European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)

A REDUCED-BASIS METHOD FOR SOLVING PARAMETER-DEPENDENT CONVECTION-DIFFUSION PROBLEMS AROUND RIGID BODIES[1]

Timo Tonn* **and Karsten Urban†**

*Politecnico di Torino, Dipartimento di Matematica
Corso Duca degli Abbruzzi 24, 10129 Torino, Italy
e-mail: timotonn@calvino.polito.it

†University of Ulm, Abteilung Numerik
Helmholtzstr. 18, 89069 Ulm, Germany
e-mail: karsten.urban@uni-ulm.de

Key words: Reduced-Basis Method, Convection-Diffusion, Rigid Bodies

Abstract. *We consider a convection-diffusion problem in a box in which rigid bodies are present. The location and orientation of these bodies are subject to a set of parameters. In order to use a reduced basis method, we perform a two-step method. In the first step, we transform the parameter-dependent geometric situation to a reference situation (also mapping the mesh). Then, we use the Empirical Interpolation Method (EIM) in order to separate the parameter from the variables of the pde. We present several numerical results that indicate the efficiency of the method. The corresponding analysis will be presented in a forthcoming paper, [2].*

1 Introduction

Flow problems around moving bodies naturally occur in several applications, e.g. propulsion systems for ships or helicopters. The flow problems are typically modeled by instationary nonlinear pde's and usually lead to highly complex, extremely high dimensional nonlinear systems. Hence, for an efficient numerical solution, a model reduction is strongly required.

There is a huge literature on reduced basis methods in particular for flow problems, a complete list goes beyond the scope of this paper. Let us just mention [4, 5, 7, 8,

[1]This work was supported by the *European Commission* within the Research Training Network *Breaking Complexity*, No. 2-2001-00574. This paper was written when the first author was in residence at the Politecnico di Torino. We are extremely grateful to Claudio Canuto for many inspiring discussions and in particular to Gianluigi Rozza (MIT, Department of Mechanical Engineering) for inspiring discussions and for providing us with useful hints and materials. We also acknowledge that he pointed out missing references to related work in an earlier version of this paper.

9, 10, 15, 16, 19, 23, 25] for fluid flow, [11, 13, 17, 18, 19, 20] for elliptic and [6, 22] for parabolic problems. All investigations are concerned either with stationary problems or with instationary problems, where the domain is fixed in time. More recently, also parameter-dependent problems have been investigated, [11, 13, 18, 19, 25]. In most of the papers the influence of these parameter is affine. Just recently, non-affine parameter dependencies have been investigated, [12, 14].

We want to develop a new reduced basis framework that allows us to construct a reduced model for instationary problems with moving bodies. As a first step in this direction, we aim to develop a reduced basis framework for linear and nonlinear convection-diffusion problems in a box containing one or more rigid bodies whose position and orientation are subject to a parameter (which can be time-dependent in a next step also playing the role of a control).

Our method consists of two steps. Firstly, we use domain decomposition and mapping to reduce the problem to reference situation. Then, we use the *Empirical Interpolation Method (EIM)* to separate the influence of the parameter (which is non-affine here) from the pde. This idea has also been used in [12, 14] to treat parametrized problems where the influence of the parameter is non-affine. The basis functions for the reduced model are then defined by taking snapshots with respect to the parameter space.

In a forthcoming paper, [2], we will present the numerical analysis (e.g. error estimates) for the presented method. Investigations in this direction can be found e.g. in [3, 21]. More details on the particular computations can be found in [24].

The remainder of this paper is organized as follows. In Section 2, we introduce the considered convection-diffusion problem and describe appropriate parameter spaces. Section 3 is devoted to the description of the domain transformation and Section 4 contains the Empirical Interpolation Method. The final reduced basis model is introduced in Section 5, Section 6 contains our numerical results and Section 7 conclusions and an outlook.

2 A convection-diffusion problem around rigid bodies

We consider a stationary convection–diffusion problem in a rectangle $\square \subset \mathbb{R}^2$ in which one or more rigid bodies are located in dependence of a parameter μ. We assume that the shape of the bodies are identical and fixed. The bodies can be interpreted as blades of a rotor or propeller and in the case of only one body $B(\mu)$ is a rotated blade of some $B = B(0)$ around an angle $\mu \in D = [0, \frac{\pi}{2}]$, see the left part of Figure 1.

In the case of several blades we also assume that they are oriented in dependence of a parameter which now is a vector. We consider here the case of five blades. To be precise, the five blades are fixed on a rotating disc (the rotor) which is rotated with an angle ϕ. Since the whole model is periodic, it is sufficient to consider $\phi \in [-\frac{\pi}{5}, \frac{\pi}{5}]$ in the case of five blades. At each rotation angle ϕ (also called *phase angle*) of the rotor, each blade B_i is oriented around an angle ν_i with respect to the tangential of the circle. For technical

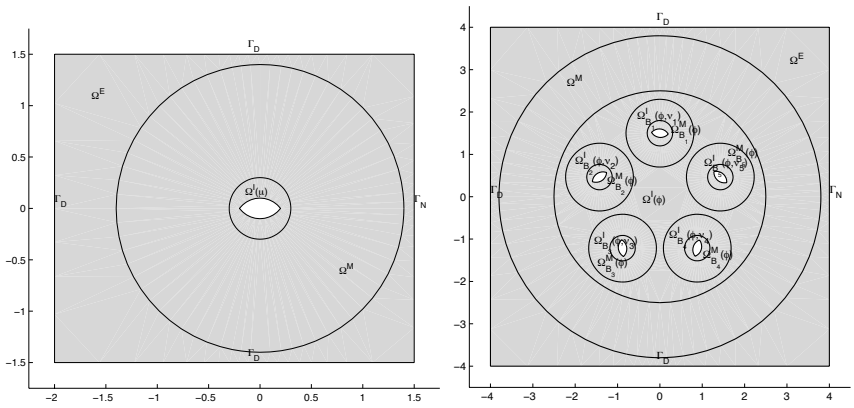

Figure 1: Geometry for one blade (left) and five (right) blades.

reasons, one typically restricts the range

$$\nu_i \in \Big[-\frac{\pi}{36}, \frac{\pi}{36} \Big] = [-5^\circ, 5^\circ].$$

Thus, the parameter vector reads

$$(\phi, \nu_1, \dots, \nu_5) \in D := \Big[-\frac{\pi}{5}, \frac{\pi}{5} \Big] \times \Big[-\frac{\pi}{36}, \frac{\pi}{36} \Big]^5.$$

Let us denote by N_B the number of blades (as already said, we detail the cases $N_B = 1$ and $N_B = 5$), then

$$B(\mu) := \bigcup_{n=1}^{N_B} B_n(\mu)$$

denotes the part of $\square$ containing the rigid bodies. Here, $\mu \in D$ is the parameter described above. We subdivide $\square \setminus B(\mu) =: \Omega(\mu)$ in subdomains according to Figure 1, so that we obtain

$$\Omega(\mu) := \bigcup_{n=1}^{N_S} \Omega_n(\mu)$$

with subdomains $\Omega_n(\mu) \subset \square$ and $N_S \in \mathbb{N}$ denoting the number of subdomains ($N_S = 3$ for one and $N_S = 13$ for five blades).

