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

A REDUCED BASIS ELEMENT METHOD FOR COMPLEX FLOW SYSTEMS

A. Emil Løvgren*, Yvon Maday† and Einar M. Rønquist††

*Department of Mathematical Sciences,
Norwegian University of Science and Technology,
7491 Trondheim, Norway
e-mail: alfemil@math.ntnu.no
web page: http://www.math.ntnu.no/∼alfemil
†Laboratoire Jacques-Louis Lions,
Université Pierre et Marie Curie, Boîte courrier 187,
75252 Paris, Cedex 05, France
e-mail: maday@ann.jussieu.fr
††Department of Mathematical Sciences
Norwegian University of Science and Technology,
7491 Trondheim, Norway
e-mail: ronquist@math.ntnu.no

Key words: reduced basis, domain decomposition, *a posteriori* error estimators, transfinite interpolation, empirical interpolation

Abstract. *The reduced basis element method is a new approach for approximating the solution of problems described by partial differential equations within domains belonging to a certain class. The method takes its roots in domain decomposition methods and reduced basis discretizations.[1–3] The basic idea is to first decompose the computational domain into smaller blocks that are topologically similar to a few reference shapes (or generic computational parts). Associated with each reference shape are precomputed solutions corresponding to the same governing partial differential equation, and similar boundary conditions, but solved for different choices of some underlying parameter. In this work, the parameters are representing the geometric shape associated with a computational part.[4,5] The approximation corresponding to the computational domain is then taken to be a linear combination of the precomputed solutions, mapped from the reference shapes for the different blocks to the actual domain. The variation of the geometry induces non-affine parameter dependence, and we apply the empirical interpolation technique to achieve an offline/online decoupling of the reduced basis procedure. Some results for incompressible flow systems have already been presented,[6–8] and the focus here will be to further improve the offline/online decoupling of problems with non-affine parameter dependence. To this end we use the empirical interpolation method[9,10] to approximate the parameter dependent operators. We also present a generalized transfinite interpolation method[11] intended to produce global C^1 mappings from the reference shapes to each corresponding block of the computational domain.*

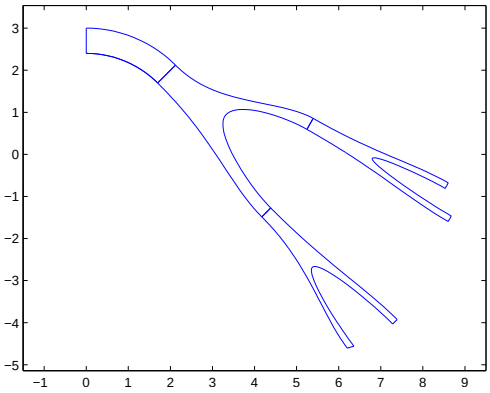


Figure 1: The computational domain Ω decomposed into building blocks.

1 INTRODUCTION

The reduced basis method is known to produce good approximations to parameter dependent problems where the solution u, or some quantity s derived from the solution, depends continuously on some parameter μ, e.g.

$$F(u(\mu);\mu) = 0, \quad s(u(\mu)) = f(u(\mu)). \tag{1.1}$$

The idea behind the reduced basis method is to split the computational process in two stages. In the first stage a lot of work is done to compute the solution of the problem (1.1) for many different choices of the parameter $\mu \in S_N = \{\mu_1, ..., \mu_N\}$. Each solution $u(\mu_i), i = 1, ..., N$, is stored and considered to be a basis function for the reduced basis approximation. In the second stage the solution of the given problem (1.1) for a new choice of μ is found as a linear combination of the basis functions computed in the first stage,

$$u_N(\mu) = \sum_{i=1}^{N} \alpha_i(\mu) u(\mu_i), \tag{1.2}$$

where the α_i are parameter dependent coefficients to be determined such that $F(u_N(\mu);\mu) = 0$. The parameters are found using a Galerkin method. Since the number of basis functions typically is very small compared to the number of degrees of freedom in the discretization used to compute the basis functions, the work needed to find the parameters is also very small.

The method is well suited for situations with many repetitive solves for reasonable variations in the parameter, like optimization problems, and problems where the traditional solution approach is too time consuming, like in control problems. Thus the first stage is called the offline stage, and the second stage is called the online stage. In order to certify the reduced basis approximation $u_N(\mu)$, *a posteriori* error estimation of the output of interest, $s_N = s(u_N(\mu))$, is used.

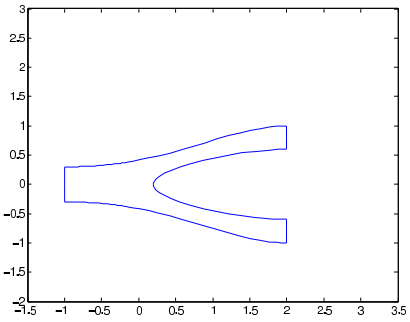


Figure 2: The reference bifurcation $\hat{\mathcal{B}}$.

In the reduced basis element method, the geometry of the computational domain is considered to be the parameter on which the solution depends.[4,5,12] The domain is decomposed into a union of non-overlapping building blocks (see Figure 1),

$$\overline{\Omega} = \overline{\Phi(\hat{\Lambda})} = \bigcup_{k=1}^{K} \overline{\Lambda^k}, \tag{1.3}$$

where each building block Λ^k is defined by a regular enough, one to one, mapping Φ^k of one of several reference domains $\hat{\Lambda}$, i.e., $\Lambda^k = \Phi^k(\hat{\Lambda})$. Throughout this text $\hat{\Lambda}$ will represent either a reference square $\hat{\Omega} = (0,1)^2$, or a reference bifurcation $\hat{\mathcal{B}}$; see Figure 2.

Corresponding to each reference domain is a set of precomputed basis functions found by solving the underlying problem (1.1) for a given set of preselected deformations $\Phi_i : \hat{\Lambda} \rightarrow \Lambda_i, i = 1, ..., N$. Each basis function $u_i = u(\Phi_i)$ is then mapped to the reference domain through an appropriate transformation, $\hat{u}_i = \Psi(u_i, \Phi_i)$, and stored.

