Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Tensor Structured Iterative Solution of Elliptic Problems with Jumping Coefficients

by

Sergey Dolgov, Boris N. Khoromskij, Ivan V. Oseledets, and Eugene E. Tyrtyshnikov

Preprint no.: 55 2010

Tensor Structured Iterative Solution of Elliptic Problems with Jumping Coefficients

Sergey Dolgov *
Moscow Institute of Physics and Technology, Russia; dc1988@mail.ru

Boris N. Khoromskij
Max-Planck-Institut für Mathematik in den Naturwissenschaften,
Inselstr. 22-26, D-04103 Leipzig, Germany; bokh@mis.mpg.de

Ivan Oseledets, Eugene E. Tyrtyshnikov *
Institute of Numerical Mathematics, Russian Academy of Sciences,
Gubkina 8, 119991 Moscow, Russia;
{ivan.oseledets@gmail.com, tee@inm.ras.ru}

September 28, 2010

Abstract

We study separability properties of solutions of elliptic equations with piecewise constant coefficients in $\mathbb{R}^d$, $d \geq 2$. Besides that, we develop efficient tensor-structured preconditioner for the diffusion equation with variable coefficients. It is based only on rank structured decomposition of the tensor of reciprocal coefficient and on the decomposition of the inverse of the Laplacian operator. It can be applied to full vector with linear-logarithmic complexity in the number of unknowns N. It also allows low-rank tensor representation, which has linear complexity in dimension d, hence, it gets rid of the "curse of dimensionality" and can be used for large values of d. Extensive numerical tests are presented.

AMS Subject Classification: 65F30, 65F50, 65N35, 65F10
Key words: structured matrices, elliptic operators, Poisson equation, matrix approximations, low-rank matrices, preconditioners, multi-dimensional matrices, tensors, finite elements, numerical methods

1 Introduction

In recent years, the numerical methods based on tensor product formats were applied for solving different classes of multi-dimensional problems related to the elliptic PDEs [11]. An

*Supported by the RFBR grants 08-01-00115, 09-01-12058, RFBR/DFG grant 09-01-91332, the Government Contracts П940, П1112 and Priority Research Grant of the Department of Mathematical Sciences of the Russian Academy of Sciences.

important ingredient for the efficient iterative solver is the construction of low-rank spectrally close preconditioners for the arising discrete elliptic systems [9, 7, 12].

In this paper, we study separability properties of solutions of elliptic equations with piecewise constant coefficients in $\mathbb{R}^d$, $d \geq 2$. Besides that, we develop efficient tensor-structured preconditioner for the diffusion equation with variable coefficients.

First consider a model elliptic boundary value problem in two dimensions,

$$\begin{aligned} -\nabla(a\nabla u) &= f \quad \text{in} \quad \Omega = [0,1]^2, \\ u|_{\partial\Omega} &= 0, \end{aligned} \tag{1.1}$$

with an assumption that f is represented by a piecewise smooth tensor decomposition

$$f(x,y) = \sum_{k=1}^{r_f} f_k^{(1)}(x)\ f_k^{(2)}(y), \tag{1.2}$$

and the diffusion coefficient $a(x,y)$ is a piecewise constant function on cells of a tensor grid in Ω. In the case of an $M \times M$ tensor tiling, reciprocals $1/a$ on these cells comprise a matrix of form

$$B = \begin{bmatrix} 1/a_{11} & \cdots & 1/a_{1M} \\ \vdots & \ddots & \vdots \\ 1/a_{M1} & \cdots & 1/a_{MM} \end{bmatrix} \tag{1.3}$$

with the notation

$$r_{1/a} = \mathrm{rank} B$$

(see Figure 1.1). Clearly, the function $1/a$ has the same separable form,

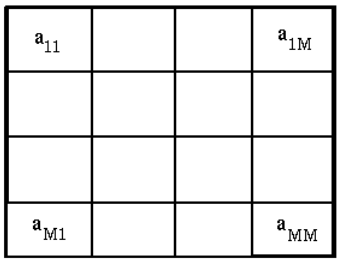

Figure 1.1: Cell structure of jumping coefficient

$$1/a(x,y) = \sum_{l=1}^{r_{1/a}} b_l^{(1)}(x) \cdot b_l^{(2)}(y) = \sum_{l=1}^{r_{1/a}} \frac{1}{a_l^{(1)}(x)} \cdot \frac{1}{a_l^{(2)}(y)}, \tag{1.4}$$

which can be shown by a constant spline interpolation. Given $\varepsilon > 0$, we approximate u by a separable decomposition

$$u_{r_u} = \sum_{k=1}^{r_u} u_k^{(1)}(x)\, u_k^{(2)}(y), \tag{1.5}$$

so that $||u - u_{r_u}|| \leq \varepsilon$.

In this paper we first investigate how r_u depends on ε, $r_{1/a}$ and r_f. Straightforward analysis in the continuous case gives the following rank estimate:

$$r_u = O(Mr_v),$$

where r_v is a maximal rank of the solution in each domain, generated by $M \times M$ tiling. Note that r_v does not depend on a, as in each domain solution satisfies Poisson equation with constant coefficient: $-a\Delta u = f$.

In 3D or higher dimensional case we formulate the problem in a similar way. Consider

$$\begin{aligned} -\nabla(a\nabla u) &= f \quad \text{in} \quad \Omega = [0,1]^d, \\ u|_{\partial\Omega} &= 0, \end{aligned} \tag{1.6}$$

and assume a separability property for the right-hand side,

$$f(\mathbf{x}) = \sum_{k=1}^{r_f} f_k^{(1)}(x_1) \cdots f_k^{(d)}(x_d), \tag{1.7}$$

and the reciprocal diffusion coefficient,

$$1/a(\mathbf{x}) = \sum_{l=1}^{r_{1/a}} b_l^{(1)}(x_1) \cdots b_l^{(d)}(x_d) = \sum_{l=1}^{r_{1/a}} \frac{1}{a_l^{(1)}(x_1)} \cdots \frac{1}{a_l^{(d)}(x_d)}. \tag{1.8}$$

Now for a given $\varepsilon > 0$, we approximate u by a separable decomposition

$$u_{r_u} = \sum_{k=1}^{r_u} u_k^{(1)}(x_1) \cdots u_k^{(d)}(x_d), \tag{1.9}$$

so that $||u - u_{r_u}|| \leq \varepsilon$.

The main result is that we can obtain an approximate discrete solution, with the rank bound

$$r_{1/a} r_v$$

uniformly in d. In the general case, the accuracy might be not sufficient to take this approximate solution as a final solution of the problem, but the corresponding solver is quite fast, and we can use it as a black box preconditioner on each iteration of, e.g., GMRES solver.

The rest of the paper is organized as follows. In the section 2 we introduce the equivalent formulation of the initial problem, and the related gradient equations, which are the start point of our work. Then we present numerical examples of separability properties of the solution of finite element method in 2D. In section 3 we introduce the discretization scheme, quasi-optimal preconditioner and estimate the condition number and eigenvalues of the preconditioned problem in general case, and in some particular cases, in which we are able to prove the better bounds. In section 4 we test the preconditioning properties of the proposed algorithm and also test the compression properties of the new QTT tensor format (see [15, 10, 12]) for coefficients, matrices and solutions.

