Guaranteed a posteriori error bounds for low rank tensor approximate solutions

Sergey Dolgov*
University of Bath, Department of Mathematical Sciences
Claverton Down, BA2 7AY Bath, United Kingdom

Tomáš Vejchodský†
Institute of Mathematics, Czech Academy of Sciences,
Žitná 25, CZ-115 67 Prague 1, Czech Republic

May 21, 2019

Abstract

We propose guaranteed and fully computable upper bound on the energy norm of the error in low rank Tensor Train (TT) approximate solutions of (possibly) high dimensional reaction-diffusion problems. The error bound is obtained from Euler–Lagrange equations for a complementary flux reconstruction problem, which are solved in the low rank TT representation using the block Alternating Linear Scheme. This bound is guaranteed to be above the energy norm of the total error, including the discretization error, the tensor approximation error, and the error in the solver of linear algebraic equations. Numerical examples with the Poisson equation and the Schrödinger equation with the Henon-Heiles potential in up to 40 dimensions are presented to illustrate the efficiency of this approach.

*S. Dolgov is thankful to the Engineering and Physical Sciences Research Council (UK) for the support through Fellowship EP/M019004/1. **Email:** s.dolgov@bath.ac.uk

†T. Vejchodský gratefully acknowledges the support of the Czech Science Foundation, project no. 18-09628S, and the institutional support RVO 67985840. **Email:** vejchod@math.cas.cz

Keywords: high-dimensional problem, reaction-diffusion, Tensor Train, Alternating Linear Scheme, flux reconstruction, complementary energy
MSC2010: 65N15, 65N30, 15A69, 15A23, 65F10, 65N22

1 Introduction

This paper deals with the linear second-order elliptic partial differential equation of a reaction-diffusion type:

$$-\Delta u + \kappa^2 u = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ is a d-dimensional hyperrectangle, i.e. a Cartesian product of d intervals. The reaction coefficient $\kappa = \kappa(x) \geq 0$ is assumed variable and nonnegative in general and the homogeneous Dirichlet boundary conditions are considered for simplicity. Besides classical reaction-diffusion models in 2–3 dimensions, implicit time stepping schemes for the Schrödinger equation [32] require to solve an equation of the form (1), in arbitrarily high dimension.

This problem is solved by the finite element method using low rank tensor approximations, see Section 4, the book [22] or surveys [20, 26] for details. This enables us to compute an approximate solution even in high dimensional cases. Since the number of degrees of freedom grows exponentially with the dimension d, the traditional approaches are prohibitively expensive for higher values of d. This phenomenon is known as the curse of dimensionality [9], and the low rank tensor approximation methods allow to break it in many practical cases, reducing the computational costs and memory demands from exponential to polynomial in d. The main idea is to consider the expansion coefficients of the finite element solution as a d-dimensional tensor and approximate it by a low rank tensor decomposition. In particular we use the simple and robust Tensor Train (TT) decomposition [35].

Low rank tensor approximations bring further error to the computed solution. If a tensor is compressed from the full representation using the singular value decomposition [35], this error can be controlled. However, in practice the full storage is not possible, and one computes a TT representation directly, using iterative interpolation or solution techniques. In this case it is very difficult to obtain guaranteed and sharp estimates of the approximation error. Some existing results [6] rely on a particular solution method that might be not the fastest one. Guaranteed a priori estimates on the

convergence rate [16, 45] and approximation error [50, 47, 21] are often too pessimistic.

In this paper we propose a guaranteed a posteriori error estimator for a low rank high dimensional solution, which is independent of a particular approximation algorithm, while being locally efficient up to higher order terms [5].

This estimator is based on a complementary problem [51]. It is a second order elliptic partial differential equation that can be naturally discretized by Raviart–Thomas finite elements. Its solution is also approximated by a low rank tensor and used to compute the sharp upper bound on the energy norm of the total error of the solution of the original reaction-diffusion problem.

This type of guaranteed a posteriori error bounds for linear second order elliptic partial differential equations is already studied for many decades in the low dimensional case. The idea can be traced back to the method of hypercircle [43]. After decades of development it attracted a lot of attention in recent years, see books [2, 34, 44], papers [1, 11, 23, 31, 40, 41], and references there in. Guaranteed and robust error bounds for the particular reaction-diffusion problem (1) were derived in [12] for the vertex-centred finite volume method and in [1, 3, 4, 5, 49] for the finite element method.

The rest of the paper is organized as follows. Section 2 derives the guaranteed upper bound on the total error of the finite element solution. Section 3 introduces the complementary problem for the flux reconstruction. Section 4 presents the main idea of low rank tensor approximations, including the Cartesian grid, indexing, tensor train decomposition, the alternating scheme for solving systems of linear algebraic equations, tensorization of the complementary problem and the block alternating linear scheme for its solution, tensorized Gauss–Legendre quadrature, and a specific procedure for evaluation of the error estimator. Section 5 shows numerical results for the Poisson problem, reaction-diffusion problem, and Schrödinger equation with Henon–Heiles potential. Finally, Section 6 draws the conclusions and ideas for further research.

2 Guaranteed a posteriori error bound

We first introduce the needed notation. The well known Sobolev space $H_0^1(\Omega)$ consists of square integrable functions with square integrable distributional derivatives and with zero traces on the boundary $\partial\Omega$. The $L^2(\Omega)$ inner

product is denoted by $(\cdot,\cdot)$. The weak solution of problem (1) is introduced as a function $u \in H^1_0(\Omega)$ such that

$$(\boldsymbol{\nabla} u, \boldsymbol{\nabla} v) + (\kappa^2 u, v) = (f, v) \quad \forall v \in H^1_0(\Omega). \tag{2}$$

To guarantee integrability we consider $\kappa \in L^\infty(\Omega)$ and $f \in L^2(\Omega)$.

This problem is solved numerically using the standard finite element method of the first order and the finite element solution is further approximated by a low rank tensor as described below in Section 4. However, at this point the particular details about the numerical solution are not important, because the guaranteed error bound we will introduce is independent of the used numerical method and applies to arbitrary conforming approximation $u_h \in H^1_0(\Omega)$ of u.

In order to introduce the guaranteed error bound, we denote by $\mathcal{T}_h$ the usual finite element mesh of the domain Ω. More precisely, $\mathcal{T}_h$ is a set of closed d-dimensional hyperrectangles, called elements, which form a face-to-face partition of Ω. Symbol $\Pi_K : L^2(K) \to \mathbb{Q}_{1,1,\dots,1}(K)$ stands for the $L^2(K)$-orthogonal projector to the space $\mathbb{Q}_{1,1,\dots,1}(K)$ consisting of functions defined in the element $K \in \mathcal{T}_h$ and being linear in each of their d variables. It is useful to introduce symbol Π for elementwise concatenation of projectors Π_K such that $(\Pi f)|_K = \Pi_K f$ for all $K \in \mathcal{T}_h$. Further, we denote by $\|\cdot\|$ the norm in $L^2(\Omega)$ and by $\|\cdot\|_K$ the norm in $L^2(K)$ for $K \in \mathcal{T}_h$. The energy norm is given as $|\!|\!| v |\!|\!|^2 = \|\boldsymbol{\nabla} v\|^2 + \|\kappa v\|^2$ for all $v \in H^1_0(\Omega)$ and its restriction to elements $K \in \mathcal{T}_h$ as $|\!|\!| v |\!|\!|^2_K = \|\boldsymbol{\nabla} v\|^2_K + \|\kappa v\|^2_K$. It is convenient to recall the Poincaré inequality

$$\|v\| \le C_{\mathrm{P}} \, \|\boldsymbol{\nabla} v\| \quad \forall v \in H^1_0(\Omega). \tag{3}$$

The optimal value of the Poincaré constant is $C_{\mathrm{P}} = \lambda_1^{-1/2}$, where λ_1 is the first eigenvalue of the Dirichlet Laplacian in Ω. Since Ω is a hyperrectangle, the optimal value of C_{P} is known to be

$$C_{\mathrm{P}} = \pi^{-1} \left(\sum_{i=1}^{d} L_i^{-2} \right)^{-1/2}, \tag{4}$$

where L_i, $i = 1, 2, \dots, d$, are lengths of sides of the hyperrectangle Ω, see [33]. Guaranteed numerical bounds for C_{P} are also available [48].

Since zero values of the reaction coefficient κ cause technical difficulties, we introduce a constant shift parameter $\kappa_0 > 0$ and define

$$\widetilde{\kappa}(x) = \kappa(x) + \kappa_0. \tag{5}$$

In order to simplify the notation, we set $r = f - \kappa^2 u_h$ and denote by h_K the diameter of K.

The *error estimator*, *local error indicators* and the *oscillation* term are defined by rules

$$\eta^2(\boldsymbol{\tau}) = \sum_{K\in\mathcal{T}_h} (\eta_K(\boldsymbol{\tau}) + \operatorname{osc}_K(r))^2, \tag{6}$$

$$\eta_K^2(\boldsymbol{\tau}) = \|\boldsymbol{\tau} - \boldsymbol{\nabla} u_h\|_K^2 + \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K^2, \tag{7}$$

$$\operatorname{osc}_K(r) = \min\left\{h_K\pi^{-1}\|r - \Pi_K r\|_K, \left\|\kappa^{-1}(r - \Pi_K r)\right\|_K\right\}, \tag{8}$$

respectively. Quantity $\boldsymbol{\tau} \in \boldsymbol{H}(\operatorname{div},\Omega)$ is called the *flux*. At this moment it can be arbitrary, but its specific choice will be discussed in Section 3 below. Note that if the norm $\|\kappa^{-1}(r - \Pi_K r)\|_K$ is not defined, typically if $\kappa = 0$ in K, then we consider this norm to be infinite and $\operatorname{osc}_K(r) = h_K\pi^{-1}\|r - \Pi_K r\|_K$.

The following theorem is a generalization of [5, Theorem 1] for the variable coefficient κ. It provides the guaranteed and fully computable upper bound on the energy norm of the total error.

Theorem 1. *Let $u \in H_0^1(\Omega)$ be the weak solution given by (2). Let $u_h \in H_0^1(\Omega)$, $\boldsymbol{\tau} \in \boldsymbol{H}(\operatorname{div},\Omega)$, and $\kappa_0 > 0$ be arbitrary. Then*

$$|||u - u_h||| \le \eta(\boldsymbol{\tau}) + \kappa_0 C_{\mathrm{P}} \left(\sum_{K\in\mathcal{T}_h} \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K^2\right)^{1/2} \tag{9}$$

with C_{P} given by (4). Moreover, if $0 < \operatorname{ess\,inf}_\Omega \kappa$ then estimate (9) holds with $\kappa_0 = 0$.

