that simple screens are not able to represent adequately 3D solution fields for complex simulations, such as multiphase or turbulent flow simulations. The use of 3D immersive facilities could be a new and effective way to visualize and apprehend a physical phenomenon.

In the present report, we experiment a new methodology to adress both issues: a method for fast estimation of nearby solutions is employed to compute at real time solution fields as the user interactively modifies some parameters. Moreover, this approach has been linked to a virtual reality software to visualize solution fields as 3D scenes in an immersive space.

A practical problem in heat transport is presented in the first section. Then, we describe the proposed methodology employed for fast estimation of nearby solutions. Finally the 3D visualization process is illustrated.

1 Problem description

1.1 Modelling

We consider as model problem the thermal transport equations, including advection and diffusion effects. Given a domain Ω closed by the boundary $\Gamma = \Gamma_D \cup \Gamma_N$, state equations read:

$$
\begin{aligned}
\frac{\partial T}{\partial t}(\mathbf{x}, t) + C\, \mathbf{a}(\mathbf{x}) \cdot \nabla T(\mathbf{x}, t) &= \nabla \cdot (\kappa \nabla T(\mathbf{x}, t)) \quad \text{in } \Omega \\
T(\mathbf{x}, t) &= T_0(\mathbf{x}, t) \text{ on } \partial\Omega_D \\
\kappa \frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}, t) &= \Phi_0(\mathbf{x}, t) \text{ on } \partial\Omega_N \\
T(\mathbf{x}, 0) &= T_{ini}(\mathbf{x}),
\end{aligned}
\tag{1}
$$

where $\mathbf{x}$ are the Cartesian coordinates and t the time, T represents the temperature field and κ the thermal conductivity. Dirichlet and Neumann boundary conditions are applied on $\partial\Omega_D$ and $\partial\Omega_N$ respectively, T_0 and Φ_0 being the imposed temperature and thermal flux ($\mathbf{n}$ unit normal outward vector). Finally, T_{ini} is the initial temperature field. For the sake of simplicity, it is assumed that the advection field $\mathbf{a}(\mathbf{x})$ does not depend neither on time nor on the temperature solution.

1.2 Test-case

We consider as test-case the heat transport in a room that contains a window, a door, a heater and a fan (see Fig. 1). We assume that a flow is entering by the bottom of the door and by the window, that is partially open, and is exiting by the fan.
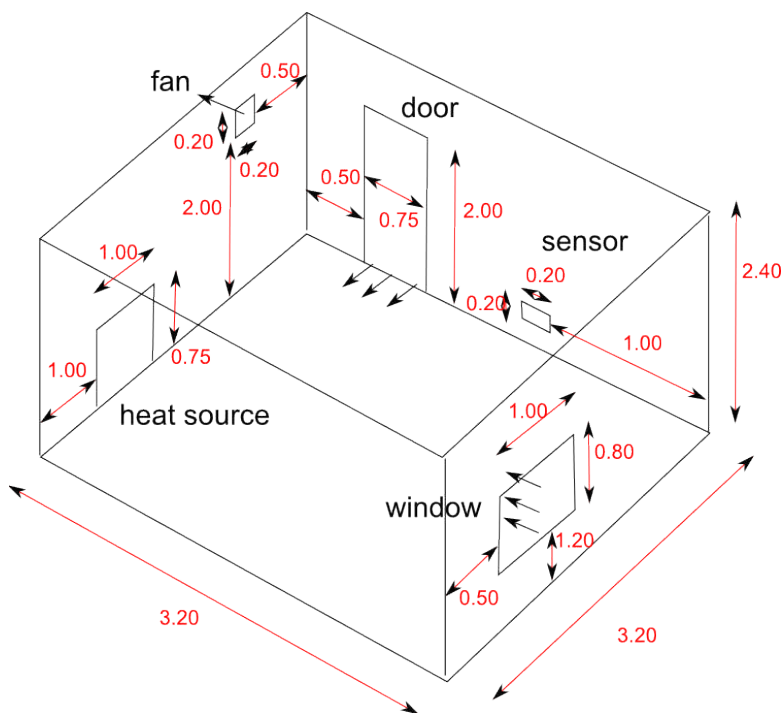

Figure 1: Configuration of the test-case.

For the thermal transport equations presented above, the following boundary conditions are adopted: we assume that the walls and the ceiling correspond to perfect adiabatic conditions:

$$
\frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}, t) = 0 \text{ on } \partial\Omega_{wall} \cup \partial\Omega_{ceiling}.
\tag{2}
$$

The door and the extraction fan are assumed to be exposed to an atmosphere of the same temperature, i.e. no heat flux is gained or lost at these points:

$$\frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}, t) = 0 \text{ on } \partial\Omega_{door} \cup \partial\Omega_{fan}. \tag{3}$$

Moreover, we assume that the window has a constant and uniform temperature, yielding a Dirichlet boundary condition:

$$T(\mathbf{x}, t) = T_{win} \text{ on } \partial\Omega_{window}. \tag{4}$$

Finally, the heater is modeled by a Neumann boundary condition. We assume it is permanently switched on to provide a heat flux that varies in accordance with the temperature of a sensor and a user-defined target temperature. Finally, the boundary condition has the form:

$$\frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}, t) = \psi(t)\varphi(\mathbf{x}) \text{ on } \partial\Omega_{heater}. \tag{5}$$