Given coefficients a and b (that may also be non-constant or even also depend on the solution itself), we consider the following convection-diffusion problem.

$$\begin{cases} -a\Delta u + b \cdot \nabla u = 0 \ , & \text{in } \Omega(\mu), \\ u = 0 \ , & \text{on } \partial B(\mu), \\ u = 1 \ , & \text{on } \Gamma_D, \\ \frac{\partial u}{\partial n} = 0 \ , & \text{on } \Gamma_N, \end{cases} \tag{2.1}$$

where $\Gamma_N := \partial\square \cap \{x = 1.5\}$ is the Neumann part of the outer boundary $\partial\square$ and $\Gamma_D := \partial\square \setminus \Gamma_N$ the Dirichlet part of $\partial\square$.

Then, defining for $\mu \in D$ and $g \in H^{1/2}(\Gamma_D)$

$$V(\mu, g) := \left\{v \in H^1(\Omega(\mu)) : v = g \text{ on } \Gamma_D,\ v = 0 \text{ on } \partial B(\mu)\right\} \tag{2.2}$$

the variational formulation of (2.1) is well-known to read as follows: Find $u \in V(\mu, 1)$, such that

$$\int_{\Omega(\mu)} a(x)\, \nabla u(x)\, \nabla v(x)\ dx + \int_{\Omega(\mu)} b(x) \cdot \nabla u(x)\, v(x)\ dx = 0 \quad \forall v \in V(\mu, 0), \tag{2.3}$$

or, equivalently in operator form

$$\langle A(\mu)u, v\rangle + \langle B(\mu)u, v\rangle = 0 \quad \forall v \in V(\mu, 0)\ , \tag{2.4}$$

where

$$\begin{aligned} \langle A(\mu)u, v\rangle &:= \sum_{n=1}^{N_S} \int_{\Omega_n(\mu)} a(x)\, \nabla u(x) \cdot \nabla v(x)\ dx, \\ \langle B(\mu)u, v\rangle &:= \sum_{n=1}^{N_S} \int_{\Omega_n(\mu)} b(x) \cdot\ \nabla u(x)\, v(x)\ dx. \end{aligned} \tag{2.5}$$

As already said, we want to investigate if a reduced basis method can be used for such a parameter-dependent problem. Reduced basis models have been studied in the literature also for bilinear forms that depend on a parameter, see e.g. [11, 13, 18, 19, 25]. However, there the parameter needs to enter the bilinear form in a very specific way, e.g. as multiple factor. Our case above does not fit into that framework. A possible wayout is described in [12, 14], by approximating the bilinear forms using the so-called empirical interpolation method (see Section 4 below).

3 Transformation to a reference domain

In our problem, the shape of the body is invariant, only its location and orientation depends on a specific parameter $\mu \in D$. Hence, it is a natural idea to transform the problem to a reference situation, say $\mu = 0$. Since in our particular problem, the parameter μ represents a rotation, the desired transformation seems obvious.

Thus, we transform the variational formulation (2.3) and (2.4), respectively, to a reference domain $\hat{\Omega}$. It turns out that this step is also crucial to derive an efficient algorithm, since it enables the application of a reduced basis method on the reference domain. We start by deriving the mapping for the case of one blade and five blades, respectively, and then perform the transformation itself. It turns out that the transformation is quite similar in both cases.

One single blade

As already pointed out, we use the splitting $\Omega(\mu)$ into $N_S = 3$ subdomains as visualized in Figure 1. Furthermore, we choose the reference domain $\hat{\Omega} = \Omega(\hat{\mu})$, where $\hat{\mu} = 0$, i.e., the blade is aligned horizontal.

We can easily define the mapping on each particular subdomain. On Ω_1, no mapping is applied, $\Omega_2(\mu)$ is rotated by an angle of $(-\mu)$, thus we have an affine transformation in this subdomain. In Ω_3, each point is rotated by an angle depending on its position, i.e., points near to the outer circle are almost not rotated, while points near to the inner circle are rotated almost by $(-\mu)$. As the rotation depends on the position this is a non-affine transformation.

Thus, one possible mapping reads:

$$\hat{x} = T(x,\mu) := \begin{cases} x, & \text{in } \Omega_1, \\ Q_1(\mu)x, & \text{in } \Omega_2(\mu), \\ Q_2(\mu,x)x, & \text{in } \Omega_3, \end{cases} \tag{3.1}$$

where $Q_1(\mu)$ is a (Givens-)rotation by $(-\mu)$ and

$$Q_2(x,\mu) := \begin{pmatrix} \cos(\mu\rho(\|x\|_2)) & \sin(\mu\rho(\|x\|_2)) \\ -\sin(\mu\rho(\|x\|_2)) & \cos(\mu\rho(\|x\|_2)) \end{pmatrix}, \tag{3.2}$$

where $\rho(z) := 1 - \frac{z-r_I}{r_O-r_I}$ and r_I,r_O are the radii of the inner and outer circle, respectively and we assume that the center of the inner and outer circle, respectively, is the origin. By definition, this choice realizes a full rotation by $(-\mu)$ for points on the inner circle ($\rho(r_I) = 1$) and no rotation for points on the outer circle ($\rho(r_O) = 0$). The effect of the (inverse) mapping is shown in Figure 2. For the transformation of the pde to the reference domain we also need the Jacobian of (3.1) which can be computed in a straightforward way.