Proof. Notation $v = u - u_h$, $r = f - \kappa^2 u_h$, weak formulation (2), and the divergence theorem yield identity

$$\begin{aligned}|||u - u_h|||^2 &= (\boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v) + (\kappa^2 u - \kappa^2 u_h, v) = \\ &\sum_{K\in\mathcal{T}_h} \left[(\boldsymbol{\tau} - \boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v)_K + (\Pi_K r + \operatorname{div}\boldsymbol{\tau}, v)_K + (r - \Pi_K r, v)_K\right],\end{aligned} \tag{10}$$

where $(\cdot,\cdot)$ and $(\cdot,\cdot)_K$ stand for the $L^2(\Omega)$ and $L^2(K)$ inner products, respectively. The last inner product on the right hand side can be estimated by the Cauchy–Schwarz inequality in two different ways:

$$\begin{aligned}(r - \Pi_K r, v)_K &\le \left\|\kappa^{-1}(r - \Pi_K r)\right\|_K \|\kappa v\|_K \le \left\|\kappa^{-1}(r - \Pi_K r)\right\|_K |||v|||_K, \\ (r - \Pi_K r, v)_K &= (r - \Pi_K r, v - \overline{v}_K)_K \le \|r - \Pi_K r\|_K \|v - \overline{v}_K\|_K \\ &\le h_K\pi^{-1}\|r - \Pi_K r\|_K |||v|||_K,\end{aligned}$$

where $\overline{v}_K = |K|^{-1}(v,1)_K$ stands for the integral average of v over K and the Poincaré inequality $\|v-\overline{v}_K\|_K \le h_K\pi^{-1}\|\boldsymbol{\nabla} v\|_K$ is used [42, 8]. Thus,

$$(r-\Pi_K r, v)_K \le \mathrm{osc}_K(r)\|\!|v|\!\|_K, \tag{11}$$

where we recall that if $\|\kappa^{-1}(r-\Pi_K r)\|_K$ is not defined due to vanishing κ then it is considered to be infinity.

Since $\widetilde{\kappa} > 0$, we multiply and divide the second inner product on the right-hand side of (10) by $\widetilde{\kappa}$, use the Cauchy–Schwarz inequality and (11) to obtain

$$\|\!|u-u_h|\!\|^2 \le \sum_{K\in\mathcal{T}_h} \left(\|\boldsymbol{\tau}-\boldsymbol{\nabla} u_h\|_K \|\boldsymbol{\nabla} v\|_K + \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K \|\widetilde{\kappa} v\|_K + \mathrm{osc}_K(r)\|\!|v|\!\|_K\right) \tag{12}$$

Triangle inequality $\|\widetilde{\kappa} v\|_K \le \|\kappa v\|_K + \|\kappa_0 v\|_K$ and bound

$$\begin{aligned}
&\|\boldsymbol{\tau}-\boldsymbol{\nabla} u_h\|_K \|\boldsymbol{\nabla} v\|_K + \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K \|\kappa v\|_K \\
&\quad \le \left(\|\boldsymbol{\tau}-\boldsymbol{\nabla} u_h\|_K^2 + \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K^2\right)^{1/2} \left(\|\boldsymbol{\nabla} v\|_K^2 + \|\kappa v\|_K^2\right)^{1/2}
\end{aligned}$$

provide an estimate

$$\|\!|u-u_h|\!\|^2 \le \sum_{K\in\mathcal{T}_h} \left(\eta_K(\boldsymbol{\tau}) + \mathrm{osc}_K(r)\right)\|\!|v|\!\|_K + \sum_{K\in\mathcal{T}_h} \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K \|\kappa_0 v\|_K . \tag{13}$$

Separate application of the Cauchy–Schwarz inequality to the first and second sum yields

$$\|\!|u-u_h|\!\|^2 \le \eta(\boldsymbol{\tau})\|\!|v|\!\| + \kappa_0 \|v\| \left(\sum_{K\in\mathcal{T}_h} \left\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\right\|_K^2\right)^{1/2} .$$

Poincaré inequality (3) and notation $v = u - u_h$ now implies the inequality (9).

Moreover, if $0 < \operatorname{ess\,inf}_\Omega \kappa$ then it is possible to choose $\kappa_0 = 0$. Function $\widetilde{\kappa}$ is then positive and the whole proof, specifically (12), remains valid. □

Note that the upper bound (9) can be simplified for the price of its slight increase. Indeed, using $\|\widetilde{\kappa}^{-1}(\Pi_K r + \operatorname{div}\boldsymbol{\tau})\|_K \leq \eta_K(\boldsymbol{\tau}) \leq \eta_K(\boldsymbol{\tau}) + \operatorname{osc}_K(r)$ in (13), we easily obtain bound

$$\|\!|u - u_h|\!\| \leq \left(1 + \kappa_0^2 C_{\mathrm{P}}^2\right)^{1/2} \eta(\boldsymbol{\tau}).$$

Further note that triangle inequality implies

$$\eta(\boldsymbol{\tau}) \leq \tilde{\eta}(\boldsymbol{\tau}) = \left(\sum_{K\in\mathcal{T}_h} \eta_K^2\right)^{1/2} + \left(\sum_{K\in\mathcal{T}_h} \operatorname{osc}_K^2(r)\right)^{1/2}, \tag{14}$$

and definition (7) gives

$$\sum_{K\in\mathcal{T}_h} \eta_K^2 = \|\boldsymbol{\tau} - \boldsymbol{\nabla} u_h\|^2 + \left\|\widetilde{\kappa}^{-1}(\Pi r + \operatorname{div}\boldsymbol{\tau})\right\|^2. \tag{15}$$

Since values κ^{-1} in (8) may cause quadrature errors and technical problems in higher dimension, we found the following estimate to be useful:

$$\operatorname{osc}_K^2(r) \leq \min\left\{h_K \pi^{-1}, \max_K \kappa^{-1}\right\} \|r - \Pi r\|_K. \tag{16}$$

Concerning the shift parameter κ_0, it should be chosen small. Ideally so small that $\kappa_0 C_{\mathrm{P}}$ is negligible with respect to 1 and $1 + \kappa_0 C_{\mathrm{P}} \approx 1$. If the reaction coefficient κ is bounded away from zero in Ω then neither the shift κ_0 nor the Poincaré constant C_{P} are needed and estimate (9) holds with $\kappa_0 = 0$.

An alternative upper bound on the energy norm of the error is based on the choice of the flux $\boldsymbol{\tau}$ such that it satisfies an equilibration condition, for example $\Pi_K r + \operatorname{div}\boldsymbol{\tau} = 0$ or $\int_K (\Pi_K r + \operatorname{div}\boldsymbol{\tau})\,\mathrm{d}\boldsymbol{x} = 0$ for all $K \in \mathcal{T}_h$, see [49]. This alternative upper bound is similar to the bound presented in Theorem 1 and the equilibration condition enables us to avoid technical difficulties with vanishing κ and neither the shift parameter κ_0 nor the Poincaré constant C_{P} are needed. However, we do not prefer this approach, because the equilibration condition cannot be satisfied exactly in practical computations due to round-off and tensor truncation errors and consequently, the upper bound on the error cannot be guaranteed. Moreover, the practical implementation of the equilibration condition is technically more involved, especially for high dimensional problems.

3 Flux reconstruction

Theorem 1 provides a guaranteed error bound for arbitrary flux $\boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, \Omega)$. However, in order to obtain a sharp error bound, the flux $\boldsymbol{\tau}$ has to be chosen carefully. A natural idea is to minimize the error estimator $\eta^2(\boldsymbol{\tau})$ over a finite dimensional subspace $\boldsymbol{W}_h \subset \boldsymbol{H}(\mathrm{div}, \Omega)$. We employ the standard finite element technology and choose the Raviart–Thomas finite element space

$$\boldsymbol{W}_h = \{\boldsymbol{\tau}_h \in \boldsymbol{H}(\mathrm{div}, \Omega) : \boldsymbol{\tau}_h|_K \in \mathbf{RT}_1(K) \quad \forall K \in \mathcal{T}_h\},$$

where

$$\mathbf{RT}_1(K) = [\mathbb{Q}_{2,1,\ldots,1}(K), \mathbb{Q}_{1,2,\ldots,1}(K), \ldots, \mathbb{Q}_{1,1,\ldots,2}(K)]^T$$

is the local Raviart–Thomas space on the d-dimensional hyperrectangle K and $\mathbb{Q}_{p_1,p_2,\ldots,p_d}(K)$ stands for the space of polynomials in K having degree at most p_s in variable x_s, $s = 1, 2, \ldots, d$.

Since the functional $\eta^2(\boldsymbol{\tau})$ is not quadratic, its minimization leads to a nonlinear problem. Therefore, we leave out the oscillation terms and define the *flux reconstruction* $\boldsymbol{\tau}_h \in \boldsymbol{W}_h$ as the minimizer of the quadratic functional (15) over the Raviart–Thomas space $\boldsymbol{W}_h$. The Euler–Lagrange equations corresponding to this minimization problem read

$$\big(\widetilde{\kappa}^{-2} \,\mathrm{div}\, \boldsymbol{\tau}_h, \mathrm{div}\, \boldsymbol{w}_h\big) + (\boldsymbol{\tau}_h, \boldsymbol{w}_h) = \big(\widetilde{\kappa}^{-2}[\kappa^2 u_h - \widetilde{\kappa}^2 u_h - f], \mathrm{div}\, \boldsymbol{w}_h\big) \qquad (17)$$

for all $\boldsymbol{w}_h \in \boldsymbol{W}_h$. The problem of finding $\boldsymbol{\tau}_h \in \boldsymbol{W}_h$ satisfying these equations is called the *complementary problem*. Note that the right-hand side of (17) is adjusted by using the divergence theorem. Interestingly, if κ is bounded away from zero in Ω and $\kappa_0 = 0$ then problem (17) is independent of u_h.