On the computational domain Ω, the global solution is found as a linear combination of the precomputed basis functions on each building block, while applying a continuity constraint on the block interfaces.[13] The global reduced basis approximation is then

$$u_N(\Phi) = \bigcup_{k=1}^{K} \sum_{i=1}^{N_k} \alpha_i(\Phi^k) \Psi^{-1}(\hat{u}_i, \Phi^k), \tag{1.4}$$

where the coefficients $\alpha_i(\Phi^k)$ are determined such that $F(u_N(\Phi); \Phi) = 0$, and such that

$$\int_{\Gamma_{kl}} (u_{|_{\Lambda_k}} - u_{|_{\Lambda_l}}) \psi dS = 0, \quad \forall\, \psi \in W_{kl}, \tag{1.5}$$

where Γ_{kl} is the interface between two adjacent blocks in the decomposition of Ω, and W_{kl} is some low-dimensional functional space defined on this interface. In (1.4) N_k represents the number of precomputed basis functions associated with each reference domain; this number may in general be different.

The regularity requirements of the mappings Φ_i and Φ^k and the definition of the transformation Ψ of the basis functions, depends on the particular problem (1.1). If the basis functions are scalar fields, the transformation of the basis functions to the reference domain is defined by $\hat{u}_i = \Psi(u_i, \Phi_i) = u_i \circ \Phi_i$.

If the basis functions are vector fields associated with incompressible fluid flow, the mapping of the basis functions to the reference domain is defined by the Piola transformation,[14]

$$\hat{\boldsymbol{u}}_i = \Psi(\boldsymbol{u}_i, \Phi_i) = \mathcal{J}_i^{-1}(\boldsymbol{u}_i \circ \Phi_i)|J_i|, \tag{1.6}$$

where $\mathcal{J}_i$ is the Jacobian of Φ_i, and J_i its determinant. In order for the mapped basis functions to be continuous, the mappings Φ_i and Φ^k must be C^1 from $\hat{\Lambda}$ to Λ_i and Λ^k, since the Jacobian contains the derivatives of the mappings with respect to the reference coordinates.

One common way to achieve C^1 mappings is to solve a Laplace problem on the reference domain, with the coordinates of the boundary of each deformed domain as Dirichlet boundary conditions to the Laplace problem.[15] One Laplace problem has to be solved for each spatial direction, and on the computational domain Ω, the mapping for each building block has to be computed in the online stage of the reduced basis element method.

As an alternative to solving several Laplace problems in the online stage, we propose in Section 2 a generalization of the traditional transfinite interpolation method[16] in order to construct C^1 mappings for more general reference domains.[11] In this case most of the work will be done in the offline stage.

In addition we will see in Section 3 that when the geometry is considered to be the parameter in the reduced basis method, the problem will depend on the parameter in a non-affine sense. Again with the goal to do most of the computations in the offline stage, we present the empirical interpolation method[9] applied to the reduced basis element method.

In Section 4 we present some of the basics of the *a posteriori* error estimation used to certify the reduced basis element approximation. The focus will be on compliant output, but error estimation for non-compliant output is also possible.[3]

In the final section we present numerical examples of the reduced basis element method used to find the solution of the steady Stokes problem on different domains Ω, where $\overline{\Omega} = \overline{\Phi(\hat{\Lambda})} = \bigcup_{k=1}^{K} \overline{\Phi^k(\hat{\Lambda})}$. The domains have an inlet boundary Γ_{in}, an outlet boundary Γ_{out}, and walled boundaries Γ_w. The steady Stokes problem is defined as: Find the velocity $\boldsymbol{u} \in X(\Omega)$ and the pressure $p \in M(\Omega)$, such that

$$\begin{array}{lcll} a(\boldsymbol{u}, \boldsymbol{v}; \Phi) + b(\boldsymbol{v}, p; \Phi) & = & l(\boldsymbol{v}; \Phi) & \forall\, \boldsymbol{v} \in X(\Omega) \\ b(\boldsymbol{u}, q; \Phi) & = & 0 & \forall\, q \in M(\Omega), \end{array} \tag{1.7}$$

where $X(\Omega) = \{\boldsymbol{v} \in (H^1(\Omega))^2, \boldsymbol{v}_{\Gamma_w} = 0, v_{t_{|\Gamma_{in}}} = v_{t_{|\Gamma_{out}}} = 0\}$, and $M(\Omega) = L^2(\Omega)$. In addition, we have the Neumann type boundary conditions given by specifying $\sigma_n = \nu \frac{\partial u_n}{\partial n} - p$ to be $\sigma_n^{in} = -1$ along Γ_{in} and $\sigma_n^{out} = 0$ along Γ_{out}; here, u_n is the normal velocity component and $\partial/\partial n$ denotes the derivative in the outward normal direction.

On the same spaces $X(\Omega)$ and $M(\Omega)$, and with the same boundary conditions as for the steady Stokes problem, we also present an example of the reduced basis element method used to find the approximation of the steady Navier-Stokes problem when Ω consists of only one building block. The steady Navier-Stokes problem is given as: Find $\boldsymbol{u} \in X(\Omega)$ and $p \in M(\Omega)$ such that

$$\begin{array}{lcll} a(\boldsymbol{u},\boldsymbol{v};\Phi) + c(\boldsymbol{u},\boldsymbol{u},\boldsymbol{v};\Phi) + b(\boldsymbol{v},p;\Phi) & = & l(\boldsymbol{v};\Phi) & \forall\, \boldsymbol{v} \in X(\Omega) \\ b(\boldsymbol{u},q;\Phi) & = & 0 & \forall\, q \in M(\Omega). \end{array} \tag{1.8}$$

All velocity basis functions found as the solutions of (1.7) and (1.8) are divergence free, and reduced basis velocity solution may be found independently of the pressure.[6] To solve the coupled reduced basis element problems corresponding to (1.7) and (1.8) for both the velocity and the pressure, we have to make sure that the reduced basis solution spaces fulfill the inf-sup condition. One way of doing this is for each pressure basis function p_i to find $\boldsymbol{v}_i^e \in X(\hat{\Lambda})$, such that

$$\boldsymbol{v}_i^e = \arg \max_{\boldsymbol{u} \in X(\hat{\Lambda})} \frac{\int_{\hat{\Lambda}} \hat{p}_i \nabla \cdot \boldsymbol{u} d\hat{\Lambda}}{|\boldsymbol{u}|_{H^1(\hat{\Lambda})}}. \tag{1.9}$$

When these velocity fields are included in the reduced basis approximation, the inf-sup condition is fulfilled.