Five blades

We assume that the orientation of the blade initially only depends on its position, i.e., on the rotation angle of the rotor. A function

$$f \in C^2_{2\pi}, \qquad f : \phi \mapsto \alpha$$

is called *blade steering curve (BSC)*, where α denotes the angle w.r.t. the tangential. We define the reference domain $\hat{\Omega}$ by $\Omega(\hat{\mu}) = \Omega(\hat{\phi}, \hat{\nu}_1, \hat{\nu}_2, \hat{\nu}_3, \hat{\nu}_4, \hat{\nu}_5)$, where $\hat{\phi} = 0$ and $\hat{\nu}_i = 0$, i.e., we have no rotation of the rotor and no deviation from the BSC in all five blades. Furthermore, $\Omega(\mu)$ is divided into $N_S = 13$ subdomains as visualized in Figure 1. For this

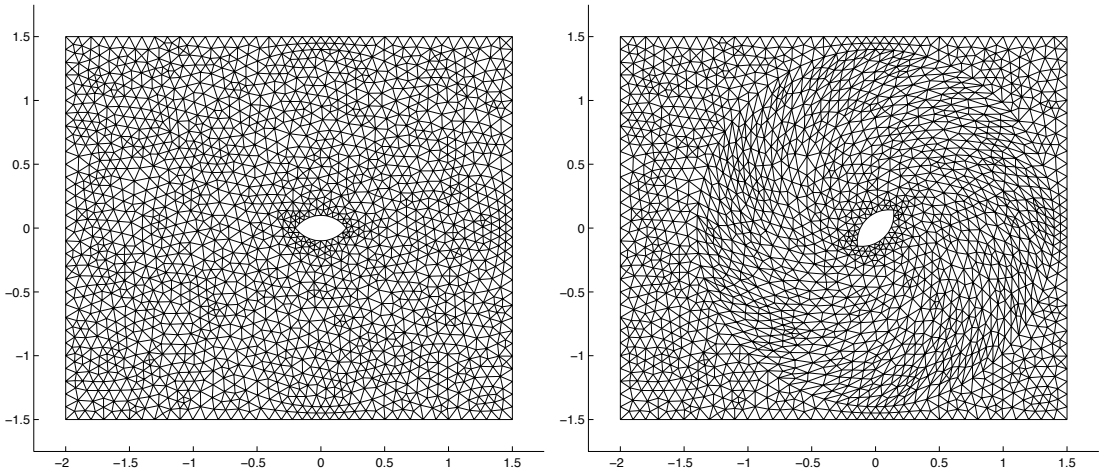

Figure 2: Effect of inverse mapping (one blade).

case one possible mapping reads $\hat{x} = T(x, \mu)$, where

$$T(x,\mu) := \begin{cases} x, & \text{in } \Omega^E, \\ Q_1(\phi)x, & \text{in } \Omega^I(\phi), \\ Q_2(x,\phi)x, & \text{in } \Omega^M, \\ Q_1(\phi)[Q_1(\Delta\nu_i)(x - M_{B_i}(\phi)) + M_{B_i}(\phi)], & \text{in } \Omega^I_{B_i}(\phi,\nu_i), \\ Q_1(\phi)[Q_2(x - M_{B_i}(\phi), \Delta\nu_i(x - M_{B_i}(\phi)) + M_{B_i}(\phi)], & \text{in } \Omega^M_{B_i}(\phi), \end{cases} \tag{3.3}$$

where $Q_1(\alpha)$ is again a (Givens-)rotation by $(-\alpha)$, $Q_2(x,\alpha)$ is defined by (3.2) and $M_{B_i}(\phi)$ is the center of the i-th blade (in the computational domain). Furthermore, $\Delta\nu_i$ is the difference of the steering angle of the i-th blade, when mapping from the original domain to the reference domain, i.e.,

$$\Delta\nu_i = \left(f\left(\hat{\phi}_i\right) + \hat{\nu}_i\right) - \left(f\left(\phi_i\right) + \nu_i\right) = f\left(2\pi\frac{i-1}{5}\right) - f\left(\phi + 2\pi\frac{i-1}{5}\right) - \nu_i,$$

where f is a given BSC. In other words, this is a concatenation of the mapping introduced for the case of one blade.

To visualize the effect of the (inverse) mapping, the left part of Figure 3 shows a mesh for the reference domain $\hat{\Omega}$ and the right part shows the resulting mesh when applying the (inverse) mapping for $\mu = (\phi, \nu_1, \nu_2, \nu_3, \nu_4, \nu_5) = (36^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$.

Again we have to compute the Jacobian of (3.3) which turns out to be a combination of the case of one blade.

Now we are ready to use these transformations within the pde.

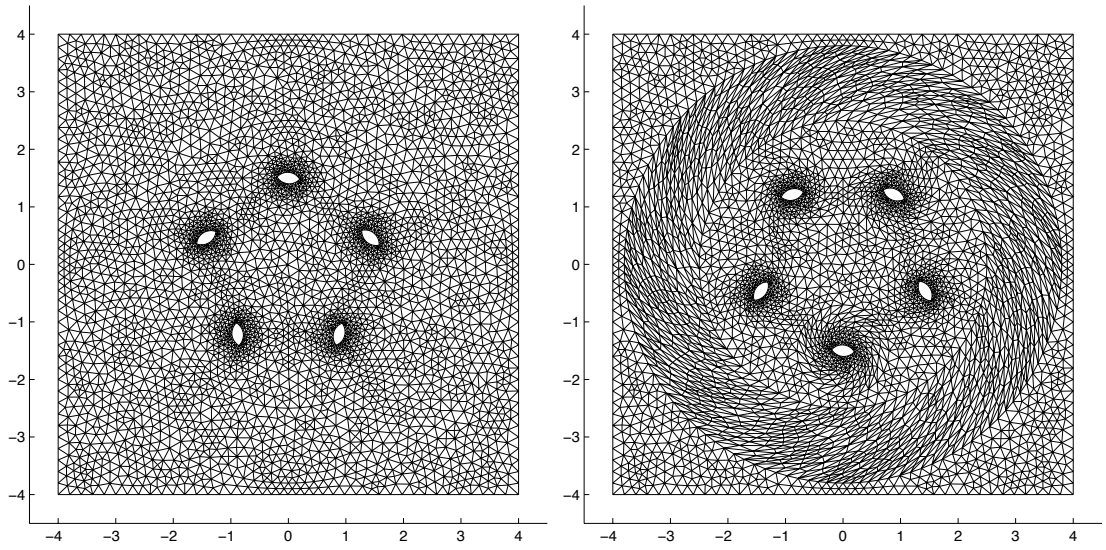

Figure 3: Mesh on reference domain $\hat{\Omega}$ (left) and effect of the inverse mapping (right) for the case of five blades.

