Sensitivity equation method for the Navier-Stokes equations applied to uncertainty propagation
Camilla Fiorini, Bruno Després, Maria Adela Puscas

To cite this version:
Camilla Fiorini, Bruno Després, Maria Adela Puscas. Sensitivity equation method for the Navier-Stokes equations applied to uncertainty propagation. International Journal for Numerical Methods in Fluids, 2021, 10.1002/fld.4875. hal-02485825v2

HAL Id: hal-02485825
https://hal.science/hal-02485825v2
Submitted on 25 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Sensitivity equation method for the Navier–Stokes equations applied to uncertainty propagation

Camilla Fiorini¹, Bruno Després¹, and Maria Adela Puscas²

¹LJLL, Sorbonne Université, Paris, France.
²CEA, Centre de Saclay, DEN/SAC/DANS/DM2S/STMF/LMSF, Saclay, France.

Abstract

This work deals with sensitivity analysis for the Navier–Stokes equations. The aim is to provide an estimate of the variance of the velocity field when some of the parameters are uncertain and then to use the variance to compute confidence intervals for the output of the model. First, we introduce the physical model and analyse its stability. The sensitivity equations are derived, and their stability analysed as well. We propose a finite element-volume numerical scheme for the state and the sensitivity, which is integrated into the open-source industrial code TrioCFD. Finally, we present some numerical results: a steady and an unsteady test case for the channel flow problem are investigated. For the steady case, we compare the results to the Monte Carlo method and show how the sensitivity analysis technique succeeds in providing very accurate estimates of the variance. For the unsteady case, a new filtering procedure is proposed to deal with a sensitivity that grows in time. The filtered sensitivity is then used to compute the variance of the output and to provide confidence intervals.

1 Introduction

Sensitivity analysis (SA) studies how changes in the input of a model affect the output, and it is essential for many engineering applications, such as uncertainty quantification, optimal design, and to answer what if questions, i.e. what happens to the solution of the model if the input parameters change. These tasks can be performed in many different ways, depending on the nature of the model considered. In this work, we consider systems that can be modelled with partial differential equations (PDEs). The sensitivity variable itself is defined as the derivative of the state (i.e., the output of the model) with respect to the parameters of interest.

In the framework of PDEs, one can distinguish two main classes of methods: the differentiate-then-discretise methods and the discretise-then-differentiate methods. As the names say, in the first case the state model is formally differentiated with respect to the parameter of interest, providing an analytical sensitivity system which can then be discretised with the most appropriate numerical scheme. The second class of methods swaps the two steps which, in the general case, are not commutative. A detailed comparison between the two classes of methods can be found in [Gun03] for optimisation problems. In this work, we focus on the first class, and, in particular, on the continuous sensitivity equation method [BB97, DPB06, DP06, CDF18, FCD19].

The main aim of this work is to give an estimate of the variance of the solution of the Navier–Stokes equations when there are uncertain parameters and then to use the estimated variance to compute confidence intervals. This goes under the name of forward uncertainty propagation, which is part of the broader field of uncertainty quantification (UQ). Many strategies and techniques have been proposed in the literature to tackle UQ
problems, particularly in the case of PDEs models: Monte Carlo method [RC13], polynomial chaos [Wal03, XK03, KM06, DPL13], random space partition [AC13], to name but a few. A review of these methods applied to fluid dynamics problems can be found in [WH02].

Methods of uncertainty propagation based on SA are particularly efficient in terms of computational time if compared, for instance, to methods like Monte Carlo. However, since SA is based on Taylor expansions of the state variable with respect to the parameter of interest, these methods are intrinsically local [Del14]: they can be used only for random variables with a small variance. The Monte Carlo approach does not require this assumption; however, it is not applicable for realistic unsteady test cases in 2D and 3D, due to its high computational cost. In this work we propose an approach based on the sensitivity equation method: under the hypothesis of small variance of the input parameters, we can provide a first-order estimate of the variance of the solution at a reasonable computational cost.

Two declinations of the channel flow test case (i.e. study of the flow past an obstacle) are investigated: a steady ($Re = 25$) and an unsteady ($Re = 100$) test case. For the steady test case, a detailed comparison with the Monte Carlo method is performed: the variance of the output is first estimated using a classical Monte Carlo technique and then with the sensitivity analysis method. The results are extremely accurate. The unsteady test case presents a sensitivity which grows in time: this is caused by the fact that the parameter considered has an influence on the frequency of vortex shedding. This problem is also described in [HEPB04] for a similar test case to the one considered here: to deal with it, they chose a pulsating inflow velocity, which imposes the frequency of vortex shedding. However, this choice is not suitable for all applications. Therefore we propose a new filtering technique, which allows us to recover the periodic physical part of the sensitivity.

The main original contributions of this work are: (a), to establish the stability in the norm $L^p(0,T; L^2(\Omega)) \cap L^2(0,T; H^1(\Omega))$ for the state and for the sensitivity, we provide an explicit function adapted to the specific geometry to homogenise the Dirichlet boundary conditions (b), we detail the discretisation of the method within a finite element-volume (FEV) framework, well adapted to the open-source industrial code TrioCFD: (c), to be able to remove the phase dependency in the sensitivity signals in the case of pulsating flows, typically in the Von Karman vortex street, we propose and justify a new filtering technique. These results are also validated with a detailed comparison with Monte Carlo simulations.

The paper is organised as follows: in section 2 we present the physical model for the state and derive the sensitivity equations. Stability estimates are provided for both the state and the sensitivity for the prescribed boundary conditions. To do that, we introduce a function to homogenise the Dirichlet boundary conditions: the computation of such a function is explicit and detailed in appendix A. In section 3, we design a FEV method for the sensitivity, which is adapted to the open-source industrial code TrioCFD [DC]. In section 4, the code implemented for the sensitivity is rigorously validated. In section 5, we show how the sensitivity can be used to give a first order estimate of the variance of the model output; then the estimated variance is used to compute confidence intervals for the output of the model. In section 6, the numerical results are presented and discussed.

## 2 The physical model

In this section, we present the physical model as well as some stability estimates for it and its sensitivity.

### 2.1 The state equations

Let us consider the domain $\Omega$ in Figure 1: it is a channel with walls on the top and the bottom and an obstacle of square section at distance $x_D$ from the inflow boundary. The Navier-Stokes system and the boundary conditions for this domain are:
If the boundary conditions depend on time, see section A for how to compute such a function.

