multiobjective optimization in hydrodynamic stability control

Frank Strauss, Jean-Antoine Desideri, Régis Duvigneau, Vincent Heuveline

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Multiobjective optimization in hydrodynamic stability control

Frank Strauß, Jean-Antoine Désidéri, Régis Duvigneau and Vincent Heuveline

N° 6608

Juillet 2008

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6608--FR+ENG

Multiobjective optimization in hydrodynamic stability control

Frank Strauß*, Jean-Antoine Désidéri†, Régis Duvigneau‡ and Vincent Heuveline§

Thème NUM — Systèmes numériques
Projet OPALE

Rapport de recherche n° 6608 — Juillet 2008 — 17 pages

Abstract: We focus on a shape optimization problem in hydrodynamic stability control. The goal is to minimize the drag in a flow regime by modification of the underlying geometry while ensuring hydrodynamic stability. We propose the extension of a numerical scheme to speed up the computations and to avoid the calculation of derivatives of the original PDE model within the optimization process. Therefore, we introduce a metamodel as approximation of the original model. A derivative-free particle swarm optimization algorithm is used to replace a gradient-based method. Moreover, in a concurrent optimization approach, the criteria of drag and stability are considered simultaneously. This is realized by the simulation of a Nash game associated with a split of design variables established from a sensitivity analysis accounting for first and second derivatives of functionals related to the metamodel.

Key-words: Optimum shape design, hydrodynamic stability, metamodel, particle swarm optimization, Nash game

* Karlsruhe Institute of Technology
† Project Team Opale
‡ Project Team Opale
§ Karlsruhe Institute of Technology

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Optimisation multiobjectif en contrôle de stabilité hydrodynamique

Résumé : On s'intéresse à un problème d'optimisation de forme en contrôle de stabilité hydrodynamique. L'objectif est de minimiser la traînée associée à un écoulement en modifiant la géométrie tout en préservant la stabilité hydrodynamique. On propose l'extension d'un schéma numérique pour en augmenter l'efficacité en coût et éviter le calcul de gradient du modèle EDP dans l'optimiseur. Pour cela, on introduit un métamodèle pour approcher le modèle d'origine. Par ailleurs, dans une approche d'optimisation concourante, on traite simultanément un critère de traînée et un critère de stabilité. Le compromis est réalisé par l'établissement d'un équilibre de Nash associé à un partage des variables de conception construit à partir d'une analyse de sensibilité s'appuyant sur des dérivées premières et secondes de fonctionnelles associées au métamodèle.

Mots-clés : Conception optimale de forme, stabilité hydrodynamique, métamodèle, optimisation par essaim de particules, jeu de Nash

Contents

1 Introduction

In some recent work [7] we addressed the question how to minimize the drag for the flow around a body in a channel by means of shape optimization while ensuring hydrodynamic stability. The governing equations are the stationary Navier-Stokes equations and the stability issues are described in terms of linear stability by the eigenvalues of the linearized Navier-Stokes operator. The shape of the body is parameterized and can be changed by means of a discrete number of design variables. Both for the Navier-Stokes equations and for the associated eigenvalue problem the solution is based on the finite element method. To solve the shape optimization problem a gradient-based method such as SQP is considered. This solution process faces certain difficulties. The numerical scheme based on the PDE model requires heavy computations since function and gradient computations are very expensive. For the eigenvalue function derivatives even may not exist in every point. In addition, relying on gradient-based optimization algorithms the iteration process might get stuck in local minima without finding the global optimum. Moreover, the formulation which was used so far focuses on the drag minimization while treating the important issue of the stability only as constraint. However, it would be interesting to see which best stability value could be achieved. The research presented here shows possibilities to modify and extend some parts of the numerical scheme, in order to address these problems.
To reduce the computational time we introduce a metamodel as approximation to the original PDE model. This metamodel is constructed using radial basis functions. For a sufficiently accurate metamodel, the evaluation of the PDE model can be replaced by the evaluation of the metamodel. The latter calculations can be performed very quickly. In that context an analytical evaluation of the gradients of the metamodel can also be used to replace the gradients of the original model. Furthermore, a gradient-free optimization technique, the

particle swarm optimization algorithm, is considered to avoid problems due to nondifferentiability. Moreover, such a method enables to promote the search in regions beyond the local minimum found by a gradient-based algorithm. Ideally, a global optimum in the design optimization process would be found. To accelerate convergence a switch to a gradient-based method after a certain number of iterations could be considered. Furthermore, techniques of multiobjective optimization are studied where the stability function is also considered as an objective function. We follow an approach which simulates a Nash game between two players as proposed in [3]. The goal is to determine an appropriate Nash equilibrium point. A continuity of Nash equilibrium points between the two criteria can also be computed.