Transformation of the pde to the reference domain

To transform the pde we follow standard lines and set $\hat{\Omega} = \Omega(\hat{\mu})$ as well as $\hat{V}(g) := V(\hat{\mu}, g)$ on the reference domain. Then, the variational formulation for (2.1) on the reference domain takes the form: Find $\hat{u} \in \hat{V}(1)$, such that

$$\left\langle \hat{A}(\mu)\hat{u}, \hat{v} \right\rangle + \left\langle \hat{B}(\mu)\hat{u}, \hat{v} \right\rangle = 0 \quad \forall\ \hat{v} \in \hat{V}(0), \tag{3.4}$$

for differential operators $\hat{A}(\mu)$ and $\hat{B}(\mu)$ that take the form

$$\begin{aligned} \left\langle \hat{A}(\mu)\hat{u}, \hat{v} \right\rangle &= \sum_{n=1}^{N_S} \int_{\hat{\Omega}_n} \nabla\hat{u}(\hat{x}) \cdot T^{(n)}(\hat{x};\mu)\, \nabla\hat{v}(\hat{x})\, d\hat{x}, \\ \left\langle \hat{B}(\mu)\hat{u}, \hat{v} \right\rangle &= \sum_{n=1}^{N_S} \int_{\hat{\Omega}_n} \hat{v}(\hat{x})\, t^{(n)}(\hat{x};\mu) \cdot \nabla\hat{u}(\hat{x})\, d\hat{x}, \end{aligned} \tag{3.5}$$

where the matrix $T^{(n)}$ and the vector $t^{(n)}$ are obtained in a straightforward way by the change of variable $x \in \Omega_n \mapsto \hat{x} \in \hat{\Omega}_n$ and the chain rule.

Hence, we have shifted the dependence on the parameter from the domain dependence to a dependence of the bilinear form, i.e., the coefficients of the corresponding bilinear forms $\hat{a}(\cdot,\cdot)$ and $\hat{b}(\cdot,\cdot)$ depend on the parameter $\mu \in D$. This allows us to use a reduced basis approach. However, we are still lacking an important requirement for obtaining an efficient algorithm, namely the separability of (3.4) w.r.t. the parameter μ. This will be discussed in the next section.

4 Empirical Interpolation Method

As already said above, reduced basis methods have already been investigated for parameter-dependent problems, see e.g. [11, 13, 18, 19, 25]. This approach, however, requires that the differential operator can be factorized so that the influence of the parameter can be separated from the bilinear form of the pde. The bilinear forms $\hat{a}(\cdot,\cdot)$ and $\hat{b}(\cdot,\cdot)$ associated with $\hat{A}$ and $\hat{B}$, respectively, do not have this form. This is basically due to the fact that the transformation is not affine linear.

In analogy to [12, 14], we aim to use the so-called *empirical interpolation method (EIM)*, [1], in order to approximate $\hat{A}$ and $\hat{B}$ by separable operators that will allow to use a reduced basis approach. Moreover, of course, a separation of the parameter from the differential operator also gives rise to efficient numerical methods. Let us now briefly describe the EIM and then describe its application to our problem.

Let $g : \Omega \times D \to \mathbb{R}$ be a function, depending on spatial coordinates $x \in \Omega$ and the parameter $\mu \in D$. The main idea of the empirical interpolation method is to construct M basis functions $q_m : \Omega \to \mathbb{R}$, $m = 1, \dots, M$, and M interpolation points ξ_m, $m = 1, \dots, M$, such that $g(x, \mu)$ is interpolated by

$$g_M(x,\mu) := \sum_{m=1}^{M} \lambda_m(\mu) q_m(x), \tag{4.1}$$

where (for one particular μ) the weights $\lambda_m(\mu)$, $m = 1, \dots, M$, are given by the solution of the following interpolation problem

$$g_M(\xi_j,\mu) = \sum_{m=1}^{M} \lambda_m(\mu) q_m(\xi_j) = g(\xi_j,\mu)\ , \quad j = 1, \dots, M. \tag{4.2}$$

In other words, we approximate a given function g by interpolating it by a tensor product g_M at appropriately chosen knots.

Furthermore, the number M of used basis functions is determined in such a way that a specified tolerance ε_{emp} is reached, i.e.,

$$\varepsilon_M(\mu) := \|g(x,\mu) - g_M(x,\mu)\|_{L^\infty(\Omega)} \leq \varepsilon_{emp}\ , \quad \forall \mu \in D. \tag{4.3}$$

One chooses the smallest M such that (4.3) is valid. Details on the construction of the basis functions and the choice of the interpolation knots can be found in [1].

We apply the EIM on each subdomain $\hat{\Omega}_n$ to approximate $T^{(n)}$ and $t^{(n)}$ by

$$\begin{aligned} \left\| T^{(n)}(\hat{x};\mu) - \sum_{m=1}^{M^{a,n}} \Theta_m^{(n)}(\mu)\, \Lambda_m^{(n)}(\hat{x}) \right\|_{L^\infty(\hat{\Omega}_n)} &\leq \varepsilon_{emp}, \\ \left\| t^{(n)}(\hat{x};\mu) - \sum_{m=1}^{M^{b,n}} \theta_m^{(n)}(\mu)\, \lambda_m^{(n)}(\hat{x}) \right\|_{L^\infty(\hat{\Omega}_n)} &\leq \varepsilon_{emp}, \end{aligned}$$

(where the norms are to be understood as those for vector fields) so that we obtain the following approximations of $\hat{A}$ and $\hat{B}$, respectively:

$$\begin{aligned}\left\langle \hat{A}^{emp}(\mu)\hat{u},\hat{v}\right\rangle &= \sum_{n=1}^{N_S}\int_{\hat{\Omega}_n}\nabla\hat{u}(\hat{x})\cdot\sum_{m=1}^{M^{a,n}}\Theta_m^{(n)}(\mu)\,\Lambda_m^{(n)}(\hat{x})\,\nabla\hat{v}(\hat{x})\,d\hat{x}\\ &=: \sum_{q=1}^{Q}\Theta^q(\mu)\,A^q(\hat{u},\hat{v})\end{aligned} \tag{4.4}$$

and similarly

$$\left\langle \hat{B}^{emp}(\mu)\hat{u},\hat{v}\right\rangle = \sum_{s=1}^{S}\Phi^s(\mu)B^s(\hat{u},\hat{v}). \tag{4.5}$$

We finally obtain the approximate variational problem: Find $\hat{u}\in\hat{V}(1)$, such that

$$\left\langle \hat{A}^{emp}(\mu)\hat{u},\hat{v}\right\rangle + \left\langle \hat{B}^{emp}(\mu)\hat{u},\hat{v}\right\rangle = 0, \tag{4.6}$$

for all $\hat{v}\in\hat{V}(0)$.