Function $\psi(t)$ allows to control the magnitude of the heat flux. It is dependent on the sensor temperature T_s and the desired temperature T_d:

$$\psi(t) = \psi_0 + \alpha(T_d - T_s(t)), \tag{6}$$

with ψ_0 and α user-defined constants. The function $\varphi(\mathbf{x})$ defines the flux shape and depends on the heater location and size. A tensor product of two piecewise cubic polynomials is chosen such that it has a compact support with respect to the dimensions of the heater (see Fig. 2) and is C^1. The choice of this model will facilitate the derivation of boundary conditions for the sensitivity fields, as we will see latter.
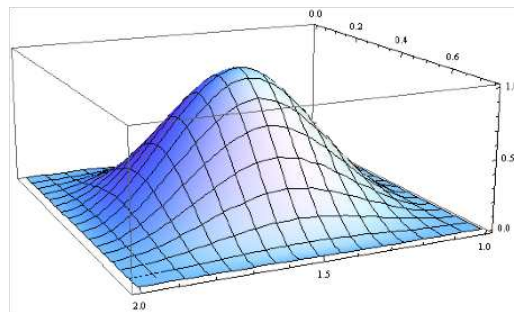

Figure 2: Shape of the flux.

The initial temperature field is assumed to be uniform:

$$T(\mathbf{x}, 0) = T_{ini}. \tag{7}$$

1.3 Visualization and interaction

The objective of these developments is to visualize the time evolution of the 3D temperature field in a virtual reality facility and allow the user to interact with the simulation by moving the heater horizontally. Therefore, we define as x_h the abscissa of the heater centroid along the wall. The temperature field is considered as a function of space, time and heater location $T(\mathbf{x}, t; x_h)$.

To compute a solution field at real time for each new heater location, one should develop a fast estimate of the temperature field. This process is presented in the following sections.

2 Computation of a reference solution

First, a reference temperature field is computed for a given heater location $\overline{x_h}$ and stored. Then, a fast estimate of the solution is constructed from the reference field, for a nearby location of the heater.

2.1 Flow field

A first step consist of computing a flow that defines the advection field **a**. Then, we will assume that the evolution of temperature field will not modify the flow. Therefore, the flow field is computed only once.

An unstructured grid is generated in the computational domain, including some refined area in the vicinity of the inlet and outlet boundaries (see Fig. 3). The grid counts 65,223 vertices and 382,407 tetrahedra. The parallel Navier-Stokes solver NUM3SIS[3], developed at INRIA, is carried out to compute a velocity field (see Fig. 4). The flow is assumed to be laminar.
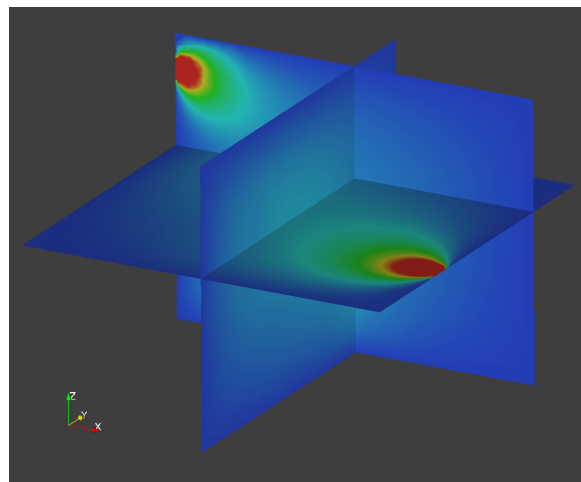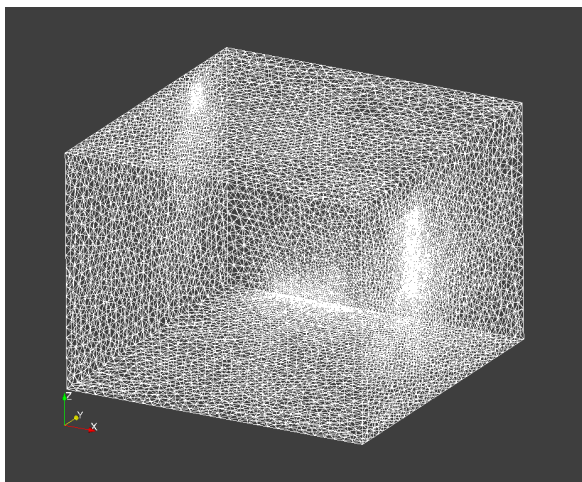

Figure 3: Unstructured grid for the flow computation.

Figure 4: Velocity magnitude in some planes.

2.2 Temperature field

Once the advection field is known, the temperature solution for a reference heater location $\overline{x_h}$ is computed by solving the PDE system (Eq. 1) with prescribed boundary conditions. This task is achieved by using a cartesian grid of size $N_x = N_y = 32$ and $N_z = 16$. The velocity field is interpolated on this grid to provide the advection field. The equations are discretized using a finite-difference scheme, with upwind advection terms, centered diffusion terms and an explicit first-order backward time integration procedure. The time step is chosen such that a CFL stability condition holds.