2 Hydrodynamic stability control

2.1 Physical model and linear stability

At first, we briefly want to introduce the setting of our problem in hydrodynamic stability control (see also [2]). As an example we study the situation of a flow around a body in a channel, where the body is assumed to be a cylinder in the initial configuration. The benchmark configuration is depicted in Fig. 1.

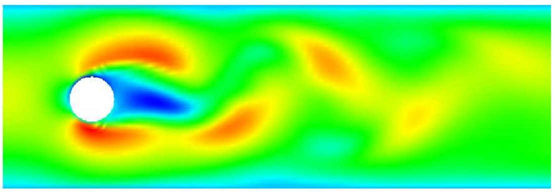

Figure 1: Benchmark flow around body in channel.

In a bounded domain $\Omega_q \subset \mathbb{R}^2$ we consider a base flow $\hat{u} := \{\hat{v}, \hat{p}\}$, determined by the stationary Navier-Stokes equations describing viscous, incompressible Newtonian fluid flow,

$$\begin{aligned} -\nu\Delta\hat{v} + \hat{v}\cdot\nabla\hat{v} + \nabla\hat{p} &= f \qquad \text{in } \Omega_q, \\ \nabla\cdot\hat{v} &= 0 \qquad \text{in } \Omega_q, \end{aligned} \tag{1}$$

where $\hat{v}$ describes the velocity vector field, $\hat{p}$ denotes the pressure, ν is the kinematic viscosity and f is a prescribed volume force. For ease of presentation we assume here that the density $\rho \equiv 1$. The subscript q in the notation Ω_q describes the parameterization of the underlying computational domain. We assume q to be a finite-dimensional vector. Furthermore, problem (1) is assumed to have a unique solution and that for a solution operator S the relation $\hat{u} = S(q)$ holds. At the boundary $\partial\Omega_q$, the usual non-slip boundary conditions are imposed along rigid parts together with suitable inflow and free-stream outflow conditions,

$$\hat{v}|_{\Gamma_{\text{rigid}}} = 0, \quad \hat{v}|_{\Gamma_{\text{in}}} = \hat{v}_{\text{in}}, \quad \nu\partial_n\hat{v} - \hat{p}n|_{\Gamma_{\text{out}}} = 0. \tag{2}$$

In our framework we consider the hydrodynamic stability by means of linear stability. This method relies on the solution of the eigenvalue problem related to the linearization of (1) about $\hat{u}$.

$$\begin{aligned} A(\hat{u})(u) = -\nu\Delta v + \hat{v}\cdot\nabla v + v\cdot\nabla\hat{v} + \nabla p &= \lambda v \qquad \text{in } \Omega_q, \\ \nabla\cdot v &= 0 \qquad \text{in } \Omega_q, \end{aligned} \tag{3}$$

for nonzero $u := \{v, p\} \in U$ and $\lambda \in \mathbb{C}$, under homogeneous boundary conditions (2). Here, $U \subset H^1(\Omega_q)^d \times L^2(\Omega_q)$ is a suitable subspace according to the prescribed boundary conditions (2). Obviously, this eigenvalue problem is non-symmetric and may possess complex eigenvalues. If an eigenvalue of (3) has $\mathrm{Re}\lambda < 0$, then the base solution $\hat{u}$ is unstable, otherwise it is said to be linearly stable (see e.g. [4]). The drag force acting on the body B is given by

$$F_D = \int_B \left(\rho\nu\frac{\partial u_t}{\partial n}n_y - pn_x\right) dS,$$

where ∂S is the surface of the object, n is the normal vector on S with its components n_x, n_y. The tangential velocity on S is denoted by u_t and the tangent vector is defined by $(n_y, -n_x)$.

2.2 Parameterization of shape

The shape of the body is described by spline functions consisting of a composition of cubic Bézier curves (see [5]). We focus on the description in one quadrant using symmetry properties for the other parts. A Bézier curve is defined by

$$b(t) = \sum_{j=0}^{3} b_j B_j^{(3)}(t)$$

with control points $b_0, \ldots, b_3 \in \mathbb{R}^2$ and Bernstein polynomials

$$B_j^{(2)}(t) = \left(\begin{array}{c} 2 \\ j \end{array}\right) t^j(1-t)^{2-j}, \quad t \in [0,1].$$

For the composition we have for $i = 0, 1$

$$b(t) = \sum_{j=0}^{3} b_{3i+j} B_j^3\left(\frac{t-t_i}{t_{i+1}-t_i}\right) \quad \text{on} \quad t \in [t_i, t_{i+1}].$$