5 Reduced-Basis Approximation

Now we are going to describe the reduced basis method we are considering. To this end, we need to define our basis functions (also called *modes*) for the reduced model. Often this is done by defining certain so-called *snapshots* as a starting point. We take snapshots corresponding to the parameter μ, i.e., we fix N values $\mu_1,\dots,\mu_N$ for the parameter and compute the corresponding snapshots as discrete solutions of (3.4), i.e., find $\hat{u}(\mu_n)\in\hat{V}_h(1)$, such that

$$\left\langle \hat{A}(\mu_n)\hat{u}(\mu_n),\hat{v}\right\rangle + \left\langle \hat{B}(\mu_n)\hat{u}(\mu_n),\hat{v}\right\rangle = 0\ , \tag{5.1}$$

for all $\hat{v}\in\hat{V}_h(0)$, where $\hat{V}_h(g)$ is a discrete subspace of $\hat{V}(g)$ (e.g. a finite element space with mesh size h). Next, we homogenize the problem (with respect to the boundary conditions) by defining

$$\tilde{u}_i := \begin{cases}\hat{u}(\mu_1)\ , & \text{if } i=0\ ,\\ \hat{u}(\mu_{i+1})-\hat{u}(\mu_1)\ , & \text{if } i=1,\cdots,N-1,\end{cases} \tag{5.2}$$

i.e., $\tilde{u}_0$ satisfies the non-homogeneous boundary conditions, whereas all others fulfill homogeneous boundary conditions. We define the approximation space (trial space) by $\hat{V}_h^N(g) := \tilde{u}_0+\hat{V}_h^N(0)$, where $\hat{V}_h^N(0) := \operatorname{span}\{\tilde{u}_i : 1\le i\le N-1\}$ is the test space.

Using these spaces for any new parameter $\mu \in D$ we could directly apply a Galerkin projection to obtain the reduced basis approximation by solving the following problem: Find $\hat{u}^N(\mu) \in \hat{V}_h^N(1)$, such that

$$\left\langle \hat{A}(\mu)\hat{u}^N(\mu), \hat{v} \right\rangle + \left\langle \hat{B}(\mu)\hat{u}^N(\mu), \hat{v} \right\rangle = 0, \tag{5.3}$$

for all $\hat{v} \in \hat{V}_h^N(0)$. As this would not lead to an efficient algorithm, we take advantage of the derived approximation for (3.4) and solve instead the approximate problem: Find $\hat{u}^N(\mu) \in \hat{V}_h^N(1)$, such that

$$\left\langle \hat{A}^{emp}(\mu_n)\hat{u}^N(\mu_n), \hat{v} \right\rangle + \left\langle \hat{B}^{emp}(\mu_n)\hat{u}^N(\mu_n), \hat{v} \right\rangle = 0 \tag{5.4}$$

for all $\hat{v} \in \hat{V}_h^N(0)$.

5.1 Linear PDEs

Let us detail our approach first for a linear pde, where we can use the full computational power of the EIM.

Algorithm 5.1 *(Reduced basis approximation for linear problems).*
Given a tolerance ε_{emp}.
Offline-Stage:

1. Use EIM to compute Θ^q and Φ^s as above as well as the separated bilinear forms A^q and B^s.

2. Fix N values $\mu_1, \ldots, \mu_N$ for the parameter and compute the snapshots by solving the approximated problem (5.1).

Online-Stage: Given a new parameter $\mu \in D$

1. Compute $\Theta^q(\mu)$ and $\Phi^s(\mu)$ using (4.2). This requires the solution of $N_S \times 2 \times 2 \times 2$ (small) linear systems of equations.

2. Assemble the linear system for the reduced model. This requires $\mathcal{O}((Q+S)N^2) + \mathcal{O}((Q+S)N)$ operations.

3. Solve the reduced $(N \times N)$-system. This can be done in general in $\mathcal{O}(N^3)$ operations.

In the online stage the computational work is reduced from solving an N_h-dimensional (finite element) problem to a N-dimensional ($N \ll N_h$) problem. Obviously this approach is in particular more efficient if the convection-diffusion problem has to be solved several times.

5.2 Nonlinear PDEs

As we also want to deal with nonlinear pde's, we consider the following equation

$$-a\Delta u + g(u)b \cdot \nabla u + h(u)u = 0, \tag{5.5}$$

using the same boundary conditions as above. One standard way to solve such equations is by an iteration of linear problems of the form

$$-a\Delta u + g(U)b \cdot \nabla u + h(U)u = 0, \tag{5.6}$$

where U typically denotes a solution of some previous iteration. Thus, we can reduce basically everything to the linear case. However, we have to take the functions $g(U)$ and $h(U)$ into account for computing the EIM since otherwise we would have to recompute these terms in the online-stage in each iteration. With this modification (which also causes more computational work in the offline-stage) we can use the presented algorithm within this framework.

6 Numerical Results

We now present some numerical results for two linear and two nonlinear problems, namely

$$\begin{aligned} 0 &= -0.1\Delta u + (1,0)^T \cdot \nabla u, \\ 0 &= -\Delta u + (x,y)^T \cdot \nabla u, \\ 0 &= -0.1\Delta u + (0.5,0.5)^T u \cdot \nabla u, \\ 0 &= -0.1\Delta u + (0.5,0.5)^T u \cdot \nabla u + u^2. \end{aligned}$$

All computations have been performed with *FEMLAB* using finite elements.

6.1 One Blade

The geometry is shown in Figure 1. The mesh size is fixed for all computations to $h_{max} = 0.1$. The snapshots are taken by a uniform subdivision of the parameter interval $D = [0, \frac{\pi}{2}]$.

For different numbers of snapshots N, we compute the reduced basis approximation $\hat{u}^N(\mu)$ and compare it to the reference solution $\hat{u}(\mu)$, which is computed on the same mesh. This is done for $M = 5$ different angels of interest. We measure the error $e^N(\mu) := \hat{u}(\mu) - \hat{u}^N(\mu)$ terms of the following quantities:

$$\left\| \frac{\partial e^N(\mu)}{\partial n} \right\|_{L^1(\partial\hat{B})}, \quad \left\| \frac{\partial e^N(\mu)}{\partial n} \right\|_{L^2(\partial\hat{B})}, \quad \left\| e^N(\mu) \right\|_{L^\infty(\hat{\Omega})}, \quad \left\| e^N(\mu) \right\|_{L^2(\hat{\Omega})}.$$

The first two quantities reflect the fact that in several applications the interesting quantities (like e.g. forces or efficiency) are measured on the boundary of the blade. We display the average values of these quantities for M different angles.