2 A GENERALIZED TRANSFINITE INTERPOLATION METHOD

When the reference domain is a square, we use the traditional transfinite interpolation method[16] to define a C^1 mapping from the reference domain to any deformed rectangle with corners smaller than 180^o. The idea is to construct the interior points of the physical domain as linear combinations of points on the boundaries.

On the reference square, $\hat{\Omega} = (0,1)^2$, we construct one-dimensional weight functions $\phi_i(r)$, such that for $r_0 = 0$ and $r_1 = 1$ we get

$$\phi_i(r_j) = \delta_{ij}, \quad 0 \leq i,j \leq 1. \tag{2.1}$$

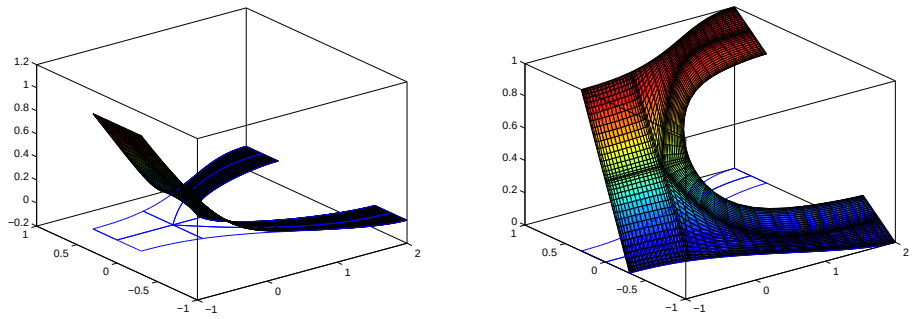
The one dimensional weight functions may be linear, but this is not a necessity. We may also use different weight functions in different spatial directions.

We assume that a representation of the boundaries of the physical domain is given with respect to the reference variables (ξ,η) by a bijective map. Each boundary will be the function of one variable, and we define the horizontal boundaries $\boldsymbol{x}(\xi,0)$ and $\boldsymbol{x}(\xi,1)$, and the vertical boundaries $\boldsymbol{x}(0,\eta)$ and $\boldsymbol{x}(1,\eta)$, where $\boldsymbol{x} = (x,y)$.

The one-dimensional transfinite interpolations (or projections), are then defined as

$$\Omega_\xi = \Phi_\xi(\hat{\Omega}) = \phi_0(\xi)\boldsymbol{x}(0,\eta) + \phi_1(\xi)\boldsymbol{x}(1,\eta) \tag{2.2}$$

$$\Omega_\eta = \Phi_\eta(\hat{\Omega}) = \phi_0(\eta)\boldsymbol{x}(\xi,0) + \phi_1(\eta)\boldsymbol{x}(\xi,1) \tag{2.3}$$



(a) The harmonic weight function associated with the left vertical side of the reference bifurcation.

(b) The harmonic projection onto the left vertical side of the reference bifurcation.

Figure 3: Non-linear harmonic functions used in the transfinite interpolation method for general reference domains.

$$\Omega_{\xi\eta} = \Phi_{\xi\eta}(\hat{\Omega}) = \sum_{i=0}^{1}\sum_{j=0}^{1} \phi_i(\xi)\phi_j(\eta)\boldsymbol{x}(r_i, r_j), \tag{2.4}$$

and the resulting two-dimensional projection is

$$\Omega = \Phi(\hat{\Omega}) = \Phi_\xi(\hat{\Omega}) + \Phi_\eta(\hat{\Omega}) - \Phi_{\xi\eta}(\hat{\Omega}). \tag{2.5}$$

The mapping (2.5) will preserve the boundaries of the physical domain, and the interior points are determined via a linear transformation of the grid points defined on the reference domain.

If the reference domain $\hat{\Lambda}$ is a deformed rectangle, or a more general geometry with more than four sides, for example a bifurcation, we can no longer use one-dimensional weight functions like the ones presented in (2.1). We now associate one weight function ϕ_i to each side $\hat{\Gamma}_i, i = 1, ...n$ of an n-sided reference domain $\hat{\Lambda}$. We let $\phi_i = 1$ on Γ_i, and solve the Laplace problem

$$\Delta\phi_i = 0 \quad \text{in } \hat{\Lambda}, \tag{2.6}$$

with homogeneous Neumann boundary conditions on the two sides of $\hat{\Lambda}$ adjacent to $\hat{\Gamma}_i$, and homogeneous Dirichlet boundary conditions on the remaining sides. On the reference square $\hat{\Omega}$, these harmonic weight functions will coincide with the one-dimensional weight functions defined in (2.1), but on a general reference domain, the weight functions will be non-linear C^1 functions; see Figure 3(a).

To define the transfinite interpolation on a general reference domain, we also need the projection of the interior onto each side $\hat{\Gamma}_i$. On the reference square these projections are

given by the reference coordinates as $(\xi, 0), (\xi, 1), (0, \eta)$, and $(1, \eta)$. On a general domain we compute the projection π_i onto the side $\hat{\Gamma}_i$ by solving the Laplace problem

$$\Delta \pi_i = 0 \quad \text{in } \hat{\Lambda}, \tag{2.7}$$

with the Dirichlet boundary condition along $\hat{\Gamma}_i$ distributed linearly from 0 to 1 with respect to arc-length. On the sides adjacent to $\hat{\Gamma}_i$ we set π_i equal to either 0 or 1, and on the remaining sides we use homogeneous Neumann boundary conditions. On the reference square this procedure would reproduce the reference coordinates, while on general reference domains we get nonlinear C^1 functions; see Figure 3(b). On the corresponding side Γ_i on a physical domain, the value of an internal point $(\hat{x}, \hat{y})$ on the reference domain is determined by the function $\psi_i(t) : [0, 1] \to \Gamma_i$, where t is found from $t = \pi_i(\hat{x}, \hat{y})$. Again, on the reference square the boundary functions are given by $\boldsymbol{x}(\xi, 0), \boldsymbol{x}(\xi, 1), \boldsymbol{x}(0, \eta)$, and $\boldsymbol{x}(1, \eta)$.