As control points for the composition of two cubic Bézier curves in the first quadrant we take in this case

$$b_0 = (0,h), \quad b_3 = (x_1, y_1), \quad b_6 = (l, 0)$$

and further

$$b_1 = (l/6, h), \quad b_5 = (l, h/6)$$

to guarantee differentiability at b_0 and b_6. The points b_2 and b_4 are chosen to ensure a smooth connection with given slope $(1 \cdot l, m \cdot h)$ at b_3, where $m \in [0, -1]$. In this context we set $t_0 = 0, t_1 = 1/2$ and $t_2 = 1$. The grid point b_3 is not fixed and depends on the choice of coefficients c_l and c_h,

$$b_3 = (x_1, y_1) = (c_l \cdot l, c_h \cdot h).$$

The volume V is approximated by the area under the control polygon on the interval $[0, l]$. It yields in good approximation the volume of the described object.

$$\begin{aligned} V &= \sum_{i=1}^{6} (b_{i,x} - b_{i-1,x})(b_{i-1,y} + b_{i,y})/2 \\ &= \left(\left(\frac{5}{12} - \frac{m}{6} \right) c_l + \left(\frac{5}{72} + \frac{7m}{72} \right) + \frac{7}{12} c_h \right) lh. \end{aligned}$$

The construction in one quadrant and several designs for constant volume are depicted in Fig. 2.

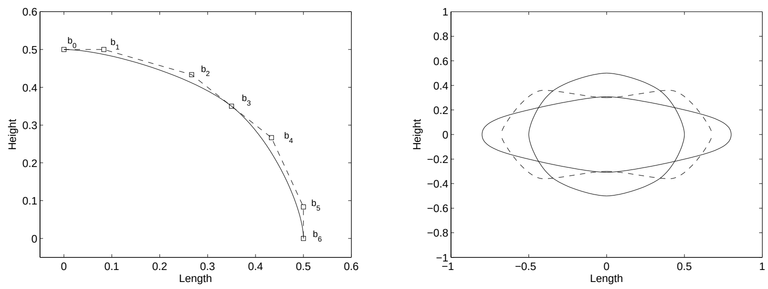

Figure 2: Construction in the case of cubic spline approximation (left) and different shapes with constant volume (right).

Within this context the shape optimization problem we want to solve is formulated as follows

$$\begin{array}{l} \min_q J(q) = F_D(q) \\ \text{s.t.} \\ \text{Re}(\lambda_{\min}(A(S(q)))) \geq 0, \\ V(q) = V^*, \\ \underline{q} \leq q \leq \bar{q}. \end{array} \tag{4}$$

where q is a design variable vector consisting of parameters for the description of the body by spline functions. Usually, we consider the five-dimensional vector

$$q = (l, h, m, c_l, c_h)$$

including the length l and height h of the body as well as slope m and position (c_l, c_h) of the control point b_3. The vectors $\underline{q}$ and $\bar{q}$ are lower and upper bounds for the parameters due to restrictions on the geometry of the body. The volume V of the body is assumed to be constant at value V^*.

3 Metamodel by radial basis functions

In [7] a numerical scheme based on the finite element method is proposed to solve the hydrodynamic, the eigenvalue and the optimization problem. This involves heavy computations and if the optimization problem is solved using evaluations of this PDE model the iteration process is very time consuming. Therefore, we are interested in finding a good approximation for the original PDE model to speed up the calculations. We intend to construct a metamodel to approximate the functions of drag and the smallest eigenvalue. Then we can either run the whole optimization process for the metamodel or switch to it for a certain number of function evaluations.
There are several ways for such an approximation as e.g. polynomial fitting, radial basis functions, neural networks and kriging. In our case, radial basis functions, which are non-polynomial interpolation methods, are used since they have been found to be very accurate for interpolation in high dimensions (see e.g. [1]). The construction of the approximation is based on a database of N_D points $x_i \in \mathbb{R}^n$ where the original model is evaluated. The approximation $\tilde{f}$ of an original function f is described as follows

$$\tilde{f}(x) = \sum_{i=1}^{N} c_i \Phi(x - x_i),$$

where $\Phi(x) = \phi(\|x\|)$ is a radial function. We choose the function Φ to be the Gaussian radial basis function

$$\Phi(x) = e^{-x^2/a^2},$$

and the parameter a is called attenuation factor. The coefficients c_i are determined by the solution of the linear system

$$Ac = F,$$

where $(A_{ji})_{i,j=1,\ldots,N_D} = \Phi(x_j - x_i)$ and $F_i = f(x_i)$, $i = 1, \ldots, N_D$. For distinct data points the matrix A is positive definite and hence there exists a solution of this linear system. Since we have an analytical expression for the approximation $\tilde{f}$ we can also determine gradients which could be used in a gradient-based optimization method. Indeed the partial derivative with respect to $x^{(k)}$ is given by

$$\frac{\partial f}{\partial x^{(k)}} = \sum_{i=1}^{N} c_i e^{-x^2/a^2} (-2(x^{(k)} - x_i^{(k)})/a^2).$$