This reference solution is stored and is used to compute sensitivity fields and fast estimate of the solution fields as the heater is moved.

3   Fast estimation of a nearby solution

Once the temperature field is known for a reference heater location $\overline{x_h}$, an estimate of the temperature field for a new heater location $\widetilde{T}(\mathbf{x}, t; x_h)$ can be obtained by using a first-order Taylor series expansion from the known solution:

$$\widetilde{T}(\mathbf{x}, t; x_h) = T(\mathbf{x}, t; \overline{x_h}) + S_T(\mathbf{x}, t; \overline{x_h}) \, (x_h - \overline{x_h}), \tag{8}$$

where S_T represents the sensitivity field of the temperature with respect to the heater location:

$$S_T(\mathbf{x}, t; \overline{x_h}) = \frac{\partial T}{\partial x_h}(\mathbf{x}, t; \overline{x_h}). \tag{9}$$

This field measures the change in the temperature field as the heater location varies. One can notice that the computation of a nearby solution using Eq. 8 is rather inexpensive and can be carried out at real time, provided that the temperature field and its sensibility are known. More details about this approach can be found in the literature (see for instance [1]).

4 The continuous sensitivity equation method

One can obtain a PDE system governing the sensitivity field by differentiating implicitly the state equation (Eq. 1) with respect to the parameter x_h. This approach is known as the continuous sensitivity equation method and has been applied to several fields for various applications [1, 2, 4, 5]

$$\frac{\partial}{\partial x_h}\left(\frac{\partial T}{\partial t}(\mathbf{x}, t; x_h) + C\, \mathbf{a}(\mathbf{x}) \cdot \nabla T(\mathbf{x}, t, x_h)\right) = \frac{\partial}{\partial x_h}\left(\nabla \cdot (\kappa \nabla T(\mathbf{x}, t; x_h))\right) \tag{10}$$

After some simplifications (permutation of partial derivatives), we obtain finally:

$$\frac{\partial S_T}{\partial t}(\mathbf{x}, t; x_h) + C\, \mathbf{a}(\mathbf{x}) \cdot \nabla S_T(\mathbf{x}, t, x_h) = \nabla \cdot (\kappa \nabla S_T(\mathbf{x}, t; x_h)) \tag{11}$$

We employ the same procedure to derive suitable boundary conditions. For the wall, ceiling, door and fan, zero Neumann boundary conditions are obtained:

$$\frac{\partial}{\partial x_h}\left(\frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}, t)\right) = 0 \text{ on } \partial\Omega_{wall} \cup \partial\Omega_{ceiling} \cup \partial\Omega_{door} \cup \partial\Omega_{fan}. \tag{12}$$

Then:

$$\frac{\partial S_T}{\partial \mathbf{n}}(\mathbf{x}, t) = 0 \text{ on } \partial\Omega_{wall} \cup \partial\Omega_{ceiling} \cup \partial\Omega_{door} \cup \partial\Omega_{fan}. \tag{13}$$

For the window, a zero Dirichlet boundary condition is derived, since the window temperature does not depend on the heater location:

$$\frac{\partial}{\partial x_h}\left(T(\mathbf{x}, t)\right) = \frac{\partial}{\partial x_h}\left(T_{win}\right) \text{ on } \partial\Omega_{window}. \tag{14}$$

$$S_T(\mathbf{x}, t) = 0 \text{ on } \partial\Omega_{window}. \tag{15}$$

Finally, we differentiate the heat flux imposed at the heater to derive sensitivity boundary conditions:

$$\frac{\partial}{\partial x_h}\left(\frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}, t)\right) = \frac{\partial}{\partial x_h}\left(\psi(t)\varphi(\mathbf{x}; x_h)\right) \text{ on } \partial\Omega_{heater}. \tag{16}$$

$$\frac{\partial S_T}{\partial \mathbf{n}}(\mathbf{x}, t) = \psi(t)\frac{\partial}{\partial x_h}\left(\varphi(\mathbf{x}; x_h)\right) \text{ on } \partial\Omega_{heater}. \tag{17}$$

Only the function $\varphi(\mathbf{x}; x_h)$, that defines the shape of the flux, depends on x_h. Its derivative yields the flux for the sensitivity, which is depicted on Fig. 5.
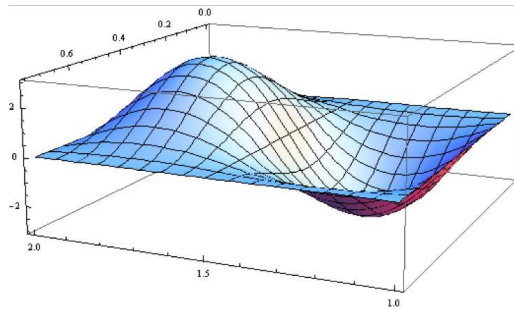

Figure 5: Shape of the flux sensitivity.