Let us note that a local efficiency result for the local error indicators $\eta_K(\boldsymbol{\tau}_h)$ is proved in [5] for the case of piecewise constant κ and for the flux reconstruction computed by minimizing local analogies of the quadratic functional (15) on patches of elements.

Since the space $\boldsymbol{W}_h$ is finite dimensional, problem (17) is equivalent to a system of linear algebraic equations. This system is solved by using low rank tensor approximations as described in the following section.

4 Low rank tensor approximations of u_h and $\boldsymbol{\tau}_h$

This section describes low rank tensor approximations of the finite element solution u_h and later also of the reconstructed flux $\boldsymbol{\tau}_h$. The standard finite element method suffers from the curse of dimensionality: both the memory requirements and the computational time grow exponentially with d. If the dimension d is much larger than 3, as in a typical Schrödinger equation, then they become prohibitively large. In this case, we approximate expansion coefficients representing u_h and $\boldsymbol{\tau}_h$ by suitable low rank tensor decompositions.

4.1 Cartesian grid and indexing

The domain $\Omega = (a_1, b_1) \times \cdots \times (a_d, b_d)$ is a Cartesian product of intervals (a_k, b_k), $k = 1, \ldots, d$. Within this domain, we introduce a Cartesian product grid,

$$z(i) = (z_1(i_1), \ldots, z_d(i_d)), \qquad a_k = z_k(0) < \cdots < z_k(i_k) < \cdots < z_k(n_k) = b_k,$$

for $k = 1, \ldots, d$, where i_k are individual indices of $n_k + 1$ nodes in the k-th direction, and i is the total index,

$$i = i_d + i_{d-1} \cdot n_d + \cdots + i_1 \cdot n_d \cdots n_2, \quad i_k = 0, \ldots, n_k. \tag{18}$$

Further, we define $h_k(i_k) = z_k(i_k) - z_k(i_k - 1)$ for $i_k = 1, 2, \ldots, n_k$.

Introducing finite elements

$$K(i) = [z_1(i_1 - 1), z_1(i_1)] \times \cdots \times [z_d(i_d - 1), z_d(i_d)], \tag{19}$$

we define the usual piecewise linear finite element space

$$V_h = \{v_h \in H_0^1(\Omega) : v_h|_{K(i)} \in \mathbb{Q}_{1,1,\ldots,1}(K(i)),\ i_k = 1, 2, \ldots, n_k,\ k = 1, 2, \ldots, d\}$$

and the finite element solution $u_h \in V_h$ of problem (1) by the identity

$$(\boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v_h) + (\kappa^2 u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h. \tag{20}$$

Defining the standard univariate piecewise linear and continuous hat functions

$$\varphi_{i_k}^{(k)}(x_k) = \begin{cases} \dfrac{x_k - z_k(i_k - 1)}{h_k(i_k)}, & \text{if } z_k(i_k - 1) \le x_k \le z_k(i_k), \\ \dfrac{z_k(i_k + 1) - x_k}{h_k(i_k + 1)}, & \text{if } z_k(i_k) \le x_k \le z_k(i_k + 1), \\ 0, & \text{otherwise,} \end{cases}$$

for $i_k = 1, 2, \ldots, n_k - 1$, $k = 1, 2, \ldots, d$, we express u_h as

$$u_h(x) = \sum_{i_1,\ldots,i_d=1}^{n_1-1,\ldots,n_d-1} \hat{u}(i_1, \ldots, i_d)\varphi_{i_1}^{(1)}(x_1) \cdots \varphi_{i_d}^{(d)}(x_d), \tag{21}$$

where $x = (x_1, x_2, \ldots, x_d)$. Expansion coefficients $\hat{u}(i_1, \ldots, i_d)$ are usually considered as elements of a vector, however, since all indices $i_1, \ldots, i_d$ can be varied independently of each other, they can be seen as a d-dimensional *tensor*.

For example, for $\kappa = 0$, plugging (21) into the finite element formulation (20), we obtain a structured linear system $\hat{A}\hat{u} = \hat{b}$, where

$$\hat{A} = L_1 \otimes M_2 \otimes \cdots \otimes M_d + \cdots + M_1 \otimes \cdots \otimes M_{d-1} \otimes L_d. \tag{22}$$

Moreover, assuming a separable factorization

$$f(x) = f^{(1)}(x_1)f^{(2)}(x_2) \cdots f^{(d)}(x_d),$$

we obtain a separable right hand side

$$\hat{b} = b_1 \otimes b_2 \otimes \cdots \otimes b_d.$$

Here,

$$L_k = \left[\left(\frac{\mathrm{d}\varphi_{i_k}^{(k)}}{\mathrm{d}x_k}, \frac{\mathrm{d}\varphi_{j_k}^{(k)}}{\mathrm{d}x_k}\right)\right] \quad \text{and} \quad M_k = \left[\left(\varphi_{i_k}^{(k)}, \varphi_{j_k}^{(k)}\right)\right]$$

are the "one-dimensional" stiffness and mass matrices, respectively, and $b_k = \left[(f^{(k)}, \varphi_{i_k}^{(k)})\right]$ is the "one-dimensional" load vector, $k = 1, 2, \ldots, d$. Symbol $\otimes$ stands for the Kronecker product, defined for any matrices $A = [a_{i,j}]$ and B as follows,

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \cdots & a_{1,m}B \\ \vdots & & \vdots \\ a_{n,1}B & \cdots & a_{n,m}B \end{bmatrix}.$$

Although the number of entries of the matrix $\hat{A}$ and of the load vector $\hat{b}$ grows exponentially with d, the number of entries in "one-dimensional" factors L_k, M_k, and b_k grows linearly. Therefore, the idea how to break the curse of dimensionality is to never actually compute Kronecker products and work directly with "one-dimensional" factors instead. The following subsection briefly describes a more general low rank decomposition for the tensors of expansion coefficients $\hat{u}$, load $\hat{b}$, and for the stiffness matrix $\hat{A}$.

4.2 TT decomposition

We employ the Tensor Train (TT) [35] low rank tensor approximation format. The tensor $\hat{u}$ is approximated by a tensor $\tilde{u} = [\tilde{u}(i_1, \dots, i_d)]$ that admits the TT decomposition

$$\tilde{u} = \sum_{\alpha_0,\dots,\alpha_d=1}^{r_0,\dots,r_d} u^{(1)}_{\alpha_0,\alpha_1} \otimes u^{(2)}_{\alpha_1,\alpha_2} \otimes \cdots \otimes u^{(d)}_{\alpha_d-1,\alpha_d}, \tag{23}$$

where the *TT blocks* $u^{(k)} = \left[u^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)\right]$ are three-dimensional tensors. Notice that if α_{k-1}, α_k are fixed then $u^{(k)}_{\alpha_{k-1},\alpha_k}$ is a vector of length $n_k - 1$. The auxiliary summation ranges $r_0, \dots, r_d$ are called *TT ranks*. For consistency with the left hand side, the border ranks are constrained to $r_0 = r_d = 1$, but the intermediate TT ranks can be larger than one, and depend on the desired approximation error $\hat{u} - \tilde{u}$. Assuming that all grid sizes and intermediate TT ranks are the same, $n_1 = \cdots = n_d = n$ and $r_d = \cdots = r_{d-1} = r$, we conclude that the TT decomposition contains $\mathcal{O}(dnr^2)$ unknowns. If the TT rank r is moderate, this can be much smaller than the original $(n-1)^d$ unknowns.

Many closed-form functions admit TT approximations (or even exact decompositions) of their coefficient tensors with low TT ranks [36, 13]. For example, the function $u(x) = \sin(\pi x_1) \cdots \sin(\pi x_d) \in H^1_0((0,1)^d)$ can be approximated on the above Cartesian grid by a rank-1 TT decomposition (23) with $u^{(k)}(i_k) = \sin(\pi i_k / n_k)$, $k = 1, \dots, d$. Smooth functions can often be approximated by truncated series, where each term is low-rank, which gives a bound on the total rank [50, 47]. For example, the solution of the Poisson equation (i.e. problem (1) with $\kappa = 0$) with a low-rank right hand side can be approximated by a tensor in the TT format with a relative error ε and TT ranks bounded by $\mathcal{O}(\log^2 \varepsilon)$ [19].

In general, any tensor can be approximated by a TT decomposition using $\mathcal{O}(dnr^2)$ entries from the tensor via cross interpolation algorithms [38, 46], allowing r to be large enough. Of course, in practice we are interested in problems where r can be taken small, e.g. independent of (or mildly dependent on) the dimension d and/or the grid size n.

Structured matrices can also be represented in a TT format. For example, one can prove [25] that the stiffness matrix (22) can be represented as

$$\hat{A} = \sum_{\beta_0,\dots,\beta_d=1}^{R_0,\dots,R_d} A^{(1)}_{\beta_0,\beta_1} \otimes \cdots \otimes A^{(d)}_{\beta_{d-1},\beta_d}. \tag{24}$$

with $R_1 = \cdots = R_{d-1} = 2$ and the following TT blocks:

$$A^{(1)} = \begin{Bmatrix} L_1 & M_1 \end{Bmatrix}, \quad A^{(k)} = \begin{Bmatrix} M_k & 0 \\ L_k & M_k \end{Bmatrix}, \quad A^{(d)} = \begin{Bmatrix} M_d \\ L_d \end{Bmatrix},$$

where $k = 2, \ldots, d-1$, the rows of the curly bracket matrices correspond to the rank index β_{k-1}, and the columns correspond to β_k (e.g. $A^{(k)}_{1,1} = A^{(k)}_{2,2} = M_k$, $A^{(k)}_{2,1} = L_k$, $A^{(k)}_{1,2} = 0$). This clearly reduces the storage costs compared to the direct summation of Kronecker product terms in (22).

The product $\hat{A}\tilde{u}$ can be explicitly expressed as another TT decomposition with TT ranks $r_0R_0, \ldots, r_dR_d$ [35], without expanding the Kronecker products. Moreover, the result can be *re-approximated* up to quasi-optimal ranks for the given accuracy using the singular value decomposition (SVD) with the cost proportional to d.

4.3 Alternating scheme for solving linear equations

First attempts to solve systems of linear algebraic equations with decomposed tensors were based on traditional iterative methods such as Richardson, CG, and GMRES with matrix-vector products and other operations implemented on TT blocks followed by the rank truncation [30, 28, 7, 14]. This approach applies to any low rank tensor decomposition format. However, in realistic problems the TT ranks of intermediate (e.g. Krylov) vectors grow rapidly, exceeding the optimal ranks of the solution significantly, unless a good preconditioner is used.