To determine the best attenuation factor a we consider a testing set of N_T additional points $(\hat{x}_j)_{j=1,\ldots,N_T}$ and evaluate the error $e_f(a)$ between the value of the original function f and the approximation $\tilde{f}$ for different attenuation factors, i.e.

$$e_f(a) = \|(f(\hat{x}_j) - \tilde{f}(\hat{x}_j; a))_{j=1,\ldots,N_T}\|.$$

We compute radial basis function approximations for the drag F_D and the stability represented by the smallest eigenvalue $\lambda_{\min}$. We use a database with $N_D = 99$ points $x_i \in \mathbb{R}^5$ and a testing set of $N_T = 8$ points. The error functions e_{F_D} and e_λ for different attenuation factors are shown in Fig. 3.

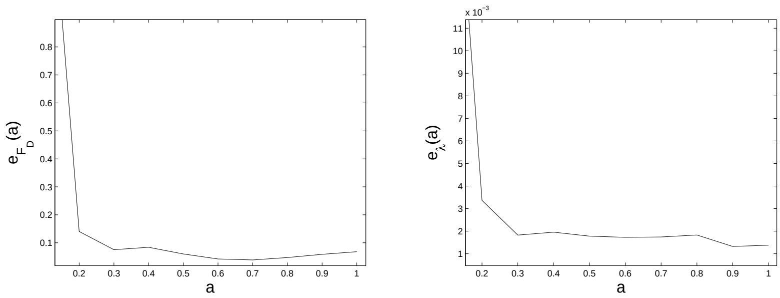

Figure 3: Error for drag and eigenvalue approximation using the Gauss function.

Further computations are performed using $a_f = 0.7$ and $a_\lambda = 0.9$.

To increase the accuracy of calculations, an enlargement of the database as well as the testing set is intended and will be addressed in future computations.

4 Particle swarm optimization

The use of gradient-based algorithms like SQP faces certain limitations. Convergence to a global optimum is only possible if we start sufficiently close to it. Usually, we do not know this optimum in advance. In addition, the treatment of functions which may be nondifferentiable in certain points can lead to inaccurate results. Therefore, in our studies we want to consider a semi-stochastic optimization method which uses function values only and aims at finding the global optimum.

In recent years work has been focused on algorithms that are inspired from natural phenomena. In this framework we want to study the Particle Swarm Optimzation (PSO) technique, which mimics the social behaviour of bird flocking (see [8]). The algorithm is based on the assumption that individual birds exchange information about their position, velocity and fitness and the behaviour of the flock is then influenced to increase the probability of migration to regions of high fitness (see [6]).

We consider a set of p particles $x_i \in \mathbb{R}^n$ that constitute the swarm. We want to follow the trajectory of each particle, adjusting it dynamically using the information collected throughout the optimization process. In each iteration step k the position of the particle x_i^k is changed by a velocity vector v_i,

$$x_i^{k+1} = x_i^k + v_i^k.$$

The velocity vector is determined by the velocity in the previous iteration step, the distance to the best position x_i^* of the particle i in iteration k and the best ever reached position x^*,

$$v_i^k = \omega^k v_i^{k-1} + c_1 r_1 (x_i^* - x_i^k) + c_2 r_2 (x^* - x_i^k).$$

Here $r_1, r_2 \in [0,1]$ are uniformly distributed numbers and c_1 and c_2 are called trust coefficients and set to the value two to have a mean of 1. Moreover, an inertia term ω is added which is set to the value of 1.3 and which is reduced successively by a factor α after a certain number of iterations. The location of the particles and the velocities are initialized randomly within lower and upper bounds $\underline{v}, \bar{v}$ and $\underline{x}, \bar{x}$ respectively. A complete algorithm reads then as follows.

1. For $i = 1, \ldots, p$ initialize randomly the locations $x_i^0 \in [\underline{x}, \bar{x}]$ and the velocities $v_i^0 \in [\underline{v}, \bar{v}]$, initialize the inertia ω^0, set $k = 0$.
2. Evaluate the function for each particle.
3. Update inertia if necessary, i.e. $\omega^k = \alpha \omega^{k-1}$.
4. Update the best position x_i^* for particle i and the global best position x^*.
5. Compute the velocities of the particles
$$v_i^k = \omega^k v_i^{k-1} + c_1 r_1 (x_i^* - x_i^k) + c_2 r_2 (x^* - x_i^k).$$
6. Check if velocities are within the boundaries, otherwise project them on $\underline{v}$ or $\bar{v}$.
7. Update the location of the particles
$$x_i^{k+1} = x_i^k + v_i^k$$
8. Check if locations of particles are within the boundaries, otherwise project them on $\underline{x}$ or $\bar{x}$.
9. If termination criterion reached then stop, otherwise set $k := k + 1$ and goto step 2.