Let \( R_g \) be a sufficiently smooth\(^1\) stationary\(^2\) function such that \( \nabla \cdot R_g = 0 \) in \( \Omega \), \( R_g = u \) on \( \Gamma_{in} \cup \Gamma_{top} \), and \( \nabla R_g n|_{\Gamma_{out}} = 0 \). Then, if \( u \cdot n \geq 0 \) on \( \Gamma_{out} \) and \( \tilde{f} \) is stationary the following stability estimate holds:

\[
\|u\|^2 \leq \|R_g\|^2 + \|\tilde{f}(t)\|^2 t + K(R_g, \tilde{f}) e^{2t|\nabla R_g|_{L^\infty}}, \tag{2}
\]

where \( \tilde{f} = f + \nu \Delta R_g - (R_g \cdot \nabla)R_g \), and the norm \( \| \cdot \| \) is defined as follows:

\[
\|u\|^2 := \|u(T)\|^2 + 2\nu \int_0^T \|\nabla u(t)\|^2 dt.
\]

**Proof.** In order to deal with the boundary terms, we start by homogenising the boundary conditions \( \tilde{u} = u - R_g \). We have: \( \|u\| \leq \|\tilde{u}\| + \|R_g\| \), therefore, since \( R_g \) is regular, controlling \( \tilde{u} \) is equivalent to controlling \( u \).

---

\(^1\)See section A for how to compute such a function.

\(^2\)If the boundary conditions depend on time, \( R_g \) cannot be taken stationary.
The equations for \( \hat{u} \) are:

\[
\begin{align*}
\hat{\partial}_t \hat{u} - \nu \Delta \hat{u} + ((\hat{u} + R_g) \cdot \nabla) \hat{u} + (\hat{u} \cdot \nabla) R_g + \nabla p &= \hat{f}, & \Omega, t > 0, \\
\nabla \cdot \hat{u} &= 0, & \Omega, t > 0, \\
\hat{u}(x, 0) &= -R_g, & \Omega, t = 0, \\
\hat{u} &= 0 \quad \text{on } \Gamma_{in} \cup \Gamma_w, \\
(\nu \nabla \hat{u} - p \mathbf{n}) &= -\nu \nabla R_g \mathbf{n} = 0 \quad \text{on } \Gamma_{out}.
\end{align*}
\]

To obtain the stability estimate, we multiply by \( \hat{u} \) and integrate by parts:

\[
\int_\Omega \hat{\partial}_t \hat{u} \cdot \hat{u} + \nu \int_\Omega \nabla \hat{u} : \nabla \hat{u} + \int_\Omega \left( [(\hat{u} + R_g) \cdot \nabla] \hat{u} \right) \cdot \hat{u} + \int_\Omega \left( [(\hat{u} \cdot \nabla) R_g] \cdot \hat{u} \right) - \int_{\Gamma_{out}} (\nu \nabla \hat{u} - p \mathbf{n}) \cdot \hat{u} = \int_{\Omega} \hat{f} \cdot \hat{u}.
\]

The nonlinear term can be rewritten and integrated by parts as follows:

\[
\int_\Omega \left( [(\hat{u} + R_g) \cdot \nabla] \hat{u} \right) \cdot \hat{u} = \int_\Omega (\hat{u} + R_g) \cdot \nabla \left( \frac{\hat{u}^2}{2} \right) = \int_\Omega \hat{u} \cdot \nabla R_g \cdot \mathbf{n} \frac{\hat{u}^2}{2},
\]

where we used the second equation \( \nabla \cdot (\hat{u} + R_g) = \nabla \cdot u = 0 \). Therefore, using the boundary conditions, one has:

\[
\frac{1}{2} \frac{d}{dt} \| \hat{u} \|^2 + \nu \| \nabla \hat{u} \|^2 + \int_{\Gamma_{out}} (\hat{u} + R_g) \cdot \nabla \hat{u} \cdot \mathbf{n} \frac{\hat{u}^2}{2} + \int_\Omega \left( [(\hat{u} \cdot \nabla) R_g] \cdot \hat{u} \right) = \int_{\Omega} \hat{f} \cdot \hat{u},
\]

where \( \| \cdot \| \) is the \( L^2 \)-norm in space.

We use the hypothesis \( (\hat{u} + R_g) \cdot \mathbf{n}|_{\Gamma_{out}} = u \cdot \mathbf{n}|_{\Gamma_{out}} \geq 0 \) to remove the integral on \( \Gamma_{out} \). Then, since \( R_g \) is regular, we have:

\[
-\int_\Omega \left( [(\hat{u} \cdot \nabla) R_g] \cdot \hat{u} \leq \| \nabla R_g \|_{L^\infty} \| \hat{u} \|^2.
\]

Therefore, we obtain:

\[
\frac{1}{2} \frac{d}{dt} \| \hat{u} \|^2 + \nu \| \nabla \hat{u} \|^2 \leq \| \nabla R_g \|_{L^\infty} \| \hat{u} \|^2 + \| \hat{f} \| \| \hat{u} \|. \tag{3}
\]

The rest of the proof consists of two steps: first, we prove an estimate for \( \| \hat{u} \| \); then we substitute this results into (3) to obtain the estimate (2). To obtain an estimate for \( \| \hat{u} \| \), we start from (3), we remove the positive term \( \nu \| \nabla \hat{u} \|^2 \), develop the time derivative, and simplify \( \| \hat{u} \| \), which is positive, obtaining:

\[
\frac{d}{dt} \| \hat{u} \| \leq \| \nabla R_g \|_{L^\infty} \| \hat{u} \| + \| \hat{f} \|,
\]

which can be rewritten as:

\[
e^{-t\| \nabla R_g \|_{L^\infty}} \left( \frac{d}{dt} \| \hat{u} \| \right) \leq e^{-t\| \nabla R_g \|_{L^\infty}} \| \hat{f} \|.
\]

We remark that the left-hand side is the time derivative of \( e^{-t\| \nabla R_g \|_{L^\infty}} \| \hat{u} \| \). We can integrate and, if \( f \) is constant in time, we obtain:

\[
\| \hat{u} \| \leq \left( \| \hat{u}(x, 0) \| + \frac{\| \hat{f} \|}{\| \nabla R_g \|_{L^\infty}} \right) e^{t\| \nabla R_g \|_{L^\infty}}
\]

\[
= \left( \| R_g \| + \frac{\| \hat{f} \|}{\| \nabla R_g \|_{L^\infty}} \right) e^{t\| \nabla R_g \|_{L^\infty}}, \tag{4}
\]

which concludes the first step. For the second step, we start from (3) and use Young’s inequality, obtaining:

\[
\frac{1}{2} \frac{d}{dt} \| \hat{u} \|^2 + \nu \| \nabla \hat{u} \|^2 \leq \left( \frac{1}{2} + \| \nabla R_g \|_{L^\infty} \right) \| \hat{u} \|^2 + \frac{1}{2} \| \hat{f} \|^2.
\]

4
We multiply by 2 and use the estimate (4), obtaining:

\[
\frac{d}{dt} \| \tilde{u} \|^2 + 2\nu \| \nabla \tilde{u} \|^2 \leq \| \tilde{f} \|^2 + (1 + 2\| \nabla R_y \|_{L^\infty}) \left( \| R_y \|_0 + \frac{\| \tilde{f} \|_{L^\infty}}{\| R_y \|_{L^\infty}} \right)^2 e^{2t\| \nabla R_y \|_{L^\infty}}.
\]