A more robust technique that is commonly adopted nowadays is the Alternating Linear Scheme (ALS) [24] and its extensions [37, 16, 29]. These methods project the equations onto bases constructed from the TT decomposition of the solution itself, and therefore avoid decompositions of auxiliary vectors with high TT ranks. We emphasize that the TT form (23) of the solution is crucial for this technique.

Let us start with the standard ALS algorithm, suitable for the primal problem (20). The TT decomposition (23) can be rewritten as a linear map from the elements of a k-th TT block to the elements of $\tilde{u}$. Given (23), let us define a partial TT decomposition by

$$U^{(q\cdots k)}_{\alpha_{q-1},\alpha_k} = \sum_{\alpha_q,\ldots,\alpha_{k-1}=1}^{r_q,\ldots,r_{k-1}} u^{(q)}_{\alpha_{q-1},\alpha_q} \otimes \cdots \otimes u^{(k)}_{\alpha_{k-1},\alpha_k} \in \mathbb{R}^{(n_q-1)\cdots(n_k-1)}, \qquad (25)$$

for any $1 \le q \le k \le d$. Notice that $U^{(1\cdots k)}$ and $U^{(q\cdots d)}$ can be seen as $\prod_{\ell=1}^{k}(n_\ell - 1) \times r_k$ and $r_{q-1} \times \prod_{\ell=q}^{d}(n_\ell - 1)$ matrices, respectively; for this reason, they are called *interface* matrices. Now we introduce the so-called *frame* matrix by replacing $u^{(k)}$ in (23) by the identity matrix I:

$$U_{\neq k} = U^{(1\cdots k-1)} \otimes I \otimes \left(U^{(k+1\cdots d)}\right)^\top \in \mathbb{R}^{\prod_{\ell=1}^{d}(n_\ell-1)\times(r_{k-1}(n_k-1)r_k)}, \tag{26}$$

where I is of size $n_k - 1$. In addition, we introduce the notation $\bar{u}^{(k)}$ for the k-th TT block stretched into a vector of length $r_{k-1}(n_k - 1)r_k$. It is then easy to prove [24] that the TT decomposition (23) is equivalent to a linear map

$$\tilde{u} = U_{\neq k}\bar{u}^{(k)} \quad \text{for each} \quad k = 1, \dots, d.$$

If we plug this decomposition into the linear system $\hat{A}\hat{u} = \hat{f}$ (replacing $\hat{u}$ by $\tilde{u}$), we obtain an overdetermined linear system for $\bar{u}^{(k)}$ (and hence for elements of $u^{(k)}$). The simplest way to resolve it is to project the equation onto the same frame matrix $U_{\neq k}$,

$$\left(U_{\neq k}^\top \hat{A} U_{\neq k}\right) \bar{u}^{(k)} = \left(U_{\neq k}^\top \hat{f}\right). \tag{27}$$

The ALS algorithm now iterates over $k = 1, \dots, d$ (hence the name alternating), solving the reduced system (27) of size $r_{k-1}(n_k - 1)r_k$ in each step. Once a TT block $u^{(k)}$ is updated from (27), it is used in the construction of the frame matrix (26) in the next step. If the matrix $\hat{A}$ is symmetric positive definite (SPD) then the projected system (27) can be rigorously related to an optimization problem. Consequently, the ALS algorithm can be related to the nonlinear block Gauss–Seidel method and the local convergence can be proved [45].

For the numerical efficiency one can notice that the reduced matrix and right hand side in (27) can be constructed from the TT blocks of $\hat{A}, \hat{u}$ and $\hat{f}$ without ever expanding the Kronecker products. In a sequential iteration $k = 1, 2, \dots, d$, partial projections can be cached such that the cost of each ALS step becomes independent of d [24, 37]. The TT blocks can be *enriched* with auxiliary vectors, such as approximate residuals [16], which gives a mechanism for increasing TT ranks and their adaptation to the desired accuracy.

4.4 TT decomposition for the complementary problem

Since the complementary solution $\boldsymbol{\tau}_h \in \boldsymbol{W}_h$ is a vector field with d components, the corresponding linear operator in (17) has a $d \times d$ block structure. Therefore it is important to adopt a particular TT decomposition for expansion coefficients of $\boldsymbol{\tau}_h$. This specific structure will be useful for the tailored iterative solver described in Subsection 4.5. As a by-product, we obtain a low rank algorithm which might be efficient for the solution of more general equations as well.

The general idea for the TT decomposition of $\boldsymbol{\tau}_h$ is similar to the decomposition of u_h. However, the construction of the Cartesian product Raviart–Thomas space $\mathbf{RT}_1$ is more complicated. The s-th component $\tau_h(s,x)$ of $\boldsymbol{\tau}_h(x)$, $s = 1,\dots,d$, is a piecewise polynomial function that is piecewise quadratic and continuous in the s-th variable and piecewise linear and discontinuous in all the other variables. Let $\hat{\varphi}^{(k)}_{j_k}(x_k)$, $j_k = 1,2,\dots,2n_k$, denote piecewise linear and discontinuous functions defined in $[z_k(j_k-1), z_k(j_k)]$ as $(z_k(j_k) - x_k)/h_k(j_k)$ for j_k odd and as $(x_k - z_k(j_k-1))/h_k(j_k)$ for j_k even and vanishing elsewhere in (a_k, b_k), $k = 1,2,\dots,d$. Similarly, let $\phi^{(k)}_{i_k}(x_k)$, $i_k = 0,1,\dots,2n_k$, $k = 1,2,\dots,d$, be the usual piecewise quadratic and continuous "one-dimensional" basis functions. Then components of the vector field $\boldsymbol{\tau}_h \in \boldsymbol{W}_h$ can be expanded in these basis functions as follows

$$\begin{aligned}\tau_h(s,x) = \sum_{j_1,\dots,j_{s-1},j_{s+1},\dots,j_d=1}^{2n_1,\dots,2n_{s-1},2n_{s+1},\dots,2n_d} \sum_{i_s=0}^{2n_s} \hat{\tau}(s, j_1,\dots,j_{s-1}, i_s, j_{s+1},\dots,j_d) \\ \cdot \hat{\varphi}^{(1)}_{j_1}(x_1)\cdots\hat{\varphi}^{(s-1)}_{j_{s-1}}(x_{s-1})\cdot\phi^{(s)}_{i_s}(x_s)\cdot\hat{\varphi}^{(s+1)}_{j_{s+1}}(x_{s+1})\cdots\hat{\varphi}^{(d)}_{j_d}(x_d). \quad (28)\end{aligned}$$

This expansion enables us to express the complementary problem (17) in the following block structure:

$$\begin{bmatrix} B_{1,1} & \cdots & B_{1,d} \\ \vdots & \cdots & \vdots \\ B_{d,1} & \cdots & B_{d,d} \end{bmatrix} \begin{bmatrix} \hat{\tau}(1) \\ \vdots \\ \hat{\tau}(d) \end{bmatrix} = \begin{bmatrix} g_1 \\ \vdots \\ g_d \end{bmatrix}, \quad (29)$$

where

$$
\begin{aligned}
B_{s,s} &= (\widetilde{\kappa}^{-2}\partial_s\psi^{(s)}, \partial_s\psi^{(s)}) + (\psi^{(s)}, \psi^{(s)}), \quad s = 1, \ldots, d, \\
B_{s,k} &= (\widetilde{\kappa}^{-2}\partial_s\psi^{(s)}, \partial_k\psi^{(k)}), \qquad s \neq k, \\
g_s &= (\widetilde{\kappa}^{-2}[\kappa^2 u_h - \widetilde{\kappa}^2 u_h - f], \partial_s\psi^{(s)}), \\
\psi^{(s)}(x) &= \left[\hat{\varphi}^{(1)}_{j_1}(x_1)\cdots\hat{\varphi}^{(s-1)}_{j_{s-1}}(x_{s-1})\phi^{(s)}_{i_s}(x_s)\hat{\varphi}^{(s+1)}_{j_{s+1}}(x_{s+1})\cdots\hat{\varphi}^{(d)}_{j_d}(x_d)\right],
\end{aligned}
\tag{30}
$$

and ∂_s stands for the partial derivative $\partial/\partial x_s$.

Since all blocks $B_{s,k}$ are nonzero, the components of $\hat{\tau}$ are coupled through (29), and they should be approximated in the same TT decomposition, in order to apply the ALS technique. This requires all components to have same dimensions. This can be achieved by expanding the ranges of the summation indices $j_1, \ldots, j_{s-1}, j_{s+1}, \ldots, j_d$ in (28) to start from zero, and by the corresponding expansion of the tensor $\hat{\tau}$ by zeros. With this slight abuse of notation, the complementary solution is approximated by

$$
\begin{aligned}
\tau_h(s, x) = &\sum_{i_1,\ldots,i_d=0}^{2n_1,\ldots,2n_d} \hat{\tau}(s, i_1, \ldots, i_d) \\
&\cdot \hat{\varphi}^{(1)}_{i_1}(x_1)\cdots\hat{\varphi}^{(s-1)}_{i_{s-1}}(x_{s-1}) \cdot \phi^{(s)}_{i_s}(x_s) \cdot \hat{\varphi}^{(s+1)}_{i_{s+1}}(x_{s+1})\cdots\hat{\varphi}^{(d)}_{i_d}(x_d),
\end{aligned}
\tag{31}
$$

and $\hat{\tau}(s, i_1, \ldots, i_d)$ are entries of the $d \times (2n_1 + 1) \times \cdots \times (2n_d + 1)$ tensor. Consequently, all blocks $B_{s,k}$, $s, k = 1, 2, \ldots, d$, are of size $\prod_{\ell=1}^{d}(2n_\ell + 1) \times \prod_{\ell=1}^{d}(2n_\ell + 1)$.

4.5 Block ALS for the complementary problem

In principle, we could consider $\hat{\tau}$ as a $(d+1)$-dimensional tensor as described above, approximate it in a TT decomposition with $d + 1$ blocks, and apply the standard ALS algorithm to the tensorized linear system (29). However, due to the special meaning of the first dimension, this ALS algorithm might be inefficient, as we demonstrate by the numerical experiments in Section 5. Instead, we incorporate the component index s into the TT block which is being evaluated in the current ALS step, using the so-called block TT format [15].