As conclusion, the sensitivity field is governed by a PDE system that is very similar to the state equations for the temperature. Only the boundary conditions differ and, in practice, one uses the same computational methods to compute the temperature field and the sensitivity field. One can underline that the sensitivity equations are written in advance of time. Therefore, one can compute at each time step the temperature first and the sensitivity field then.

5 Visualization in the virtual reality space

5.1 Presentation of the virtual reality space at INRIA Sophia-Antipolis

The immersive space at INRIA (see Fig. 6) regroups two display devices, a CadWall for its ease of use and an iCube (see Fig. 7) for its immersion quality. Both devices offer 6:1 sound restitution, Infitec stereo images and optical tracking. Image generation can also be performed by two different syst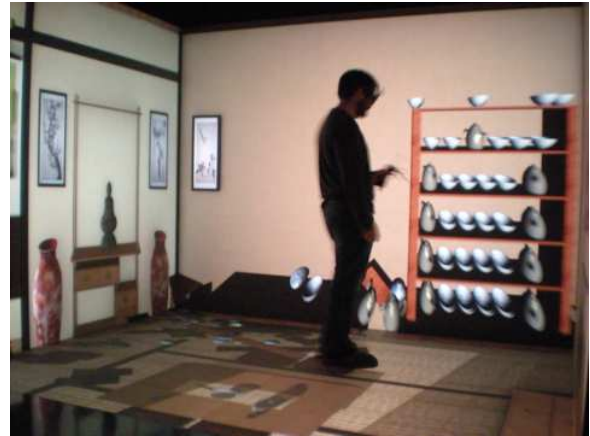em, a single PC with multiple graphics card or a cluster. More details can be found at `http://www-sop.inria.fr/dream/ImmersiveSpace/ImmersiveSpace`.

Figure 6: Immersive space at INRIA Sophia-Antipolis.

Figure 7: Example of 3D scene with interaction.

5.2 Virtual reality software (SimplySim company)

The 3D scene visualization is carried out using the software SimplyCube developed by SimplySim company (`http://www.simplysim.net`).

The SimplyCube is a product developed by SimplySim since 2008, dedicated to creating applications for real-time 3D simulations, virtual reality, serious games and video games. It is composed of two main parts: a powerful and easy-to-use SDK to create 3D application and a set of different editors to build fully interactive 3D environments.

There are a lot of advantages to using the SimplyCube. First, it is focused on realism, both for 3D rendering and physics computing by using state of the art 3D techniques and multi-physics engine technology. Then, thanks to its SDK design and the editors, the SimplyCube is more user friendly than any other software available on the market. With the Microsoft Visual Programming Language compatibility, there is no need to know anything about 3D programming. The application is designed visually and the code is automatically generated for. Finally, the Service Oriented Architecture used by the SimplyCube makes the creation of interactions very easy. Each service can be associated with virtual entities present in the scene, giving specific behavior to this object. For example, we can bind a MotorizedHinge service to a door to create an open/close interaction.

For virtual reality applications like SimViz, we generally use the virtual reality pack which adds functionalities like VRPN binding (tracking, analog, force feedback devices ...), stereo rendering (asymmetric frustum, DirectX quadbuffer) and clustering thanks to the Scalable Graphics's technology. More information can be found at `http://www.simplysim.net`.

5.3 Overall procedure

Finally, the overall procedure for the interactive and immersive computation and visualization are summarized in the following procedure, illustrated by Fig. 8:

1. Pre-processing phase (reference heater location):

   - Compute the advection flow field using a Navier-Stokes solver and storage;
   - Compute the temperature field by solving Eq. 1 and storage;

- Compute the sensitivity field by solving Eq. 11 and storage;

2. Interactive simulation

- Identify the current heater location (provided by Simplysim server);
- Fast estimate of the solution field using Eq. 8 and stored fields;
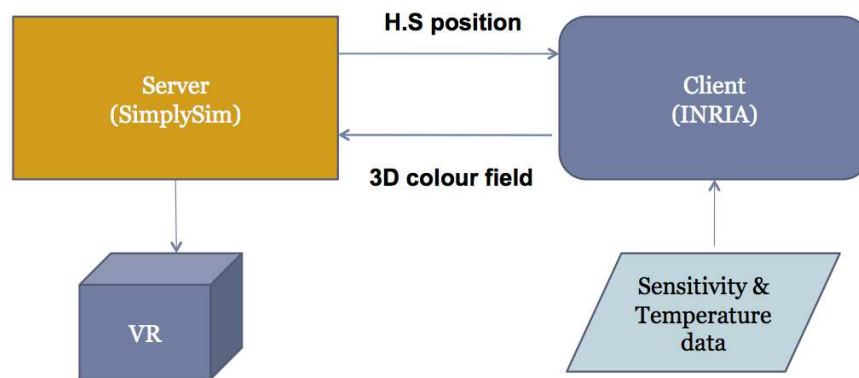- Update of the scene (sent to simplysim server);

Figure 8: Workflow for interactive computation and visualization.

