Universität Stuttgart

SimTech
Cluster of Excellence

N. Jung[a] · **B. Haasdonk**[b] · **D. Kröner**[c]

Reduced Basis Method for Quadratically Nonlinear Transport Equations

Stuttgart, July 2009

[a] Institute of Automatic Control, Technical University of Munich
Boltzmannstraße 15, 85748 Garching/ Germany
nadine.jung@tum.de
http://www.rt.mw.tum.de

[b] Institute of Applied Analysis and Numerical Simulation, University of Stuttgart
Pfaffenwaldring 57, 70569 Stuttgart/ Germany
haasdonk@mathematik.uni-stuttgart.de
http://www.ians.uni-stuttgart.de/agh

[c] Institute of Applied Mathematics, University of Freiburg
Hermann-Herder-Str. 10, 79104 Freiburg/ Germany
dietmar@mathematik.uni-freiburg.de
http://www.mathematik.uni-freiburg.de/IAM

Abstract If many numerical solutions of parametrized partial differential equations have to be computed for varying parameters, usual finite element methods (FEM) suffer from too high computational costs. This can occur in optimization, control, parameter estimation, etc. The Reduced Basis Method (RBM) is a technique for model reduction of such highly resolved discretization schemes. The RBM allows to solve parametrized problems faster than by a direct FEM. Consequently, the RBM is suitable for such multi-query settings.
In the current presentation we extend the RBM for the stationary viscous Burgers equation to the time-dependent case and general quadratically nonlinear transport equations. We present the RB-algorithm and its offline/online decomposition. A posteriori error estimators justify the approach. Numerical experiments on a parameter-dependent transport problem, discretized with backward Euler, Newton Method and FEM, demonstrate the applicability of the model reduction technique. Comparison of the CPU times for RBM and FEM demonstrates the efficiency, in particular we observe an acceleration factor of up to 66.

Keywords Model Reduction · Reduced Basis Methods · Parameter Dependent Transport Equation · Time-dependent Viscous Burgers Equation · A Posteriori Error Estimates

Preprint Series **Issue No. 2009-15**
Stuttgart Research Centre for Simulation Technology (SRC SimTech)
SimTech – Cluster of Excellence
Pfaffenwaldring 7a
70569 Stuttgart
publications@simtech.uni-stuttgart.de
www.simtech.uni-stuttgart.de

REDUCED BASIS METHOD FOR QUADRATICALLY NONLINEAR TRANSPORT EQUATIONS

N. JUNG*, B. HAASDONK+ AND D. KRÖNER†

* Lehrstuhl für Regelungstechnik, Technische Universität München, Boltzmannstraße 15, 85748 Garching, Germany.
E-mail: nadine.jung@tum.de

+ Institut für Angewandte Analysis und Numerische Simulation, Universität Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany.
E-mail: haasdonk@mathematik.uni-stuttgart.de

† Abteilung für Angewandte Mathematik, Albert-Ludwigs-Universität Freiburg, Hermann-Herder-Str. 10, 79104 Freiburg, Germany.
E-mail: dietmar@mathematik.uni-freiburg.de

Abstract. If many numerical solutions of parametrized partial differential equations have to be computed for varying parameters, usual finite element methods (FEM) suffer from too high computational costs. This can occur in optimization, control, parameter estimation, etc. The Reduced Basis Method (RBM) is a technique for model reduction of such highly resolved discretization schemes. The RBM allows to solve parametrized problems faster than by a direct FEM. Consequently, the RBM is suitable for such multi-query settings.
In the current presentation we extend the RBM for the stationary viscous Burgers equation to the time-dependent case and general quadratically nonlinear transport equations. We present the RB-algorithm and its offline/online decomposition. A posteriori error estimators justify the approach. Numerical experiments on a parameter-dependent transport problem, discretized with backward Euler, Newton Method and FEM, demonstrate the applicability of the model reduction technique. Comparison of the CPU times for RBM and FEM demonstrates the efficiency, in particular we observe an acceleration factor of up to 66.

1. Introduction. We present the *Reduced Basis Method (RBM)* applied to a parametrized quadratically nonlinear transport equation of the form

$$\partial_t \bar{u}(\mu) - \operatorname{div}[\eta(\mu)\nabla\bar{u}(\mu) - \vec{c}(\mu)(\bar{u}(\mu))^2] = p(\mu) \qquad \text{in } \Omega\times[0,T] \tag{1.1}$$

$$\bar{u}(\mu,0) = u_0(\mu) \qquad \text{in } \Omega \tag{1.2}$$

$$\bar{u}(\mu) = u_d(\mu) \qquad \text{on } \partial\Omega\times[0,T]. \tag{1.3}$$

Here, $\Omega \subset \mathbb{R}^2$ denotes a domain with a Lipschitz-continuous boundary $\partial\Omega$. With $T > 0$ we denote the end time. The parameter vector is denoted by $\mu \in D \subset \mathbb{R}^d$, where D is the parameter domain. In engineering applications the parameter vector expresses geometrical, physical or control parameters. The parameter dependent viscosity $\eta(\mu) \geq I_\eta > 0$ which is assumed to be strictly positive with lower bound I_η, the velocity $\vec{c}(\mu)$, the Dirichlet values $u_d(\mu)$, the source term $p(\mu)$ and the initial data function $u_0(\mu)$ may in general be space- and time-dependent. We generally assume that all parameter functions depend affinely on the parameter. This means, e.g. for the viscosity $\eta(\mu)$, that for appropriately chosen parameter-dependent functions $\Theta_\eta^q(\mu,t)$ and space-dependent functions $\eta^q(x)$ for $q = 1,\dots,Q_\eta$ with low integer Q_η we have $\eta(\mu,x,t) = \sum_{q=1}^{Q_\eta} \Theta_\eta^q(\mu,t)\eta^q(x)$. The RBM is depending on a so called reduced basis. The simplest way of obtaining such a basis for time-dependent problems, can be described as follows: Initially, one chooses in a suitable way N parameters $\mu_i \in D$ and time instants $t_i \in [0,T]$ for $i = 1,\dots,N$ and computes the finite element (FE) approximations $u_H(\mu_i,t_i)$ on a fine mesh, the so called snapshots. The number N is typically small compared to the dimension $\mathcal{N}$ of the FE space. A new space X_N

is constructed as span of these N FE approximations. Then, in order to compute approximations for many arbitrary new parameters $\mu \in D$, possibly different from $\mu_1, ..., \mu_N$, an RB approximation $u_N(\mu)$ is computed in the lower dimensional space X_N. Because of the lower dimension, the computations of the RB approximation $u_N(\mu, t) \in X_N$ is faster than the computation of the FE approximation $u_H(\mu, t)$. The reduced simulation scheme is based on certain matrices, vectors, and in our case tensors. These quantities can be decomposed in an offline/online procedure. Offline means that we precompute and store parameter- and time-independent quantities. In the online phase, these offline quantities are combined with few parameter-dependent coefficients, which yield the final parameter-dependent system-matrices and vectors. This online phase is therefore very fast. An analytical quantification of the error between the RB approximation $u_N(\mu)$ and the FE approximation $u_H(\mu)$ can be specified by a posteriori error estimators.

The RBM has been applied to various kinds of PDEs, for a comprehensive overview see [12]. In [6] and [4] linear and nonlinear parabolic PDEs are discretized with the RBM. In the latter reference the nonlinearity is assumed to be monotone and approximated by the so called empirical interpolation. In [14, 15] the focus is geometric parameterization of the domain. The RBM has been extended from FEM to Finite Volume discretizations for linear schemes and affine parameter dependence [7] and more general parameter dependence [8]. In [16] the RBM is applied to the stationary, viscous Burgers equation. It contains an a posteriori error estimator between $u_N(\mu)$ and $u_H(\mu)$ as well as an existence proof. The current presentation condenses the results of the study [9], which extends this method and the results of [16]. We focus on the time-dependent case, and allow a wider class of quadratically nonlinear transport equations. The results presented in [4] are not applicable in our case, as the nonlinearity of equation (1.1) can not be assumed to be monotone. Independently of our study, [11] also treats the viscous time-dependent Burgers equation. Our study, however, is not limited to one space dimension and we present an energy-norm error estimator, analogous to [6] and [11].