The block TT approximation $\tilde{\tau}(s, i_1, \ldots, i_d) \approx \hat{\tau}(s, i_1, \ldots, i_d)$ is defined

as

$$\tilde{\tau}(s) = \sum_{\alpha_0,\ldots,\alpha_d=1}^{r_0,\ldots,r_d} \tau^{(1)}_{\alpha_0,\alpha_1} \otimes \cdots \otimes \tau^{(k-1)}_{\alpha_{k-2},\alpha_{k-1}} \otimes \bar{\tau}^{(k)}_{\alpha_{k-1},\alpha_k}(s) \otimes \tau^{(k+1)}_{\alpha_k,\alpha_{k+1}} \otimes \cdots \otimes \tau^{(d)}_{\alpha_{d-1},\alpha_d}, \tag{32}$$

where s can appear in any k-th TT block for $k = 1, \ldots, d$. Both TT blocks $\tau^{(k)}_{\alpha_{k-1},\alpha_k}$ and $\bar{\tau}^{(k)}_{\alpha_{k-1},\alpha_k}(s)$ have lengths $2n_k + 1$ for fixed α_{k-1}, α_k, and s. The benefit of this block TT decomposition for the ALS method stems from the form of the frame matrix. Similarly to (25), we introduce

$$T^{(q\cdots k)}_{\alpha_{q-1},\alpha_k} = \sum_{\alpha_q,\ldots,\alpha_{k-1}=1}^{r_q,\ldots,r_{k-1}} \tau^{(q)}_{\alpha_{q-1},\alpha_q} \otimes \cdots \otimes \tau^{(k)}_{\alpha_{k-1},\alpha_k}$$

and notice that the k-th frame matrix $T_{\neq k} = T^{(1\cdots k-1)} \otimes I \otimes \left(T^{(k+1\cdots d)}\right)^\top$, cf. (26), consists of TT blocks without the component index s. This allows us to reduce the block matrix in (29) component by component, preserving and exploiting its block structure for efficient solution of the reduced problem.

Specifically, the *block ALS* method [10] computes the block TT approximation of $\hat{\tau}$ by iterating over $k = 1, \ldots, d$ and solving in each step the reduced system

$$\begin{bmatrix} T_{\neq k}^\top B_{1,1} T_{\neq k} & \cdots & T_{\neq k}^\top B_{1,d} T_{\neq k} \\ \vdots & \cdots & \vdots \\ T_{\neq k}^\top B_{d,1} T_{\neq k} & \cdots & T_{\neq k}^\top B_{d,d} T_{\neq k} \end{bmatrix} \begin{bmatrix} \bar{\tau}^{(k)}(1) \\ \vdots \\ \bar{\tau}^{(k)}(d) \end{bmatrix} = \begin{bmatrix} T_{\neq k}^\top g_1 \\ \vdots \\ T_{\neq k}^\top g_d \end{bmatrix} \tag{33}$$

of size $r_{k-1}(2n_k + 1)r_k d$. Note that when we switch to the next step ($k - 1$ or $k + 1$), the index s in the block TT format (32) needs to be moved to the corresponding ($k - 1$ or $k + 1$) TT block, such that the new frame matrix remains independent of s. This operation can be performed using SVD.

For example, suppose we need to move the index s from the k-th TT block to the $(k + 1)$-th TT block. First, we stretch $\bar{\tau}^{(k)}$ into a $r_{k-1}(2n_k + 1) \times dr_k$ matrix by grouping indices α_{k-1} and i_k into a new row index, and s and α_k into a new column index,

$$T^{(k)}(\alpha_{k-1}, i_k;\ s, \alpha_k) = \bar{\tau}^{(k)}_{\alpha_{k-1},\alpha_k}(i_k, s).$$

Computing the SVD of $T^{(k)}$ and truncating it up to a desired accuracy, we

obtain

$$T^{(k)}(\alpha_{k-1}, i_k;\ s, \alpha_k) \approx \sum_{\alpha'_k=1}^{r'_k} P(\alpha_{k-1}, i_k;\ \alpha'_k)\sigma(\alpha'_k)Q(\alpha'_k;\ s, \alpha_k),$$

where P and Q are matrices of left and right singular vectors, respectively, and σ are singular values. In the new block TT decomposition of $\tilde{\tau}(s)$ the TT block $\bar{\tau}^{(k)}(s)$ in (32) is replaced by

$$\tau^{(k)}_{\alpha_{k-1},\alpha'_k}(i_k) = P(\alpha_{k-1}, i_k;\ \alpha'_k)$$

and the TT block $\tau^{(k+1)}$ is changed to

$$\bar{\tau}^{(k+1)}_{\alpha'_k,\alpha_{k+1}}(i_{k+1}, s) = \sum_{\alpha_k=1}^{r_k} \sigma(\alpha'_k)Q(\alpha'_k;\ s, \alpha_k)\tau^{(k+1)}_{\alpha_k,\alpha_{k+1}}(i_{k+1}).$$

Notice that the new block TT decomposition has the same form as (32) except that s is located in the $(k+1)$-th block, and the k-th TT rank is r'_k. This new rank may differ from r_k in general, but in practical computations the difference is often insignificant. This procedure can be continued or reversed in order to place s into any desired block.

The size of (33) might still be rather large for the direct solution. Typical values of TT ranks range from 10 to 50, and the grid sizes n_k can range from tens to hundreds. However, the matrix inherits the TT decomposition, and hence a fast matrix-vector product can be implemented for iterative solvers [37]. Specifically, we use GMRES with a block Jacobi preconditioner with respect to the rank indices α_{k-1}, α_k in the solution. Recall that $\bar{\tau}^{(k)}(s)$ is enumerated by three independent indices $\alpha_{k-1}, \alpha_k, i_k$, and hence each matrix $\hat{B}_{s,l} = T^{\top}_{\neq k} B_{s,l} T_{\neq k}$ can in turn be seen as a three-level block matrix, $\hat{B}_{s,l} = [\hat{B}_{s,l}(\alpha_{k-1}, \alpha_k, i_k;\ \alpha'_{k-1}, \alpha'_k, i'_k)]$. The preconditioner is constructed by extracting the diagonal with respect to the first two levels, i.e. $\tilde{B}_{s,l} = [\hat{B}_{s,l}(\alpha_{k-1}, \alpha_k, i_k;\ \alpha_{k-1}, \alpha_k, i'_k)\delta(\alpha_{k-1}, \alpha'_{k-1})\delta(\alpha_k, \alpha'_k)]$.

In order to construct (33) efficiently, we need to decompose the components of the matrix $B_{s,l}$ and the right hand side g_s in the TT formats (24) and (23). This step relies crucially on the approximation of $\widetilde{\kappa}^{-2}(x)$, f, and other functions. We collocate them on a special quadrature grid using the TT Cross method [38]. This method together with the Cartesian structure of the finite elements enables us to easily construct the linear and bilinear forms in TT formats for both the primal and the complementary problem.

4.6 Tensorized Gauss–Legendre quadrature

Error indicators (7), oscillation terms (8), as well as entries of involved matrices and vectors are given as integrals over finite elements. These integrals are computed by tensorized Gauss–Legendre quadrature. Utilizing the Cartesian structure of the finite elements, we use one-dimensional Gauss–Legendre quadrature in each interval $[z_k(i_k-1), z_k(i_k)]$, constituting the element (19). Assuming that m quadrature nodes are introduced in each interval, we end up with the total of mn_k quadrature nodes $a_k \le y_k(1) < \cdots < y_k(mn_k) \le b_k$ for the k-th direction, and with the corresponding quadrature weights $w_k(j_k)$, $j_k = 1, \dots, mn_k$.

Now, all integrated functions in weak formulations (2), (17) and definitions (7), (8) are approximated by tensors of collocation values at $y_k(j_k)$. For example, to evaluate the first scalar product in (30), we introduce a tensor

$$\hat\sigma = [\hat\sigma(j_1, \dots, j_d)], \quad \hat\sigma(j_1, \dots, j_d) = \widetilde\kappa^{-2}(y_1(j_1), \dots, y_d(j_d))$$

of values of $\widetilde\kappa^{-2}$ at the quadrature nodes and approximate it by the TT Cross algorithm as

$$\hat\sigma \approx \tilde\sigma = \sum_{\alpha_0,\dots,\alpha_d=1}^{r_0,\dots,r_d} \sigma^{(1)}_{\alpha_0,\alpha_1} \otimes \cdots \otimes \sigma^{(d)}_{\alpha_{d-1},\alpha_d}.$$

This enables us to approximate entries of the corresponding matrix as

$$(\widetilde\kappa^{-2}\partial_s\psi^{(s)}_i, \partial_s\psi^{(s)}_j) \approx \sum_{\ell_1,\dots,\ell_d=1}^{mn_1,\dots,mn_d} w_1(\ell_1)\cdots w_d(\ell_d)\tilde\sigma(\ell)\partial_s\psi^{(s)}_i(y(\ell))\partial_s\psi^{(s)}_j(y(\ell)), \tag{34}$$

where $y(\ell) = (y_1(\ell_1), \dots, y_d(\ell_d))$ and i, j are multiindices with components i_q, j_q in range $0, 1, \dots, 2n_q+1$, $q = 1, 2, \dots, d$, see (31). Due to the tensor product structure of $\tilde\sigma$ and $\psi^{(s)}$, all terms in (34) corresponding to a fixed direction can be grouped together, and the sum over ℓ can be implemented as a product of sums over individual ℓ_k with the total computational cost being linear in d. Moreover, the loops over i and j can be factorized as well, such that the whole matrix on the left hand side of (34) can be written in the matrix TT format (24) block by block, e.g. $\sum_{\ell_s} w_s(\ell_s)\sigma^{(s)}_{\alpha_{s-1},\alpha_s}(\ell_s)\partial_s\phi^{(s)}(y_s(\ell_s))\partial_s\phi^{(s)}(y_s(\ell_s))$ becomes the s-th TT block, and so on. Since the elements have finite support, the total cost of this operation is $\mathcal{O}(dnr^2)$, where r is the maximal TT rank of $\tilde\sigma$.