6   Results

6.1   Validation of the fast estimation procedure

First, we validate the fast estimation procedure by comparing the temperature field for a given heater location x_h obtained from a simulation $T(\mathbf{x}, t; x_h)$, to the estimate $\widetilde{T}(\mathbf{x}, t; x_h)$ obtained by extrapolation from reference fields $T(\mathbf{x}, t; \overline{x_h})$ and $S_T(\mathbf{x}, t; \overline{x_h})$.

Fig. 9 and 10 show the comparison of the fields in a 2D plane, for a small displacement $|\overline{x_h} - x_h| = 0.1m$, whereas the corresponding sensitivity field is represented in Fig. 11. Positive sensitivity values yield an increase of the temperature, whereas negative values yield a decrease. As can be seen, the fast estimate is in satisfactory agreement with the simulation. The comparison of temperature isolines (Figs. 12 and 13) shows that a small discrepancy exists, that is related to the omission of higher-order terms in the Taylor series expansion.
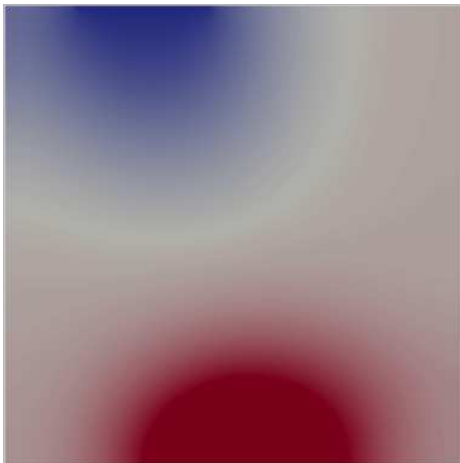
Figure 9: Colormap of the temperature field for a small displacement ($t = 5s$).
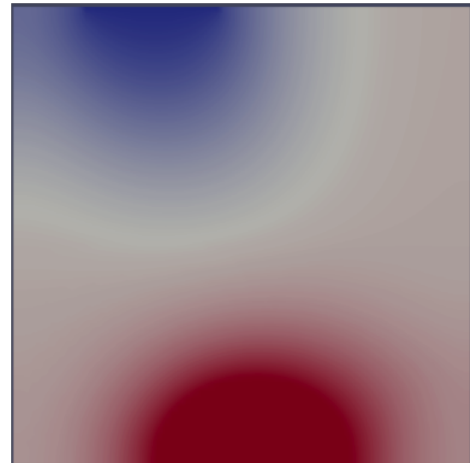
Figure 10: Colormap of the fast estimate for a small displacement ($t = 5s$).
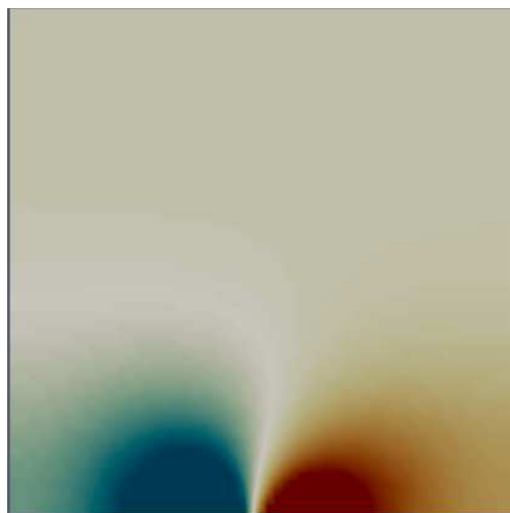
Figure 11: Colormap of the sensitivity field ($t = 5s$).

However, if one considers a larger displacement $|\overline{x_h} - x_h| = 0.5m$, the accuracy of the estimate degrades and the comparison of the solutions exhibits significant discrepancies (see Fig. 14 to 17). This behavior was expected, since we use first-order extrapolations only. This difficulty could be overcome by using extrapolations of higher order or by combining linear extrapolations from several reference points.
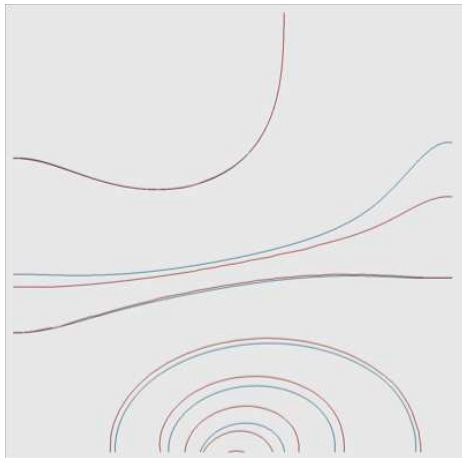
Figure 12: Isolines of the temperature field for a small displacement ($t = 10s$). Blue lines correspond to the simulation, red lines to the fast estimate.
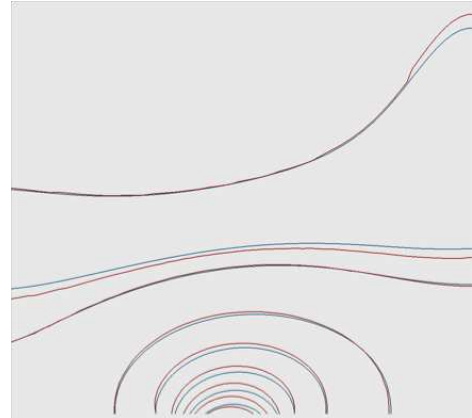