The paper is organized as follows. In the next section, § 2, we first give the discretization of the problem and based on this we introduce the RB algorithm. We describe details on the offline/online decomposition in § 3, which results in a fast online simulation scheme. To justify our approach analytically, we present two a posteriori error estimators in § 4 that allow an efficient computation. Additionaly, in § 5 we present some numerical experiments. In particular, we investigate the computational gain and the error convergence. We conclude our study in § 6.

2. RB Approximation for Quadratically Nonlinear Transport Equations. We will first give the detailed discretization, which includes Dirichlet value treatment, backward Euler for time discretization and the Newton method for the nonlinearity. Based on this we introduce the RB approximation.

2.1. Boundary Value Treatment. In order to use the solution spaces with zero boundary values, we transform the problem (1.1)–(1.3) to a problem with homogeneous Dirichlet boundary values by substituting $u(\mu) := \bar{u}(\mu) - u_d(\mu)$. This yields

$$\begin{aligned} \partial_t u(\mu) - \operatorname{div}[\eta(\mu)\nabla u(\mu) - 2\vec{c}(\mu)u_d(\mu)u(\mu) - \vec{c}(\mu)(u(\mu))^2] &= g(\mu) && \text{in } \Omega \times [0,T] \\ u(\mu, 0) &= u_0(\mu) && \text{in } \Omega \\ u(\mu) &= 0 && \text{on } \partial\Omega \times [0,T] \end{aligned} \tag{2.1}$$

with $g(\mu) := -\partial_t u_d(\mu) + \mathrm{div}[\eta(\mu)\nabla u_d(\mu) - \vec{c}(\mu)(u_d(\mu))^2] + p(\mu)$.

2.2. Time Discretization. The time is discretized by backward Euler. Therefore, we get a quadratically nonlinear elliptic PDE in each time instant. The index k refers to the time instant $t^k = k\Delta t$ for given time step size $\Delta t > 0$. The end-time T refers to t^K, i.e. $k \in \{0, ..., K\}$. The solutions for fixed time reside in the Sobolev space $X := H_0^1(\Omega)$. For the detailed discretization we introduce the FE-basis space $X_H \subset X$. The FE space X_H contains space-dependent basis functions $\phi_i,\ 1 \le i \le \mathcal{N}$. The dimension $\mathcal{N}$ depends on the fineness of the mesh and the order of the polynoms. The time-discrete finite element approximation for the time instant t^k is $u_H^k(\mu) := u_H(\mu, t^k)$. The weak exact solution of the given problem is denoted by $u_e^k(\mu)$. For simplicity we omit the parameter dependence of u if confusion is unlikely.

DEFINITION 2.1. *(Functions f, $\tilde{f}$ and the Fréchet-Derivative $d_u f$)*
The parameter domain is D. Let

$$f^k : X \times X \times D \to \mathbb{R}, \quad \tilde{f}^k : X \times X \times D \to \mathbb{R} \text{ and } d_u f^k : X \times X \times X \times D \to \mathbb{R}$$

be mappings with the explicit definitions for all $v, u \in X,\ \mu \in D$ for a given $u_e^{k-1}(\mu)$ and $u_H^{k-1}(\mu) \in X$

$$\begin{aligned}
f^k(u, v; \mu) &:= \int_\Omega (u - u_e^{k-1}(\mu))v - \Delta t \int_\Omega \vec{c}^k(\mu)\nabla(u^2)v + \Delta t \int_\Omega \eta^k(\mu)\nabla u \nabla v \\
&- \Delta t \int_\Omega 2\vec{c}^k(\mu)\nabla(u_d^k(\mu)u)v - \Delta t \int_\Omega \left(g^k(\mu)\right) v; \qquad (2.2)\\
\tilde{f}^k(u, v; \mu) &:= \int_\Omega (u - u_H^{k-1}(\mu))v - \Delta t \int_\Omega \vec{c}^k(\mu)\nabla(u^2)v + \Delta t \int_\Omega \eta^k(\mu)\nabla u \nabla v \\
&- \Delta t \int_\Omega 2\vec{c}^k(\mu)\nabla(u_d^k(\mu)u)v - \Delta t \int_\Omega \left(g^k(\mu)\right) v \text{ and} \qquad (2.3)\\
d_u f^k(u, v; w; \mu) &:= \int_\Omega uv + \Delta t \int_\Omega \eta^k(\mu)\nabla u \nabla v - 2\Delta t \int_\Omega \vec{c}^k(\mu)\nabla(uw)v \\
&- \Delta t \int_\Omega 2\vec{c}^k(\mu)\nabla(u_d^k(\mu)u)v,
\end{aligned}$$

where $u_d^k(\mu) := u_d(\mu, t^k)$, $\vec{c}^k(\mu) := \vec{c}(\mu, t^k)$, $\eta^k(\mu) := \eta(\mu, t^k)$ and $g^k(\mu) := g(\mu, t^k)$ are time discrete values of the parameter functions. The mapping $d_u f^k(u, v; w; \mu)$ is the partial derivative in the Fréchet sense with respect to u of $f^k(u, v; \mu)$ at w. The weak exact solution in the time instant k is defined as solution to the equation

$$f^k(u_e^k(\mu), v; \mu) = 0 \quad \text{for all } v \in X,\ \mu \in D \qquad (2.4)$$

in the infinite dimensional space $H_0^1(\Omega)$.

REMARK 2.2. *The results of this paper are applicable to nonlinear parameter-dependent semidiscrete PDEs in weak form on a normed space $(X, \|.\|_X)$ which have the form*

$$\begin{aligned}
m(u, v) - m(u^{k-1}, v) + \Delta t b^k(u, v; \mu) + \Delta t a^k(u, u, v; \mu) &= \Delta t q^k(v; \mu) \\
m(u^0, v) &= h(v; \mu)
\end{aligned}$$

for all $k \in \{0, ..., K\}$ and $v \in X,\ \mu \in D$ where $a^k(., ., .; \mu)$ is a trilinear form and $m(., .)$ as well as $b^k(., .; \mu)$ are bilinear forms. The linear forms are $h(.; \mu)$ and $q^k(.; \mu)$. The

last argument in a^k, b^k, q^k *and* h *is the parameter vector* μ. *The forms are supposed to fulfill the conditions*

$$a^k(u,v,w;\mu) = a^k(v,u,w;\mu) \quad \textit{for all } u,v,w \in X \tag{2.5}$$

$$\textit{and } m(v,u) = m(u,v) \quad \textit{for all } u,v \in X. \tag{2.6}$$

The bilinear form $d(.,.;\mu) = m(.,.) + b(.,.;\mu) + 2\inf_{w\in X} a(.,.;w;\mu)$ *needs to have a positive coercivity constant for all* $k \in \{0,...,K\}$ *and all* $\mu \in D$. *One easily shows that the Burgers equation fulfills these conditions.*

2.3. Newton Method. The nonlinearity is linearized with the Newton Method.

The initial value of the Newton method is the approximation of the previous time instant, $u_e^{k,0}(\mu) := u_e^{k-1}(\mu)$, if L_k is the index of the last Newton iteration. Then, the starting vector for the next time instant is $u_e^{k+1,0}(\mu) = u_e^{k,L_k}(\mu)$. In the l-th iteration step the linearized equation for determining $u_e^{k,l+1}(\mu)$ is

$$d_u f^k(u_e^{k,l+1}(\mu) - u_e^{k,l}(\mu), v; u_e^{k,l}(\mu);\mu) = -f^k(u_e^{k,l}(\mu), v;\mu) \quad \text{for all } v \in X. \tag{2.7}$$

The stopping criterion for the Newton iteration is $\left\|u_e^{k,l+1}(\mu) - u_e^{k,l}(\mu)\right\|_{L^2} < \epsilon_{tol}$.

For the analytical result of Propositon 2.4 and Lemma 2.5 we require Definition 2.3. The results are generally derived and therefore valid for the reduced basis treatment as well as the FEM. We consider the Sobolev space $H_0^1(\Omega)$ with the well-defined norm

$\|u\|_{\Delta t} := \left(\|u\|^2_{L^2(\Omega)} + \Delta t I_\eta \|\nabla u\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}$ for the timestep Δt and the given lower bound I_η for the viscosity.