Finally, we can integrate in time, obtaining:

\[
\| \tilde{u} \|^2 + 2\nu \int_0^t |\nabla \tilde{u}(s)|^2 ds \leq \| R_y \|^2 + \| \tilde{f} \|^2 + (1 + 2\| \nabla R_y \|_{L^\infty}) \left( \| R_y \|_0 + \frac{\| \tilde{f} \|_{L^\infty}}{\| R_y \|_{L^\infty}} \right)^2 e^{2t\| \nabla R_y \|_{L^\infty}}.
\]

\[\square\]

### 2.2 The sensitivity equations

We now consider \( u \) as a function of space, time and a scalar uncertain parameter \( a \), \( u = u(x, t; a) \) and we write a formal Taylor expansion with respect to \( a \):

\[
u(x, t; a + \delta a) = \sum_{k=0}^\infty u_k(x, t; a) \delta a^k,
\]

where \( u_0 = u \) and the coefficient \( u_k \) is the \( k \)-th derivative of \( u \) with respect to \( a \):

\[
u(x, t; a) := \frac{d^k}{da^k} u(x, t; a),
\]

and it is called the \( k \)-th order sensitivity. To consider more than one parameter of interest, the sensitivity should be defined as the gradient of the state with respect to the vector of parameters and a multi-dimensional Taylor expansion would be necessary, but this is not treated in this work. A similar expansion can be done for the pressure \( p \), and the data \( f \), \( d \), and \( g \). In order to write the equations for the sensitivities, one can replace (5) into (1) and then factorise according to the powers of \( \delta a \). For \( k = 0 \) we obtain the state system (1). For \( k = 1 \), we obtain the first order sensitivity equations. In this work, we consider only first-order sensitivity and the notation \( u_1 = u_a \) will be employed. This choice is common [BB97, DPB06, DP06, CDF18, FCD19] because in most cases first order sensitivities provide enough information. It is possible to consider higher order sensitivities if necessary, but this is not investigated in this work. The first order sensitivity equations, referred to as the sensitivity equations in short, are:

\[
\begin{align*}
\tilde{\nu} \cdot \mathbf{u}_a - \nu \Delta \mathbf{u}_a + (\mathbf{u}_a \cdot \nabla) \mathbf{u}_a + (\mathbf{u} \cdot \nabla) \mathbf{u}_a + \nabla p_a &= \mathbf{f}_a, & \Omega, t > 0, \\
\nabla \cdot \mathbf{u}_a &= 0 & \Omega, t > 0, \\
\mathbf{u}_a(x, 0) &= 0 & \Omega, t = 0, \\
\mathbf{u}_a &= -\mathbf{g}_a(y) \mathbf{n} & \text{on } \Gamma_{in}, \\
\mathbf{u}_a &= 0 & \text{on } \Gamma_w, \\
(\nu \nabla \mathbf{u}_a - p_a I) \mathbf{n} &= 0 & \text{on } \Gamma_{out},
\end{align*}
\]

where \( \Gamma_w := \Gamma_{obst} \cup \Gamma_{top} \cup \Gamma_{bottom} \). These are known as the Oseen equations: an introduction on the subject can be found in [Vol14], both for the theoretical and the numerical aspects. A similar problem is investigated, although only from a numerical point of view, in [DP05, DPB06], where they use the sensitivity for shape optimization problems: in their case an additional term \( \nu_a \nabla \mathbf{u} \cdot \mathbf{n} \), but this case is not considered in this work.
We want to obtain a stability estimate for the sensitivity, similar to the one for the state (2). To do this, we need the following hypothesis:

\[ 2\kappa = \kappa(u, \Omega) > 0 : \left| \int_{\Omega} [(u_a \cdot \nabla)u] \cdot u_a \right| \leq \kappa \|u_a\|^2. \quad (7) \]

We remark that hypothesis (7) is trivial if the \( L^\infty \)-norm of the gradient \( \nabla u \) is controlled (we would have \( \kappa = \| \nabla u \|_{L^\infty} \)). However, in the time dependent case we have a control only on \( \int_0^t \| \nabla u(s) \|^2 ds \), which is not sufficient. A similar hypothesis (although less restrictive than ours) can be found in [Ray07]: in the section about linearised Navier–Stokes equations around an stationary state, they suppose

\[ u \in L^2(0, T; H^1(\Omega)) \cap L^\infty(0, T; L^4(\Omega)). \]

From the estimate (2) and the triangular inequality we only have

\[ u \in L^2(0, T; H^1(\Omega)) \cap L^\infty(0, T; L^2(\Omega)). \]

Asking for \( \kappa > 0 \) on the boundary terms on \( \Gamma_{in} \), we start by homogenising the boundary conditions, as we did earlier for the state equations. We define a new variable \( \tilde{u}_a = u_a - R_{g_a} \). The new variable \( \tilde{u}_a \) verifies the following equations:

\[
\begin{align*}
\partial_t \tilde{u}_a - \nu \Delta \tilde{u}_a + (\tilde{u}_a \cdot \nabla)u + (u \cdot \nabla)\tilde{u}_a + \nabla p_a &= \tilde{f}_a, & \Omega, t > 0, \\
\nabla \cdot \tilde{u}_a &= 0, & \Omega, t > 0, \\
\tilde{u}_a(x, 0) &= -R_{g_a}, & \Omega, t = 0, \\
\tilde{u}_a &= 0, & \Omega, t > 0, \quad \text{on } \Gamma_{in} \cup \Gamma_w \\
\nu \nabla \tilde{u}_a - p_a) \cdot n &= -\nu \nabla R_{g_a} \cdot n = 0, & \Omega, t > 0, \quad \text{on } \Gamma_{out}.
\end{align*}
\]

We can multiply by \( u_a \) and integrate by parts as done for the state, and we obtain:

\[
\frac{1}{2} \int_0^t \|u_a\|^2 + \nu \|\nabla u_a\|^2 + \int_{\Omega} [(u_a \cdot \nabla)u] \cdot u_a + \int_{\Omega} [(u \cdot \nabla)\tilde{u}_a] \cdot \tilde{u}_a = \int_{\Omega} \tilde{f}_a u_a. \quad (8)
\]

All the integrals on the boundary are zero thanks to the boundary conditions. For the first integral, we use hypothesis (7) and obtain

\[
\int_{\Omega} [(u_a \cdot \nabla)u] \cdot u_a \geq -\kappa \|u_a\|^2. \quad (9)
\]

The second integral can be treated as we did for the nonlinear term of the state, obtaining:

\[
\int_{\Omega} [(u \cdot \nabla)\tilde{u}_a] \cdot \tilde{u}_a = \int_{\Omega} u \cdot \nabla \left( \frac{|\tilde{u}_a|^2}{2} \right) = \int_{\Gamma_{out}} u \cdot n \|\tilde{u}_a\|^2 / 2 \geq 0. \quad (10)
\]

By plugging (9) and (10) into (8), we have:

\[
\frac{1}{2} \int_0^t \|u_a\|^2 + \nu \|\nabla u_a\|^2 \leq \kappa \|u_a\|^2 + \|\tilde{f}_a\| \|\tilde{u}_a\|,
\]

which has exactly the same structure of (3). Therefore, using the same technique, we obtain:

\[
\|u_a\|^2 + 2\nu \int_0^t |\nabla u_a(s)|^2 ds \leq \|R_{g_a}\|^2 + \|\tilde{f}_a\|^2 t + \frac{1 + 2\kappa}{2\kappa} \left( \|R_{g_a}\| + \|\tilde{f}_a\| \right)^2 e^{2\nu t}.
\]

\[ \square \]
3 Numerical schemes

In order to discretise the systems (1) and (6), we use a finite element-volume (FEV) method in space and forward Euler in time. For more details about FEV, cf. [Emo92, Hei03, APFC17, ABF15]. The numerical schemes described below are implemented in the industrial open source code TRUST TrioCFD [DC].