We now number the sides of an n-sided reference domain $\hat{\Lambda}$, with corresponding physical domain Λ, in a clockwise manner, and let $\boldsymbol{x}_i$ denote the corner between sides $\hat{\Gamma}_i$ and $\hat{\Gamma}_{i+1}$. Furthermore we let $\hat{\Gamma}_{n+1} = \hat{\Gamma}_1$, and define the transfinite interpolation

$$\Lambda = \Phi(\hat{\Lambda}) = \sum_{i=1}^{n} [\phi_i(\hat{\Lambda}) \psi_i(\pi_i(\hat{\Lambda})) - \phi_i(\hat{\Lambda}) \phi_{i+1}(\hat{\Lambda}) \boldsymbol{x}_i]. \tag{2.8}$$

The big benefit in using the extended transfinite interpolation in the context of the reduced basis element method, is that all the harmonic functions may be computed in the offline stage. In the online stage we only need to find the boundary functions ψ_i, and perform the linear combination of the harmonic functions in (2.8).

3 EMPIRICAL INTERPOLATION APPLIED TO THE REDUCED BASIS ELEMENT METHOD

One of the goals of the reduced basis method is to do all computations involving high resolution in the offline stage. When a Galerkin method is used to find the coefficients of the reduced basis approximation, the operators in the given problem operate on basis functions stored in some high resolution space. The elemental contribution to the stiffness matrix of the reduced basis problem on $\Omega = \Phi(\hat{\Lambda})$ from, say, the diffusion operator is expressed as

$$a(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_j; \Phi) = \nu \int_{\hat{\Lambda}} \mathcal{J}^{-T} \hat{\nabla} (\frac{1}{|J|} \mathcal{J} \hat{\boldsymbol{u}}_i) \cdot \mathcal{J}^{-T} \hat{\nabla} (\frac{1}{|J|} \mathcal{J} \hat{\boldsymbol{u}}_j) |J| d\hat{\Lambda}, \quad \text{for } i, j = 1, ..., N, \tag{3.1}$$

where $\tilde{\boldsymbol{u}}_i = \Psi^{-1}(\hat{\boldsymbol{u}}_i; \Phi)$ is a precomputed basis function mapped to the physical domain; see (1.6). Since the Jacobian of the mapping from the reference domain to the physical domain is involved in the operator, we can not precompute the elemental contribution of the operator directly. Instead we separate the parameter dependent contributions from

the precomputed basis functions by first decomposing each operator, such that for any $\boldsymbol{v}, \boldsymbol{w} \in X(\Omega)$

$$a(\boldsymbol{v}, \boldsymbol{w}; \Phi) = \nu \sum_{q=1}^{Q} \int_{\hat{\Lambda}} g^q(\Phi) a^q(\hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}) d\hat{\Lambda}, \tag{3.2}$$

where $Q = 17$ and all $a^q(\cdot, \cdot)$ are independent of Φ.[8] Similar decompositions are also possible for other operators with non-affine parameter dependence.

The idea is then to use empirical interpolation[9] to approximate each of the parameter dependent functions $g^q(\Phi)$ as a linear combination of only a few functions, $\tilde{g}_m^q = \tilde{g}^q(\Phi_m^q)$, where Φ_m^q are found within a predefined set of mappings. We get

$$g^q(\Phi) \approx \sum_{m=1}^{M_q} \beta_m^q(\Phi) \tilde{g}_m^q, \tag{3.3}$$

where only the M_q parameter dependent coefficients $\beta_m^q(\Phi)$ have to be computed in the online stage. Furthermore, the construction of the empirical interpolation is such that these coefficients can be found by sampling the function $g^q(\Phi)$ at M_q isolated points, and then invert a lower triangular matrix.

On the geometry given by the mapping Φ, the elemental contribution of the diffusion operator is then approximated by

$$a(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_j; \Phi) \approx \nu \sum_{q=1}^{Q} \sum_{m=1}^{M_q} \beta_m^q(\Phi) \int_{\hat{\Lambda}} \tilde{g}^q(\Phi_m) a^q(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{v}}_j) d\hat{\Lambda}. \tag{3.4}$$

Since both N, Q and M_q are small integers, it is reasonable to precompute the value of (3.4) for all combinations of i, j, q, and m. For the diffusion operator we then need to store QM_q matrices of size N^2. This is also done for the other operators in the given problem. In the online stage we find all the coefficients for the empirical interpolation approximations, and then assemble the total stiffness matrix for the reduced basis problem by adding the stored matrices together with the given coefficients as weights. The resulting N^2 stiffness matrix is then solved by a direct method to find the reduced basis coefficients.

We will in the next section see how the reduced basis coefficients may be used to compute the output of interest, and also to find error estimators of this output of interest, without assembling the reduced basis approximation itself. In this way, we never have to do any computations involving the high resolution basis used to compute the basis functions for the reduced basis element method. In shape optimization, say, we may compute the output of interest for several choices of the geometry, and choose the geometry best suited without ever explicitly constructing the underlying solution u_N.

4 *A POSTERIORI* ERROR ESTIMATION

We assume that we are not interested in the reduced basis solution itself, but rather some quantity derived from it. When the coefficients of the reduced basis solution is

found, the output of interest $s(\boldsymbol{u}_N) = f(\boldsymbol{u}_N; \Phi)$ can be found as

$$s(\boldsymbol{u}_N) = \sum_{i=1}^{N} \alpha_i(\Phi) f(\boldsymbol{u}_i; \Phi), \tag{4.1}$$

where the evaluation of the parameter dependent functional is done using the empirical interpolation method described in Section 3. Thus the computational effort needed to compute the output of interest is independent of the resolution of the underlying basis functions.