2 Gradient equations

2.1 Approximate solution of the operator equation

An approximate discrete solution, with rank only $r_{1/a}r_v$, but not $M^\alpha r_v$ is based on the following equivalent formulation of the initial problem:

$$\begin{aligned} &-\nabla(a\nabla u) = f = -\Delta v \\ &u|_{\partial\Omega} = v|_{\partial\Omega} = 0. \end{aligned} \tag{2.1}$$

The main reason of this formulation is that it has similar objects in right and left parts:

$$-\frac{\partial}{\partial x}a\frac{\partial}{\partial x}u - \frac{\partial}{\partial y}a\frac{\partial}{\partial y}u = -\frac{\partial^2}{\partial x^2}v - \frac{\partial^2}{\partial y^2}v \tag{2.2}$$

The main idea is to cancel some portion of information in these equations, which provides sufficiently good accuracy or preconditioning properties, but allows efficient algorithmic implementation.

Consider the equation in the following form:

$$-\nabla^T(a\nabla u) = -\nabla^T\nabla v$$

This holds under the following condition:

$$-a\nabla u = -\nabla v + \vec{\psi},$$

where $\nabla^T\vec{\psi} = 0$. This is similar to the orthogonal Helmholtz decomposition [2]: any vector field $\vec{V}$ can be decomposed to the following orthogonal parts:

$$\vec{V} = \vec{V}_{div} + \vec{V}_{curl},$$

where $\texttt{div}\ \vec{V}_{div} = \nabla^T\vec{V}_{div} = 0$, and $\texttt{curl}\ \vec{V}_{curl} = \nabla \wedge \vec{V}_{curl} = 0$. In our case $\vec{\psi} = \vec{V}_{div}$. Then we just omit $\vec{\psi}$, that is, consider the problem only on curl subspace. Then,

$$-\nabla u \approx -\nabla\tilde{u} = -\frac{1}{a}\nabla v. \tag{2.3}$$

In terms of continuous functions we can integrate this gradient equation using the Newton-Leibniz formula:

$$\tilde{u}(x,y) - \tilde{u}(0,y) + \tilde{u}(0,y) - \tilde{u}(0,0) = \int\limits_0^y \frac{\partial\tilde{u}(0,\eta)}{\partial y}d\eta + \int\limits_0^x \frac{\partial\tilde{u}(\xi,y)}{\partial x}d\xi.$$

Due to the Dirichlet boundary condition $u|_{\partial\Omega} = 0$, the first term is equal to 0. From the equation (2.3) we obtain:

$$\tilde{u}(x,y) = \int\limits_0^x \frac{1}{a(\xi,y)}\frac{\partial v(\xi,y)}{\partial x}d\xi$$

As $v = (-\Delta)^{-1} f$, it can be approximated by the canonical decomposition:

$$v \approx v_{r_v} = \sum_{k=1}^{C|\log\varepsilon|} \sum_{p=1}^{r_f} (D_k^{(1)} f_p^{(1)}) \otimes (D_k^{(2)} f_p^{(2)}) = \sum_{k=1}^{r_v} v_k^{(1)} \otimes v_k^{(2)},$$

where $D_k^{(q)}$ are canonical factors of Δ^{-1}, providing the ε-accuracy of separation approximation. Using also the decomposition for $1/a$ (1.8), we can write:

$$\tilde{u} \approx \tilde{u}_{r_u}(x,y) = \sum_{k=1}^{C|\log\varepsilon| r_f} \sum_{l=1}^{r_{1/a}} b_l^{(2)}(y) v_k^{(2)}(y) \int_0^x b_l^{(1)}(\xi) \frac{\partial v_k^{(1)}(\xi)}{\partial x} d\xi,$$

from which we can easily estimate the rank $r_u \leq C|\log\varepsilon| r_f r_{1/a}$.

To obtain a symmetric resolving operator we should use some other approach, than the Newton-Leibniz integration of the gradient equation. Multiply both parts of the gradient equation (2.3) by $-\Delta^{-1}\nabla^T$:

$$\Delta^{-1}\nabla^T \nabla \tilde{u} = \Delta^{-1}\Delta\tilde{u} = \tilde{u} = -\Delta^{-1}\nabla^T \frac{1}{a}\nabla v.$$

From the equation $f = -\Delta v$ we get $v = -\Delta^{-1} f$. Then

$$u \approx \tilde{u} = \Delta^{-1}\left(\nabla^T \frac{1}{a}\nabla\right)\Delta^{-1} f,$$

i.e., we have the following operator, which we consider in discrete case as a candidate for preconditioner:

$$\left(\nabla^T a \nabla\right)^{-1} \approx P = \Delta^{-1}\left(\nabla^T \frac{1}{a}\nabla\right)\Delta^{-1}. \tag{2.4}$$

The same result can be obtained as a solution of the minimization problem

$$J = \left\| \nabla u - \frac{1}{a}\nabla v \right\|^2 \to \texttt{min},$$

i.e. the solution of gradient equation (2.3) in the least-squares formulation. In the case of a low-rank reciprocal coefficient, we have the following approximation for this operator:

$$P = \Delta^{-1}\left(\nabla^T \frac{1}{a}\nabla\right)\Delta^{-1} \approx \sum_{k=1}^{r_P} A_k^{(1)} \otimes A_k^{(2)},$$

where

$$r_P \leq \mathrm{rank}(\Delta^{-1})^2 \cdot d\ r_{1/a} \leq O(d r_{1/a} |\log(\varepsilon)|^2),$$

as approximation of Δ^{-1} can be obtained from the ε-approximation of corresponding potential with $O(|\log(\varepsilon)|)$ rank. If we apply this operator to the right-hand side f with the rank r_f, we obtain the rank of u

$$r_u \leq O(d\ r_{1/a} |\log(\varepsilon)|^2\ r_f). \tag{2.5}$$

We see, that its separation rank is bounded by $|\log(\varepsilon)|^2$, but the separation rank in the case of functions, obtained by a Newton-Leibniz integration grows linearly with $|\log(\varepsilon)|$.

In the next subsection we present some numerical examples (see Tables 2.1-2.3), that shows, that the last rank bound holds in many practical cases.