Figure 13: Isolines of the temperature field for a small displacement ($t = 30s$). Blue lines correspond to the simulation, red lines to the fast estimate.
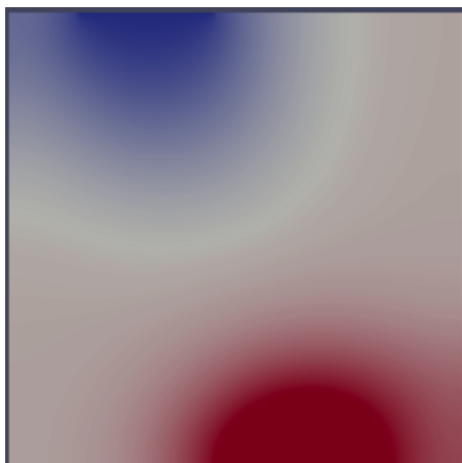

Figure 14: Colormap of the temperature field for a large displacement ($t = 5s$).
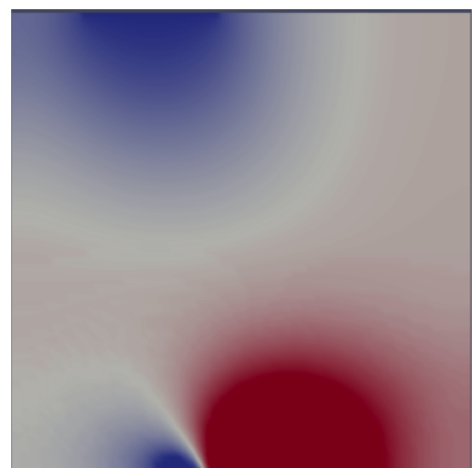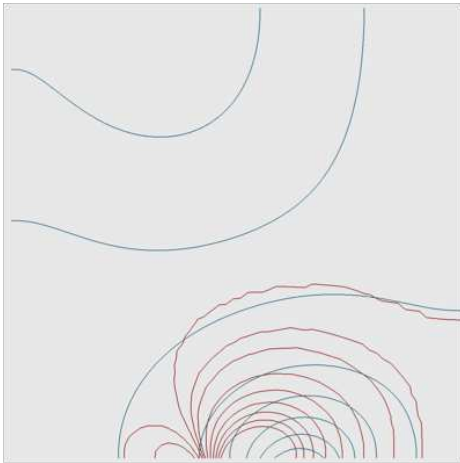

Figure 15: Colormap of the fast estimate for a large displacement ($t = 5s$).

Figure 16: Isolines of the temperature field for a large displacement ($t = 10s$). Blue lines correspond to the simulation, red lines to the fast estimate.

Figure 17: Isolines of the temperature field for a large displacement ($t = 30s$). Blue lines correspond to the simulation, red lines to the fast estimate.

6.2 Visualization in virtual reality facility

A 3D scene representing the computational domain and including the various components (heater, window, fan, etc) is created thanks to the software described above (see Fig. 18). The heater can be moved easily by hand (see Fig. 19). Translation and rotation can be applied, although we only consider horizontal translation along the wall for this project.
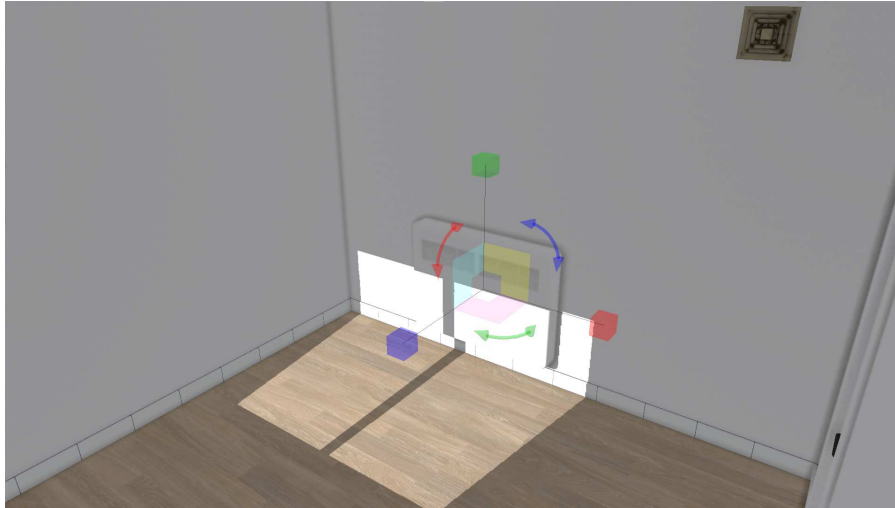
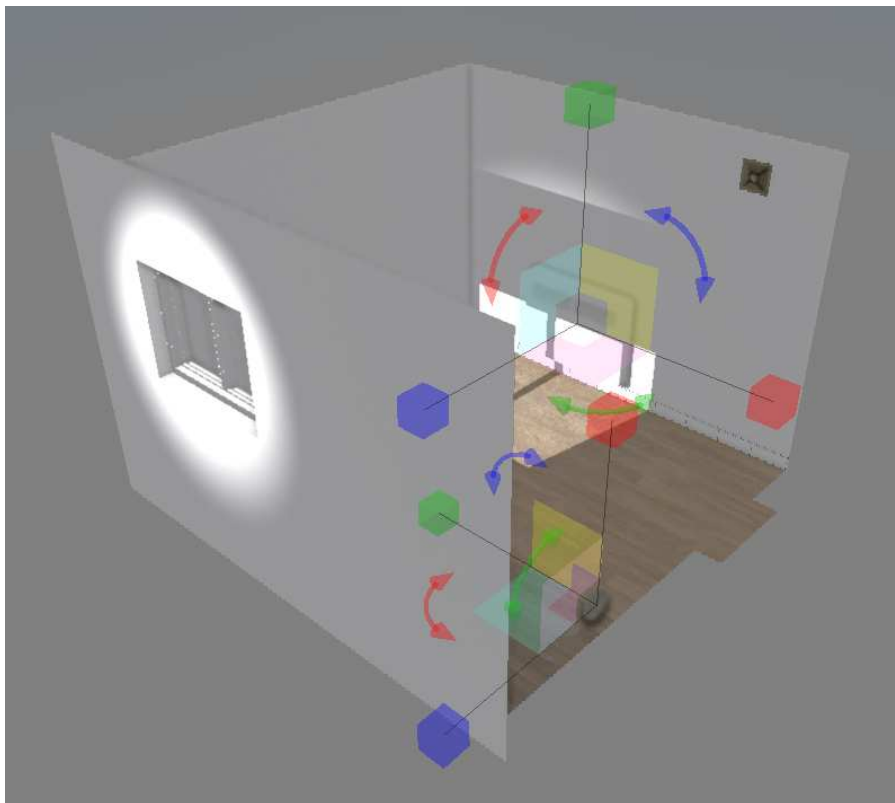Figure 18: 3D scene with the heater, the fan and the door. Sunlight is entering through the window.