In order to assess the quality of this output, we also compute *a posteriori* error bounds $s^+(\boldsymbol{u}_N)$ and $s^-(\boldsymbol{u}_N)$,[3,17] such that for all Φ within the span of the parameters used to precompute the basis functions, we get

$$s^-(\boldsymbol{u}_N) \leq s(\boldsymbol{u}) \leq s^+(\boldsymbol{u}_N). \tag{4.2}$$

We present here the bounds in the compliant case $s(\boldsymbol{u}) = l(\boldsymbol{u})$, on domains with one building block.[8] For computational domains consisting of several building blocks, the work is ongoing.

We introduce an alternative diffusion operator on the reference domain $\hat{\Lambda}$,

$$\hat{a}(\boldsymbol{v}, \boldsymbol{w}; \Phi) = \int_{\hat{\Lambda}} g(\Phi) \hat{\nabla}(\boldsymbol{v} \circ \Phi) \cdot \hat{\nabla}(\boldsymbol{w} \circ \Phi) d\hat{\Lambda}, \tag{4.3}$$

where $\boldsymbol{v}$ and $\boldsymbol{w}$ are functions on Ω, and $g(\Phi)$ is a positive function depending on the mapping $\Phi : \hat{\Lambda} \rightarrow \Omega$, $g(\Phi)$ is chosen such that

$$\alpha_0 ||\boldsymbol{v}||_X^2 \leq \hat{a}(\boldsymbol{v}, \boldsymbol{v}) \leq a(\boldsymbol{v}, \boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in X(\Omega), \tag{4.4}$$

for some positive real constant α_0.

Since the Stokes operator is symmetric, the output bounds defined by

$$\begin{array}{rcl} s^-(\boldsymbol{u}_N) & = & l(\boldsymbol{u}_N) \\ s^+(\boldsymbol{u}_N) & = & l(\boldsymbol{u}_N) + \hat{a}(\boldsymbol{e}, \boldsymbol{e}), \end{array} \tag{4.5}$$

satisfy (4.2). Here the velocity field $\boldsymbol{e}$ is defined as the field that satisfies the residual equation

$$\hat{a}(\boldsymbol{e}, \boldsymbol{v}; \Phi) = l(\boldsymbol{v}; \Phi) - a(\boldsymbol{u}_N, \boldsymbol{v}; \Phi) - b(\boldsymbol{v}, p_N; \Phi) \quad \forall\, \boldsymbol{v} \in \tilde{X}(\Omega), \tag{4.6}$$

where $\tilde{X}(\Omega) = \{\boldsymbol{v} \circ \Phi \in (H^1(\hat{\Lambda}))^2,\ \boldsymbol{v}_{|\Gamma_w} = 0\}$.

For the Navier-Stokes operator we no longer have symmetry, and we need the reduced basis approximation $(\boldsymbol{\psi}_N, \lambda_N)$ of the linearized dual problem:[18] Find $\boldsymbol{\psi} \in X(\Omega)$ and $\lambda \in M(\Omega)$ such that

$$\begin{array}{lcll} a(\boldsymbol{v}, \boldsymbol{\psi}; \Phi) + c_1(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\psi}; \Phi) + b(\boldsymbol{v}, \lambda; \Phi) & = & -l(\boldsymbol{v}; \Phi) & \forall\, \boldsymbol{v} \in X(\Omega) \\ b(\boldsymbol{\psi}, q; \Phi) & = & 0 & \forall\, q \in M(\Omega), \end{array} \tag{4.7}$$

where

$$c_1(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\psi};\Phi) = c(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\psi};\Phi) + c(\boldsymbol{v},\boldsymbol{u},\boldsymbol{\psi};\Phi), \tag{4.8}$$

and $\boldsymbol{u}$ is the velocity solution of (1.8). To this end we need to compute basis functions for the problem (4.7), in addition to the basis functions found for the primal problem (1.8).

We define the primal and dual residuals

$$\begin{array}{rcl} \mathcal{R}^{pr}((\boldsymbol{u}_N,p_N);\boldsymbol{v};\Phi) & = & l(\boldsymbol{v};\Phi) - a(\boldsymbol{u}_N,\boldsymbol{v};\Phi) - c(\boldsymbol{u}_N,\boldsymbol{u}_N,\boldsymbol{v};\Phi) - b(\boldsymbol{v},p_N;\Phi), \\ \mathcal{R}^{du}(\boldsymbol{u}_N;(\boldsymbol{\psi}_N,\lambda_N);\boldsymbol{v};\Phi) & = & -l(\boldsymbol{v};\Phi) - a(\boldsymbol{v},\boldsymbol{\psi}_N;\Phi) - c_1(\boldsymbol{u}_N,\boldsymbol{v},\boldsymbol{\psi}_N;\Phi) - b(\boldsymbol{v},\lambda_N;\Phi), \end{array} \tag{4.9}$$

and solve the two residual equations similar to (4.6),

$$\begin{array}{lll} 2\hat{a}(\boldsymbol{e}^{pr},\boldsymbol{v};\Phi) & = \mathcal{R}^{pr}((\boldsymbol{u}_N,p_N);\boldsymbol{v};\Phi) & \forall \boldsymbol{v}\in\tilde{X}(\Omega) \\ 2\hat{a}(\boldsymbol{e}^{du},\boldsymbol{v};\Phi) & = \mathcal{R}^{du}(\boldsymbol{u}_N;(\boldsymbol{\psi}_N,\lambda_N);\boldsymbol{v};\Phi) & \forall \boldsymbol{v}\in\tilde{X}(\Omega). \end{array} \tag{4.10}$$

Then the output bounds for the steady Navier-Stokes problem defined as

$$s_N^{\pm} = l(\boldsymbol{u}_N;\Phi) - \mathcal{R}^{pr}((\boldsymbol{u}_N,p_N);\boldsymbol{\psi}_N;\Phi) \pm \kappa\hat{a}(\boldsymbol{e}^{\pm},\boldsymbol{e}^{\pm};\Phi), \tag{4.11}$$

where κ is a strictly positive number, and

$$\boldsymbol{e}^{\pm} = \boldsymbol{e}^{pr} \mp \frac{1}{\kappa}\boldsymbol{e}^{du}, \tag{4.12}$$

satisfies (4.2).