3.1 Spatial discretisation

Here, we recall the main ideas of the FEV method for the state and we adapt it to the sensitivity equations. For this purpose, in this subsection we only consider the spatial part of the systems (1) and (6) (i.e. $\partial_t u = \partial_t u_a = 0$). Let $\Gamma_D = \Gamma_{in} \cup \Gamma_w$ be the Dirichlet boundary and $\Gamma_N = \Gamma_{out}$ the Neumann boundary. Let $T_h$ be a triangulation of the domain $\Omega$ compatible with the boundary conditions, and $K_j \in T_h$ a triangle $(j = 1, \ldots, N_T)$: a triangulation is said to be compatible with the boundary conditions when, if a triangle $K_j$ has a side on the boundary, than that side belongs entirely to $\Gamma_D$ or entirely to $\Gamma_N$. We denote with $x_i$ the nodes $(i = 1, \ldots, N_N)$, which are the middle points of the edges of the triangles. A control volume $\omega_i$ is associated with each node (cf. Figure 2 for the definition of $\omega_i$).

We introduce the following spaces:

$$Q_h = \{ q_h : \forall K \in T_h, \quad q_h|_K \in P_0(K) \},$$

$$V_h = \{ v_h \text{ continuous at } x_i : \forall K \in T_h, \quad v_h|_K \in P_1(K) \},$$

$$V_h = \{ w_h = (w^x, w^y)^t : w^x, w^y \in V_h \} = V_h^2.$$  

The space $Q_h$ is spanned by the indicator functions of the triangles, $\chi_K$, and $V_h$ is spanned by $\varphi_i(x)$, with $\varphi_i \in V_h$ and $\varphi_i(x_j) = \delta_{ij}$. We look for an approximate solution for the systems (1) and (6), respectively $(u_h, p_h) \in V_h \times Q_h$ and $(u_{a,h}, p_{a,h}) \in V_h \times Q_h$. The two components of the discrete velocity field and its sensitivity will be denoted $u_h = (u_h^x, u_h^y)^t$ and $u_{a,h} = (u_{a,h}^x, u_{a,h}^y)^t$.

In order to have a discrete formulation, we integrate the mass equation and its sensitivity over the triangles $K_j$ and the momentum equation and its sensitivity over the control volumes $\omega_i$:

$$\int_{\partial K_j \cap \Gamma_D} u_h \cdot n = \int_{\partial K_j \cap \Gamma_D} g, \quad \forall K_j \in T_h,$$

$$- \int_{\partial \omega_i \cap \Gamma_D} (\nu \nabla u_h - p_h I) n + \int_{\partial \omega_i \cap \Gamma_D} (u_h \otimes u_h) n = \int_{\partial \omega_i} f - \int_{\partial \omega_i \cap \Gamma_{in}} (g n \otimes gn) n, \quad \forall \omega_i,$$

$$\int_{\partial K_j \cap \Gamma_N} u_{a,h} \cdot n = \int_{\partial K_j \cap \Gamma_N} g_n, \quad \forall K_j \in T_h,$$

$$- \int_{\partial \omega_i \cap \Gamma_N} (\nu \nabla u_{a,h} - p_{a,h} I) n + \int_{\partial \omega_i \cap \Gamma_N} (u_{a,h} \otimes u_h + u_h \otimes u_{a,h}) n = \int_{\partial \omega_i} f_n - \int_{\partial \omega_i \cap \Gamma_{in}} (g_n n \otimes gn + gn \otimes g_n n) n, \quad \forall \omega_i. \quad (11)$$

Then, one can expand $(u_h, p_h)$ and $(u_{a,h}, p_{a,h})$ in the bases of the correspondent spaces as follows:

$$u_h(x) = \sum_{i=1}^{N_N} u_h(x_i) \varphi_i(x), \quad p_h(x) = \sum_{j=1}^{N_T} p_h(K_j) \chi_{K_j},$$

$$u_{a,h}(x) = \sum_{i=1}^{N_N} u_{a,h}(x_i) \varphi_i(x), \quad p_{a,h}(x) = \sum_{j=1}^{N_T} p_{a,h}(K_j) \chi_{K_j}. \quad (12)$$

We divide the nodes into two sets: the Dirichlet nodes $\mathcal{D} := \{ i : x_i \in \Gamma_D \}$ and all the other nodes which are referred to as the internal nodes $\mathcal{I} := \{ i : x_i \in \Omega \cup \Gamma_N \}$. By plugging (12)
into (11) for the linear terms and using the identity \((a \otimes b) c = a(b \cdot c)\) for the nonlinear terms, we have:

\[
\begin{align*}
\sum_{i \in I} u_i(x_i) \cdot \int_{\partial K_j \cap \Gamma_D} \varphi_i \cdot n = \int_{\partial K_j \cap \Gamma_{in}} g \\
- \sum_{j \in I} \int_{\partial \omega_j \cap \Gamma_N} (\nu u_{b,h}(x_j) + \nabla \varphi_j) \cdot n + \sum_{j=1}^{N_T} p_{h}(K_j) \int_{\partial \omega_j} \chi_{K_j} \cdot n \\
+ \int_{\partial \omega_j \cap \Gamma_{in}} u_b(u_b \cdot n) = \int_{\omega_j} f - \sum_{j=1}^{N_T} \int_{\partial \omega_j \cap \Gamma_{in}} g^2 n \end{align*}
\]

\(\forall K_j \in \mathcal{T}_h, \quad \forall \omega_j, \quad \forall \omega_l, \)  

(13)

We can define the vectors of unknowns as follows:

\[
U_h = \left[ u_h^x(x_i) \right]_{i \in I}, \quad V_h = \left[ u_h^y(x_i) \right]_{i \in I}, \quad P_h = \left[ p_h(K_j) \right]_{j=1, \ldots, N_T}, \quad U_h = \left[ U_h \right]_{V_h} = \left[ U_h \right]_{V_h},
\]

\[
U_{a,h} = \left[ u_{a,h}(x_i) \right]_{i \in I}, \quad V_{a,h} = \left[ v_{a,h}(x_i) \right]_{i \in I}, \quad P_{a,h} = \left[ p_{a,h}(K_j) \right]_{j=1, \ldots, N_T}, \quad U_{a,h} = \left[ U_{a,h} \right]_{V_{a,h}}.
\]