4.7 Evaluation of the error estimator

Importantly, Theorem 1 holds for arbitrary $u_h \in H_0^1(\Omega)$ and arbitrary $\boldsymbol{\tau}_h \in \boldsymbol{H}(\mathrm{div}, \Omega)$. Therefore, the upper bound property (9) holds even if u_h and $\boldsymbol{\tau}_h$ are polluted by various errors, including iteration errors in the ALS algorithm and tensor truncation errors. However, the upper bound (9) needs to be evaluated exactly. Given the approximate solution u_h and the complementary solution $\boldsymbol{\tau}_h$ in the TT format, it might be difficult to evaluate $\eta(\boldsymbol{\tau}_h)$ in (9) without tensor truncation errors. Specifically, if we are interested in the *elementwise* error estimation, we run the TT Cross to approximate the terms in (7), then we approximate the pointwise square root of $\eta_K^2(\boldsymbol{\tau})$, and finally we use the TT Cross again to compute $\eta(\boldsymbol{\tau})$ from (6). If we choose the TT Cross approximation tolerance much smaller than that used for the ALS solvers then this approximation of (9) will likely remain an upper bound on the error. However, it is not guaranteed and, moreover, it might result in large TT ranks of some of intermediate quantities.

On the other hand, a guaranteed bound can be computed if $\eta(\boldsymbol{\tau})$ is replaced by its upper bound $\tilde{\eta}(\boldsymbol{\tau})$ given by (14)–(15), because $\tilde{\eta}(\boldsymbol{\tau})$ can be evaluated exactly (up to round-off and possible quadrature errors) due to its favourable separable structure. After solving primal and complementary problems, we are given TT approximations of $\boldsymbol{\tau}_h$, $\mathrm{div}\ \boldsymbol{\tau}_h$, $\boldsymbol{\nabla} u_h$, $\widetilde{\kappa}^{-1}$ and Πr. We interpolate them at quadrature nodes $y_k(\ell_k)$ and multiply the error terms by square roots of the quadrature weights, e.g.

$$\begin{aligned}
\tilde{\eta}_1(\ell_1, \dots, \ell_d) &= \sqrt{w_1(\ell_1) \cdots w_d(\ell_d)}\, (\boldsymbol{\tau}_h(y(\ell)) - \boldsymbol{\nabla} u_h(y(\ell)))\,, \\
\tilde{\eta}_2(\ell_1, \dots, \ell_d) &= \sqrt{w_1(\ell_1) \cdots w_d(\ell_d)} \widetilde{\kappa}^{-1}(y(\ell))\, (\Pi r(y(\ell)) + \mathrm{div}\ \boldsymbol{\tau}_h(y(\ell)))\,,
\end{aligned} \tag{35}$$

$\ell_k = 1, 2, \dots, mn_k$, $k = 1, 2, \dots, d$. These tensors depend *polylinearly* on $\boldsymbol{\tau}_h, u_h, \tilde{\kappa}^{-1}$ and r, and thus their TT decompositions can be constructed *exactly*, followed by taking exact Frobenius norms [35] of the TT formats,

$$\sum_{K \in \mathcal{T}_h} \eta_K^2(\boldsymbol{\tau}_h) = \|\tilde{\eta}_1\|_F^2 + \|\tilde{\eta}_2\|_F^2,$$

where we recall that $\eta_K^2(\boldsymbol{\tau}_h)$ satisfy (15). An upper bound on the oscillation term can be easily computed by using (16), where $\|r - \Pi r\|$ and $\min\{h_K \pi^{-1}, \max_K \kappa^{-1}\}$ are computed separately.

5 Numerical examples

5.1 Constant reaction coefficient

As an example, we consider the reaction-diffusion problem (1) in the unit cube $\Omega = (0,1)^d$ with the constant reaction coefficient κ^2, and the right hand side

$$f(x) = \sum_{k=1}^{d} \left[8 + \kappa^2 \left(1 - 4(x_k - 0.5)^2\right)\right] \prod_{\substack{i=1 \\ i \neq k}}^{d} \left(1 - 4(x_i - 0.5)^2\right).$$

chosen such that the exact solution is

$$u(x) = \prod_{k=1}^{d} \left(1 - 4(x_k - 0.5)^2\right).$$

We are to investigate how the accuracy of the proposed error estimator depends on the number of grid points n, the dimension d, and the reaction coefficient κ^2.

For the computation of matrix elements and error estimators as described in Section 4.6, we use $m = 4$ Gauss–Legendre points in each interval of the grid. Since κ is constant and f is a polynomial of degree at most 2 in each variable, this quadrature rule is exact for our computations.

For comparison, we present error estimators $\eta(\boldsymbol{\tau}_h)$ and $\tilde{\eta}(\boldsymbol{\tau}_h)$ given by (6) and (14) as well as the corresponding guaranteed error bounds given by the right hand side of (9):

$$E(\boldsymbol{\tau}_h) = \eta(\boldsymbol{\tau}_h) + \kappa_0 C_{\mathrm{P}} \left(\sum_{K \in \mathcal{T}_h} \left\| \widetilde{\kappa}^{-1} (\Pi_K r + \operatorname{div} \boldsymbol{\tau}_h) \right\|_K^2 \right)^{1/2},$$

$$\tilde{E}(\boldsymbol{\tau}_h) = \tilde{\eta}(\boldsymbol{\tau}_h) + \kappa_0 C_{\mathrm{P}} \left(\sum_{K \in \mathcal{T}_h} \left\| \widetilde{\kappa}^{-1} (\Pi_K r + \operatorname{div} \boldsymbol{\tau}_h) \right\|_K^2 \right)^{1/2}.$$

Estimator $\tilde{\eta}(\boldsymbol{\tau}_h)$ and error bound $\tilde{E}(\boldsymbol{\tau}_h)$ are evaluated as described in Section 4.7. Estimator $\eta(\boldsymbol{\tau}_h)$ and error bound $E(\boldsymbol{\tau}_h)$ cannot be computed without tensor truncation errors and their approximate values are computed using the TT Cross algorithm. The accuracy of these error estimators and error

bounds is measured by indices of effectivity $I_{\text{eff}}^{(\mathcal{E})} = \mathcal{E}/|||u - u_h|||$, where $\mathcal{E}$ stands for η, E, $\tilde{\eta}$, and $\tilde{E}$, respectively.

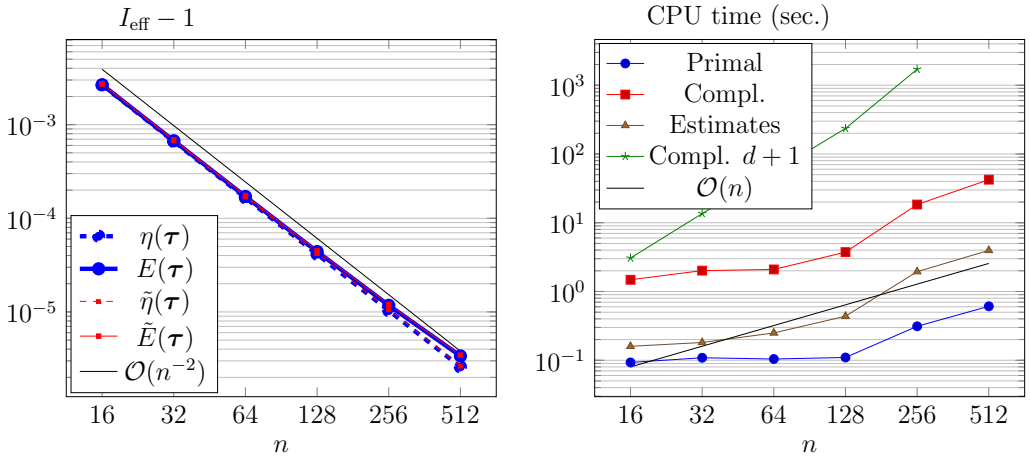
We start with the dependence of indices of effectivity on the number of grid points n. We fix $\kappa = 0$ and $d = 3$. The shift parameter is chosen as $\kappa_0 = 0.1$ and TT approximation thresholds as $\delta_p = 10^{-3}$ and $\delta_c = 10^{-6}$ for the primal and complementary problem, respectively. Figure 1 (left) shows differences of I_{eff} from 1. We can notice that all estimators are almost indistinguishable for this example and converge with the second order $\mathcal{O}(h^2)$ as the grid is refined. This is due to sufficiently small κ_0. The only and very minor difference caused by the second term in (9) can be spotted only for very fine grids. Error bounds E and $\tilde{E}$ are guaranteed by Theorem 1 to have indices of effectivity above one. Interestingly, all indices of effectivity of estimators η and $\tilde{\eta}$ are above one as well. Further notice that the optimal estimator $\eta(\boldsymbol{\tau}_h)$ is indistinguishable from the simpler version $\tilde{\eta}(\boldsymbol{\tau}_h)$.

In Figure 1 (right) we plot CPU times for individual parts of the scheme: the assembly and solution of the primal problem (20) ("Primal"), the assembly and solution of the complementary problem (17) ("Compl."), and the evaluation of estimators η, $\tilde{\eta}$, E and $\tilde{E}$ ("Estimates"). Note that the complementary problem is solved using the block TT version described in Section 4.5. For comparison we also present the CPU time for the solution of the complementary problem by the simple ALS applied to a $(d+1)$-dimensional problem, mentioned at the beginning of Section 4.5 ("Compl. $d+1$").

We can notice that the computational cost is asymptotically linear in n although we are solving a three-dimensional problem with approximately n^3 degrees of freedom. This is due to uniform boundedness of TT ranks of all TT approximations. In particular, the TT ranks of $\boldsymbol{\tau}_h$ are bounded by 5 and TT ranks of $\eta_K(\boldsymbol{\tau}_h)^2$ are bounded by 14 in this example. Not only the approximate solution is computed quickly, error estimator $\tilde{E}(\boldsymbol{\tau}_h)$ guarantees its high accuracy. Concerning the complementary problem, the simple ALS is much slower than the block TT solver due to a larger number of GMRES iterations needed for solving (27) compared to (33). Therefore, we proceed with the block TT version in the rest of the paper.

Next, we present how the estimators behave with respect to the dimension d. We keep $\kappa = 0$ and fix $n = 128$. Due to the conditioning of the complementary problem in higher dimensions, we set a larger shift $\kappa_0 = 1$ and relax the tolerance of the TT solver for the complementary problem to $\delta_c = 10^{-4}$. With this setting the block TT solver converges, but the quality

Figure 1: Differences of effectivity indices from 1 (left) and CPU times (right) vs. the number of grid points in each direction for the 3D Poisson problem.