This algorithm is well-suited for parallelization since each trajectory could be computed on a different processor. Only the global best position x^* has to be communicated between the processors.

This particle swarm optimization algorithm is used to solve the optimization problem (4). The volume constraint and the box constraints are treated by adding suitable penalty terms to the objective functional, i.e.

$$J(q) = F_D(q) + \sum_{i \in I} \mu_i \max\{0, c_i(q)\} + \sum_{j \in E} \nu_j c_j(q)^2,$$

where the index set of the inequality constraints $c_i(q) \leq 0$ is denoted by I and the set of equality constraints by E. In our case the objective functional reads as

$$J(q) = F_D(q) + \mu \max\{0, -\lambda_{\min}(q)\} + \nu(V(q) - V^*)^2$$

for penalty parameters μ and ν.

5 Multiobjective optimization

5.1 Concurrent optimization

In a concurrent optimization problem we want to optimize two criteria where we select a primary criterion J_A and a secondary criterion J_B. We want to study if we can improve the secondary criterion without deteriorating the primary one too much. We follow the approach which is developed in [3] and simulate a Nash game where one player utilizes a subspace $U \subset \mathbb{R}^n$ to optimize J_A and the other one its complement W to optimize J_B. It is the goal to obtain a Nash equilibrium point $q = (\bar{u}, \bar{w})$ for which the strategy of one player is best given the strategy of the other player [9]. The optimal strategies are denoted by $\bar{u}$ and $\bar{w}$, respectively. The corresponding concurrent optimization problem is then written as follows

$$\begin{array}{llcll} \min_{u \in U} & J_A(u, \bar{w}) & \qquad & \min_{w \in W} & J_B(\bar{u}, w) \\ \text{s.t.} & c(u, \bar{w}) = 0 & & \text{s.t.} & \text{no constraint} \end{array} \tag{5}$$

where c is a vector of length n_c.

As a starting point to solve (5) we consider the optimum q_A^* of the target functional J_A. From this design point we consider variations with respect to orthogonal directions ω_i, summarized in the splitting matrix

$$S = (\omega_1, \ldots, \omega_n).$$

Variations related to the subspace U are denoted by the coefficients $u_1, \ldots, u_p$ and those to subspace W by $w_1, \ldots, w_{n-p}$. The vector q is then written as

$$q = q_A^* + S \begin{pmatrix} u \\ w \end{pmatrix}$$

and

$$q = q_A^* + u_1\omega_1 + \ldots + u_p\omega_p + w_1\omega_{p+1} + \ldots + w_{n-p}\omega_n$$

respectively. The design variables related to J_A are now $u_1, \ldots, u_p$ and for J_B they are $w_1, \ldots w_{n-p}$. The crucial point is now the choice of the splitting matrix S and the decision in which way to split into u and w.

As proposed in [3] we want to choose the splitting matrix according to the eigenvectors of the Hessian matrix at the optimum q_A^* of the objective functional J_A. This approach is based on the following idea. We consider the expansion of the primary functional J_A about its optimized value q_A^* in the direction of a unit vector $\omega \in \mathbb{R}^n$.

$$J_A(q_A^* + \varepsilon\omega) = J_A(q_A^*) + \varepsilon(\nabla J_A^*)^T\omega + \frac{\varepsilon^2}{2}\omega^T H_A^*\omega + O(\varepsilon^3),$$

where $H_A^* = H_A(q_A^*)$. The objective is to propose a splitting associated with the choice of a basis of vectors ω^j such that the tail $n-p$ vectors are associated with variations that are, for fixed small-enough ε, as small as possible. Since $(\nabla J_A^*)^T\omega_i = 0$ (for $i > n_c$) we have

$$J_A(q_A^* + \varepsilon\omega) - J_A(q_A^*) = \frac{\varepsilon^2}{2}\omega^T H_A^*\omega + O(\varepsilon^3) \tag{6}$$

and we propose to consider the eigenvectors associated to the smallest eigenvalues of H_A^* as design space for J_B.