2.2 Numerical separability properties in 2D

We can solve the equation (1.1) using Galerkin method: choose appropriate basis functions $\varphi_1(x), \ldots, \varphi_n(x)$ and seek the solution as a linear combination

$$u_h(x,y) = \sum_{i_1,i_2=1}^{n} u(i_1,i_2)\varphi_{i_1}(x)\varphi_{i_2}(y),$$

with unknown coefficients $u(i_1,i_2)$ to be obtained from the linear system

$$\begin{aligned} \sum_{i_1,i_2=1}^{n} u(i_1,i_2)\left(a\nabla\varphi_{i_1}(x)\varphi_{i_2}(y), \nabla\varphi_{j_1}(x)\varphi_{j_2}(y)\right)_{L_2(\Omega)} &= \left(f, \varphi_{j_1}(x)\varphi_{j_2}(y)\right)_{L_2(\Omega)}, \\ j_1, j_2 &= 1, \ldots, n. \end{aligned} \tag{2.6}$$

Remark 2.1 *Although we denote basis functions by φ both for x and y directions (for the ease of presentation), in fact, number of grid points and grid cell size can be different for different directions, hence, in such case there will be different sets of basis functions $\varphi_{i_1}(x)$ and $\psi_{i_2}(y)$.*

We can write (2.6) in the following form:

$$AU = F,$$

where

$$A = \left[\left(a\nabla\varphi_{i_1}(x)\varphi_{i_2}(y), \nabla\varphi_{j_1}(x)\varphi_{j_2}(y)\right)_{L_2(\Omega)}\right],$$

$$F = \left[\left(f, \varphi_{j_1}(x)\varphi_{j_2}(y)\right)_{L_2(\Omega)}\right] = \sum_{k=1}^{r_f}\left[\left(f_k^{(1)}, \varphi_{j_1}(x)\right)_{L_2(0,1)}\right] \otimes \left[\left(f_k^{(2)}, \varphi_{j_2}(y)\right)_{L_2(0,1)}\right]$$

Let us gather coefficients $u(i_1,i_2)$ into a matrix $U = [u(i_1,i_2)] \in R^{nxn}$ and decompose it using the SVD:

$$u(i_1,i_2) = \sum_{k=1}^{n} \sigma_k U_{i_1,k}^{(1)} U_{i_2,k}^{(2)},$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ are the singular values, and $U_{i_1,k}^{(1)}$, $U_{i_2,k}^{(2)}$ are the k-th singular vectors. In order to obtain a reduced representation for the solution, we can truncate this sum keeping only summands with a certain number of senior singular values and neglecting summands with smaller singular values. In this way approximation to U of a lower rank $U_{r_u} = [u_{r_u}(i_1,i_2)]$ is obtained:

$$u_{r_u}(i_1,i_2) = \sum_{k=1}^{r_u} \sigma_k \; U_{i_1,k}^{(1)} \; U_{i_2,k}^{(2)}.$$

Given an accuracy parameter ε, we can choose r_u so that the estimate $||U - U_{r_u}|| \leq \varepsilon$ is guaranteed to hold with a minimal possible r_u. Then, it is easy to derive that

$$\hat{u}_{r_u}(x,y) = \sum_{i_1,i_2=1}^{n} u_{r_u}(i_1,i_2)\varphi_{i_1}(x)\varphi_{i_2}(y) = \sum_{k=1}^{r_u} \sigma_k \left(\sum_{i_1=1}^{n} U_{i_1,k}^{(1)}\varphi_{i_1}(x)\right)\left(\sum_{i_2=1}^{n} U_{i_2,k}^{(2)}\varphi_{i_2}(y)\right)$$

approximates $u_h(x, y)$ with accuracy $O(\varepsilon)$.

In numerical examples below, we are interested to find relations between r_u and ε, $r_{1/a}$, r_f, and their dependence on n. In the following we assume that a has constant values on $M \times M$ cells. We take piecewise linear hat elements as basis functions $\varphi_i(x)$.

1. Dependence on ε and n (table 2.1). We can deduce that practical dependence

Table 2.1: r_u versus ε and n; $r_{1/a} = 1$; $M = 8$.

	$\log_{10}(1/\varepsilon)$						
n	4	5	6	7	8	9	10
16	2	4	5	5	6	7	7
32	3	5	5	7	7	9	9
64	2	4	4	6	6	9	9
128	2	4	5	6	8	10	11
256	2	4	5	6	8	10	12
512	3	4	5	7	8	11	13
1024	3	4	6	8	9	12	14

is of the form

$$r_u(\varepsilon) = C \cdot \log(1/\varepsilon). \tag{2.7}$$

If we make a linear fit of $r_u(|\log(\varepsilon)|)$ for $n = 1024$, using the least squares method, the dependence is $r_u = 1.86 \cdot \log(1/\varepsilon) - 5$. Also we can see that if the approximation tolerance ε is greater than the discretization error $O(1/n^2)$, then r_u does not depend on n (e.g., see the column with $\varepsilon = 10^{-5}$).

2. Dependence on $r_{1/a}$ (table 2.2). Now the least squares linear fitting gives a

Table 2.2: r_u versus ε and $r_{1/a}$; $M = 8$; $n = 256$.

	$\log_{10}(1/\varepsilon)$						
$r_{1/a}$	4	5	6	7	8	9	10
1	3	4	6	8	9	12	14
2	5	8	14	21	28	34	41
3	5	8	14	20	30	37	47
4	7	13	22	35	45	56	67
5	8	17	31	46	60	73	85
6	8	17	30	46	65	80	93
7	11	19	34	54	72	91	107
8	11	23	41	60	81	96	112

dependence $r_u = 13.95 \cdot r_{1/a} + 7.96$
(for $\varepsilon = 10^{-10}$). Thus,

$$r_u(r_{1/a}) = C \cdot r_{1/a}. \tag{2.8}$$

3. Dependence on M (table 2.3). In this example we have used randomly generated

Table 2.3: r_u versus ε and M; $r_{1/a} = 1$; $n = 256$.

	$\log_{10}(1/\varepsilon)$			
M	4	5	8	11
2	2	3	7	12
3	2	4	9	16
4	3	4	11	17
8	3	5	12	18
12	4	5	12	19
16	3	5	11	18
32	3	5	11	18

values in the closed interval $[1, 7]$ for a with rank 1. We see that for sufficiently large M ($M > 4$), the rank r_u does not depend on M. As a matter of fact, if the rank $r_{1/a}$ is fixed, then r_u becomes a constant, no matter whatever big jumps and high oscillations in a might occur (see Fig. 2.1). In this examples we have taken a separable function f with $r_f = 1$, but

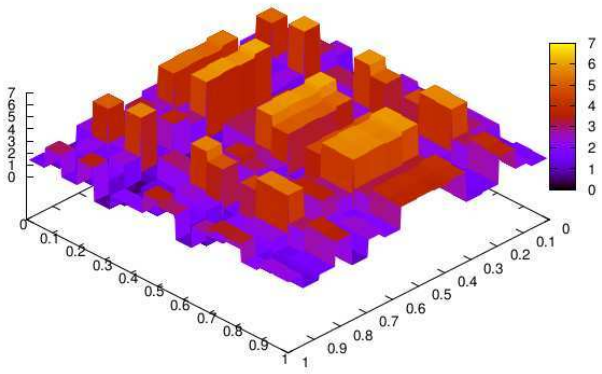

Figure 2.1: Randomly filled coefficient a with rank 1 and 16x16 domain splitting

the same results are observed as well with $r_f > 1$. Consequently, from equations (2.7)-(2.8) we observe an estimate of the form

$$r \leq C \cdot r_{1/a} \cdot \log(1/\varepsilon).$$

3 On the construction of quasi-optimal preconditioner

3.1 Discretization scheme

First, we present more detailed description of the discretization scheme of the diffusion equation. We prove that the spectrum preconditioned matrix, which arises from the

discretization of the diffusion operator with the constants, does not depend on the grid size, so, the number of iterations does not depend on a grid size. Moreover, we prove in some simple cases, that eigenvalues of the preconditioned matrix form a finite amount of separate clusters. As in previous sections, two and higher dimensional problems differ only in some technical details, but the whole concept is the same. So, we show full proofs only for the 2D case, and make remarks, how to generalize them to the higher dimensional case.