Definition 2.3. *(Coercivity Constant, Lipschitz Constant and Radius)*
Let $\mu \in D$ *be given and* X *a suitable function space. The positive coercivity constant is defined as* $0 < \alpha^k_{\Delta t}(\mu) := \inf_{w\in X} \inf_{v\in X, v\neq 0} \frac{d_u f^k(v,v;w;\mu)}{\|v\|^2_{\Delta t}}$ *and the Lipschitz constant is* $L(\Delta t;\mu) := \sup_{u,v,w^1,w^2\in X} \frac{|d_u f^k(u,v;w^1;\mu) - d_u f^k(u,v;w^2;\mu)|}{\|v\|_{\Delta t}\|u\|_{\Delta t}\|w^1 - w^2\|_{\Delta t}}$.

The radius is $r^k(\Delta t;\mu) := \frac{\alpha^k_{\Delta t}(\mu)}{L(\Delta t;\mu)}\left(1 - \sqrt{1 - \tau^k(\Delta t;\mu)}\right)$, *where* $\tau^k(\Delta t;\mu)$ *is defined as* $\tau^k(\Delta t;\mu) := \frac{2L(\Delta t,\mu)\epsilon^{k,k-1}(\Delta t;\mu)}{\left(\alpha^k_{\Delta t}(\mu)\right)^2}$. *We define the residual as*
$\epsilon^{k,k'}(\Delta t;\mu) = \sup_{v\in X} \frac{-\tilde{f}^k(u_H^{k'}(\mu), v;\mu)}{\|v\|_{\Delta t}}$.

The convergence of the Newton Method with using the previous time step as starting point is demonstrated in the Proposition 2.4. .

Proposition 2.4. *(Convergence of the Newton Method)*
If the bilinear form $d_u f^k(.,.;w;\mu)$ *has a positive coercivity constant* $\alpha^k_{\Delta t}(\mu)$ *and a Lipschitz constant* $L(\Delta t;\mu)$ *with respect to* w *for all* $w \in X$, *and the time step size* Δt *is small enough, then the Newton Method (2.7) converges to the exact solution,* $u_e^k(\mu)$*of the equation*

$$f^k(.,v;\mu) = 0 \quad \textit{for all } v \in X$$

i.e. $u_e^{k,l}(\mu) \to u_e^k(\mu)$ *for* $l \to \infty$ *with*

$$\left\|u_e^{k,l+1}(\mu) - u_e^k(\mu)\right\|_{\Delta t} \leq C'(\Delta t, k;\mu)\left\|u_e^{k,l}(\mu) - u_e^k(\mu)\right\|_{\Delta t} \textit{ for all } \mu \in D$$

where $C'(\Delta t, k;\mu) := \frac{L_{\Delta t}(\Delta t;\mu)}{2\left(\alpha^k_{\Delta t}(\mu) - L_{\Delta t}(\Delta t;\mu) r^k(\Delta t;\mu)\right)}$. *The constant* $C'(\Delta t, k;\mu)$ *converges to zero for* $\Delta t \to 0$.

Proof. We apply the ideas of [13, Chapter 10.3.1, page 353] and [3, page 301, Lemma 3.3 and Theorem 3.1], for details see [9, Chapter 2]. □

Additionally, we want to show that the error which appears by solving $\tilde{f}^k(.,v;\mu) = 0$ instead of $f^k(.,v;\mu) = 0$, i.e. putting the approximate previous time instant $u_H^{k-1}(\mu)$ instead of the exact previous time instant $u_e^{k-1}(\mu)$, is small.

LEMMA 2.5. *(Convergence in Time)*
If the bilinear form $d_u f^k(.,.;u_H^{k-1}(\mu);\mu)$ is for a fixed $\mu \in D$ coercive and the time step size Δt is sufficiently small, then the approximation $u_H^{k,l}(\mu)$ converges to the exact weak solution $u_e^k(\mu)$ in each time instant with

$$\left\| u_e^k(\mu) - u_H^{k,L_k}(\mu) \right\|_{\Delta t} \leq \Delta_e^k(\Delta t;\mu) \quad \text{for all } k \in \{0,...,K\}. \tag{2.8}$$

$$\text{with } \Delta_e^k(\Delta t;\mu) \quad := \epsilon_{tol} \sum_{k'=1}^{k} C_1(\Delta t,k';\mu) \frac{1}{\Pi_{k''=k'}^{k} \alpha_{\Delta t}^{k''}(\mu)},$$

$C'(\Delta t,k;\mu) := \frac{L(\Delta t;\mu)}{\alpha_{\Delta t}^k(\mu)} r^k(\Delta t;\mu)$, $C_1(\Delta t,k;\mu) := \frac{C'(\Delta t,k;\mu)}{1-C'(\Delta t,k;\mu)}$ and ϵ_{tol} is the stopping criterion for the Newton iteration. The constant $C_1(\Delta t,k;\mu)$ converges to zero for $\Delta t \to 0$.

Proof. For details see [9, Chapter 2]. □

2.4. RB Projection. The RB algorithm consists of a projection to a lower dimensional subspace and an effective decomposition. In this section we concentrate on the RB projection that is given explicitly below. Given a parameter-dependent PDE, for example Eqn. (2.1), where the parameter vector μ resides in D, we construct the parameter sample set S_N by arbitrarily choosing N tuples $(\mu_i, t^{k_i}) \in D \times [0,T]$.

DEFINITION 2.6. *(Parameter Sample Set S_N, Reduced Basis Space X_N, Reduced Basis Approximation $u_N(\mu)$)*
In general, the parameter domain $D \subset \mathbb{R}^d$ is a cartesian product of intervals and the parameter $\mu \in D$. The parameter sample set is given by
$S_N := \{(\mu_n, t^{k_n}) = (\mu_n^1,...,\mu_n^p, t^{k_n}) : \ t^{k_n} := k_n \Delta t, \ k_n \in \{1,...,K\}, \ 1 \leq n \leq N\}$.
The RB space is defined as $X_N := \text{span}\{u_H^{k_1,l_1}(\mu_1),...,u_H^{k_N,l_N}(\mu_N)\}$ with $(k_i,l_i) \in \{1,...,K\} \times \{1,...,L_k\}$. For simplicity, we use the abbrevation $\xi_n(x) := u_H^{k_n}(\mu_n,x)$. After Galerkin projection of the underlying PDE, in our case (2.4), to the smaller dimensional space X_N, the RB approximation can be decomposed as $u_N^k(\mu) = \sum_{n=1}^N \underline{u}_{N_n}^k(\mu)\xi_n(x)$. Here the coefficient vector is $\underline{u}_N^k(\mu) = (\underline{u}_{N_1}^k(\mu),...,\underline{u}_{N_N}^k(\mu))$.

We solve the underlying PDE N-times with the FEM for all (μ_n, t^{k_n}) of S_N and get N FE-approximations $u_H^{k_n}(\mu_n)$, $1 \leq n \leq N$. The RB space X_N is constructed with the above computed FE approximations. It holds that $X_N \subset X_H \subset X$.

If we now look for several approximations of many different tuples, possibly $(\mu, t^k) \notin S_N$, but $(\mu, t^k) \in D \times [0,T]$, we compute the coefficient vector $\underline{u}_N^k(\mu)$ for $\mu \in D$.

3. Offline/Online Decomposition. A fundamental ingredient in reduced basis approximation of P^{2}DEs is the effective decomposition of the computations in an offline and online phase. The offline phase prepares parameter-independent quantities, the computation of those quanties is (typically heavily) depending on H. The offline computed quantities are stored. The online phase assembles the final parameter-dependent matrices and vectors for the RB algorithm, which is ideally independent of the complexity H, the dimension of the FE space X_H. In each Newton step we solve a linearized equation (2.7) in X_N.

The construction of X_N is the first step in the offline stage, see Definition 2.6. Secondly, we computed the offline matrices, vectors and a tensor. In order to discriminate the offline quantities are denoted with a 'hat' as a superscript.