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$,	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 2$	4.5761e-01	3.3881e-01	1.2539e-01	5.4620e-03
$N = 4$	1.6949e-02	6.4470e-04	3.7161e-03	8.0362e-06
$N = 6$	3.6594e-04	3.7995e-07	6.6941e-05	4.2811e-09
$N = 8$	5.2275e-06	7.5078e-11	1.1455e-06	1.0824e-12
$N = 10$	2.6008e-07	2.6797e-13	6.5634e-08	5.0167e-15

Table 1: Results for one blade and example 1.

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$,	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 2$	1.1024e-01	2.3864e-02	1.0885e-02	1.3184e-04
$N = 4$	3.7916e-03	2.8802e-05	1.8383e-04	3.4300e-08
$N = 6$	2.6304e-05	3.7832e-09	3.4220e-06	1.4072e-11
$N = 8$	5.7911e-07	1.7271e-12	7.8264e-08	1.1656e-14
$N = 10$	4.0959e-08	7.9742e-15	4.9690e-09	5.5172e-17

Table 2: Results for one blade and example 2.

The tables 1-4 show exponential decreasing errors. This is also shown in Figure 4, which shows the decay of all four error quantities for example 4. We also see that, as expected, the convergence is slower for the quantities on the boundary.

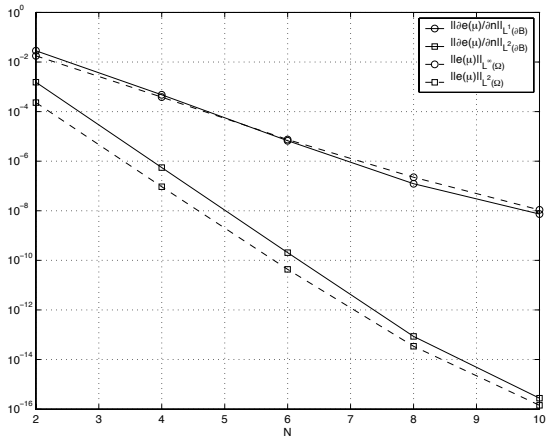

Figure 4: Example of one blade: Decay of the errors.

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$,	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 2$	1.7430e-01	5.3947e-02	5.0205e-02	1.4509e-03
$N = 4$	3.2225e-03	2.2602e-05	8.9602e-04	6.3792e-07
$N = 6$	4.3833e-05	6.0672e-09	2.1547e-05	4.2519e-10
$N = 8$	3.2445e-07	4.5052e-13	3.6249e-07	1.0633e-13
$N = 10$	1.4655e-08	1.2759e-15	1.9834e-08	4.1186e-16

Table 3: Results for one blade and example 3.

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$,	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 2$	2.8339e-02	1.5279e-03	1.7724e-02	2.2992e-04
$N = 4$	4.7547e-04	5.5136e-07	3.8283e-04	9.3087e-08
$N = 6$	6.6904e-06	2.0499e-10	7.4213e-06	4.3219e-11
$N = 8$	1.2306e-07	8.5338e-14	2.2431e-07	3.4243e-14
$N = 10$	7.3634e-09	2.7582e-16	1.0780e-08	1.4195e-16

Table 4: Results for one blade and example 4.

Another aspect of the reduced basis method is that once the snapshots are computed and the matrices are assembled, the reduced basis approximation can be obtained very fast. Though it is perfectly suited whenever rapid, repeated and reliable solutions of parameterized pde's are needed. To underline this, Figure 5 shows computing timings to obtain N_{sol} solutions (i.e., the solution for N_{sol} different parameter values μ). On the one hand, they are computed directly, on the other hand reduced basis method using $N = 10$ snapshots is applied. The left part of Figure 5 shows computing times for a linear problem (corresponding to example 2) and the right part shows computing timings for the nonlinear problem in example 4.

Here, we have not taken into account that for smaller mesh sizes h_{max} the time for obtaining the solution directly increases, while the time for obtaining the reduced basis approximation is constant, since the online-stages are independent of the used mesh. On the other hand, timings for obtaining the snapshots and assembling the needed matrices will increase too, but this only has to be done once.

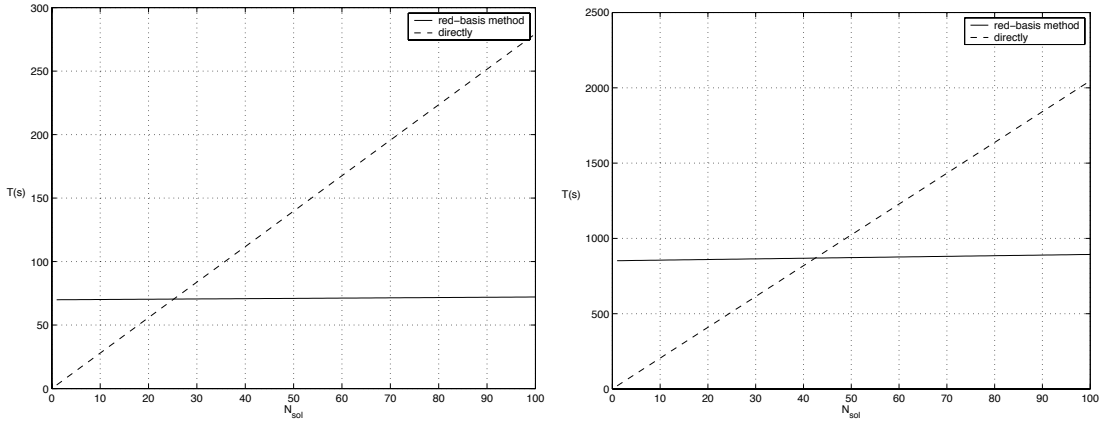

Figure 5: Comparison of computing times for one blade and for the linear example 2 (left) and the nonlinear problem example 4 (right).

6.2 Five Blades

The geometry is again shown Figure 1. The mesh size is fixed for all computations to $h_{max} = 0.2$. Again, we obtain snapshots by uniformly subdividing the parameter space D. We compare the same error quantities as above for $M = 5$ parameters.