For brevity, denote the matrix of discretized operator $\nabla^T a \nabla$ as $\Gamma(a)$:

$$\Gamma(a) = \nabla_h^T a_h \nabla_h, \tag{3.1}$$

where ∇_h and a_h are matrices of discretized operator ∇ and the coefficient a (see (3.3)).

Consider the finite-difference discretization scheme on the uniform grid:

$$\frac{\partial_h u_h}{\partial_h x} = \frac{u_{i+1,j} - u_{i,j}}{h},$$

where $u_{i,j}$ is the value of the function u_h in the grid point (i, j) with coordinates

$$(x_i, x_j) = (ih, jh), \quad h = 1/(n+1), \quad i, j = 1, ..., n.$$

We also require, that interface points (points of jumps in coefficient) belong to the set of nodes of the grid. Since a is not defined on interfaces, we choose shifted grid for the discretized coefficient:

$$a(i,j) = a_{i-1/2,j-1/2} = a(x_i - h/2, y_j - h/2) = a((i-1/2)h, (j-1/2)h),$$

i.e. we consider the coefficient a in the grid point $(i - 1/2, j - 1/2)$, which is the center of cell, corresponding to the following values of u_h: $u_{i-1,j-1}$, $u_{i-1,j}$, $u_{i,j-1}$, $u_{i,j}$ (see Fig. 3.1). Then

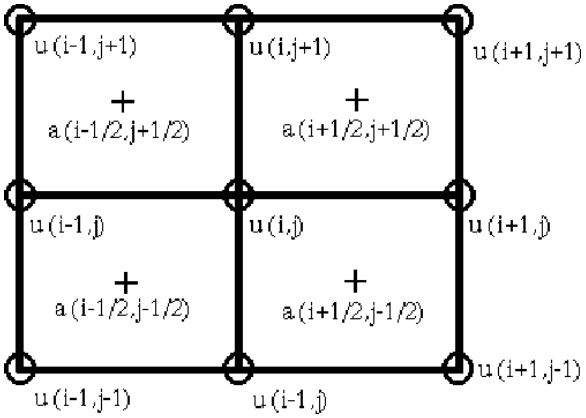

Figure 3.1: Discretization grids for a and u

$$\frac{\partial_h}{\partial_h x} a_h \frac{\partial_h u_h}{\partial_h x} = \frac{a_{i+1/2,j} \dfrac{u_{i+1,j} - u_{i,j}}{h} - a_{i-1/2,j} \dfrac{u_{i,j} - u_{i-1,j}}{h}}{h},$$

where as $a_{i-1/2,j}$ we take the averaged value in the direction j:

$$a_{i-1/2,j} = \frac{a_{i-1/2,j-1/2} + a_{i-1/2,j+1/2}}{2}. \tag{3.2}$$

In the same way we formulate discrete derivatives for another variable j, and, for the whole gradient:

$$\nabla_h u_h = \begin{bmatrix} \dfrac{u_{i+1,j} - u_{i,j}}{h} \\ \dfrac{u_{i,j+1} - u_{i,j}}{h} \end{bmatrix}, \tag{3.3}$$

$$\nabla_h = \frac{1}{h} \begin{bmatrix} \nabla_h^1 & & & & \\ & \nabla_h^1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \nabla_h^1 \\ -I & I & & & \\ & -I & I & & \\ & & \ddots & \ddots & \\ & & & -I & I \\ & & & & -I \end{bmatrix} \in \mathbb{R}^{2n^2 \times n^2}, \text{ where } \nabla_h^1 = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & -1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

is a 1D gradient (derivative), $I \in \mathbb{R}^{n \times n}$ is identity matrix. Introduce also the averaged value in the direction i:

$$a_{i,j-1/2} = \frac{a_{i-1/2,j-1/2} + a_{i+1/2,j-1/2}}{2}. \tag{3.4}$$

Define matrix a_h in the following way:

$$a_h = \begin{bmatrix} A1_1 & & & & & & & \\ & A1_2 & & & & & & \\ & & \ddots & & & & & \\ & & & A1_n & & & & \\ & & & & A2_{1/2} & & & \\ & & & & & A2_{3/2} & & \\ & & & & & & \ddots & \\ & & & & & & & A2_{n-1/2} \end{bmatrix} \in \mathbb{R}^{2n^2 \times 2n^2},$$

where

$$A1_j = \begin{bmatrix} a_{1/2,j} & & & \\ & a_{3/2,j} & & \\ & & \ddots & \\ & & & a_{n-1/2,j} \end{bmatrix}, \text{ and } A2_{j-1/2} = \begin{bmatrix} a_{1,j-1/2} & & & \\ & a_{2,j-1/2} & & \\ & & \ddots & \\ & & & a_{n,j-1/2} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

are diagonal matrices with averaged values (3.2)) and (3.4) on the diagonal. Then the matrix representation (3.1) $\Gamma(a) = \nabla_h^T a_h \nabla_h$ holds in terms of usual matrix multiplication for the

case of Dirichlet-Neumann boundary conditions. In case of Dirichlet-Dirichlet conditions, there also will be the following additional term:

$$\Gamma(a) = \nabla_h^T a_h \nabla_h + L_e^T a_h L_e, \tag{3.5}$$

where

$$L_e = \frac{1}{h^2}\begin{bmatrix} L_e^1 \otimes I \\ I \otimes L_e^1 \end{bmatrix} \in \mathbb{R}^{2n^2 \times n^2}, \qquad L_e^1 = \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & 1 \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{3.6}$$

Then we obtain the following discretization scheme:

$$\begin{aligned} [\Gamma(a)u_h]_{(ij)} &= \frac{-a_{i-1/2,j}u_{i-1,j} + (a_{i-1/2,j} + a_{i+1/2,j})u_{i,j} - a_{i+1/2,j}u_{i+1,j}}{h^2} + \\ &+ \frac{-a_{i,j-1/2}u_{i,j-1} + (a_{i,j-1/2} + a_{i,j+1/2})u_{i,j} - a_{i,j+1/2}u_{i,j+1}}{h^2}, \end{aligned} \tag{3.7}$$

where (ij) is a joint 2D index: $(ij) = i + (j-1)n$. So, the matrix Γ has the following elements:

$$\Gamma(a)_{(ij)(km)} = \frac{1}{h^2}\begin{cases} -a_{i,j-1/2}, & k = i,\ m = j-1, \\ -a_{i-1/2,j}, & k = i-1,\ m = j, \\ a_{i-1/2,j-1/2} + a_{i+1/2,j-1/2} + a_{i-1/2,j+1/2} + a_{i+1/2,j+1/2}, & k = i,\ m = j, \\ -a_{i+1/2,j}, & k = i+1,\ m = j, \\ -a_{i,j+1/2}, & k = i,\ m = j+1, \\ 0, & otherwise, \end{cases}$$