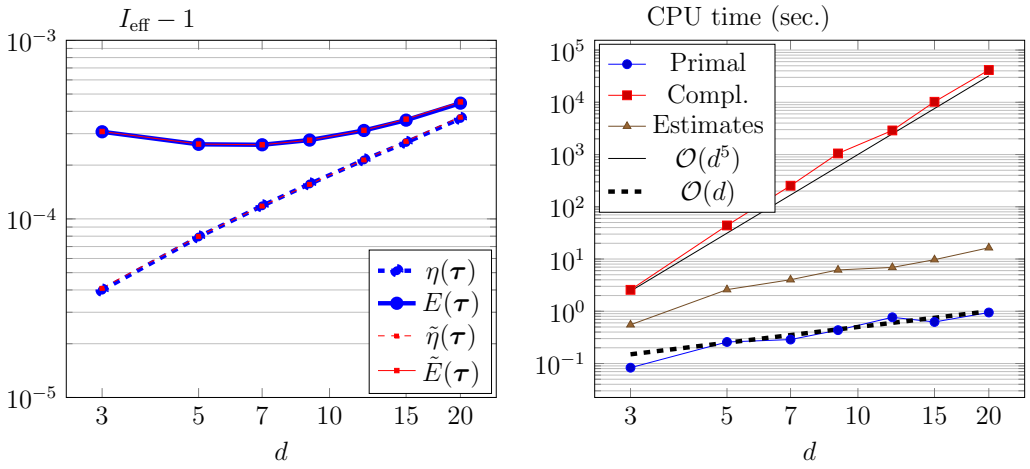


of the reconstructed flux $\boldsymbol{\tau}_h$ is lower. The tolerance of the primal problem is kept to be $\delta_p = 10^{-3}$. Figure 2 presents differences of indices of effectivity from 1 (left) and the corresponding CPU times (right).

The inefficiency of the error estimators grows slowly with the dimension, but even for the 20-dimensional problem the estimators remain accurate up to 3 decimal digits, which is essentially perfect for practical purposes. Indeed, the error estimator $\widetilde{E}(\boldsymbol{\tau}_h)$ guarantees that the relative error $|||u - u_h|||/|||u|||$ is below 0.00782, while the exact error is approximately 0.00781 irrespectively of the dimension d. However, the CPU time of solving the complementary problem grows significantly. The observed order is about $\mathcal{O}(d^5)$. This is partially because the complementary solution is a vector function of size d and the TT ranks and the number of GMRES iterations for the reduced problems (33), both seem to grow linearly in d. Nevertheless, this polynomial growth of the complexity is still much better than the exponential growth, but much worse than the linear cost of solving just the primal problem.

Finally, we test the influence of the reaction coefficient κ^2 on the accuracy of the error estimator. We fix $n = 128$ and $d = 3$ and consider various positive values of κ. Therefore the shift κ_0 is not needed and we formally set $\kappa_0 = 0$. Here, we use the TT approximation tolerance for the complementary problem $\delta_c = 10^{-7}$ in order to solve the problem for the smallest $\kappa^2 = 10^{-3}$. Note that the reaction-diffusion problem (1) is singularly perturbed for large κ and its solution often has steep boundary layers of width κ. If

Figure 2: Differences of indices of effectivity from 1 (left) and CPU times (right) vs. dimension d for the Poisson problem.



these layers are not resolved by the mesh, the finite element solution exhibits spurious oscillations. The proposed estimator captures this behaviour well and estimates the error accurately even in the singularly perturbed case. Figure 3 (left) shows that the index of effectivity is almost 1 for small values of κ^2 and grows for larger values. However, even for $\kappa^2 = 10^6$ all indices of effectivity are below 1.4.

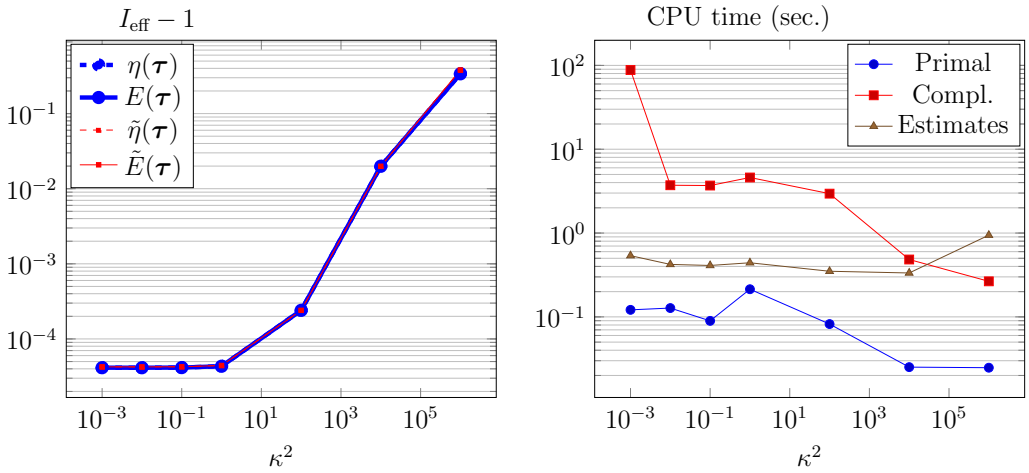
In Figure 3 (right) we see that the CPU time grows significantly for small κ^2, since the complementary problem becomes severely ill-conditioned. This motivates future research on more efficient preconditioning techniques. However, even with a simple block Jacobi preconditioner, the CPU time remains stable for a range of orders of κ^2, and even decreases for large κ^2, where the operator approaches a perturbed identity.

5.2 Schrödinger equation with Henon–Heiles potential

In the second example we consider a variable reaction coefficient. We carry out one step of the Shift-and-Invert iteration for searching the ground state in the molecular Schrödinger equation with the model Henon–Heiles potential [32]. The elliptic PDE (1) is posed on a hypercube $\Omega = (-5, 5)^d$. The reaction coefficient is the following polynomial of degree three:

$$\kappa^2(x) = V_{\text{h}}(x) + V_{\text{u}}(x) + 1,$$

Figure 3: Differences of indices of effectivity from 1 (left) and CPU times (right) vs. the coefficient κ^2 for the 3D reaction-diffusion problem.



where the harmonic potential $V_{\text{h}}(x)$ and its non-harmonic perturbation $V_{\text{u}}(x)$ are given by

$$V_{\text{h}}(x) = \sum_{k=1}^{d} x_k^2 \quad \text{and} \quad V_{\text{u}}(x) = 0.223606 \sum_{k=1}^{d-1} \left(x_k^2 x_{k+1} - \frac{1}{3} x_{k+1}^3 \right).$$

This corresponds to the Henon–Heiles potential $V_{\text{h}}(x) + V_{\text{u}}(x)$ plus a unit shift due to the Shift-and-Invert power method.

The exact solution is chosen to be the Gaussian function

$$u(x) = \prod_{k=1}^{d} \exp\left(-\frac{x_k^2}{2} \right),$$

which represents the typical form of the lowest eigenfunction of the Schrödinger operator. The right hand side $f(x) = (d + 1 + V_{\text{u}}(x))\, u(x)$ is computed accordingly. Despite the cubic nonlinearities, the reaction coefficient is nonnegative (and actually greater than 1) in Ω. From the modelling point of view, this domain is also sufficiently large such that the solution is negligibly small (below 10^{-5}) on the boundary, which justifies the use of the homogeneous Dirichlet boundary conditions.

The high-dimensional Henon–Heiles Schrödinger model was approached with Tensor Train eigenvalue solvers and a spectral discretization [32, 27, 15],

but no attempts were made to estimate a posteriori errors in a systematic way (instead, a fixed grid was chosen which proved to be overwhelmingly fine in low-dimensional tests). Here we compute the guaranteed error estimator using the complementary solution and illustrate the performance of the described approach in high dimensions.

Since the reaction coefficient is strictly positive, we set $\kappa_0 = 0$. Consequently, $E(\boldsymbol{\tau}) = \eta(\boldsymbol{\tau})$ and $\tilde{E}(\boldsymbol{\tau}) = \tilde{\eta}(\boldsymbol{\tau})$. We continue to use $m = 4$ quadrature points in each direction of each element. This gives a relative quadrature error of the order of 10^{-8}, which is many orders of magnitude smaller than the discretization error. The approximation thresholds are fixed to $\delta_p = \delta_c = 10^{-3}$ and grids with either $n = 16$ or $n = 64$ elements in each dimension are used.

Figure 4: Differences of effectivity indices from 1 (left) and CPU times (right) vs. dimension d for the Schrödinger equation. Solid lines: $n = 64$, dashed lines: $n = 16$.

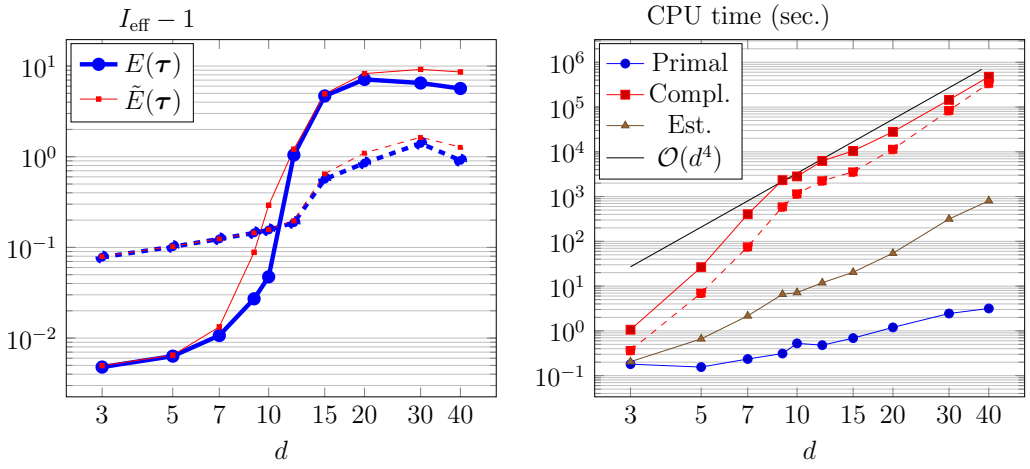


Figure 4 shows differences of indices of effectivity from 1 (left) and the corresponding CPU times (right). The efficiency of error estimators deteriorates with the dimension. However, indices of effectivity tend to stabilize below 10, which still provides a reasonable error bound. This deterioration is mainly due to the condition number of the complementary system. For the smaller n the matrix is well conditioned and the effectivity indices are much closer to 1. Interestingly, the cost of solving the complementary problem grows with a slower rate $\mathcal{O}(d^4)$ than that for the Poisson problem. The

reason for this is roughly the same number of GMRES iterations for the reduced system (33) for all dimensions, since the potential term improves the conditioning of this system.