We introduce the following vectors:

\[
F^i = \int_{\omega_i \cap \Gamma_{in}} g^2 n^i, \quad D_i = \int_{\partial K_j \cap \Gamma_{in}} g_i,
\]

\[
F^{a,i} = \int_{\omega_i \cap \Gamma_{in}} g a n^i, \quad D_{a,i} = \int_{\partial K_j \cap \Gamma_{in}} g a_i,
\]

and the following matrices:

\[
\tilde{A}_{i,j} := -\int_{\partial \omega_j \cap \Gamma_N} \nu \nabla \varphi_j \cdot n, \quad B^{x}_{i,j} := \int_{\partial K_j \cap \Gamma_D} \varphi_i n^x, \quad C^{x}_{i,j} := \int_{\partial \omega_j \cap \Gamma_N} \chi_{K_j} n^x
\]

where the superscript \(i = x, y\) indicates the \(x\)- and \(y\)- components of the normal vector \(n = (n^x, n^y)\), of the state source term \(f = (f^x, f^y)\), and of the sensitivity source term \(f_a = (f_{a,x}, f_{a,y})\).

For the convection terms, an upwind-type approach is applied: for each edge of \(\partial \omega_j\), \(u_b \cdot n\) is computed and, according to its sign, it is multiplied by either \(u_b(x_i)\) or \(u_b(x_h)\), where \(\omega_j\) is the control volume adjacent to the considered edge. We remark that using an upwind-type scheme leads to a CFL condition \([\text{CFL67}]\) on the time step \(\Delta t\) when considering the time discretisation: this CFL condition is the same for the state and the sensitivity equations because the transport speed in both equations is \(u\).

For the sake of having a more compact and readable notation, we define \(x_a\) as follows:

\[
x_a = \begin{cases} 
x_i & \text{if } u_b \cdot n_i > 0 \text{ on } \partial \omega_i \cap \partial \omega_k, \\
x_h & \text{otherwise},
\end{cases}
\]

which truly depends on \(x_i\), \(x_h\) and \(u_b\), but we use a simplified notation. The context will make it non ambiguous. In the same way, we can define \(x_{a,b}\) according to the sign of \(u_{a,h} \cdot n\).

Therefore, for the convection term in the state equation, we can write:

\[
\int_{\partial \omega_i} u_b(u_b \cdot n) \approx \int_{\partial \omega_i} u_b(x_a) (u_b \cdot n) = \int_{\partial \omega_i} u_b(x_a) \left( \sum_{j \in I} u_b(x_j) \varphi_j \cdot n \right).
\]
We consider the first component of the vector here above:

\[
\int_{\mathcal{Q}_a} u_h^x(x_j) \left( \sum_{j \in \mathcal{J}} u_h(x_j) \varphi_j \cdot n \right) = \\
= \sum_{j \in \mathcal{J}} u_h^x(x_j) \int_{\mathcal{Q}_a} u_h(x_j) \varphi_j n^x + \sum_{j \in \mathcal{J}} u_h^y(x_j) \int_{\mathcal{Q}_a} u_h(x_j) \varphi_j n^y.
\]

We can define the following matrices:

\[
L^x_{i,j}(U_h) = \int_{\mathcal{Q}_a} u_h^x(x_j) \varphi_j n^x, \quad L^y_{i,j}(U_h) = \int_{\mathcal{Q}_a} u_h^y(x_j) \varphi_j n^y.
\]

The second component would give as a result \(L^x_{i,j}(V_h)\) and \(L^y_{i,j}(V_h)\). Concerning the sensitivity, doing the same computations would lead to eight matrices, four of them identical to the ones introduced for the state, and four others as follows:

\[
L^a_{i,j}(U_{a,h}) = \int_{\mathcal{Q}_a} u_{a,h}^x(x_j) \varphi_j n^x, \quad L^b_{i,j}(U_{b,h}) = \int_{\mathcal{Q}_a} u_{b,h}^x(x_j) \varphi_j n^x
\]

\[
L^a_{i,j}(V_{a,h}) = \int_{\mathcal{Q}_a} u_{a,h}^x(x_j) \varphi_j n^y, \quad L^b_{i,j}(V_{b,h}) = \int_{\mathcal{Q}_a} u_{b,h}^x(x_j) \varphi_j n^y
\]

Finally, by introducing the following notation:

\[
A = \begin{pmatrix} \hat{A} & 0 \\ 0 & A \end{pmatrix}, \quad B = \begin{pmatrix} B^x & B^y \end{pmatrix}, \quad C = \begin{pmatrix} C^x \cr C^y \end{pmatrix}, \quad F = \begin{pmatrix} F^x \\ F^y \end{pmatrix}, \quad F_a = \begin{pmatrix} F_a^x \\ F_a^y \end{pmatrix},
\]

\[
L(U_h) = \begin{pmatrix} L^x(U_h) & L^y(U_h) \\ L^x(V_h) & L^y(V_h) \end{pmatrix}, \quad L_a(U_{a,h}) = \begin{pmatrix} L_a^x(U_{a,h}) & L_a^b(U_{a,h}) \\ L_a^a(U_{a,h}) & L_a^b(U_{a,h}) \end{pmatrix}
\]

and observing that \(C = B^T\), the discrete system can be written in the compact form:

\[
\begin{align*}
AU_h + B^T P_h + L(U_h) U_h &= F, \\
BU_h &= D, \\
AU_{a,h} + B^T P_{a,h} + L_a(U_{a,h}) U_h + L(U_h) U_{a,h} &= F_a, \\
BU_{a,h} &= D_a.
\end{align*}
\]

The first two equations correspond to the state system, and are independent of the last two: the first one is the discretisation of the momentum equation and the second one is the discretisation of the mass equation \(\nabla \cdot \mathbf{u} = 0\), where the vector \(D\) accounts for the Dirichlet boundary conditions. The last two equations correspond to the sensitivity and, in particular, the first of these depends on the solution of the state through the convective matrix \(L(U_h)\).