$i, j, k, m = 1, ..., n$. In full representation the matrix $\Gamma(a)$ has the following symmetric form:

$$\Gamma(a) = \begin{bmatrix} A_1^0 & A_{\frac{3}{2}}^1 & & & & & \\ A_{\frac{3}{2}}^1 & A_2^0 & A_{\frac{5}{2}}^1 & & & & \\ & \cdots & \cdots & \cdots & & & \\ & & & A_{n-\frac{3}{2}}^1 & A_{n-1}^0 & A_{n-\frac{1}{2}}^1 & \\ & & & & A_{n-\frac{1}{2}}^1 & A_n^0 \end{bmatrix} \in \mathbb{R}^{n^2 \times n^2},$$

where A_j^0, A_j^1 are the following matrices:

$$A_j^0 = \begin{bmatrix} 4a_{1,j} & -a_{\frac{3}{2},j} & & & & & \\ -a_{\frac{3}{2},j} & 4a_{2,j} & -a_{\frac{5}{2},j} & & & & \\ & \cdots & \cdots & \cdots & & & \\ & & & -a_{n-\frac{3}{2},j} & 4a_{n-1,j} & -a_{n-\frac{1}{2},j} & \\ & & & & -a_{n-\frac{1}{2},j} & 4a_{n,j} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$A_{j-1/2}^1 = \begin{bmatrix} -a_{1,j-1/2} & & & & & \\ & -a_{2,j-1/2} & & & & \\ & & \ddots & & & \\ & & & -a_{n-1,j-1/2} & & \\ & & & & -a_{n,j-1/2} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where $a_{i,j}$ is the averaged value in both directions:

$$a_{i,j} = 1/4(a_{i-1/2,j-1/2} + a_{i+1/2,j-1/2} + a_{i-1/2,j+1/2} + a_{i+1/2,j+1/2}).$$

This scheme is known to have the approximation property

$$|u(x_i, y_j) - u_{i,j}| = O(h^2)$$

for smooth enough data, where $u(x_i, y_j)$ is the exact solution u at the grid node with the index (i, j). Notice, that

$$\Gamma(1) = \Delta_h = \nabla_h^T \nabla_h + L_e^T L_e$$

is just a discretized Dirichlet-Dirichlet Laplace operator.

3.2 Spectral equivalence and condition number of preconditioner

First, prove the spectral equivalence of $\Gamma(a)$ and $\Gamma(1) = \Delta_h$.

Lemma 3.1

$$\min a \ \Delta_h \le \Gamma(a) \le \max a \ \Delta_h, \qquad \frac{1}{\max a}\Delta_h \le \Gamma(\frac{1}{a}) \le \frac{1}{\min a}\Delta_h, \tag{3.8}$$

where min *and* max *are taken by the indices* (ij) *from the array* $a_{i-1/2,j-1/2}$.

Proof. Consider the application of Γ to a vector u:

$$\begin{aligned}(\Gamma(a)u, u) &= \frac{1}{h^2}\sum_{i,j=1}^{n} -a_{i-1/2,j}u_{i-1,j}u_{i,j} - a_{i,j-1/2}u_{i,j-1}u_{i,j} + \\ &+ (a_{i-1/2,j} + a_{i,j-1/2} + a_{i+1/2,j} + a_{i,j+1/2})u_{i,j}u_{i,j} - \\ &- a_{i+1/2,j}u_{i+1,j}u_{i,j} - a_{i,j+1/2}u_{i,j+1}u_{i,j}.\end{aligned}$$

By shifting indices of $a_{i-1/2,j}$, $a_{i,j-1/2}$ to $i + 1/2$, $j + 1/2$, with the corresponding shift of an index of u in the sum, we obtain:

$$\begin{aligned}(\Gamma(a)u, u) &= \frac{1}{h^2}\sum_{i,j=1}^{n-1} a_{i+1/2,j}u_{i,j}u_{i,j} + a_{i+1/2,j}u_{i+1,j}u_{i+1,j} - 2a_{i+1/2,j}u_{i,j}u_{i+1,j} + \\ &+ a_{i,j+1/2}u_{i,j}u_{i,j} + a_{i,j+1/2}u_{i,j+1}u_{i,j+1} - 2a_{i,j+1/2}u_{i,j}u_{i,j+1} \\ &= \frac{1}{h^2}\sum_{i,j=1}^{n-1} a_{i+1/2,j}(u_{i+1,j} - u_{i,j})^2 + a_{i,j+1/2}(u_{i,j+1} - u_{i,j})^2.\end{aligned}$$

Indices in sums vary in the range $1, ..., n-1$, since terms with indices 0 and $n+1$ are equal to zero due to the Dirichlet boundary conditions. We see, that $(\Gamma(a)u, u)$ depends linearly on a. By choosing $a = 1$ we obtain the similar representation of $(\Delta_h u, u)$:

$$(\Delta_h u, u) = \frac{1}{h^2}\sum_{i,j=1}^{n-1}(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2.$$

So, we can make the following estimate:

$$(\min a)(\Delta_h u, u) \le (\Gamma(a)u, u) \le (\max a)(\Delta_h u, u),$$

Also, using the same considerations to $\Gamma(1/a)$, and noting that $\min(\frac{1}{a}) = \frac{1}{\max a}$, $\max(\frac{1}{a}) = \frac{1}{\min a}$, we obtain the statement of lemma. ■

Remark 3.2 *Spectral equivalence estimate* (3.8) *is valid for the wide class of Galerkin and finite difference types of discretization of elliptic problems in* $\mathbb{R}^d$*:*

$$\min a(\nabla\phi, \nabla\psi) \le (a\nabla\phi, \nabla\psi) \le \max a(\nabla\phi, \nabla\psi), \qquad \forall\phi, \psi \in H_0^1(\Omega),$$

where $(a\nabla\phi, \nabla\psi)$ *is a Galerkin discretization of the diffusion operator on basis functions* ϕ *and test functions* ψ*.*

Now we prove our main

Theorem 3.3 *Suppose we have a problem* (1.1)*, discretized using the scheme* (3.7)*, and the preconditioner* (2.4) *is used. Then the preconditioned matrix has the following spectral equivalence:*

$$\frac{\min a}{\max a} I \le \Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}\Gamma(a) \le \frac{\max a}{\min a} I.$$

Proof. Using the Lemma 3.1, estimate $\Gamma(1/a)$ and $\Gamma(a)$:

$$\min a\Delta_h \le \Gamma(a) \le \max a\Delta_h,$$

$$\frac{1}{\max a}\Delta_h \le \Gamma(1/a) \le \frac{1}{\min a}\Delta_h.$$

Then for the preconditioned matrix:

$$\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}\Gamma(a) \ge \Delta_h^{-1}(\frac{1}{\max a}\Delta_h)\Delta_h^{-1}(\min a\Delta_h) = \min a\frac{1}{\max a} I.$$

Similar upper bound holds:

$$\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}\Gamma(a) \le \Delta_h^{-1}(\frac{1}{\min a}\Delta_h)\Delta_h^{-1}(\max a\Delta_h) = \max a\frac{1}{\min a} I.$$

■

Corollary 3.4

$$\mathrm{cond}(\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}\Gamma(a)) = O\left(\left(\frac{\max a}{\min a}\right)^2\right)$$

Remark 3.5 *The numerical examples show, that lower spectral bound is sufficiently better:*

$$\lambda_{min}(\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}\Gamma(a)) \ge 1,$$

hence, the condition number in fact is bounded by $O\left(\frac{\max a}{\min a}\right)$*. Although we have no proof of this statement in the general case, in some special cases, such as 1D or the case of one interface (see below) it can be proved.*

3.3 Refined condition number estimate for 1D and 2D problems

In this subsection we present more detailed spectral analysis of the preconditioner in some special cases.