Figure 19: Manipulation of the scene components (heater and sensor)

The 3D temperature field is represented by a shader colored by the temperature values (see Fig. 20 and 21). Actually, a HSV (hue, saturation, value) description of the colormap is employed, the temperature value defining the hue.
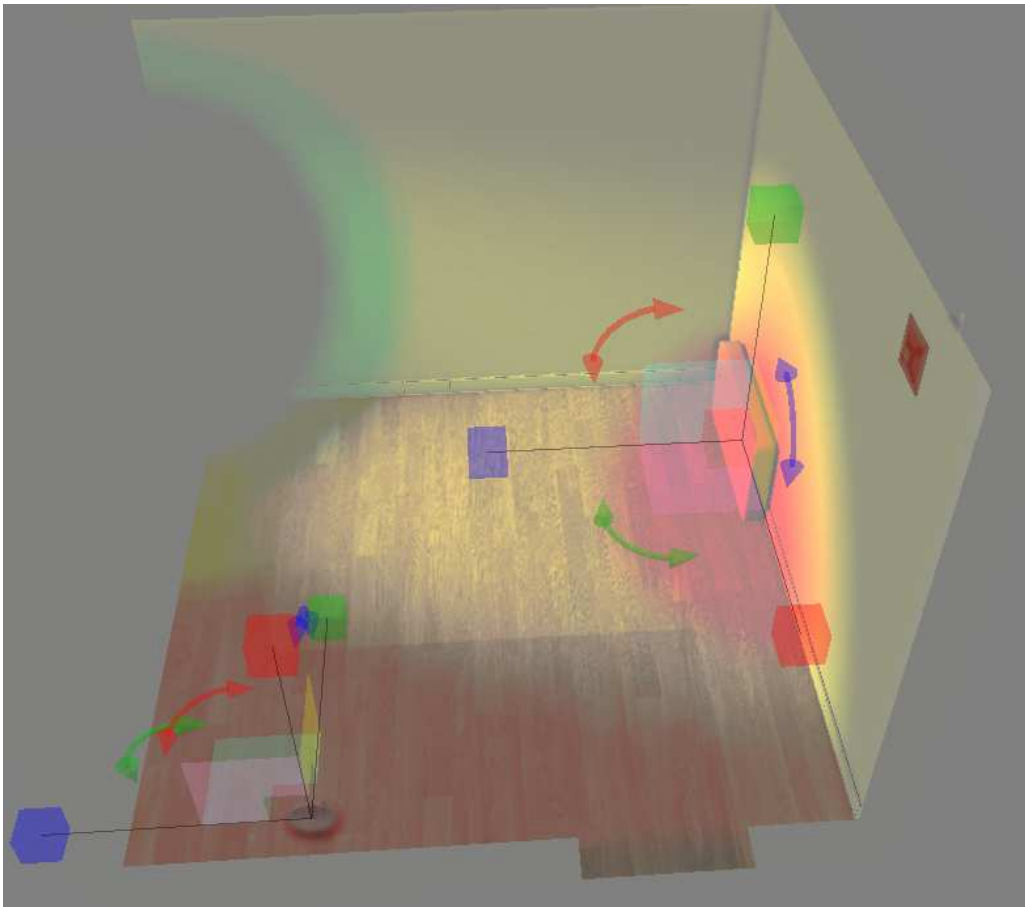