The *a posteriori* output bounds can be used in order to select the basis functions which in an optimal way represent the set of all parameters in the parameter space.[19] Based on a greedy algorithm, the first parameter is selected at random. If we assume that m parameters with corresponding basis functions are already selected, the next parameter is found among the remaining parameters as the one whose corresponding solution maximizes the output bound

$$s^{+}(\boldsymbol{u}_{N_m}) - s^{-}(\boldsymbol{u}_{N_m}). \tag{4.13}$$

By repeating this process until (4.13) is below a certain limit, we reduce the total number of basis functions in the reduced basis approximation.

By applying the empirical interpolation method described in the previous section, we may compute the solutions of (4.6), or (4.10), for each basis function in the offline stage, and in the online stage we find the output error bounds by assembling all contributions to (4.5), or (4.11), together with the reduced basis coefficients.

5 NUMERICAL EXAMPLES

We now present some examples of the reduced basis element method applied to different geometric structures. The first is a bifurcation, where the solution of the steady Stokes problem is found as a linear combination of global basis functions. The second structure

is a hierarchical system consisting of one pipe and three bifurcations. The solution of the steady Stokes problem is now found as a linear combination on each block structure, i.e. pipe or bifurcation, and glued together with Lagrange multipliers across the block interfaces. The third structure is a "bypass" system with three pipe blocks and two bifurcation blocks. Again we present the solution of the steady Stokes problem, with the linear combinations on each block glued together with Lagrange multipliers across block interfaces. The final structure is a deformed quarter annulus, and we now solve the steady Navier-Stokes problem using global basis functions.

The offline/online decoupling is illustrated below in all its detail, but for the examples, not all steps are included. In all the examples we have used spectral elements and the $\mathbb{P}_{\mathcal{N}} \times \mathbb{P}_{\mathcal{N}-2}$ method[20] with $\mathcal{N} = 20$ to compute the basis functions and the reference solution on the physical domains.
Offline:

1 Solve $\Delta\phi_j = 0$ and $\Delta\pi_j = 0$ on $\hat{\Lambda}$, for $j = 1, ..., n$.

2 Compute the basis geometries $\Lambda_i = \Phi_i(\hat{\Lambda})$, for $i = 1, ...N$ using the generalized transfinite interpolation (2.8).

3 Compute the basis functions $(\boldsymbol{u}_i, p_i)$ by solving (1.7) or (1.8) on Λ_i, for $i = 1, ...N$.

4 Enrich the velocity basis by solving (1.9), for $i = 1, ..., N$.

5 Use the selection algorithm to choose the optimal basis functions.

6 Compute and store the interpolation points needed in the empirical interpolation, and store the corresponding lower triangular matrices.

Online:

7 Compute the different blocks Λ^k of the generic domain Ω using the generalized transfinite interpolation (2.8).

8 Compute the coefficients needed in the empirical interpolation by sampling the different parameter dependent functions in (3.2) in the stored interpolation points and solving the corresponding lower triangular matrices.

9 Assemble all contributions to the reduced basis stiffness matrix and the right hand side.

10 Solve for the reduced basis coefficients.

11 Assemble all contributions to the output of interest and the *a posteriori* error estimators.

N	$\|\boldsymbol{u}_N - \boldsymbol{u}\|_{H^1}$	$\|\|p_N - p\|\|_{L^2}$	$s_N^+ - s$	$s - s_N^-$
1	$1.4 \cdot 10^{-2}$	$8.8 \cdot 10^{-2}$	$1.9 \cdot 10^{-3}$	$2.1 \cdot 10^{-4}$
5	$5.0 \cdot 10^{-4}$	$4.8 \cdot 10^{-3}$	$1.1 \cdot 10^{-5}$	$2.5 \cdot 10^{-7}$
10	$9.9 \cdot 10^{-6}$	$7.2 \cdot 10^{-5}$	$7.3 \cdot 10^{-9}$	$9.8 \cdot 10^{-11}$
15	$4.0 \cdot 10^{-6}$	$7.3 \cdot 10^{-6}$	$1.5 \cdot 10^{-10}$	$1.6 \cdot 10^{-11}$

Table 1: The reduced basis error on a single bifurcation. N is the number of basis functions after the selection algorithm is applied.

N	N_1	N_2	$\|\boldsymbol{u}_N - \boldsymbol{u}\|_{H^1}$	$\|\|p_N - p\|\|_{L^2}$
36	9	9	$2.6 \cdot 10^{-3}$	$4.0 \cdot 10^{-1}$
44	11	11	$1.7 \cdot 10^{-3}$	$6.6 \cdot 10^{-2}$
52	13	13	$1.2 \cdot 10^{-3}$	$4.9 \cdot 10^{-2}$
65	15	15	$1.1 \cdot 10^{-3}$	$3.7 \cdot 10^{-2}$
105	15	30	$4.2 \cdot 10^{-4}$	$6.3 \cdot 10^{-3}$

Table 2: The error in the reduced basis steady Stokes solution on a multi-block system corresponding to Figure 1. $N = N_1 + 3N_2$ is the total number of basis functions used. N_1 is the number of basis geometries used to generate the basis functions on the pipe block, N_2 is the number of basis functions used on the bifurcation blocks.

5.1 Steady Stokes: Bifurcation

We consider bifurcations characterized by the length and angle of the upper leg relative to the length and angle of the lower leg. In the tensor product parameter space generated by eight relative lengths and eight relative angles, we generate 64 bifurcations. We precompute the steady Stokes solutions on these bifurcations, and store them on $\hat{\Omega}$. Again we compute the associated enriched velocity solutions, but before we find the reduced basis solution we apply the selection algorithm described earlier. For all the steady Stokes problem presented, we have omitted step 6 in the offline stage and step 8 in the online stage. The assembling in steps 9 and 11 is done by computing all products online. The time spent in the online stage is still negligible compared to the computation of the reference solution.