In 1D problem the matrix Γ (3.1) has the following form:

$$\Gamma(a) = \frac{1}{h^2}\begin{bmatrix} a_{1/2}+a_{3/2} & -a_{3/2} & & & & & \\ -a_{3/2} & a_{3/2}+a_{5/2} & -a_{5/2} & & & & \\ & \cdots & \cdots & \cdots & & & \\ & & & -a_{n-3/2} & a_{n-3/2}+a_{n-1/2} & -a_{n-1/2} & \\ & & & & -a_{n-1/2} & a_{n-1/2}+a_{n+1/2} \end{bmatrix}.$$

In terms of the shift matrix it can be represented in the following way:

$$\Gamma(a) = \frac{1}{h^2}(\operatorname{diag}(a) + S\operatorname{diag}(a)S^T + L_e^{1T}\operatorname{diag}(a)L_e^1 - S\operatorname{diag}(a) - \operatorname{diag}(a)S^T),$$

where the shift matrix S is specified by:

$$S = \begin{bmatrix} 0 & 1 & 0 & & & \\ 0 & 0 & 1 & & & \\ & \cdots & \cdots & \cdots & & \\ & & & 0 & 0 & 1 \\ & & & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

and

$$\operatorname{diag}(a) = \begin{bmatrix} a_{1/2} & & & & & \\ & a_{3/2} & & & & \\ & & \ddots & & & \\ & & & a_{n-3/2} & & \\ & & & & a_{n-1/2} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

is just the matrix with values of a in appropriate places on the diagonal (one dimensional analog of a_h), and $L_e^1 = L_e^{1T}$ is a matrix with 1 at position (n,n), see (3.6). One-dimensional gradient ∇_h can be represented in the following way: $\nabla_h = \dfrac{I - S^T}{h}$. Then one can easily show, that this representation is equivalent to $\nabla_h^T a_h \nabla_h$ up to the one element:

$$\begin{aligned} \Gamma(a) &= \frac{1}{h^2}((I-S)\operatorname{diag}(a) + (S-I)\operatorname{diag}(a)S^T + L_e^{1T}\operatorname{diag}(a)L_e^1) \\ &= \frac{I-S}{h}\operatorname{diag}(a)\frac{I-S^T}{h} + \frac{L_e^{1T}\operatorname{diag}(a)L_e^1}{h^2} \\ &= \nabla_h^T\operatorname{diag}(a)\nabla_h + \frac{L_e^{1T}\operatorname{diag}(a)L_e^1}{h^2}. \end{aligned}$$

This representation is useful for some tensor formats, such as QTT format (see [15, 10]), which may sufficiently reduce the amount of memory and computational cost, if the tensor ranks of vectors and matrices are not large. The shift matrix has QTT rank 2, and the diagonal matrix has rank 1, so if we have the diffusion coefficient in low QTT-rank representation, the QTT-rank of $\Gamma(a)$ is also bounded:

$$\text{QTT-rank}(\Gamma(a)) \le 7\,\text{QTT-rank}(a).$$

In 1D case, the elements of Δ_h^{-1} can be written explicitlly:

$$[\Delta_h^{-1}]_{ij} = (n+1) \cdot \begin{cases} x_i\, y_j, & i \geq j, \\ y_i\, x_j, & i < j, \end{cases}$$

where

$$x_i = 1 - \frac{i}{n+1}, \quad y_j = \frac{j}{n+1}.$$

Consider the application of matrices $\Gamma(1/a)\Delta_h^{-1}$ and $\Gamma(a)\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}$ to an arbitrary vector f:

$$g = \Gamma(1/a)\Delta_h^{-1} f, \quad v = \Gamma(a)\Delta_h^{-1} g = \Gamma(a)\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1} f.$$

Then straightforward calculations gives the following

Lemma 3.6

$$g_i = \frac{1}{a_{i-1/2}} f_i - (\frac{1}{a_{i+1/2}} - \frac{1}{a_{i-1/2}}) \sum_{j=i+1}^{n} f_j + (\frac{1}{a_{i+1/2}} - \frac{1}{a_{i-1/2}}) c(f),$$

where

$$c(f) = \sum_{j=1}^{n} y_j f_j,$$

and

$$v_i = f_i - (a_{i+1/2} - a_{i-1/2})(\frac{1}{a_{n-1/2}} c(f) - c(g(f)).$$

The formula $\frac{1}{a_{n-1/2}} c(f) - c(g(f))$ is a linear functional of f. So,

$$\Gamma(a)\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1} = I + R,$$

where $\mathrm{rank}(R) = 1$. That gives the convergence of iterative solvers like PCG or GMRES in 2 iterations.

Remark 3.7 *In this Lemma we used the conception of right preconditioning:*

$$\Gamma(a)u = f \;\; \rightarrow \;\; (\Gamma(a)P)\; (P^{-1}u) = f,$$

but in numerical examples below we use $P = \Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}$ as a left preconditioner. But, as the matrices $\Gamma(a)$ and P are symmetric, it can be easily shown, that the left preconditioner has the same spectral properties:

$$P\Gamma(a) = P^T\Gamma(a)^T = (\Gamma(a)P)^T = (I+R)^T = I + R^T,$$

and R^T also has rank 1. Eigenvalues of the preconditioned matrix are also the same for left and right preconditioning. So, we consider later only right preconditioning.

Now consider 2D case. In the particular case of one interface in 2D, i.e. the coefficient has only one jump:

$$a(x,y) = \begin{cases} a_1, & y \le y_0, \\ a_2, & y > y_0. \end{cases},$$

we can prove, that eigenvalues of the preconditioned matrix form a finite amount of clusters.

Introduce the following matrix D_l:

$$D_l = \frac{1}{h^2} \begin{bmatrix} 0 & 0 & & & & & & \\ \cdots & \cdots & \cdots & & & & & \\ & 0 & 0 & 0 & & & & \\ & & 0 & D^0_{j_0} & D^1_{j_0} & & & \\ & & & D^1_{j_0} & 0 & 0 & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & 0 & 0 & \end{bmatrix} \in \mathbb{R}^{n^2 \times n^2},$$

where $D^0_{j_0}$ and $D^1_{j_0}$ are the following matrices:

$$D^0_{j_0} = \begin{bmatrix} 2 & -1/2 & & & & & \\ -1/2 & 2 & -1/2 & & & & \\ & \cdots & \cdots & \cdots & & & \\ & & & -1/2 & 2 & -1/2 & \\ & & & & -1/2 & 2 & \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$D^1_{j_0} = \begin{bmatrix} -1 & & & & \\ & -1 & & & \\ & & \ddots & & \\ & & & -1 & \\ & & & & -1 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

and block $D^0_{j_0}$ is in j_0, j_0 position in matrix D_l. Using this notation we can write the following

Lemma 3.8 *In the case of one interface the matrix Γ has the following representation:*

$$\Gamma(a) = \operatorname{diag}[a]\, \Delta_h + (a_2 - a_1) D_l,$$

where $\operatorname{diag}[a]$ *is a diagonal matrix with values $a_{i-1/2,j-1/2}$ on $(ij),(ij)$ place, and zeros otherwise.*