DEFINITION 3.1. *(Offline Quantities)*
The mass matrix is defined as $\hat{M} := (\int_\Omega \xi_i \xi_j)_{i,j=1}^N$ *and the offline part of the stiffness matrix is given by* $\hat{B}^q := (\int_\Omega \hat{\eta}^q(x)\nabla\xi_i\nabla\xi_j - \int_\Omega \bar{c}^q(x)\nabla(u_d(x)\xi_i)\xi_j)_{i,j=1}^N,\ 1 \le q \le Q_b$. $\hat{H}^q := (\int_\Omega u_0^q(x)\xi_i)_{i=1}^N,\ 1 \le q \le Q_h$ *and* $\hat{G}^q := (\int_\Omega \hat{g}^q(x)\xi_i)_{i=1}^N,\ 1 \le q \le Q_g$ *are parameter-independent vectors. The tensor is* $\hat{A}^q := (\hat{A}^q_{imj})_{i,m,j=1}^N$ *with* $\hat{A}^q_{imj} := -\int_\Omega \bar{c}^q(x)\nabla(\xi_i\xi_m)\xi_j,\ 1 \le q \le Q_a = Q_c + Q_cQ_d, 1 \le i,m,j \le N$.

DEFINITION 3.2. *(Online Quantities - Matrices and Vectors) The online quantities are the mass matrix* $\underline{M} = \hat{M}$*, the stiffness matrix* $\underline{B}^k(\mu) = \sum_{q=1}^{Q_b} \Theta_b^q(\mu, t^k)\hat{B}^q$ *and the right hand side* $\underline{G}^k(\mu) = \sum_{q=1}^{Q_g} \Theta_g^q(\mu, t^k)\hat{G}^q$*. The vector* $\underline{H}(\mu)$ *is defined as* $\underline{H}(\mu) = \sum_{q=1}^{Q_h} \Theta_h^q(\mu)\hat{H}^q$.

The matrix $\underline{B}^k(\mu)$ is a sum of products that consists of the parameter- and time-dependent factor $\Theta_b^q(\mu, t^k)$ and the offline matrix $\hat{B}^q$. The number Q_b indicates the number of summands of the affine decomposition of $\underline{B}^k(\mu)$. Here, the parameter- and time-dependent factor $\Theta_b^q(\mu, t^k)$ and the number Q_b depends on the assumed decomposition of the parameter function $\eta(\mu, t, x) = \sum_{q=1}^{Q_\eta} \Theta_\eta^q(\mu, t)\hat{\eta}^q(x)$. It holds that $Q_b = Q_\eta + Q_cQ_d$ where Q_c is the number of the summands of $\bar{c}(\mu)$ and Q_d is the number of the summands of $u_d(\mu)$. Analogously, the decomposition of $\underline{G}^k(\mu)$ and $\underline{H}(\mu)$ consists of the parameter-independent vectors $\hat{H}^q$ and $\hat{G}^q$ as well as the coefficients.

The nonlinearity appears in a trilinear form in the weak formulation, see the second integral in equation (2.2), explicitly

$$a^k(u, v; w; \mu) = -\int_\Omega \sum_{q=1}^{Q_c} \Theta_c^q(\mu, t^k)\bar{c}^q(x)\nabla(uw)v.$$

The space-independent factors could be written outside the integral. In the offline stage we compute a tensor that results from the space-dependent integral and store it. Consequently, we use the matrix $\underline{A}^k(\underline{u}_N^k(\mu))(\mu) := \sum_{q=1}^{Q_A} \Theta_A^q(\mu, t^k)\hat{A}^q\underline{u}_N^k(\mu)$, where $\hat{A}^q\underline{u}_N^k(\mu) \in \mathbb{R}^{N\times N}$ denotes the matrix obtained from the tensor-vector product between the tensor $\hat{A}^q$ and the vector $\underline{u}_N^k(\mu)$ along the first dimension, i.e. $(\hat{A}^q\underline{u}_N^k(\mu))_{m,j} = \sum_{i=1}^N \hat{A}^q_{imj}\underline{u}^k_{N_i}(\mu)$.

The above defined quantities are contained in the matrix
$\underline{DF}^k(w)(\mu) := [\underline{M}(\mu) + 2\Delta t\underline{A}^k(w)(\mu) + \Delta t\underline{B}^k(\mu)]$ and in the vector
$\underline{F}^k(w)(\mu)\underline{u}^k(\mu) := [\underline{M}(\mu) + \Delta t\underline{A}^k(w)(\mu) + \Delta t\underline{B}^k(\mu)]\underline{u}^k(\mu) - \Delta t\underline{G}^k(\mu) - \underline{M}(\mu)\underline{u}^{k-1}(\mu)$.
Consequently, we get the following system of equations

$$\underline{DF}^k(\underline{u}_N^{k,l+1}(\mu) - \underline{u}_N^{k,l})(\mu)\cdot\underline{u}_N^{k,l}(\mu) = -\underline{F}^k(\underline{u}_N^{k,l}(\mu))(\mu)\cdot\underline{u}_N^{k,l}(\mu) \tag{3.1}$$

for each Newton step $1 \le l \le L_k$ and time instant $1 \le k \le K$. In the online stage we compute the RB approximation for a new parameter vector, $\mu_{new} \in D$.

There are two loops. One loop over the discrete time instants, i.e. $k = 0, ..., K$, and in every time instant we perform a loop over all Newton steps, i.e. $l = 1, ..., L_k$. The computational complexity depends cubicly on N.

The matrices derived from the tensor are computed in every Newton step. Therefore this matrix enlarges the complexity of the algorithm used to solve quadratically

nonlinear PDEs compared to one that approximates linear PDEs. In every Newton step we multiply the stored tensor with the current coefficient vector $\underline{u}_N^{k,l}(\mu)$, which is the tensor vector product. Higher nonlinearities demand another method, *empiricial interpolation* see [1] and [4] for details. In the following we summarize the algorithm on which the RBM implementations are based. The overall online algorithm is as follows:

Algorithm (On-line stage)
Let $\epsilon_{tol} > 0$ and the offline computed quantities $\hat{H}^q$, $\hat{G}^q$, $\hat{B}^q$, $\hat{M}$ as well as $\hat{A}^q$ be given.

1. Choose a new parameter vector μ_{new} in D.
2. Compute the time- and parameter-dependent factors $\Theta_a^q(\mu_{new}, t^k)$ for all $1 \leq q \leq Q_a$ and all $1 \leq k \leq K$, analogously we compute $\Theta_b^q(\mu_{new}, t^k)$, $\Theta_g^q(\mu_{new}, t^k)$ and $\Theta_h^q(\mu_{new})$.
3. Multiply the parameter- and time-dependent factors $\Theta_b^q(\mu, t^k)$ with the offline computed space-dependent matrix $\hat{B}^q$ as well as the space-dependent vectors $\hat{G}^q$ and $\hat{H}^q$. Then we sum up over all q's, i.e.

$$\underline{B}^k(\mu_{new}) = \sum_{q=1}^{Q_b} \Theta_b^q(\mu_{new}, t^k)\hat{B}^q, \ \underline{M} = \hat{M}$$

$$\underline{G}^k(\mu_{new}) = \sum_{q=1}^{Q_g} \Theta_g^q(\mu_{new}, t^k)\hat{G}^q \text{ and}$$

$$\underline{H}^k(\mu_{new}) = \sum_{q=1}^{Q_h} \Theta_h^q(\mu_{new}, t^k)\hat{H}^q.$$

4. In the first time instant $k = 0$ we compute the starting coefficient vector $\underline{u}_N^0(\mu_{new})$ by solving $\underline{H}(\mu_{new}) = \underline{M}\ \underline{u}_N^0(\mu_{new})$.
5. In the next time instant $k := k + 1$ we set the previous approximation as starting vector for the Newton iteration, i.e. $\underline{u}_N^{k,0}(\mu_{new}) := \underline{u}_N^{k-1}(\mu_{new})$ and $l = 0$.
6. In every Newton step we assemble the matrix $\underline{A}^k(\underline{u}_N^{k,l}(\mu_{new}))(\mu_{new})$ by

$$\underline{A}^k(u_N^{k,l}(\mu_{new}))(\mu_{new}) = \sum_{q=1}^{Q_q} \Theta_A^q(\mu_{new}, t^k)\hat{A}^q\underline{u}_N^{k,l}(\mu_{new}).$$

7. We compute

$$\underline{F}^k(\underline{u}_N^{k,l}(\mu_{new}))(\mu_{new})u_N^{k,l}(\mu_{new}) = [\underline{M} + \Delta t\underline{A}^k(\underline{u}_N^{k,l}(\mu_{new}))(\mu_{new}) + \Delta t\underline{B}^k(\mu_{new})]\underline{u}_N^{k,l}(\mu_{new}) - \Delta t\underline{G}^k(\mu_{new}) - \underline{M}u_N^{k-1}(\mu_{new})$$

and the Newton increment

$$\underline{\delta u}_N^{k,l}(\mu_{new}) = -[\underline{M} + 2\Delta t\underline{A}^k(\underline{u}_N^{k,l}(\mu_{new}))(\mu_{new}) + \Delta t\underline{B}^k(\mu_{new})]^{-1}\underline{F}^k(\mu_{new}) \quad .$$

8. The next Newton iterate is $\underline{u}_N^{k,l+1}(\mu_{new}) := \underline{u}_N^{k,l}(\mu_{new}) + \underline{\delta u}_N^{k,l}(\mu_{new})$.
9. We compute $\epsilon := \left\| u_N^{k,l+1}(\mu_{new}) - u_N^{k,l}(\mu_{new}) \right\|_{L^2(\Omega)}$
   if $\epsilon \geq \epsilon_{tol} \ \rightarrow \ l := l + 1$, go to 5.
   if $\epsilon \leq \epsilon_{tol} \ \rightarrow \ \underline{u}_N^k(\mu_{new}) := \underline{u}_N^{k,l+1}(\mu_{new})$, go to 6.