6 Conclusions

This paper presents fully computable a posteriori error bounds for low rank tensor approximate solutions of high dimensional reaction-diffusion problem. These bounds are guaranteed in the sense that they are proved to be above the energy norm of the total error including the discretization error, iteration error in the linear solver, and tensor truncation errors. This provides a tool for a reliable error control in high dimensional computations.

The guaranteed error bound is based on an approximate solution of the complementary problem. Numerical experiments show that the CPU time to solve it grows considerably faster with the dimension than the CPU time needed for the original reaction-diffusion problem. However, the growth is polynomial, which is still feasible in comparison with the exponential complexity of classical approaches.

Interestingly, the error bound is guaranteed for *any* conforming approximation of the primal and complementary solution. Consequently, these solutions can be polluted by arbitrary errors and the estimator still provides a guaranteed error bound.

For efficient computational algorithms estimates or at least indicators of sizes of individual components of the total error are desirable, see e.g. [18, 39]. Separate estimates of the discretization, algebraic, and tensor truncation errors would enable us to adaptively identify the major source of the error and automatically decide whether to refine the mesh, do more iterations of the algebraic solver, or increase ranks of tensor approximations [17]. These separate estimates may be a subject of further research.

Another line of further research should focus on a more efficient way how to compute the complementary solution. For low dimensional problems, efficient local flux reconstructions are know. However, an efficient generalization of these ideas to high dimensions is still open.

References

[1] Mark Ainsworth and Ivo Babuška. Reliable and robust a posteriori error estimating for singularly perturbed reaction-diffusion problems. *SIAM J. Numer. Anal.*, 36(2):331–353 (electronic), 1999.

[2] Mark Ainsworth and J. Tinsley Oden. *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000.

[3] Mark Ainsworth and Tomáš Vejchodský. Fully computable robust a posteriori error bounds for singularly perturbed reaction–diffusion problems. *Numer. Math.*, 119(2):219–243, 2011.

[4] Mark Ainsworth and Tomáš Vejchodský. Robust error bounds for finite element approximation of reaction-diffusion problems with non-constant reaction coefficient in arbitrary space dimension. *Comput. Methods Appl. Mech. Engrg.*, 281:184–199, 2014.

[5] Mark Ainsworth and Tomáš Vejchodský. A simple approach to reliable and robust a posteriori error estimation for singularly perturbed problems. *Preprint arXiv:1812.07972*, page 33 p., 2018.

[6] M. Bachmayr, A. Cohen, and W. Dahmen. Parametric PDEs: sparse or low-rank approximations? *IMA J. Num. An.*, 2017.

[7] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, 2013.

[8] M. Bebendorf. A note on the Poincaré inequality for convex domains. *Z. Anal. Anwendungen*, 22(4):751–756, 2003.

[9] R. E. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[10] P. Benner, S. Dolgov, A. Onwunta, and M. Stoll. Low-rank solvers for unsteady Stokes-Brinkman optimal control problem with random data. *Computer Methods in Applied Mechanics and Engineering*, 304:26–54, 2016.

[11] Zhiqiang Cai and Shun Zhang. Flux recovery and a posteriori error estimators: conforming elements for scalar elliptic equations. *SIAM J. Numer. Anal.*, 48(2):578–602, 2010.

[12] I. Cheddadi, R. Fučík, M. I. Prieto, and M. Vohralík. Guaranteed and robust a posteriori error estimates for singularly perturbed reaction–diffusion problems. *M2AN Math. Model. Numer. Anal.*, 43:867–888, 2009.

[13] S. Dolgov and B. Khoromskij. Two-level QTT-Tucker format for optimized tensor calculus. *SIAM J. on Matrix An. Appl.*, 34(2):593–623, 2013.

[14] S. V. Dolgov. TT-GMRES: solution to a linear system in the structured tensor format. *Russ. J. Numer. Anal. Math. Model.*, 28(2):149–172, 2013.

[15] S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, and D. V. Savostyanov. Computation of extreme eigenvalues in higher dimensions using block tensor train format. *Computer Phys. Comm.*, 185(4):1207–1216, 2014.

[16] S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM J. Sci. Comput.*, 36(5):A2248–A2271, 2014.

[17] M. Eigel, M. Pfeffer, and R. Schneider. Adaptive stochastic Galerkin FEM with hierarchical tensor representations. *Numer. Math.*, pages 1–39, 2016.

[18] Alexandre Ern and Martin Vohralík. Adaptive inexact Newton methods with a posteriori stopping criteria for nonlinear diffusion PDEs. *SIAM J. Sci. Comput.*, 35(4):a1761–a1791, 2013.

[19] L. Grasedyck. Existence and computation of low Kronecker-rank approximations for large systems in tensor product structure. *Computing*, 72:247–265, 2004.

[20] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitt.*, 36(1):53–78, 2013.

[21] M. Griebel and H. Harbrecht. Approximation of bi-variate functions: singular value decomposition versus sparse grids. *IMA Journal of Numerical Analysis*, 2013.

[22] W. Hackbusch. *Tensor Spaces And Numerical Tensor Calculus.* Springer–Verlag, Berlin, 2012.

[23] Antti Hannukainen, Rolf Stenberg, and Martin Vohralík. A unified framework for a posteriori error estimation for the Stokes problem. *Numer. Math.*, 122(4):725–769, 2012.

[24] S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM J. Sci. Comput.*, 34(2):A683–A713, 2012.

[25] V. A. Kazeev and B. N. Khoromskij. Low-rank explicit QTT representation of the Laplace operator and its inverse. *SIAM J. Matrix Anal. Appl.*, 33(3):742–758, 2012.

[26] B. N. Khoromskij. Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications. *ESAIM: Proc.*, 48:1–28, 2015.

[27] B. N. Khoromskij and I. V. Oseledets. DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator. Preprint 69, MPI MIS, Leipzig, 2010.

[28] B.N. Khoromskij and Ch. Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM J. Sci. Comput.*, 33(1):1–25, 2011.

[29] D. Kressner, M. Steinlechner, and A. Uschmajew. Low-rank tensor methods with subspace correction for symmetric eigenvalue problems. *SIAM J Sci. Comput.*, 36(5):A2346–A2368, 2014.

[30] D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM J. Matrix Anal. Appl.*, 32(4):273–290, 2011.

[31] R. Luce and B. I. Wohlmuth. A local a posteriori error estimator based on equilibrated fluxes. *SIAM J. Numer. Anal.*, 42(4):1394–1414, 2004.

[32] H.-D. Meyer, F. Gatti, and G. A. Worth. *Multidimensional Quantum Dynamics: MCTDH Theory and Applications*. Wiley-VCH, Weinheim, 2009.

[33] Solomon G. Mikhlin. *Constants in some inequalities of analysis*. John Wiley & Sons., 1986.

[34] P. Neittaanmäki and S. Repin. *Reliable methods for computer simulation*, volume 33 of *Studies in Mathematics and its Applications*. Elsevier Science B.V., Amsterdam, 2004. Error control and a posteriori estimates.

[35] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.

[36] I. V. Oseledets. Constructive representation of functions in low-rank tensor formats. *Constr. Approx.*, 37(1):1–18, 2013.

[37] I. V. Oseledets and S. V. Dolgov. Solution of linear systems and matrix inversion in the TT-format. *SIAM J. Sci. Comput.*, 34(5):A2718–A2739, 2012.

[38] I. V. Oseledets and E. E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.*, 432(1):70–88, 2010.

[39] J. Papež, Z. Strakoš, and M. Vohralík. Estimating and localizing the algebraic and total numerical errors using flux reconstructions. *Numer. Math.*, 138(3):681–721, 2018.

[40] N. Parés and P. Díez. A new equilibrated residual method improving accuracy and efficiency of flux-free error estimates. *Comput. Methods Appl. Mech. Engrg.*, 313:785–816, 2017.

[41] N. Parés, H. Santos, and P. Díez. Guaranteed energy error bounds for the Poisson equation using a flux-free approach: solving the local problems in subdomains. *Internat. J. Numer. Methods Engrg.*, 79(10):1203–1244, 2009.

[42] L. E. Payne and H. F. Weinberger. An optimal Poincaré inequality for convex domains. *Arch. Rational Mech. Anal.*, 5:286–292 (1960), 1960.

[43] W. Prager and J. L. Synge. Approximations in elasticity based on the concept of function space. *Quart. Appl. Math.*, 5:241–269, 1947.

[44] Sergey Repin. *A posteriori estimates for partial differential equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. de Gruyter, Berlin, 2008.

[45] T. Rohwedder and A. Uschmajew. On local convergence of alternating schemes for optimization of convex problems in the tensor train format. *SIAM J. Num. Anal.*, 51(2):1134–1162, 2013.

[46] D. V. Savostyanov. Quasioptimality of maximum–volume cross interpolation of tensors. *Linear Algebra Appl.*, 458:217–244, 2014.

[47] R. Schneider and A. Uschmajew. Approximation rates for the hierarchical tensor format in periodic sobolev spaces. *Journal of Complexity*, 2013.

[48] Ivana Šebestová and Tomáš Vejchodský. Two-sided bounds for eigenvalues of differential operators with applications to Friedrichs, Poincaré, trace, and similar constants. *SIAM J. Numer. Anal.*, 52(1):308–329, 2014.

[49] I. Smears and M. Vohralík. Simple and robust equilibrated flux a posteriori estimates for singularly perturbed reaction-diffusion problems. *Preprint*, hal-01956180:22 p., 2018.

[50] E. E. Tyrtyshnikov. Tensor approximations of matrices generated by asymptotically smooth functions. *Sbornik: Mathematics*, 194(6):941–954, 2003.

[51] Tomáš Vejchodský. Complementarity based a posteriori error estimates and their properties. *Math. Comput. Simulation*, 82(10):2033–2046, 2012.