The resulting errors in velocity and pressure are presented in Table 1, and we see that the convergence is very good. In this single-block case we may apply the *a posteriori* error analysis described in Section 4, and we compute both the upper and the lower bound gaps.

5.2 Steady Stokes: Hierarchical flow system

An example of a multi-block domain comprising both pipe blocks and bifurcation blocks, is the complex flow system shown in Figure 1. For the pipes we compute basis functions on deformed squares comprising two pipe blocks, but we only use the restrictions to the inflow domain, while we for the bifurcations precompute the solutions on the

N	N_1	N_2	$\|\boldsymbol{u}_N - \boldsymbol{u}\|_{H^1}$	$\|\|p_N - p\|\|_{L^2}$
45	9	9	$9.3 \cdot 10^{-3}$	$3.3 \cdot 10$
55	11	11	$3.1 \cdot 10^{-3}$	$5.3 \cdot 10^{-1}$
65	13	13	$2.3 \cdot 10^{-3}$	$9.0 \cdot 10^{-2}$
75	15	15	$1.4 \cdot 10^{-3}$	$5.3 \cdot 10^{-2}$
105	15	30	$5.4 \cdot 10^{-4}$	$3.0 \cdot 10^{-2}$

Table 3: The error in the reduced basis steady Stokes solution on a multi-block bypass with three pipe blocks and two bifurcation blocks. $N = 3N_1 + 2N_2$ is the total number of basis functions used. N_1 is the number of basis geometries used to generate the basis functions on the pipe block, N_2 is the number of basis functions used on the bifurcation blocks.

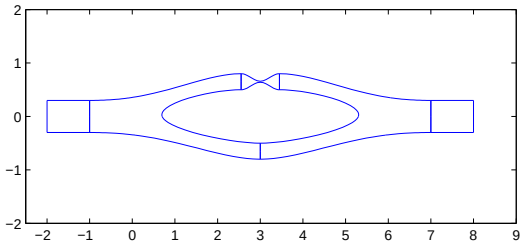


Figure 4: The bypass with three pipe blocks and two bifurcation blocks.

bifurcations described above by adding pipe blocks to the inflow and outflow boundaries in order to get the right boundary conditions. Only the restrictions of the solutions to the bifurcation block are stored and used as basis solutions. For the pipe block on the physical domain we use all 15 precomputed solutions, while we for the bifurcation blocks again use the selection process to limit the number of precomputed solutions to 30. To glue the blocks together across block interfaces, we use Lagrange multipliers. In Table 2 we see how the errors in velocity and pressure behave as the number of basis functions increases.

5.3 Steady Stokes: A "bypass"

As the final example we combine both block structures in the bypass system shown in Figure 4. Here the upper branch illustrates the effect of a clogged vein, while the lower branch is the bypass-vein. To model this domain with the reduced basis element method, we use snapshot solutions computed on three-domain pipes to generate the basis functions for the pipe blocks. The restriction of the snapshot solutions to each of the three sub-domains are now used as basis functions on their respective pipe block in the bypass system. As basis functions for the bifurcation blocks we use the same basis functions that were used on the hierarchical flow system in the previous example.

In this case we have two more block-interfaces compared to the hierarchical flow system,

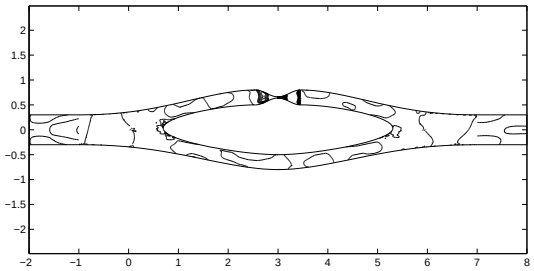


Figure 5: The contour of the error in the reduced basis pressure solution p_N when $N_1 = 15$ and $N_2 = 30$.

$N/2$	$\|\boldsymbol{u}_N - \boldsymbol{u}\|_{H^1}$	$\|\|p_N - p\|\|_{L^2}$	$s_N^+ - s$	$s - s_N^-$
1	$1.5 \cdot 10^{-1}$	$1.2 \cdot 10^{-1}$	1.1	$3.9 \cdot 10^{-3}$
2	$1.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	$1.0 \cdot 10^{-3}$
3	$2.1 \cdot 10^{-3}$	$8.7 \cdot 10^{-4}$	$3.7 \cdot 10^{-4}$	$2.7 \cdot 10^{-4}$
4	$6.7 \cdot 10^{-5}$	$2.4 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	$5.4 \cdot 10^{-6}$
5	$2.2 \cdot 10^{-5}$	$1.3 \cdot 10^{-6}$	$8.6 \cdot 10^{-7}$	$5.9 \cdot 10^{-7}$
6	$2.1 \cdot 10^{-5}$	$1.3 \cdot 10^{-6}$	$6.4 \cdot 10^{-7}$	$4.7 \cdot 10^{-7}$
7	$2.1 \cdot 10^{-5}$	$1.2 \cdot 10^{-6}$	$2.4 \cdot 10^{-7}$	$1.3 \cdot 10^{-7}$

Table 4: The error in the reduced basis solution of the Navier-Stokes problem on a single-block domain. Here we have used the polynomial degree $\mathcal{N} = 20$, but we have not used the empirical interpolation method.

each contributing eight constraints on the reduced basis velocity solution $\boldsymbol{u}_N$ (2 constraints in each spatial direction for each half of one interface). We see in Table 3 that the error convergence is good, but if too few basis functions are used we get spurious pressure modes due to the severe constraints on the reduced basis velocity space $X_N(\Omega)$.

In Figure 4 we present a contour plot of the error in the reduced basis pressure solution p_N when $N_1 = 15$ and $N_2 = 30$. Most of the error is located around the pipe block modeling the clogged vein.

5.4 Steady Navier-Stokes: Pipe

The experiment on the steady Navier-Stokes problem is done on a monodomain pipe. The basis functions are found on a deformed quarter annulus by varying the deformation of the inner curved boundary. To solve the steady Navier-Stokes problem, we consider the corresponding time-dependent problem, and iterate in time until we reach a steady state solution. In this way we find seven basis function, and a reference solution. The convergence of the reduced basis method is presented in Table 4.