Proof. Since a has one interface, all rows (and columns) in $\Gamma(a)$, with indices (ij), $j < j_0$ are equal to

$$\frac{1}{h^2} \begin{pmatrix} -a_1 & \cdots & -a_1 & 4a_1 & -a_1 & \cdots & -a_1 \end{pmatrix},$$

which is equal to the corresponding row of Δ_h, multiplied by a_1. In the same way we obtain rows with $j > j_0$. As for the interface rows, they can be represented in the following way (here we write only nonzero elements for brevity):

$$\begin{aligned} & \frac{1}{h^2} \begin{pmatrix} -a_1 & -\dfrac{a_1+a_2}{2} & 4\dfrac{a_1+a_2}{2} & -\dfrac{a_1+a_2}{2} & -a_2 \end{pmatrix} \\ = \; & a_1 \; [\Delta_h]_{(ij)} + \frac{1}{h^2} \begin{pmatrix} 0 & -\dfrac{a_2-a_1}{2} & 4\dfrac{a_2-a_1}{2} & -\dfrac{a_2-a_1}{2} & -(a_2-a_1) \end{pmatrix} \\ = \; & a_1 \; [\Delta_h]_{(ij)} + (a_2 - a_1) \; [D_l]_{(ij)} \end{aligned}$$

■

From this Lemma one can obviously deduce

Corollary 3.9 *In the case of one interface*

$$\Gamma(a)\ \Delta_h^{-1} = \mathrm{diag}\,[a] + (a_2 - a_1) D_l \Delta_h^{-1}.$$

Now we consider preconditioned matrix $\Gamma(a)$ in the terms of right preconditioning: $\Gamma(a)P = \Gamma(a)\Delta_h^{-1}\Gamma(\frac{1}{a})\Delta_h^{-1}$. We are to investigate the spectral properties of this matrix.

Lemma 3.10 *In the case of one interface*

$$\Gamma(a)P = I + \left(\frac{a_1}{a_2} + \frac{a_2}{a_1} - 2\right)(D_l\Delta_h^{-1}) - \left(\frac{a_1}{a_2} + \frac{a_2}{a_1} - 2\right)(D_l\Delta_h^{-1})^2.$$

Proof. Using Corollary 3.9 we write the preconditioned matrix in the following way:

$$\begin{aligned}
\Gamma(a)P &= \left(\mathrm{diag}\,[a] + (a_2 - a_1) D_l \Delta_h^{-1}\right)\left(\mathrm{diag}\left[\frac{1}{a}\right] + \left(\frac{1}{a_2} - \frac{1}{a_1}\right) D_l\Delta_h^{-1}\right) \\
&= I + \left(\frac{1}{a_2} - \frac{1}{a_1}\right)\mathrm{diag}\,[a]\, D_l\Delta_h^{-1} + (a_2 - a_1) D_l\Delta_h^{-1}\,\mathrm{diag}\left[\frac{1}{a}\right] + \\
&+ (a_2 - a_1)\left(\frac{1}{a_2} - \frac{1}{a_1}\right)(D_l\Delta_h^{-1})^2.
\end{aligned}$$

$\mathrm{diag}\,[a]$ multiplies rows or columns of the matrix $D_l\Delta_h^{-1}$, depending on its respective position in matrix multiplication, by the corresponding values of a. However, nonzero elements in $D_l\Delta_h^{-1}$ stay only in positions, corresponding to (ij_0), where $a = a_1$ (by the definition of $\mathrm{diag}\,[a]$, it has $a_{i-1/2,j_0-1/2} = a_1$ in $(ij_0), (ij_0)$ place). Hence, matrices $\mathrm{diag}\,[a]$ and $\mathrm{diag}\left[\frac{1}{a}\right]$ in the second and third terms of $\Gamma(a)P$ produce just multiplication by a_1 and $1/a_1$, respectively. Then we obtain the statement of Lemma. ■

Using the last Lemma, we can easily deduce, that if $D_l\Delta_h^{-1}$ has an eigenvector x and the corresponding eigenvalue λ, then the preconditioned matrix has the same vector x as an eigenvector, and the eigenvalue

$$\mu = 1 + \left(\frac{a_1}{a_2} + \frac{a_2}{a_1} - 2\right)\lambda - \left(\frac{a_1}{a_2} + \frac{a_2}{a_1} - 2\right)\lambda^2.$$

So, now we are to proof, that $D_l\Delta_h^{-1}$ has only a few different eigenvalues, hence, eigenvalues of preconditioned matrix form a cluster.

Lemma 3.11 *The matrix $D_l\Delta_h^{-1}$ has just two different eigenvalues: 0 and $1/2$.*

Proof. To prove the statement, we are to show, that the minimal characteristic polynomial of matrix $D_l\Delta_h^{-1}$ has the following form:

$$(D_l\Delta_h^{-1})^2 - \frac{1}{2} D_l\Delta_h^{-1} = 0,$$

that is, for any vector x the following holds:

$$(D_l\Delta_h^{-1})^2x - \frac{1}{2}D_l\Delta_h^{-1}x = 0. \tag{3.9}$$

Using the discrete sine Fourier transform, one can easily deduce, that any vector x can be decomposed as follows:

$$x = \sum_{k,m} \alpha_{k,m}F_kF_m + \beta_{k,m}G_kF_m + \gamma_{k,m}F_kG_m + \delta_{k,m}G_kG_m,$$

with some constants $\alpha, \beta, \gamma, \delta$, and basis functions

$$F_{k\ i} = \sin(\pi hki), \quad G_{k\ i} = \cos(\pi hki).$$

It is known, that the set $\{\sin(\pi hki)\}$ is an orthogonal basis of eigenvectors of Δ_h^{-1}. Moreover, it is orthogonal to the corresponding cos-set:

$$\sum_{i=1}^{n} \sin(\pi hki)\cos(\pi hmi) = 0, \quad for\ any\ k, m, \tag{3.10}$$

and

$$\sum_{i=1}^{n} \sin(\pi hki)\sin(\pi hmi) = 0, \quad if\ k \neq m.$$

Since the characteristic polynomial has matrix Δ_h^{-1} in it, it eliminates all cos-functions in x:

$$\Delta_h^{-1}x = \sum_{k,m} \alpha_{k,m}\lambda_{k,m}(\Delta_h^{-1})F_kF_m.$$

So, we should check the equation (3.9) only on the following basis functions:

$$x = \{x_{i,j}\} = \{\sin(\pi hki)\sin(\pi hmj)\},$$

where i, j, k, m vary in the range $1, ..., n$. If it holds for all these functions, then it holds for any vector x.