If constraints appear we assume linear independance and knowledge of the gradient in q_A^*

$$\nabla c_i^* = \nabla c_i(q_A^*), \quad i = 1, \ldots, n_c.$$

Furthermore, by a Gram-Schmidt orthogonalization process we want to construct an orthonormal basis $\{\omega_1, \ldots, \omega_{n_c}\}$ of the subspace generated by them, i.e. the subspace tangent to the constraint hyper-surfaces $c_i = 0, i = 1, \ldots, n_c$,

$$\omega_1 = \frac{\nabla c_1^*}{\|\nabla c_1^*\|},$$

$$\tilde{\omega}_2 = \nabla c_2^* - (\nabla c_2^*, \omega_1)\omega_1, \qquad \omega_2 = \frac{\tilde{\omega}_2}{\|\tilde{\omega}_2\|},$$

$$\tilde{\omega}_i = \nabla c_i^* - \sum_{j=1}^{i-1} k_j\omega_j, \quad k_j \quad \text{s.t.} \quad \tilde{\omega}_i \perp \omega_j \qquad \tilde{\omega}_i = \frac{\tilde{\omega}_i}{\|\tilde{\omega}_i\|}, \qquad i = 3, \ldots, n_c.$$

The orthogonal projection onto the subspace tangent to the constraint hyper-surfaces is given by the matrix P, described as follows,

$$P = I - \sum_{i=0}^{n_c} \omega_i\omega_i^T.$$

We then define the matrix H'_A, which is real and symmetric,

$$H'_A = PH^*_A P$$

and denote its eigenvalues by μ'_j, $j = 1, \dots, n$. These include the vectors that belong to the null space of P and those of H'_A. Equation (6) changes to

$$J_A(q^*_A + \varepsilon\omega_j) - J_A(q^*_A) = \frac{\varepsilon^2 \mu'_j}{2} + O(\varepsilon^3).$$

The a priori best definition of a $n-p$-dimensional subspace as a strategy to minimize the secondary functional is then the subspace spanned by the $n-p$ eigenvectors associated with the smallest nonzero absolute values $|\mu'_j|$.

5.2 Application to hydrodynamic problem

In the previous research [7], only the drag has been considered as objective function, while the stability, represented by the smallest eigenvalue, was treated as constraint. However, the stability aspect is of great importance and we would be interested in the best value which could be achieved. This is e.g. necessary to decide whether for a given inflow velocity hydrodynamic stability could be guaranteed.

In a first step we consider two optimization problems, one with the drag as objective and one with the eigenvalue function as objective. In a next step we want to consider both problems simultaneously and start a game between the two of them to determine Nash equilibrium points. All calculations are performed running a PSO algorithm with 128 particles and 1000 iterations for the metamodel. The results are later compared with the original PDE model.

The drag minimization problem (4) is now formulated without the stablity constraint while keeping the volume and the box constraint.

$$\begin{array}{ll} \min_q & F_D(q) \\ \text{s.t.} & V(q) = V^* \\ & \underline{q} \leq q \leq \bar{q} \end{array}$$

The lower and upper box constraints are in our application given by

$$\underline{q} = (0.03, 0.03, -1.0, 0.5, 0.5), \qquad \bar{q} = (0.08, 0.08, -0.5, 0.8, 0.8).$$

Since the stability constraint is not active at the optimum we found in the previous calculations we expect to find a similar result. Indeed we have

$$q^* = (0.08, 0.030277, -1, 0.8, 0.64325)$$

which yields

$$F_D = 5.35967, \quad \mathrm{Re}(\lambda_{\min}) = 0.0778279, \quad V = 0.00197193.$$

A comparison with the original model shows a good agreement of the results

$$F_D = 5.36048, \quad \mathrm{Re}(\lambda_{\min}) = 0.0777728, \quad V = 0.00197193.$$

The first component of the eigenfunction for the flow around the optimal design q^* is plotted in Fig. 4.

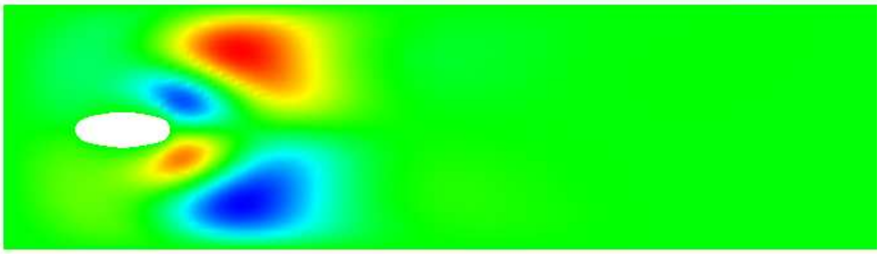

Figure 4: Eigenfunction associated to the smallest eigenvalue for design q^*

The problem of the maximization of the smallest eigenvalue is written as minimization of the negative smallest eigenvalue.

$$\begin{array}{ll} \min_q & -\mathrm{Re}(\lambda_{\min}(q)) \\ \text{s.t.} & V(q) = V^* \\ & \underline{q} \le q \le \bar{q}. \end{array}$$

As optimum we obtain

$$q^{**} = (0.08, 0.03, -0.660609, 0.8, 0.677202)$$

which yields

$$F_D = 5.41806 \quad \mathrm{Re}(\lambda_{\min}) = 0.0783713, \quad V = 0.001972.$$

Not surprisingly, for this point we have a larger eigenvalue in combination with a larger drag indicating a trade-off between the two goals. However, the differences are quite small, showing that the two goals are not totally antagonistic. Of course, it would now be interesting to see to what extent it is possible to keep the value of one objective without losing in the other objective too much. This is done in the framework of Nash game simulation which was described in Section 5.1.