4. A Posteriori Error Estimators. We justify the results of the RB algorithm by two a posteriori error estimators that quantify the error between the RB and the FE approximation. Futhermore, an efficient computation of the a posteriori error estimators is demonstrated, i.e. the online computation is independent of $\mathcal{N}$.

In this section we present two different a posteriori error estimators. The first one assumes that the dual norm of the residual is smaller than a problem specific constant. The second one does not require a similar restriction.

DEFINITION 4.1. *(Residuals)*
The dual norm of the residual $\epsilon_N^k(\Delta t;\mu)$ *with the RB appoximation* $u_N^k(\mu)$ *is defined as* $\epsilon_N^k(\Delta t;\mu) := \sup_{v\in X_H, v\neq 0} \frac{f_N^k(u_N^k(\mu),v;\mu)}{\|v\|_{\Delta t}} \in \mathbb{R}$ *with* $f_N(u,v;\mu) : X_N \times X_N \times D \to \mathbb{R}$. *The explicit definition of* f_N *is identical to Definition 2.1, equation (2.2), but using the previous RB approximation* $u_N^{k-1}(\mu)$ *instead of the exact previous time iteration* $u_e^{k-1}(\mu)$. *The mapping* $f_H(.,.;.) : X_H \times X_H \times D \to \mathbb{R}$ *is identical to Definition 2.1, equation (2.2), but using the previous approximation* $u_H^{k-1}(\mu)$ *instead of the exact previous time iteration* $u_e^{k-1}(\mu)$. *The radius is* $r_N^k(\Delta t;\mu) := \frac{\alpha_{\Delta t}^k(\mu)}{L(\Delta t;\mu)}\left(1-\sqrt{1-\tau_N^k(\Delta t;\mu)}\right)$, *where* $\tau_N^k(\Delta t;\mu)$ *is defined as* $\tau_N^k(\Delta t;\mu) := \frac{2L(\Delta t,\mu)\epsilon_N^{k,k-1}(\Delta t;\mu)}{\left(\alpha_{\Delta t}^k(\mu)\right)^2}$. If Δt is suitably small, $0 < \Delta t < 1$ according to the viscosity $\eta(\mu)$ and the scaling vector $\vec{c}(\mu)$, the bilinear form $d_u f^k(.,.;u_N^{k,L_k}(\mu);\mu)$ is coercive. With this we can derive the following error estimators.

PROPOSITION 4.2. *(1. Error Estimator)*
If the bilinear form $d_u f^k(.,.;u_N^{k,L_k}(\mu);\mu)$ *is coercive for all* $\mu \in D$ *where* $u_N^{k,L_k}(\mu) \in X_N$ *is the approximation to* $f_N^k(.,v;\mu) = 0 \quad$ *for all* $v \in X_N$ *and* $u_H^{k,L_k}(\mu) \in X_H$ *is the approximation to* $f_H^k(.,v;\mu) = 0 \quad$ *for all* $v \in X_H$ *and the dual norm of the residual is bounded,*

$$\epsilon_N^k(\Delta t;\mu) := \sup_{v\in X_H, v\neq 0} \frac{f_N^k(u_N^k(\mu),v;\mu)}{\|v\|_{\Delta t}} < \frac{\left(\alpha_{\Delta t}^k(\mu)\right)^2}{2L(\Delta t;\mu)},$$

then the error between the RB solution $u_N^k(\mu)$ *and the FE approximation* $u_H^k(\mu)$ *is bounded as follows,*

$$\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{\Delta t} \leq D_N^k(\Delta t;\mu)$$

with $D_N^k(\Delta t;\mu) := \sum_{k'=1}^k \left(\frac{r_N^{k'}(\Delta t;\mu)}{\Pi_{k''=k'+1}^k \alpha_{\Delta t}^{k''}(\mu)}\right)$. *By* $\Pi_{k=1}^l a(k)$ *we denote the product of* $a(k)\cdot ... \cdot a(l)$.

Proof. We apply the idea of [16, Proposition 2.1], for details see [9, Chapter 4.1]. □

PROPOSITION 4.3. *(2. Error Estimator)*
If the bilinear form $d_u f^k(.,.;w;\mu)$ *is coercive for all* $\mu \in D, k \in \{0,...,K\}$ *and* $w \in X$, *then we have* $\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{\Delta t} \leq \Delta_N^k(\Delta t;\mu) \quad$ *for all* $\mu \in D,\ k \in \{0,...,K\}$ *with* $\Delta_N^k(\Delta t;\mu) := \sum_{k'=1}^k \frac{\epsilon_N^{k'}(\Delta t;\mu)}{\Pi_{k''=k'}^k \alpha_{\Delta t}^{k''}(\mu)}$.

Proof. For details see Appendix A or [9, Chapter 4.1]. □

The error estimators should be computable by an offline/online decomposition, in order to guarantee the efficiency. The factors contained in the error estimators $\Delta_N^k(\Delta t,\mu)$ and $D_N^k(\Delta t,\mu)$ are $\epsilon_N^k(\Delta t;\mu)$, $\alpha_{\Delta t}^k(\mu)$, $L(\Delta t;\mu)$ and constants. We assume the coercivity constant, the Lipschitz constants and all the other constants to be

given. The residuals $\epsilon_N^k(\Delta t, \mu)$ and $\epsilon_N^{k,k'}(\Delta t, \mu)$ can be computed via an online/offline decomposition similar to [16], [10], [5] and [6], see [9]. The online computations are independent of the dimension $\mathcal{N}$.

5. Numerical Experiments. In this section we apply the above described algorithm. We consider the unit square $\Omega = [0,1]^2$. The finite element space X_H has the dimension $\mathcal{N} = 10201$. The detailed implementation as well as all offline computations are based on the C++ environment DUNE (Distributed and Unified Numerics Environment). For details see www.dune.org and [2]. Just the online algorithm is realized by Matlab routines. The implementations run on an x86 64 AMD Athlon(tm) 64 X2 Dual Core Processor 3800+ AuthenticAMD. We used a BICGSTAB scheme in the FE-algorithm.
The numerical solution obtained by this implementation approximate equation (2.1). Originally, we intended to solve a PDE with inhomogeneous Dirichlet boundary value $u_d(\mu) \neq 0$ and denote the solution as $\bar{u}(\mu)$. We transformed the solution as follows $u(\mu) := \bar{u}(\mu) - u_d(\mu)$ and get the third term in equation (1.1), homogeneous Dirichlet boundary values and the below defined right hand side. The right hand side is

$$g(\boldsymbol{\mu}) = -\int_\Omega (\partial_t u_d(\mu))v + \int_\Omega \vec{c}(\mu)(u_d(\mu))^2 \nabla v + \int_\Omega \nabla(\eta(\mu)\nabla u_d(\mu) + p(\mu))v.$$

5.1. First Example. In this example we demonstrate the nonlinear evolution. The parameter functions of (2.1) are

$$\begin{aligned} &\eta(\mu, x, t) = \mu_1, \quad p(\mu) = 0, \\ &u_d(\mu, x, t) = (2-\mu_4)\mu_5 e^{-100((y-0.1)^2+x^2)} + (\mu_4 - 1)\mu_5 e^{-100((y-0.2)^2+x^2)} \text{ and} \\ &c(\mu, x, t) = \begin{pmatrix} \mu_2 y \\ \mu_3 x \end{pmatrix}. \end{aligned}$$