In the last computations we have included the empirical interpolation, and we see from Table 5 that the convergence is still very good.

$N/2$	$\lvert u_N - u_{\mathcal{N}}\rvert_{H^1}$	$\lvert\lvert p_N - p_{\mathcal{N}}\rvert\rvert_{L^2}$	$s_N^+ - s$	$s - s_N^-$
1	$1.5 \cdot 10^{-1}$	$1.2 \cdot 10^{-1}$	1.1	$3.9 \cdot 10^{-3}$
2	$1.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$1.0 \cdot 10^{-3}$
3	$2.0 \cdot 10^{-3}$	$8.8 \cdot 10^{-4}$	$3.8 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$
4	$1.1 \cdot 10^{-4}$	$2.5 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	$5.8 \cdot 10^{-6}$
5	$5.3 \cdot 10^{-5}$	$3.6 \cdot 10^{-6}$	$2.5 \cdot 10^{-6}$	$1.5 \cdot 10^{-6}$
6	$5.1 \cdot 10^{-5}$	$3.4 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$	$1.4 \cdot 10^{-6}$
7	$3.9 \cdot 10^{-5}$	$2.1 \cdot 10^{-6}$	$2.2 \cdot 10^{-7}$	$4.2 \cdot 10^{-7}$

Table 5: The error in the reduced basis solution of the Navier-Stokes problem on a single-block domain. Results when we use empirical interpolation, and the polynomial degree $\mathcal{N} = 34$.

6 CONCLUSIONS

We have seen how the reduced basis element method works on fluid flow problems when the geometry is considered to be a parameter. An extension of the traditional transfinite interpolation method has been presented in order to improve the offline/online decoupling of the reduced basis problem. In addition we have used empirical interpolation for problems with non-affine parameter dependence, and we have shown that this method allows for offline/online decoupling of both the reduced basis solution, and the *a posteriori* error bounds. The *a posteriori* error bounds have been presented for domains comprising one building block. The multi-block case will be treated in a forthcoming paper. Other remaining issues include the extension to time-dependent problems, possibly with moving boundaries, and extension to three dimensional domains.

Acknowledgement. *This work has been supported by the Research Council of Norway under contract 10278900 and through the BeMatA programme under contract 147044/431, by the RTN project HaeMOdel HPRN-CT-2002-00270, and the ACI project "le-poumon-vous-dis-je" granted by the Fond National pour la Science. The support is gratefully acknowledged.*

REFERENCES

[1] J.P. Fink and W.C. Rheinboldt. On the error behaviour of the reduced basis technique in nonlinear finite element approximations. *Z. Angew. Math. Mech.*, Vol. **63**, 21–28, (1983).

[2] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA J.*, Vol. **18**(4), 455–462, (1980).

[3] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced basis output bound methods. *J. Fluids Engineering*, Vol. **124**, 70–80, (2002).

[4] Y. Maday and E.M. Rønquist. A reduced-basis element method. *J. Sci. Comput.*, Vol. **17**, 447–459, (2002).

[5] Y. Maday and E.M. Rønquist. The reduced-basis element method: Application to a thermal fin problem. *SIAM J. Sci. Comput.*, Vol. **26**(1), 240–258, (2004).

[6] A.E. Løvgren, Y.Maday, E.M. Rønquist. A Reduced Basis Element Method for the Steady Stokes Problem. *To appear in M2AN*, (2006).

[7] A.E. Løvgren, Y.Maday, E.M. Rønquist. A Reduced Basis Element Method for the Steady Stokes Problem: Application to hierarchical flow systems. *To appear in MIC*, (2006).

[8] A.E. Løvgren. Reduced basis modeling of hierarchical flow systems. *Ph.D. thesis, Norwegian University of Science and Technology*, (2005).

[9] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Serie I*, Vol. **339**, 667–672, (2004).

[10] M.A. Grepl, Y. Maday, N.C. Nguyen and A.T. Patera. Efficient approximation and *a posteriori* error estimation for reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Submitted to M2AN*

[11] A.E. Løvgren, Y. Maday, E.M. Rønquist. Transfinite interpolation based on non-tensorized reference domains. *In progress*, (2006).

[12] G. Rozza. Reduced-basis methods for elliptic equations in sub-domains with a posteriori error bounds and adaptivity. *To appear in Applied Numerical Mathematics*, (2005).

[13] B.F. Belgacem, C. Bernardi, N. Chorfi, and Y. Maday. Inf-sup conditions for the mortar spectral element discretization of the Stokes problem. *Numer. Math.*, Vol. **85**, 257–281, (2000).

[14] F. Brezzi and M. Fortin. Mixed and Hybrid Finite Element Methods. *Springer Verlag*, (1991).

[15] L. Formaggia and F. Nobile. A stability analysis for the Arbitrary Lagrangian Eulerian formulation with finite elements. *East-West J. Numer. Math.*, Vol. **7**, 105–131, (1999).

[16] W.J. Gordon and C.A. Hall. Transfinite Element Methods: Blending-Function Interpolation over Arbitrary Curved Element Domains. *Numer. Math.*, Vol. **21**, 109–129, (1973).

[17] D.V. Rovas. Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations. *Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA*, (2002).

[18] L. Machiels, J. Peraire, and A.T. Patera. *A Posteriori* Finite-Element Output Bounds for the Incompressible Navier-Stokes Equations: Application to a Natural Convection Problem. *J. Comput. Phys.*, Vol. **172**, 401–425, (2001).

[19] K. Veroy, C. Prud'homme, D.V. Rovas, A.T. Patera. *A Posteriori* Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations (AIAA Paper 2003-3847). *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, (2003).

[20] Y. Maday, A.T. Patera, and E.M. Rønquist. The $P_N \times P_{N-2}$ method for the approximation of the Stokes Problem. *Technical Report No. 92009, Department of Mechanical Engineering, Massachusetts Institute of Technology*, (1992).