In our case we consider the drag as primary functional, $J_A = F_D$ and the smallest eigenvalue as secondary functional. Since we want the functionals to have positive values we focus on the modified functional, $J_B = e^{\beta(-\mathrm{Re}(\lambda_{\min})+\mathrm{Re}(\lambda^*_{\min}))}$,

$$\begin{array}{lll} \min_{u\in U} & J_A(u) = F_D(u) \\ \text{s.t.} & V(u) = V^* \\ & \underline{q} \le q \le \bar{q} \end{array} \qquad \begin{array}{lll} \min_{w\in W} & J_B(w) = e^{\beta(-\mathrm{Re}(\lambda_{\min}(w))+\mathrm{Re}(\lambda^*_{\min}))} \\ \text{s.t.} & \underline{q} \le q \le \bar{q} \end{array} \tag{7}$$

However, at q^* we have 4 active constraints, namely the volume constraint and three box constraints. After projection on these constraints there is only one degree of freedom left, which can hardly be used for a split of territories. Due to this unfavourable situation we want to modify the formulation and assume one fixed box constraint expecting that other box constraints turn inactive then. Indeed, if we keep $q_4 = 0.7$ fixed and optimize J_A over the remaining four variables we obtain

$$q^{***} = (0.08, 0.0319559, -0.842778, 0.7, 0.675183)$$

which yields

$$F_D = 5.39037 \quad \mathrm{Re}(\lambda_{\min}) = 0.073795, \quad V = 0.00197192.$$

In this case, apart from the fixed value for q_4, we have only one more active box constraint on q_1 at q^{***} which appears in all calculations and seems to be inevitable. Hence we consider a reduced problem of four design variables. The eigenvalues of the Hessian matrix H'_A of $J_A = F_D$ with projection on the gradient of the volume constraint at q^{***} and on the active box constraint as explained above are then given by

$$\mu'_1 = 0.0000, \quad \mu'_2 = 0.0000, \quad \mu'_3 = 8.2654, \quad \mu'_4 = 0.8681.$$

The corresponding eigenvectors are denoted by

$$\{\omega_1, \ldots, \omega_4\},$$

where ω_1 belongs to the box constraint $q^*_1 = \bar{q}_1$ and ω_2 to the gradient of the volume constraint. The projection leads to an elimination of the first variable in all other eigenvectors. Therefore, we can neglect this variable in the forthcoming calculations and we are left with three variables for which we can define a split of territories. Since μ'_4 is small compared to μ'_3 we assign the space spanned by the eigenvector corresponding to μ'_4 to the secondary functional J_B. Therefore, we decide for a 2+1-split ($u_1 = \omega_2, u_2 = \omega_3, w_1 = \omega_4$) and start a Nash game to solve the problem (7). We obtain the following Nash equilibrium point,

$$q = (0.08, 0.0317536, -0.74314, 0.7, 0.686912).$$

For this design point we have

$$F_D = 5.39474, \quad \mathrm{Re}(\lambda_{\min}) = 0.0742049, \quad V = 0.00197192.$$

We observe that there is only a modest increase of the smallest eigenvalue by 0.56%. But, as desired, the drag changes less, with a relative increase of 0.08%.

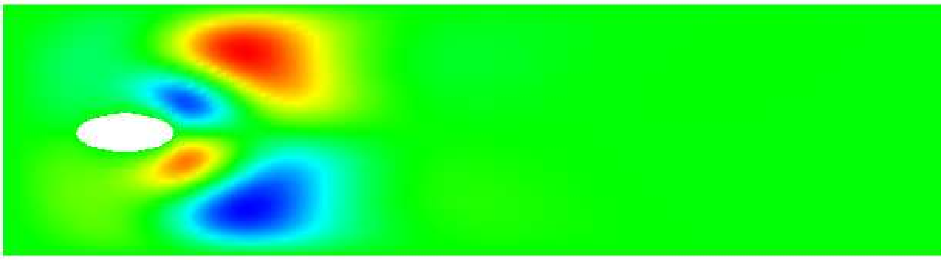

Figure 5: Eigenfunction associated to the smallest eigenvalue of Nash equilibrium design q

The design of the Nash equilibrium point together with the first component of the eigenfunction of the smallest eigenvalue is shown in Fig. 5.