The RB space X_N contains the five first Newton iterations of the detailed simulation for the first time instant, $K = 1$ and $L_1 = 5$.

$$\begin{aligned} &S_N := \{(\mu, 0.5) : \mu = (0.1, 10, 10, 3, 1)\} \\ &X_N := \text{span}\left\{u_H^{1,1}(\mu), ..., u_H^{1,5}(\mu)\right\} \text{ with } \mu = (0.1, 10, 10, 3, 1),\ N = \dim(X_N) = 5. \end{aligned}$$

The RB and FE approximations, $u_N^{1,5}(\mu)$ and $u_H^{1,5}(\mu)$, are visualized with GRAPE in Figure 5.1, a) and b). Figure 5.1 c) shows the inhomogeneous approximation $\bar{u}(\mu) = u_H^{1,5}(\mu) + u_d^1(\mu)$. The viscosity is small compared to the scaling function $\vec{c}(\mu, x, t) = \vec{c}(\mu_2, \mu_3, x, t)$ for the nonlinear convective term. The L^2-error between the fifth Newton iteration is $||u_H^{1,5}(\mu) - u_N^{1,5}(\mu)||_{L^2} = 5.57494e-05$ and the corresponding relative L^2-error $\frac{||u_H^{1,5}(\mu) - u_N^{1,5}(\mu)||_{L^2}}{||u_H^{1,5}(\mu)||_{L^2}} = 0.000372078$.
The detailed approximation lasts 24.6255 sec for five Newton steps, while the RB simulation for five Newton steps takes 0.180968 sec. The computations of the offline quantities is done in 1722.48 sec. This example demonstrates the accuracy of the RBM concerning the nonlinearity.

5.2. Second Example. In this example we construct a RB space that allows a wide choice of parameter variation. We show the results of three different parameter configurations, $\mu = (0.1, 0, 5, 4, 1)$, $\mu = (0.1, 2, 3, 2.46, 1)$ and $\mu = (0.1, 4.2, 1.3, 5.3, 1)$. The parameter space is $D := \{0.1\} \times [0, 5] \times [0, 5] \times [2, 6]$. The endtime is $T = 0.5$ and

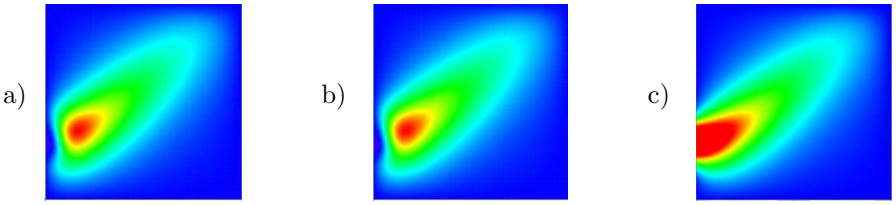

FIGURE 5.1. *a) RB approximation* $u_N^{1,5}(\mu)$ *for* $\mu = (0.1, 10, 10, 3, 1)$ *and the first time instant* $t^1 = 0.5$ *and for the fifth Newton step* $l = 5$ *b) FE approximation* $u_H^{1,5}(\mu)$ *c) inhomogeneous FE approximation* $u_H^{1,5}(\mu) + u_d^1(\mu)$

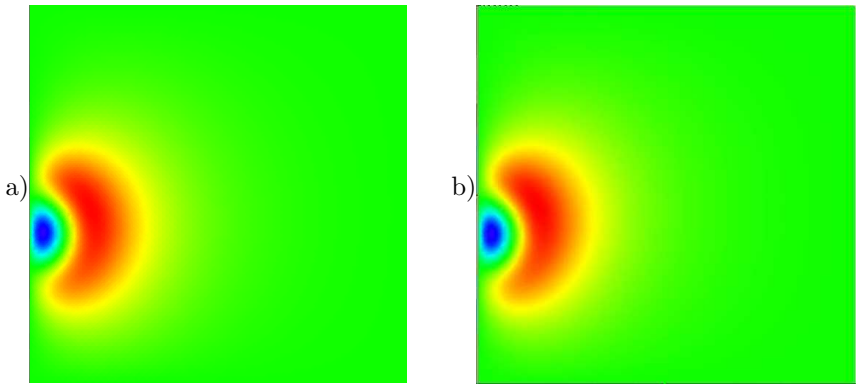

FIGURE 5.2. *a) RB approximation* $u_N^k(\mu)$ *for* $\mu = (0.1, 0, 5, 4, 1)$ *and* $t^k = 0.5$ *and b) FE approximation* $u_H^k(\mu)$ *for* $\mu = (0.1, 0, 5, 4, 1)$ *and* $t^k = 0.5$

the time step size $\Delta t = 0.5$. The RB space X_N consists of 20 basis functions, the FE approximations $u_H^k(\mu)$ for all $(\mu, t) \in S_N$ with $S_N := \{(0.1, \mu_2, \mu_3, \mu_4, 1, t^1) : \ \mu_2, \mu_3 \in \{0, 5\}, \ \mu_4 \in \{2, 3, 4, 5, 6\}\}$.
First we compute the FE and the RB approximation for $\mu = (0.1, 0, 5, 4, 1) \in D$. The L^2-error between the FE and RB approximation is $\left\|u_H^1(\mu) - u_N^1(\mu)\right\|_{L^2} = 0.00203646$ and the relative L^2 error is $\frac{\left\|u_H^1(\mu) - u_N^1(\mu)\right\|_{L^2}}{\left\|u_H^1(\mu)\right\|_{L^2}} = 0.0467045$. The visualization is nearly identical, see Figure 5.2. This demonstrates the accuracy of the RBM with respect to parameter variation.

The FE simulation is computed in 5.01631 sec, $K = 1$, $L_k = 1$, the RB Simulation needs just 0.118945 sec. In Figure 5.3 one can see that the FE and RB approximation for the parameter vector $\mu = (0.1, 2, 3, 2.46, 1)$ are almost equal, although three parameter components are varied compared to the training parameter. The relative L^2 error is $\frac{\left\|u_H^1(\mu) - u_N^1(\mu)\right\|_{L^2}}{\left\|u_H^1(\mu)\right\|_{L^2}} = 0.00051514$. In Figure 5.4 the FE and RB approximation for the parameter vector $\mu = (0.1, 4.2, 1.3, 5.3, 1)$ are presented. The time savings are obvious, because the RB algorithm needs just 0.118945 sec, while the FE algorithm needs 5.01631 sec.

5.3. Third Example. In this example we demonstrate the computational gain by using the RBM instead of the FEM. The main goal of RB-approaches is an accelerated online phase compared to the full simulation. The data functions are the same as in the first example. The time step size is $\Delta t = 0.05$ and the end time is $T = 0.5$. The

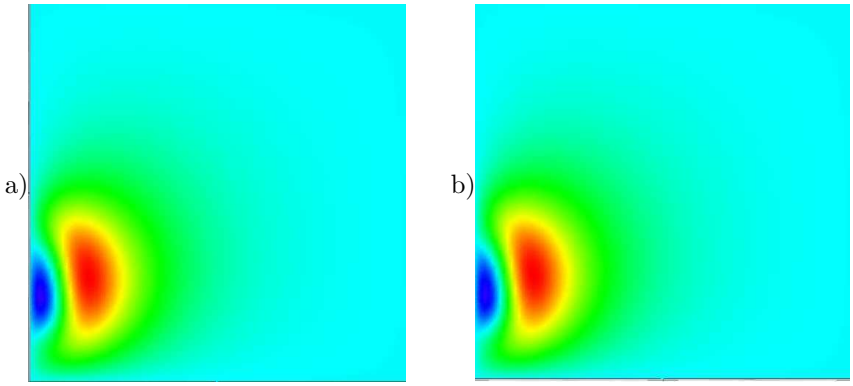

FIGURE 5.3. *a) RB approximation* $u_N^k(\mu)$ *for* $\mu = (0.1, 2, 3, 2.46, 1)$ *and* $t^k = 0.5$ *b) FE approximation* $u_H^k(\mu)$ *for* $\mu = (0.1, 2, 3, 2.46, 1)$ *and* $t^k = 0.5$