### 3.2 Time discretisation

For the time scheme, we use a forward Euler discretisation [QSS10] with an implicit treatment of the diffusion operator: an explicit treatment would require a very restrictive time step for stability. Let \(t^n\) be the \(n\)-th time step, which is computed according to the CFL condition [CFL67], and let \(U_h^n\) (respectively \(U_{a,h}^n\)) be an approximation of \(\mathbf{u}(x,t^n)\) (respectively, \(\mathbf{u}_a(x,t^n)\)). By coupling this with the spatial scheme described in the previous subsection, we obtain the following system:

\[
\begin{align*}
\frac{U_h^{n+1} - U_h^n}{\Delta t} + AU_h^{n+1} + B^T P_h^{n+1} + L(U_h^n) U_h^n &= F^n, \\
BU_h^{n+1} &= D^n, \\
\frac{U_{a,h}^{n+1} - U_{a,h}^n}{\Delta t} + AU_{a,h}^{n+1} + B^T P_{a,h}^{n+1} + L_a(U_{a,h}^n) U_h^n + L(U_h^n) U_{a,h}^n &= F_a^n, \\
BU_{a,h}^{n+1} &= D_a^n.
\end{align*}
\]
Figure 2: The control volume $\omega_i$ associated with the node $x_i$ is shaded in blue. $C_j$ and $C_\ell$ are the centres of gravity of the two triangles $K_j$ and $K_\ell$ adjacent to the edge on which $x_i$ lies.

where $M$ is the mass matrix with coefficients $M_{i,j} = \int_{\omega_i} \varphi_j$. To improve the efficiency in the computations, $M$ is lumped, i.e. it is approximated with a diagonal matrix whose coefficients are $|\omega_i| \delta_{i,j}$. The system (14) is solved with a prediction correction procedure [Cho68, Tem68, APFC17]. First, an intermediate speed $U^\ast_{h}$ and its sensitivity $U^\ast_{a,h}$ (which are not solenoidal) are computed by solving the following systems:

\begin{align*}
(I + \Delta t M^{-1} A) U^n_h &= U^n_h - \Delta t M^{-1} (L(U^n_h) U^n_h + B^\prime U^n_h - F^n), \\
(I + \Delta t M^{-1} A) U^n_{a,h} &= U^n_{a,h} - \Delta t M^{-1} (L_a(U^n_{a,h}) U^n_a + L(U^n_h) U^n_{a,h} + B^\prime P^n_{a,h} - F^n).
\end{align*}

Then, we perform the correction step:

\begin{align*}
U^{n+1}_h &= U^n_h + \Delta t M^{-1} B^\prime (P^{n+1}_h - P^n_h), \\
U^{n+1}_{a,h} &= U^n_{a,h} + \Delta t M^{-1} B^\prime (P^{n+1}_{a,h} - P^n_{a,h}).
\end{align*}

(15)

Finally, by left multiplying by $B$ and imposing the second and fourth equation of (14), one obtains:

\begin{align*}
BM^{-1} B^\prime P^{n+1}_h &= BM^{-1} B^\prime P^n_h + \frac{D^n_{h+1} - BU^n_h}{\Delta t}, \\
BM^{-1} B^\prime P^{n+1}_{a,h} &= BM^{-1} B^\prime P^n_{a,h} + \frac{D^n_{a,h+1} - BU^n_{a,h}}{\Delta t},
\end{align*}

(16)

and $P^{n+1}_h$ and $P^{n+1}_{a,h}$ can be used to compute $U^{n+1}_h$ and $U^{n+1}_{a,h}$ from (15). We remark that in order to solve the pressure problem (16), some technical discrete boundary conditions for the pressure are needed. To do this, in the code TrioCFD [DC] the Neumann boundary condition is split into two parts: the pressure and the gradient of the velocity are set to zero independently of each other.

4 Validation of the implementation of the sensitivity equation

In order to validate the sensitivity equation method, we consider the following Taylor expansion:

$$u(x, T; a + \delta a) = u(x, T; a) + \delta a u_a(x, T; a) + O(\delta a^2),$$

10
and we define the following quantity:

$$\text{err}(u) = u(x,T;\delta a) - u(x,T;a) - \delta a u_\delta(x,T;a).$$

If the sensitivity equation are correctly solved, one should have $\text{err}(u) = O(\delta a^2)$. We considered a straightforward test case: a channel as the one in Figure 1 but without the obstacle and with a parabolic velocity profile imposed at the inlet:

$$g(y) = \frac{4A}{\ell^2}g(\ell - y)$$

The maximal value of the velocity at the inlet is the parameter of interest $a$ (i.e. $a = A$). We solved this on two Cartesian grids with $h = 0.001$ and $h = 0.002$. In Figure 3, we show the $L^2$ and $L^\infty$ norms for the two components of the velocity on the two different meshes: for larger $\delta a$ we observe the expected convergence rate, but starting at $\delta a \geq 0.05$ the curve flattens (especially the ones corresponding to the horizontal component of the velocity). This is due to the fact that the quantity shown is $\text{err}(u_\delta)$ and not $\text{err}(u)$: the spatial discretisation is an additional source of error, and this part of the error is constant for a given mesh, therefore the curve tend to a plateau. The plateau value is smaller for finer meshes, as one can observe in Figure 3.

5 Uncertainty propagation

In this section, we want to show how the sensitivity can be used to give a first order estimate of the variance of the model output. In this context, the parameter $a$ is a random variable with a known distribution, expected value $\mu_a$, and variance $\sigma_a^2$. Let $X(x,t;a)$ be a physical variable (i.e. the horizontal or vertical velocity, or the pressure), whose expected value $\mu_X$ and variance $\sigma_X^2$ we want to estimate. To do this, we start from a Taylor expansion of $X$ with respect to the parameter $a$ centred in the expected value of $a$, $\mu_a$:

$$X(x,t;a) = X(x,t;\mu_a) + (a - \mu_a)X_\delta(x,t;a) + o(|a - \mu_a|^2),$$

(17)

where $X_\delta = \partial_a X$ is the sensitivity of $X$ with respect to the parameter $a$. Computing the expected value of the right and left-hand side of (17), one obtains the following first order estimate:

$$\mu_X(x,t) = E[X(x,t;\mu_a)] = X(x,t;\mu_a) + E[(a - \mu_a)]X_\delta(x,t;a) = X(x,t;\mu_a).$$

(18)
Using again the Taylor expansion (17) and the estimate just obtained for the average (18), we obtain an estimate of the variance:

$$\sigma_x^2(x,t) = E[(X(x,t) - \mu_x(x,t))^2] \approx E[(a - \mu_a)^2]X_*^2(x,t; a) = \sigma_a^2 X_*^2(x,t; a).$$  \hfill (19)

The estimates (18)-(19) are valid only where the Taylor expansion (17) holds, i.e. for small variances of the random parameter \(\sigma_a^2\). However, one can have an estimate of the variance with just one simulation of the state and one of the sensitivity, which is a minimal computational cost when compared to methods such as Monte Carlo that require thousands of simulations of the state to estimate the variance. In the general case, when more than one parameter is uncertain, to provide an estimate of the variance, one would need one simulation for the state and as many simulations of the sensitivity as the number of uncertain parameters [Fio18]. This makes the sensitivity approach really affordable and highly competitive when the number of uncertain parameters is small enough. In the next subsection, we compare the results of the SA approach with the well-known Monte Carlo method [RC13].