In a next step we want to study how the Nash equilibrium points evolve when we gradually move from one objective to another. Therefore, we introduce a functional J_{AB} together with some parameter ε and consider the following concurrent optimization problem.

$$\begin{array}{rl} \min_{u\in U} & J_A(u) = F_D(u) \\ \text{s.t.} & V(u) = V^* \\ & \underline{q} \leq q \leq \bar{q} \end{array}$$

and

$$\begin{array}{rl} \min_{w\in W} & J_{AB}(w) := \frac{J_A(w)}{J_A^*} + \varepsilon(\theta \frac{J_B(w)}{J_B^*} - \frac{J_A(w)}{J_A^*}) \\ \text{s.t.} & \underline{q} \leq q \leq \bar{q} \end{array}$$

where $J_B(w) = e^{\beta(-\mathrm{Re}(\lambda_{\min}(w))+\mathrm{Re}(\lambda^*_{\min}))}$ and $0 < \theta \leq 1$.

The behaviour is illustrated in Fig. 6 which shows the parameter ε on the range $[0, 1]$. We depict the values obtained for the metamodel at the various design points as well as the values of the original model. We can see that for small values of ε the eigenvalue can be increased, while the drag remains almost constant. However, the magnitude of the change is quite small. We have $F_D'(\varepsilon) = 0$ at $\varepsilon = 0$ as stated in [3].

6 Summary

For a shape optimization problem in hydrodynamic stability control we have extended an existing numerical scheme. We have constructed a metamodel based on radial basis functions to approximate the functions of drag and stability to speed up the calculations. A particle swarm optimization algorithm was implemented to enforce the search for a global optimum. Moreover, a concurrent optimization approach was set up to consider two objective functionals. A Nash game was then simulated to determine Nash equilibrium points.

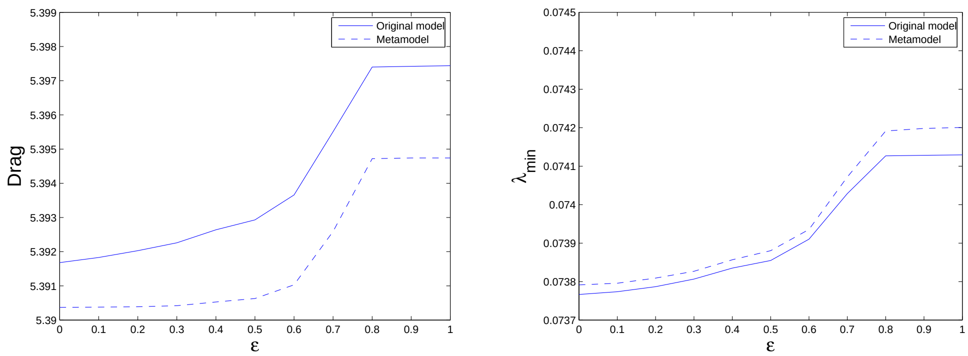

Figure 6: Dependance of drag F_D (left) and smallest eigenvalue $\lambda_{\min}$ (right) on ε

These approaches could be successfully applied to the hydrodynamic stability control problem. For a moderate database size a metamodel in good approximation of the original model was obtained. Using the metamodel, it is possible to run a full optimization in a few seconds. The application of the PSO algorithm yields the optima without any calculation of the derivatives. Furthermore, the multiobjective optimization approach results in Nash equilibrium points which indicate a possibility to improve the secondary criterion with only a small degradation of the primary criterion. However, the changes in absolute values for both functionals are rather small such that a choice of more antagonistic criteria is suggested. Moreover, an increase of the number of design variables can lead to more flexibility and larger differences in the objective function values. These aspects are currently under research.

Acknowledgements

The authors gratefully acknowledge the support of the Karlsruhe House of Young Scientists for a stay of Frank Strauß at Inria in Sophia Antipolis.

References

[1] Buhmann, M.: Radial basis functions. Acta Numerica **9**, 1–38 (2000)

[2] Chandrasekhar, S.: Hydrodynamic and hydromagnetic stability. Dover Publ. (1981)

[3] Désidéri, J.A.: Split of territories in concurrent optimization. Tech. Rep. 6108, INRIA (2007)

[4] Drazin, P., Reid, W.: Hydrodynamic stability. Cambridge University Press (1982)

[5] Farin, G.: Curves and Surfaces for Computer Aided Geometric Design. Academic Press, Boston (1988)

[6] Fourie, P., Groenwold, A.: The particle swarm optimization algorithm in size and shape optimization. Struct Multidisc Optim **23**, 259–267 (2002)

[7] Heuveline, V., Strauß, F.: Shape optimization towards stability in constrained hydrodynamic systems. To appear in Journal of Computational Physics (2008)

[8] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Int. Conf. on Neural Networks, pp. 1942–1948 (1995)

[9] Nash, J.: Non-cooperative games. Annals of Mathematics **54**(2), 286–295 (1951)

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399