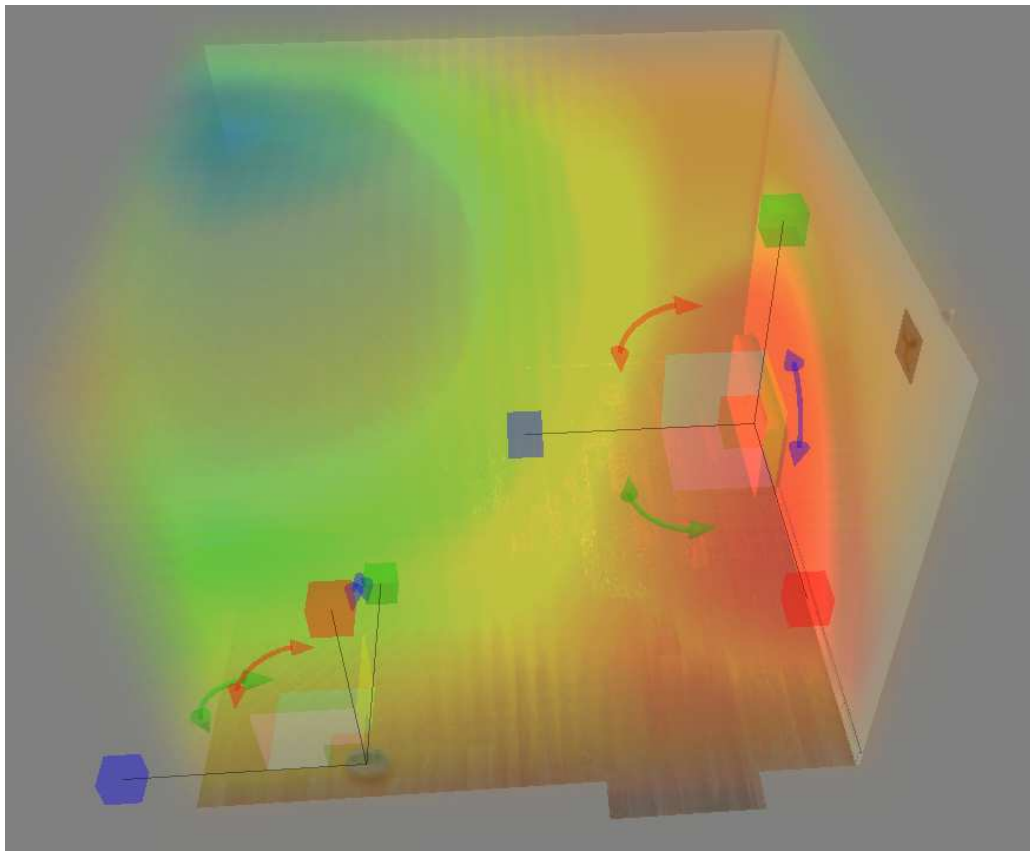Figure 20: Temperature field estimated for a given heater position ($t = 10s$).

Figure 21: Temperature field estimated for a given heater position ($t = 30s$).

7 Conclusion

In this work, we have used the continuous sensitivity equation method to compute fast estimate of nearby solutions as some simulation parameters vary. This allows the user to modify interactively a parameter and visualize at real time the changes in the solution fields.

Morevover, a 3D immersive space has been used in conjunction with a virtual reality software (from SimplySim company) to visualize solution fields.

A validation phase has shown that a more sophisticated solution reconstruction method should be employed for large parameter changes. For instance, one should combine several linear extrapolations from different pre-computed solutions, or employ second-order extrapolations.

The 3D solution rendering in the immersive space can also be improved, in order to facilitate the visualization of some precise solution values. Some on-going works are currently focused on this issue.

References

[1] DUVIGNEAU, R., AND PELLETIER, D. A sensitivity equation method for fast evaluation of nearby flows and uncertainty analysis for shape parameters. *Int. J. of Computational Fluid Dynamics 20*, 7 (August 2006), 497–512.

[2] DUVIGNEAU, R., PELLETIER, D., AND BORGGAARD, J. An improved continuous sensitivity equation method for optimal shape design in mixed convection. *Numerical Heat Transfer part B : Fundamentals 50*, 1 (July 2006), 1–24.

[3] KLOCZKO, T. Concept, architecture and performance study for a parallel code in cfd. In *Parallel CFD Conference, May 19-22, Lyon, France.*

[4] TURGEON, É., PELLETIER, D., AND BORGGAARD, J. A general continuous sensitivity equation formulation for the $k - \epsilon$ model of turbulence. In *31st AIAA Fluid Dynamics Conference and Exhibit* (Anaheim, CA, Jun. 2001). AIAA Paper 2001-3000.

[5] TURGEON, É., PELLETIER, D., AND BORGGAARD, J. Sensitivity and uncertainty analysis for variable property flows. In *39th AIAA Aerospace Sciences Meeting and Exhibit* (Reno, NV, Jan. 2001). AIAA Paper 2001-0139.