The estimated variance can be used for multiple purposes. In this work, we use it to compute some confidence intervals for the physical variables, i.e. find an interval \(\text{CI}_X\) such that the probability that \(X\) falls into \(\text{CI}_X\) is bigger than 1 - \(\alpha\). Standard choices for \(\alpha\) are 0.05 or 0.01. If the distribution of the random variable \(X\) is known, some precise estimates for the extrema of the interval exist. However, SA does not provide any insight of what the distribution of the output is. Therefore we start from Chebyshev inequality [JP12], which states that for any random variable with finite expected value and variance

$$P(|X - \mu_X| \geq \lambda) \leq \frac{\sigma_X^2}{\lambda^2}.$$  \hfill (20)

By imposing the desired level for confidence interval, i.e. \(\alpha = \frac{\sigma_X^2}{\lambda^2}\), one gets \(\lambda = \frac{\sigma_X}{\sqrt{\alpha}}\), therefore

$$\text{CI}_X = \left[\mu_X - \frac{\sigma_X}{\sqrt{\alpha}}, \mu_X + \frac{\sigma_X}{\sqrt{\alpha}}\right].$$  \hfill (20)

### 6 Numerical results

The domain used for the numerical simulations is the one in Figure 1, and the values of the parameters are the ones in Table 1. The equations are solved on the mesh with a spatial step varying from 0.005 to 0.01. Finally, we consider the following parabolic inflow condition:

$$g(y) = \frac{4A}{\ell^2} y (y - \ell),$$  \hfill (21)

where \(A\) is the maximal value of the inflow velocity, and it is the uncertain parameter (i.e. \(a = A\)). \(A\) is a Gaussian random variable of average \(\mu_A\) and variance \(\sigma_A^2\).

#### 6.1 Steady test case

For this first test case, we consider a small inflow velocity, \(\mu_A = 0.25\), which corresponds to \(Re = 25\) and leads to a stationary solution. The probability density function of \(A\) is shown in Figure 4: the standard deviation \(\sigma_A = 7.5 \times 10^{-3}\) is small enough to apply the sensitivity method described above to compute the variance of the output.

In Figure 5, we show the numerical results for the horizontal velocity and its sensitivity. For this test case, we were able to make a Monte Carlo approach as well: 1300 simulations
Figure 4: Probability density function of the uncertain parameter $A$.

Figure 5: Results for $Re = 25$. 
of the state were necessary. In Figures 6-7, we compare the variance estimated using the sensitivities and the one computed with Monte Carlo on two cross sections of the domain. As one can see, the two strategies give very similar results. In Figures 8-9, we show the confidence intervals computed according to (20) for $\alpha = 0.05$: in blue the confidence intervals are obtained with the average and variance estimated with SA, in red with Monte Carlo.

For this test case, the first order approximations provided by the SA are more than satisfactory: with only two simulations, we obtain results comparable to the ones obtained with the Monte Carlo approach, which requires 1300 simulations.

### 6.2 Unsteady test case

We now consider a case with a higher Reynolds number ($Re = 100$, which corresponds to $\mu_A = 1$); thus, we are in the range in which a Von Karman vortex street occurs. In this situation, it is reasonable to assume that the velocity behaves as follows:

$$\mathbf{u}(\mathbf{x}, t; a) \approx \sum_{k=0}^{N} \mathbf{u}_{0,k}(\mathbf{x}; a) \cos(\omega_k(a) t).$$

(22)
Figure 8: Confidence intervals on the horizontal cross section $y = 0.2$, with $\alpha = 0.05$. Comparison between Monte Carlo and the SA approaches.

Figure 9: Confidence intervals on the vertical cross section $x = 1$, with $\alpha = 0.05$. Comparison between Monte Carlo and the SA approaches.
We remark that (22) does not solve (1) exactly: some terms have been neglected. Although, from physical knowledge of the phenomenon and looking at numerical results downstream to the obstacle, we can say that the hypothesis (22) is reasonable and that \( N \) can be as small as 2 or 3.

To simulate this test case, we use the following procedure: first, we simulate the state until it reaches a periodic solution of the form (22); then, this state is injected into the sensitivity equations and the sensitivity simulation is run.

We remark that by differentiating (22) with respect to the parameter \( a \), one obtains the following behaviour for the sensitivity:

\[
\mathbf{u}(x, t; a) = \sum_{k=0}^{N} \mathbf{u}_{0,k}(x; a) \cos(\omega_k(a)t) - t \sum_{k=0}^{N} \mathbf{u}_{0,k}(x; a) \omega'_k(a) \sin(\omega_k(a)t).
\]

We can distinguish two contributions: the first sum corresponds to the influence that the input parameter \( a \) has on the amplitude of the oscillations \( \mathbf{u}_{0,k} \), the second sum corresponds to the influence on the frequency \( \omega \). The second effect leads to a sensitivity that grows linearly in time and difficult to work with. This problem is described in [HEPB04] for a similar test case: to deal with it, they chose a pulsating inflow velocity, which imposes the frequency of vortex shedding (i.e. \( \omega_k \) does not depend on \( a \)). However, this choice is not suitable for all applications. This is why, in the next subsection, we propose a filter to recover the first part of the sensitivity (23).

In Figures 10-11, we show the state and sensitivity in different points of the domain: as one can see, hypotheses (22) and (23) are verified.
6.2.1 Filter

In this subsection, we propose a filter to recover the bounded part of the sensitivity

\[ h(x, t; a) := \sum_{k=0}^{N} u_{0,a,k}(x; a) \cos(\omega_k(a)t). \]

To the best of our knowledge, this is the first work in which a filtering procedure is proposed in this context. We focus on this part of the sensitivity because, for industrial applications, the variance of the amplitude of the oscillations is more interesting than the variance of their phase.

We suppose that the ratio between the \( \omega_k \) is rational, i.e. that the state (22) and \( h \) are periodic in time. By computing the discrete Fourier transform of the state we can see that not only this is true, but also that we have a fundamental frequency \( \omega \) and then multiples of that, \( 2\omega, 3\omega \). In Figure 12, only the second harmonic \( 2\omega \) is visible. One can identify the period \( T \) of the state using the Fourier transform, and therefore the following equality holds:

\[ \int_{t-T}^{t} \frac{u_a(x, s; a)}{s} \, ds = \int_{t-T}^{t} \frac{h(x, s; a)}{s} \, ds =: \mathcal{I}(t). \]

(24)

The left-hand side of (24) can be computed numerically. To obtain \( h(x, t; a) \), one can compute the derivative of \( \mathcal{I}(t) \) and use the fact that \( h \) is \( T \)-periodic in time:

\[ \frac{d}{dt} h(x, t; a) = \frac{h(x, t; a)}{t} - \frac{h(x, t; a)}{T - t}, \]

(25)

and finally one has

\[ h(x, t; a) = \frac{t(T - t) \frac{d}{dt} \mathcal{I}}{T}. \]

The derivative (25) can be computed numerically by finite differences. In Figure 13, we applied the filter just described to a case where \( u_a \) and \( h \) are known analytically: as one can see in the left Figure, the filter precision degrades with time. However, since we know that \( h \) is periodic, one can consider the first period and repeat it, which is what is shown in the right Figure. Of course, this leads to a small jump where the two periods are glued together.
We remark that this filter could be build on multiple periods as well. However, some considerations on the computational costs are necessary: in order to filter on $N$ periods, one has to run the simulation for at least $N+1$ periods. We tested the filter on multiple periods and the differences observed do not justify the additional computational cost for this test case. The following results are obtained using a one-period filter.