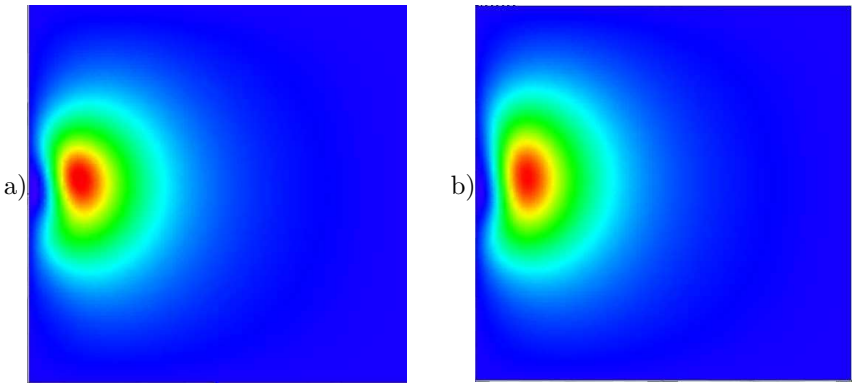

FIGURE 5.4. *Approximation for the parameter vector* $\mu = (0.1, 4.2, 1.3, 5.3, 1)$ *at the first time step* $t^1 = 0.5$ *and first Newton step* $l = 1$ *a) with the RBM b) with the FEM*

computations are done for one time instant and one Newton step $L_k = 1$. We consider the following parameter domain $D := \{0.01\} \times [4.5, 5.5] \times [4.5, 5.5] \times \{1\} \times \{1\} \subset \mathbb{R}^5$, the sample set $S_N := \{(\mu, t^k) |\ \mu = (0.01, 5, 5, 1, 1),\ t^k = k\Delta t,\ 1 \leq k \leq 10\}$ and the RB space X_N consists of 10 time iterations for parameter vector $\mu = (0.01, 5, 5, 1, 1) \in D$. Compared to the second example, we just vary two components μ_2 and μ_3 of the parameter vector, but consider a larger timespan. The other components of the parameter vector remain fixed. We compute the RB approximation for a parameter vector $\mu = (0.01, 5.2, 4.8, 1, 1)$. Table 5.1 demonstrates that the error and the relative error are still acceptable, although we computed the approximation corresponding to new parameter vector that is different from the reference parameters.

The CPU times of the FE simulation are 45.2068 sec, RB simulation 0.680140 sec and offline computations take 5556.04 sec. In Figure 5.6 we demonstrate the efficiency of the RBM compared to FEM. We see, that the RBM really is useful for many-query settings. For sufficiently many simulations, the runtime for offline phase plus the multiple online computations, is lower than the computational time for the multiple FEM simulations. The L^2-error and relative L^2-error is presented in Table 5.2.

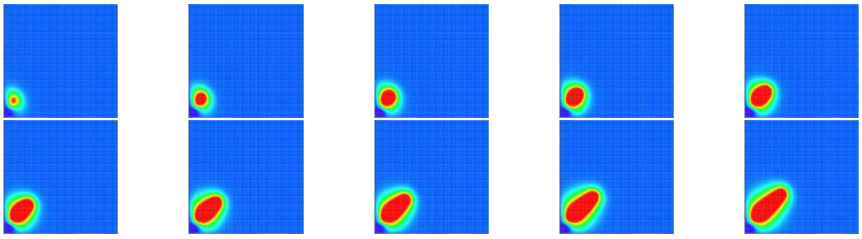

FIGURE 5.5. *RB trajectory for the parameter vector* $\mu = (0.01, 5.2, 4.8, 1, 1)$*, with final time* 0.5 *and with a time step size* $\Delta t = 0.05$ *and* $N = 10$ *for* $t^1 = 0.05$ *,* $t^2 = 0.1$*, ... ,* $t^{10} = 0.5$

TABLE 5.1
Absolute and relative L^2 *error between RB and FEM approximation, parameter vector* $\mu = (0.01, 5.2, 4.8, 1, 1), T = 0.5,\ \Delta t = 0.05$ *and* $N = 10$

time instant t^k	$\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{L^2(\Omega)}$	$\frac{\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{L^2(\Omega)}}{\left\|u_H^k(\mu)\right\|_{L^2(\Omega)}}$
$t^0 = 0$	0	0
$t^1 = 0.05$	0.000549648	0.0281908
$t^2 = 0.1$	0.00113435	0.031538
$t^3 = 0.15$	0.00173473	0.0349544
$t^4 = 0.2$	0.0023629	0.0388154
$t^5 = 0.25$	0.00298377	0.0425092
$t^6 = 0.3$	0.00361479	0.0463289
$t^7 = 0.35$	0.00422896	0.0499113
$t^8 = 0.4$	0.00484625	0.0535105
$t^9 = 0.45$	0.00544773	0.0569121
$t^{10} = 0.5$	0.00608852	0.0606843

TABLE 5.2
Absolute and relative L^2 *error between RB and FEM approximation, parameter vector* $\mu = (0.01, 5, 5, 1, 1), T = 0.5,\ \Delta t = 0.05$ *and* $N = 10$

time instant t^k	$\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{L^2(\Omega)}$	$\frac{\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{L^2(\Omega)}}{\left\|u_H^k(\mu)\right\|_{L^2(\Omega)}}$
$t^0 = 0$	0	0
$t^1 = 0.05$	$2.7705 \cdot 10^{-11}$	$1.46338 \cdot 10^{-9}$
$t^2 = 0.1$	$4.09397 \cdot 10^{-11}$	$1.17237 \cdot 10^{-9}$
$t^3 = 0.15$	$4.7922 \cdot 10^{-11}$	$9.94021 \cdot 10^{-10}$
$t^4 = 0.2$	$5.40204 \cdot 10^{-11}$	$9.12845 \cdot 10^{-10}$
$t^5 = 0.25$	$6.0319 \cdot 10^{-11}$	$8.83424 \cdot 10^{-10}$
$t^6 = 0.3$	$6.60415 \cdot 10^{-11}$	$8.69686 \cdot 10^{-10}$
$t^7 = 0.35$	$7.2702 \cdot 10^{-11}$	$8.81296 \cdot 10^{-10}$
$t^8 = 0.4$	$7.81986 \cdot 10^{-11}$	$8.86577 \cdot 10^{-10}$
$t^9 = 0.45$	$8.46411 \cdot 10^{-11}$	$9.0773 \cdot 10^{-10}$
$t^{10} = 0.5$	$8.72021 \cdot 10^{-11}$	$8.92068 \cdot 10^{-10}$

The experiments show that a specific, problem dependent construction of the space X_N is essential. The construction of the offline quantities is expensive. Therefore, the RBM is applicable if the user is interested in many computations. In gen-

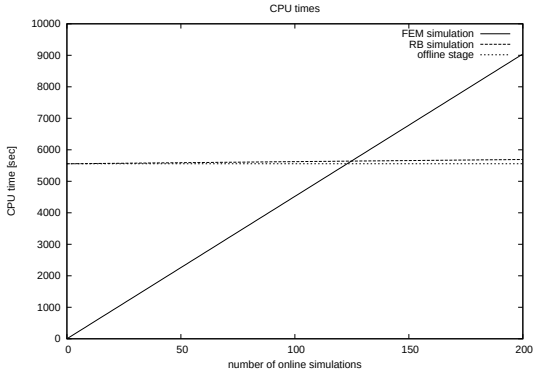

FIGURE 5.6. *CPU time [sec] depends on the number of computations,* $N = 10$

eral, the comparison between the FE and RB simulation time shows the time savings, achieved by applying the RBM.

6. Conclusion. We have presented a Reduced Basis Method for parametrized quadratically nonlinear transport equations with implicit time discretizations. The main ingredient is an offline/online decomposition of the trilinear form, which is required for solving the Newton loop. We have derived a posteriori error estimators, that can be used to give rigorous quantification of the reduced model's error in the online phase. The implementation, experimental comparison and validation of the estimators remains to be done. Experimentally, we have applied the method to a viscous Burgers Equation with 1st Order FEM discretization. The results indicate that the RB method gives good approximations compared to the FEM solutions. The RB method gives a considerable CPU gain for sufficiently many simulations.

Acknowledgement. The second author is funded by the Landesstiftung Baden-Württemberg gGmbH. We acknowledge further funding by a research scholarship of the International Office of the University of Freiburg and we thank Prof E. Süli at the University of Oxford for his hospitality.