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 20$	1.1964e-02	3.6420e-04	5.2278e-03	9.8293e-05
$N = 40$	9.1587e-04	2.0902e-06	4.2153e-04	2.9216e-07
$N = 60$	1.4821e-04	4.8827e-08	6.0365e-05	5.1978e-09
$N = 80$	4.0223e-05	4.1482e-09	1.4194e-05	3.7492e-10
$N = 100$	1.5575e-05	6.9286e-10	4.9104e-06	4.5514e-11
$N = 120$	6.8332e-06	1.2272e-10	2.3183e-06	1.0518e-11
$N = 140$	3.0928e-06	2.4590e-11	1.1610e-06	2.6126e-12
$N = 160$	1.1671e-06	3.5936e-12	4.2677e-07	3.8431e-13
$N = 180$	7.8726e-07	1.6177e-12	2.3441e-07	1.5186e-13

Table 5: Results for four blades, example 1.

Again, all tables show exponential decreasing errors, also graphically shown in Figure 6 for example 2. Due to the more complex space of parameters, one observes an increased number of snapshots needed to obtain a certain accuracy.

To compare computing timings we proceed as above. Figure 7 (left) shows computing

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 10$	5.0502e-04	6.8106e-07	6.3083e-05	3.1175e-09
$N = 20$	7.2797e-05	1.2904e-08	9.8053e-06	1.0385e-10
$N = 30$	1.6501e-05	7.7402e-10	1.6965e-06	2.3149e-12
$N = 40$	6.0813e-06	9.5160e-11	5.1187e-07	2.7740e-13
$N = 50$	2.3656e-06	1.4870e-11	2.0042e-07	6.2629e-14
$N = 60$	8.0245e-07	2.0297e-12	8.4087e-08	1.0942e-14
$N = 70$	3.6383e-07	3.3420e-13	3.0846e-08	2.0882e-15

Table 6: Results for four blades, example 2.

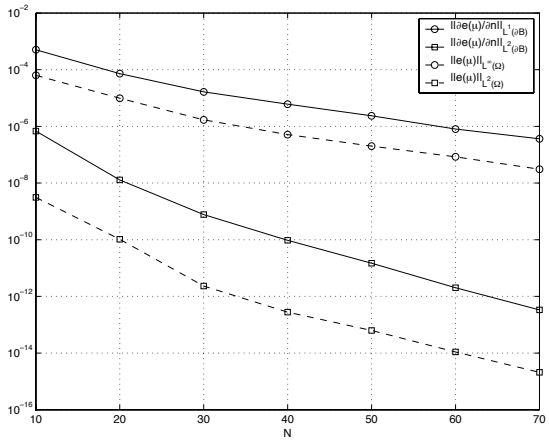

Figure 6: Five blades: decay of the errors.

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 20$	1.7994e-03	6.6581e-06	1.0316e-03	1.7604e-06
$N = 40$	1.3098e-04	3.6979e-08	7.9048e-05	8.2982e-09
$N = 60$	1.3817e-05	3.4869e-10	9.3197e-06	1.0002e-10
$N = 80$	5.1865e-06	5.5496e-11	3.2915e-06	1.6076e-11
$N = 100$	2.3986e-06	1.0469e-11	1.3001e-06	2.5292e-12
$N = 120$	8.1416e-07	1.3232e-12	4.5038e-07	2.8584e-13
$N = 140$	3.9471e-07	2.8873e-13	2.0356e-07	7.6517e-14
$N = 160$	2.0887e-07	8.1595e-14	1.0668e-07	1.9364e-14
$N = 180$	1.3694e-07	3.7260e-14	6.3443e-08	1.0606e-14

Table 7: Results for four blades, example 3.

timings needed for a linear problem (example 2) and the right part shows computing timings for a nonlinear problem (example 4, taking $N = 50$ snapshots). Compared to the case of one blade, the 'overhead' is slightly larger, once again because of the more complex parameter space and the more complex mapping, respectively. On the other hand, once the snapshots are computed and the empirical interpolation is done, we are able to obtain a reliable reduced basis approximation very fast.

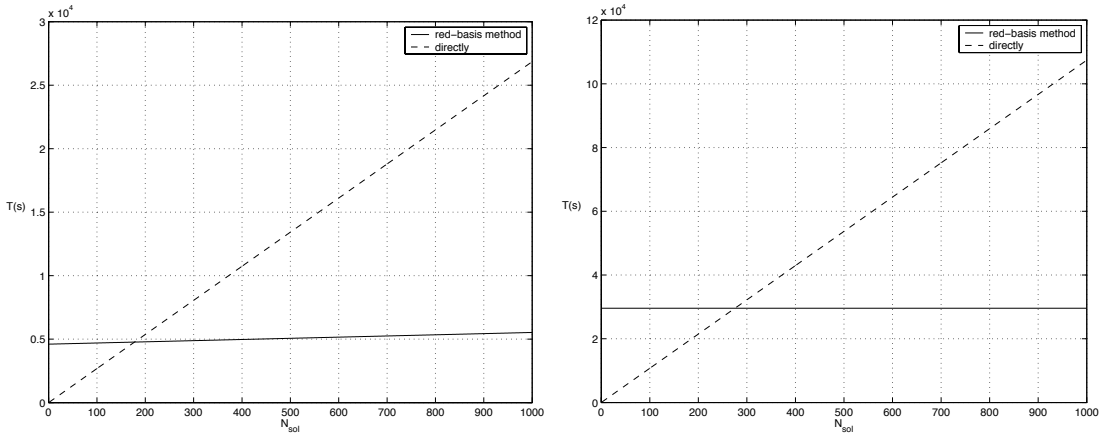

Figure 7: Five blades: Comparison of computing times for the linear example 2 (left) and the nonlinear problems 4 (right). We use $N = 50$ snapshots.

Finally, note that for a large number of snapshots N and a nonlinear problem the offline stage may require a huge amount of memory. The timings for saving and loading these

	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^1(\partial\hat{B})}$	$\left\\| \frac{\partial e^N(\mu)}{\partial n} \right\\|_{L^2(\partial\hat{B})}$	$\left\\| e^N(\mu) \right\\|_{L^\infty(\hat{\Omega})}$	$\left\\| e^N(\mu) \right\\|_{L^2(\hat{\Omega})}$
$N = 10$	8.5161e-04	1.4593e-06	5.8696e-04	3.6237e-07
$N = 20$	8.2204e-05	1.6552e-08	7.5417e-05	6.0244e-09
$N = 30$	1.8548e-05	6.9106e-10	1.2778e-05	1.3441e-10
$N = 40$	9.1968e-06	1.6472e-10	7.1211e-06	3.1829e-11
$N = 50$	1.8992e-06	7.8555e-12	1.7168e-06	2.3689e-12
$N = 60$	8.0668e-07	1.3775e-12	7.3932e-07	4.9827e-13
$N = 70$	3.6563e-07	2.3958e-13	3.5814e-07	9.1114e-14
$N = 80$	1.9058e-07	7.7435e-14	1.6374e-07	2.9236e-14
$N = 90$	9.2633e-08	1.6731e-14	9.2139e-08	8.2077e-15
$N = 100$	5.4029e-08	5.5711e-15	5.0178e-08	3.1985e-15
$N = 110$	2.8019e-08	1.5905e-15	2.4575e-08	1.0008e-15

Table 8: Results for four blades, example 4.

matrices is not taken into account in Figure 7. Furthermore, this is the reason for taking only $N = 50$ snapshots, as we obtain adequate accuracy, while keeping the complexity of the initial computation (offline-stage) and the complexity of the online-stage low.