In Figure 14, we show the results of the filter applied to the sensitivities shown previously (Figures 10-11). The sensitivities obtained in this way can now be used for different applications such as, for instance, the computation of a confidence interval as done in the previous section (cf. Figure 15). However, one must be careful, as the filtered sensitivities take into account only the variance of the amplitude of the oscillations and not the variance of the frequency.

7 Conclusion

In this paper, we propose an efficient computational strategy to deal with problems of uncertainty propagation for the Navier–Stokes equations. First, the state equations are presented, and the sensitivity equations are derived according to the continuous sensitivity equation method. Stability estimates in $L^r(0, T; L^2(\Omega)) \cap L^2(0, T; H^1(\Omega))$ for both the state and the sensitivity are rigorously proven. We homogenise the Dirichlet inflow boundary conditions using an auxiliary function which is explicitly computed. A FEV numerical scheme is designed for the sensitivity and implemented in the open-source code TrioCFD. A classical test case of flow past a square-section cylinder is investigated in two different regimes: a steady case ($Re = 25$) and a Karman vortex street ($Re = 100$). For both cases, the sensitivity is used to estimate the variance of the velocity field, and 95% confidence intervals are computed. In the steady case, a detailed comparison with a Monte Carlo method is performed: the results of the sensitivity based method are extremely accurate, and the computational gain is significant. The unsteady test case raises an issue: due to the periodic nature of the velocity, the sensitivity grows in time and is therefore unusable. We propose a new procedure, which consists in filtering the different contributions to the sensitivity in order to recover its physical parts. The filtered sensitivity is then used to compute the confidence intervals.

Questions such as the investigation of different parameters of interest, 3–dimensional computations, the addition of the energy equation to deal with the temperature and the extension to the unsteady Reynolds-Averaged Navier–Stokes (URANS) equations for the study of turbulent flows will be addressed in future works, to tackle more realistic problems.
Figure 14: Filter applied to the sensitivity of the velocity in three different points of the domain. Top row: horizontal component. Bottom row: vertical component.

Figure 15: Confidence intervals ($\alpha = 0.05$) for the velocity in three different points of the domain. Top row: horizontal component. Bottom row: vertical component.
Appendix

A Computation of the auxiliary function $R_y(x, y)$

We need a vector field $R_y(x, y) = (R^x_y(x, y), R^y_y(x, y))^t$ which is solenoidal, smooth enough (at least continuous with $\nabla R_y \in L^\infty(\Omega)$) and equal to $u$ on the Dirichlet boundary, i.e. on $\Gamma_{in} \cup \Gamma_{in}$. The same can be done for $u_o$, in particular, for the inflow boundary condition used in this work (21).

First of all, one needs to define a function $g_F(y)$ with the following properties:

$$g_F(0) = g_F(\ell) = 0,$$

$$g_F(y) = 0 \ \forall y \in \left[\frac{\ell - d}{2}, \frac{\ell + d}{2}\right].$$

(iii) The field equals $u$ on the obstacle.

Proposition 3. The following vector field is solenoidal, continuous and equals $u$ at the boundary:

$$R^x_y(x, y) = g(y)\mathcal{H}(x) + g_F(y)(1 - \mathcal{H}(x)) \quad \text{and} \quad R^y_y(x, y) = -\mathcal{H}'(x)(G(y) - G_F(y)).$$

Furthermore, $\nabla R_y \in L^\infty(\Omega)$ and $\nabla R_y \cdot n|_{\Gamma_{out}} = 0$.

Proof. (i) The field is solenoidal:

$$\nabla \cdot R_y = \partial_x R^x_y + \partial_y R^y_y = g(y)\mathcal{H}'(x) - g_F\mathcal{H}'(x) - \mathcal{H}'(x)(g(y) - g_F(y)) = 0.$$

(ii) The field is continuous because each component is a product and sum of continuous functions.

(iii) The field equals $u$ on $\Gamma_D$.

(a) The $x$ component of $u$ is equal to $g(y)$ on $\Gamma_{in}$ and zero on $\Gamma_D \setminus \Gamma_{in}$. On $\Gamma_{in}$, we have:

$$R^x_y(0, y) = g(y)\mathcal{H}(0) + g_F(y)(1 - \mathcal{H}(0)) = g(y),$$

because $\mathcal{H}(0) = 1$. On $\Gamma_{top}$ and $\Gamma_{bottom}$,

$$R^x_y(\ell, y) = R^x_y(0, y) = 0,$$

because $g(0) = g(\ell) = 0$ and $g_F(0) = g_F(\ell) = 0$. Finally, on the obstacle we have

$$R^x_y(x, y) = g_F(y),$$

which is zero $\forall y \in \left[\frac{\ell - d}{2}, \frac{\ell + d}{2}\right]$ thanks to property (26).
Finally, we chose the following function for \( G \):

\[
G(y) = \frac{\ell}{2} y^2 - \frac{1}{3} y^3.
\]

Then, we can define the function \( g_F \) as follows:

\[
g_F(y) = \begin{cases} 
ky \left( \frac{\ell}{2} - y \right) & y \in [0, \frac{\ell}{2}), \\
0 & y \in \left[ \frac{\ell}{2}, \frac{\ell}{4} \right], \\
k\ell - y \left( y - \frac{\ell}{4} \right) & y \in \left( \frac{\ell}{4}, \ell \right] .
\end{cases}
\]

where \( k \) is chosen in order to respect the property (27) and it is \( k = \frac{4\ell^3}{(\ell - d)^2} \). The primitive \( G_F \) is:

\[
G_F(y) = \begin{cases} 
k\frac{\ell}{12} y^3 - k \frac{\ell}{4} y^2 & y \in [0, \frac{\ell}{2}), \\
\frac{\ell}{12} y^3 - k \left( \frac{\ell}{12} y^3 + \frac{1}{4} \frac{\ell}{4} y^2 + \frac{1}{4} (\frac{\ell}{4} + d) y \right) - k \left( \frac{\ell}{4} y \right)^2 \left( \frac{4 - \ell}{12} \right) & y \in \left[ \frac{\ell}{4}, \frac{\ell}{2} \right], \\
\ell - k \left( \frac{\ell}{4} \right)^3 & y \in \left( \frac{\ell}{4}, \ell \right].
\end{cases}
\]

Finally, we chose the following function for \( \mathcal{H}(x) \):

\[
\mathcal{H}(x) = \begin{cases} 
2 \left( \frac{x}{x_D} \right)^3 - 3 \left( \frac{x}{x_D} \right)^2 + 1 & x \in [0, x_D], \\
0 & x \in (x_D, L],
\end{cases}
\]

which has all the required properties. The auxiliary function just computed is shown in Figure 16. Figure 17 shows the two components of \( \mathbf{R}, R_y^s \) and \( R_y^s \).
Figure 16: Solenoidal field from the example.

Figure 17: Solenoidal field from the example, $x$ and $y$ components.
References


[DC] DEN-CEA. TRUST TrioCFD code.