References.

[1] 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. *Comptes Rendus Mathematique, Acad. Sci. Paris*, 339(9):667–672, 2004.

[2] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Kloefkorn, R. Kornhuber, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. II: Implementation and tests in DUNE. *Computing*, 82(2-3):121–138, 2008.

[3] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations. Theory and algorithms*. Springer-Verlag, 1986.

[4] M. A. Grepl. *Reduced-basis Approximations and a Posteriori Error Estimation for Parabolic Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, May 2005.

[5] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear differential equations. *Mathematical Modelling and Numerical Analysis (M2AN)*, 41(3):575–605, 2007.

[6] M. A. Grepl and A. T. Patera. Reduced-basis approximation for time-dependent parametrized partial differential equations. *Mathematical Modelling and Numerical Analysis (M2AN)*, 39(1):157–181, 2005.
[7] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *Mathematical Modelling and Numerical Analysis (M2AN)*, 42(2):277–302, 2008.
[8] B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *Electronic Transactions on Numerical Analysis (ETNA)*, 32:145–161, 2008.
[9] N. Jung. Anwendung der Reduzierten Basis Methode auf quadratisch nichtlineare Transportgleichungen. Diploma thesis, Albert-Ludwigs-Universität, Freiburg, 2008.
[10] C. N. Nguyen, K. Veroy, and A. T. Patera. Certified real-time solution of parametrized partial differential equations. In S. Yip, editor, *Handbook of Materials Modeling*, pages 1523–1558. Springer, 2005.
[11] C.N. Nguyen, G. Rozza, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers equation. *Calcolo*, 2008. Submitted.
[12] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parameterized Partial Differential Equations*. Copyright MIT 2006, to appear in MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2007.
[13] A. Quarterioni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 1994.
[14] G. Rozza. *Shape design by optimal flow control and reduced basis techniques: Applications to bypass configurations in haemodynamics*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2005.
[15] T. Tonn and K. Urban. A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies. In *Proc. of ECCOMAS CFD 2006*. Delft University of Technology, The Netherlands, 2006.
[16] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.

Appendix A. Error Estimator. We define general forms that we use in the subsequent proof of Proposition 4.3.

DEFINITION A.1. *(General Forms) The bilinear forms* $m^k : X \times X \times D \to \mathbb{R}$ *and* $b^k : X \times X \times D \to \mathbb{R}$ *are defined as*

$$m(u, v; \mu) := \int_\Omega uv \text{ and } b^k(u, v, \mu) := \int_\Omega \eta^k(\mu)\nabla u \nabla v - \int_\Omega \vec{c}^k(\mu)\nabla(u_d(\mu)u)v.$$

The linear form $q^k : X \times D \to \mathbb{R}$ *that represents the right hand side is*

$$q^k(v; \mu) := \int_\Omega \big(- \frac{u_d^k(\mu) - u_d^{k-1}(\mu)}{\Delta t} + \text{div}[\eta^k(\mu)\nabla u_d^k(\mu) - \vec{c}^k(\mu)(u_d^k(\mu))^2] + p^k(\mu)\big)v$$

PROPOSITION A.2. *(2. Error Estimator)*
If the bilinear form $d_u f^k(.,.; w; \mu)$ *is coercive for all* $\mu \in D, k \in \{0, ..., K\}$ *and* $w \in X$, *then we have* $\left\|u_N^k(\mu) - u_H^k(\mu)\right\|_{\Delta t} \leq \Delta_N^k(\Delta t; \mu)$ *for all* $\mu \in D,\ k \in \{0, ..., K\}$

with $\Delta_N^k(\Delta t;\mu) := \sum_{k'=1}^{k} \frac{\epsilon_N^{k'}(\Delta t;\mu)}{\Pi_{k''=k'}^{k} \alpha_{\Delta t}^{k''}(\mu)}$.

Proof. Let $v \in X_H$ and $\mu \in D$, then we have using Definition 2.1, 4.1 and A.1

$$
\begin{aligned}
0 &= f_H^k(u_H^k(\mu), v;\mu) - f_N^k(u_N^k(\mu), v;\mu) + f_N^k(u_N^k(\mu), v;\mu) \\
&= m(u_H^k(\mu), v) - m(u_H^{k-1}(\mu), v) \\
&\quad + \Delta t b^k(u_H^k(\mu), v;\mu) + \Delta t a^k(u_H^k(\mu), u_H^k(\mu), v;\mu) - \Delta t q^k(v;\mu) \\
&\quad - m(u_N^k(\mu), v) + m(u_N^{k-1}(\mu), v) \\
&\quad - \Delta t b^k(u_N^k(\mu), v;\mu) - \Delta t a^k(u_N^k(\mu), u_N^k(\mu), v;\mu) + \Delta t q^k(v;\mu) + f_N^k(u_N^k(\mu), v;\mu) \\
&= m(u_H^k(\mu) - u_N^k(\mu), v) - m(u_H^{k-1}(\mu) - u_N^{k-1}(\mu), v) + \Delta t b^k(u_H^k(\mu) - u_N^k(\mu), v;\mu) \\
&\quad + \Delta t a^k(u_H^k(\mu), u_H^k(\mu), v;\mu) - \Delta t a^k(u_N^k(\mu), u_N^k(\mu), v;\mu) + f_N^k(u_N^k(\mu), v;\mu).
\end{aligned}
$$

We choose the test function to be $v = u_H^k(\mu) - u_N^k(\mu) \in X_H$, and get

$$
\begin{aligned}
m(v,v) \quad &+ \Delta t b^k(v, v;\mu) + \Delta t a^k(u_H^k(\mu), u_H^k(\mu), v;\mu) - \Delta t a^k(u_N^k(\mu), u_N^k(\mu), v;\mu) \\
&= m(u_H^{k-1}(\mu) - u_N^{k-1}(\mu), v) - f_N^k(u_N^k(\mu), v;\mu).
\end{aligned}
$$

We assumed that the trilinear form $a^k(u, v, w;\mu)$ is symmetric with respect to the first two arguments. Therefore, it follows,

$$
\begin{aligned}
m(v,v) \quad &+ \Delta t b^k(v, v) + \Delta t a^k(v, u_H^k(\mu), v;\mu) + \Delta t a^k(v, u_N^k(\mu), v;\mu) \\
&= m(u_H^{k-1}(\mu) - u_N^{k-1}(\mu), v) - f_N^k(u_N^k(\mu), v;\mu).
\end{aligned}
$$

Based on Definition 2.1 the following relation between f^k and $d_u f^k$ holds

$$
\begin{aligned}
\alpha_{\Delta t}^k(\mu) \left\| v \right\|_{\Delta t}^2 &\leq \frac{1}{2}\Big[d_u f^k(v, v; u_H^k(\mu);\mu) + d_u f^k(v, v; u_N^k(\mu);\mu) \Big] \\
&= -f_N^k(u_N^k(\mu), v;\mu) + m(u_H^{k-1}(\mu) - u_N^{k-1}(\mu), v) \\
&\leq -f_N^k(u_N^k(\mu), v;\mu) + \left\| v \right\|_{\Delta t} \left\| u_H^{k-1}(\mu) - u_N^{k-1}(\mu) \right\|_{\Delta t}.
\end{aligned}
$$

Without loss of generalization we assume $v \neq 0$, otherwise this statement is clear. Dividing by $\left\| v \right\|_{\Delta t} \alpha_{\Delta t}^k(\mu)$ and inserting the definition of the dual norm of the residuum $\epsilon_N^k(\Delta t;\mu)$ results in

$$
\begin{aligned}
\left\| v \right\|_{\Delta t} &\leq \frac{\epsilon_N^k(\Delta t;\mu)}{\alpha_{\Delta t}^k(\mu)} + \frac{1}{\alpha_{\Delta t}^k(\mu)} \left\| u_H^{k-1}(\mu) - u_N^{k-1}(\mu) \right\|_{\Delta t} \\
&\leq \frac{\epsilon_N^k(\Delta t;\mu)}{\alpha_{\Delta t}^k(\mu)} + \frac{\Delta_N^{k-1}(\Delta t;\mu)}{\alpha_{\Delta t}^k(\mu)} = \Delta_N^k(\Delta t;\mu).
\end{aligned}
$$

□