7 Summary, Conclusion and Outlook

We have presented a reduced basis method for convection-diffusion problems around rigid bodies whose position and orientation is subject to the choice of a parameter. We have used a domain decomposition and mapping approach to reduce the problem to a reference situation. By the Empirical Interpolation Method we obtain a separation of the parameter from the differential operator. Then, we define basis functions for the reduced model by taking snapshots with respect to the parameter. We show numerical results that indicate exponential rate of convergence with respect to the number of snapshots.

As already indicated above, the error analysis of the above method will be described in [2]. The numerical results presented in this paper are very promising and make us confident that reduced basis methods can be used efficiently for such problems. Several directions for further research are obvious. The time-dependent case is currently under investigation and also systems of pde's (like e.g. in [5, 14, 16]) are considered. Finally, we did not yet investigate how the snapshot basis can be optimized e.g. by a Proper Orthogonal Decomposition (POD) or by using a-posteriori error estimates. The ultimate goal of this project is to use this method also for optimization problems.

REFERENCES

[1] Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.; *An 'Empirical Interpolation' Method: Application to Efficient Reduced-Basis Discretization of PDEs*, (2004).

[2] Canuto, C., Tonn, T., Urban, K.: *Error estimates for the POD method for convection-diffusion problems around rigid bodies*, in preparation.

[3] Fink, J.P., Rheinboldt, W.C.; *On the error behavior of the reduced basis technique for nonlinear finite element approximations*, Z. Angew. Math. Mech., **63**, 1, pp21-pp28 (1983).

[4] Holmes, P., Lumley, J.L., Berkooz, Gal; *Turbulence, Coherent Structures and Symmetry*, Cambridge University Press (1996).

[5] Ito, K., Ravindran, S.S.; *A reduced-order method for simulation and control of fluid flow*, Journal of Computational Physics, **143**, 2, pp403-425 (1998).

[6] Kunisch, K., Volkwein, S.; *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numer. Math. **90** (2001), no. 1, 117–148.

[7] Kunisch K., Volkwein, S.; *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Numer. Anal. **40** (2002), no. 2, 492–515.

[8] Lumley, J.L.; *The structure of inhomogeneous turbulence* in Atmospheric Turbulence and Radio Wave Propagation, A.M. Yaglom adn V.I. Tatarski, eds., Nauka, Moscow (1967).

[9] Lumley, J.L.; *Coherent structures in turbulence* in R.E. Meyer, editor, Transition and Turbulence, Academic Press, New York (1981).

[10] Lumley, J.L., Poje, A.; *Low-dimensional models for flows with density fluctuations*, Phys. Fluids **9** (1997) no. 7, 2023–2031.

[11] Maday, Y., Patera, A.T., Turicini, G.; *A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations*, J. Sci. Comput., **17**, 1, pp437-pp446 (2002).

[12] Nguyen, N.C., *Reduced-Basis Approximation and A Posteriori Error Bounds for Nonaffine and Nonlinear Partial Differential Equations: Application to inverse Analysis*, PhD Thesis, Singapore-MIT Alliance, National University of Singapore (2005).

[13] Nguyen, N.C., Veroy, K., Patera, A.T.; *Certified Real-Time Solution of Parametrized Partial Differential Equations*, Handbook of Materials Modeling. R. Catlow, H.Shercliff and S. Yip Eds., Kluwer Academic Publishing, to appear (2005).

[14] Patera, A.T., Rozza, G., Veory, K.; *Reduced Basis Methodologies for Stokes Equations in parametrized domains*, EPFL-IACS report 22.2004, pre-print (2004).

[15] Peraire, J., Willcox, K.; *Balanced model reduction via the proper orthogonal decomposition*, AIAA 2001-2611, (2001).

[16] Peterson, J.S.; *The reduced basis method for incompressible viscous flow calculations*, SIAM, J. Sci. Stat. Comput., **10**, 4, pp777-pp784 (1989).

[17] Porsching, T.A.; *Estimation of the Error in the Reduced Basis Method Solution of Nonlinear Equations*, Math. Comp. **45** (1985), no. 172, 487–496.

[18] Prud'homme, C., Patera, T.A.; *Reduced-basis output bounds for approximately parametrized elliptic coercive partial differential equations*, Computing and Visualization in Science, **6**, 2-3, pp147-pp162 (2004).

[19] Prud'homme, C., Rovas, D.V., Veroy, K., Maday, Y., Patera, A.T., Turicini, G.; *Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods*, J. Fluids Engineering, **172**, pp70-pp80 (2002).

[20] Prud'homme, C., Rovas, D.A., Veroy, K., Patera, T.A.; *Mathematical and Computational Framework for Reliable Real-Time Solution of Parametrized Partial Differential Equations*, M2AN, **36**, 5, pp747-pp771 (2002).

[21] Rheinboldt, W.C.; *On the theory and error estimation of the reduced basis method for multi-parameter problems*, Nonlinear Analysis, Theory, Methods and Applications, **21**, 11, pp849-pp858 (1993).

[22] Rovas, D.V., Machiels, L., Maday, Y.; *Reduced-basis output bound methods for parabolic problems*, Preprint Laboratoire Jacques-Louis Lions (2004).

[23] Sirovich, L.; *Turbulence and the dynamics of coherent structures* in Part I–III, Quarterly of Applied Mathematics, **45** (1987) no. 3.

[24] Tonn, T.: *Application of Reduced-Basis Method for Solving Time-Dependent Convection-Diffusion Problems with Moving Bodies*, internal report, Poiltecnico di Torino, Dipartimento die Matematica, 2006.

[25] Veroy, K., Patera, A.T.; *Reduced-basis approximation of the viscosity-parametrized incompressible Navier-Stokes equation: Rigorous a posteriori error bounds*, Proceedings of Singapore-MIT Alliance Symposium (2004).