First of all, consider the application of Δ_h to a vector from this set:

$$\begin{aligned}
[h^2\Delta_h x]_{(ij)} &= -\sin(\pi hki)\sin(\pi hm(j-1)) - \sin(\pi hk(i-1))\sin(\pi hmj) + \\
&+ 4\sin(\pi hki)\sin(\pi hmj) - \\
&- \sin(\pi hk(i+1))\sin(\pi hmj) - \sin(\pi hki)\sin(\pi hm(j+1)).
\end{aligned}$$

After the summation of the first and the last terms, and of the second and the fourth, and so on, we obtain:

$$\Delta_h x = \frac{1}{h^2}(4 - 2\cos(\pi hk) - 2\cos(\pi hm))x.$$

Denote $\frac{1}{h^2}(4 - 2\cos(\pi hk) - 2\cos(\pi hm)) = \lambda(\Delta_h)$. Then the application of Δ_h^{-1} on test functions is:

$$\Delta_h^{-1}x = \lambda^{-1}(\Delta_h)x.$$

Now, consider the application of D_l to the test functions:

$$\begin{aligned}
[h^2 D_l x]_{(ij)} &= -1/2\ \sin(\pi hk(i-1))\sin(\pi hmj) + 2\sin(\pi hki)\sin(\pi hmj) - \\
&- 1/2\ \sin(\pi hk(i+1))\sin(\pi hmj) - \sin(\pi hki)\sin(\pi hm(j+1)) \\
&= -\sin(\pi hki)\sin(\pi hmj)\cos(\pi hk) + 2\sin(\pi hki)\sin(\pi hmj) - \\
&- \sin(\pi hki)\sin(\pi hmj)\cos(\pi hm) - \sin(\pi hki)\cos(\pi hmj)\sin(\pi hm).
\end{aligned}$$

Recalling the eigenvalues of Δ_h:

$$[D_l x]_{(ij)} = 1/2\ \lambda(\Delta_h) x_{(ij)} - \frac{1}{h^2}\sin(\pi hki)\cos(\pi hmj)\sin(\pi hm).$$

Since the sin-set of functions is orthogonal to the cos-set (3.10),

$$\Delta_h^{-1}\{\sin(\pi hki)\cos(\pi hmj)\} = 0.$$

Then

$$[D_l \Delta_h^{-1} x]_{(ij)} = 1/2\ x_{(ij)} - \frac{\lambda^{-1}(\Delta_h)}{h^2}\sin(\pi hki)\cos(\pi hmj)\sin(\pi hm),$$

$$\Delta_h^{-1} D_l \Delta_h^{-1} x = \Delta_h^{-1}(1/2\ x - \frac{\lambda^{-1}(\Delta_h)}{h^2}\{\sin(\pi hki)\cos(\pi hmj)\}\sin(\pi hm)) = 1/2\ \lambda^{-1}(\Delta_h)x,$$

and

$$(D_l\Delta_h^{-1})^2 x = 1/4\ x - 1/2\ \frac{\lambda^{-1}(\Delta_h)}{h^2}\{\sin(\pi hki)\cos(\pi hmj)\}\sin(\pi hm) = 1/2\ D_l\Delta_h^{-1}x.$$

So, the characteristic equation holds on the set which forms a basis, and, hence, on any vector. ■

Summarizing the Lemmas 3.10 and 3.11 we obtain the main

Theorem 3.12 *The preconditioned matrix $\Gamma(a)P$ has two clusters of eigenvalues: in*

$$\lambda_1 = 1,$$

and

$$\lambda_2 = 1 + 1/4\left(\frac{a_1}{a_2} + \frac{a_2}{a_1} - 2\right).$$

This leads to the convergence of GMRES in 2, or maybe 3 iterations.

The proposed proofs in 2D can be easily generalized to the higher dimensional case. The discretization scheme (3.7) will contain also derivatives in the other directions, the spectral equivalence will still be the same (in the proof of Lemma 3.1 terms like $a_{i,j,k+1/2}(u_{i,j,k+1} - u_{i,j,k})^2$ will arise, and so on). The clustering of eigenvalues in the case of one interface will also hold, as the set of separable test functions $\{\sin(\pi hki)\sin(\pi hmj)\}$ can be generalized obviously. There will be other eigenvalues, not just 0, 1/2 in 2D, but they also form clusters.

4 Numerical tests on preconditioning and rank bounds

EXAMPLE 1. First of all, we investigate the spectral properties of the proposed preconditioner on the 1-interface case in 2D. We choose the following coefficient

$$a(x,y) = \begin{cases} \alpha, & if\ y \le 0.5, \\ 1, & if\ y > 0.5. \end{cases}$$

The PCG and GMRES solvers converge in 2 iterations to the accuracy in the order of machine precision with any α and random right-hand side. The distribution of spectrum of preconditioned matrix is shown on Fig. 4.1 We see, that eigenvalues form two separate

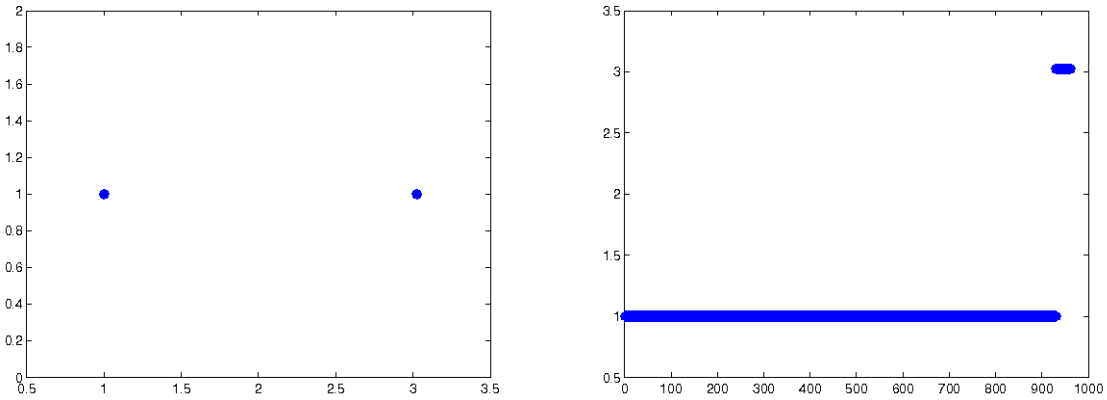

Figure 4.1: The distribution of eigenvalues along the real axis (left), and the eigenvalue versus its number (right) in the case of 1 interface

clusters, confirming Theorem 3.12.

In next examples we check the convergence of PCG iterations for preconditioned problems with various coefficients. In each case, the several runs of program with random right-hand sides were made. It shows the same number of iterations to achieve the desired relative accuracy in each run, that confirms the clustering structure of eigenvalues. In each case, the full representation of the matrix and the vectors is used, so, the timings scale as n^d. All computations are done using the MATLAB 7.9 (R2009b) and Intel C Compiler (icc) (for MATLAB MEX-functions) on a Linux Dual Core AMD Opteron machine with clock-speed 2.6 GHz, and cache size 1Mb.

EXAMPLE 2 (TABLE 4.1, 4.2). 2D Dirichlet problem, $1/a(x,y) = \mathtt{chk}(x) + \mathtt{chk}(y)$, where

$$\mathtt{chk}(x) = \begin{cases} 1, & if\ [x \cdot 16]\ is\ odd, \\ \alpha, & if\ [x \cdot 16]\ is\ even \end{cases}$$

(see Fig. 4.2).

EXAMPLE 3 (TABLE 4.3, 4.4). 3D Neumann problem, the projection of a on each of planes xy, yz, xz is shown on Fig. 4.4, the dark regions has $a(x,y,z) = \alpha$, the others $a = 1$, $f = \cos(\pi x)\cos(\pi y)\cos(\pi z)$.

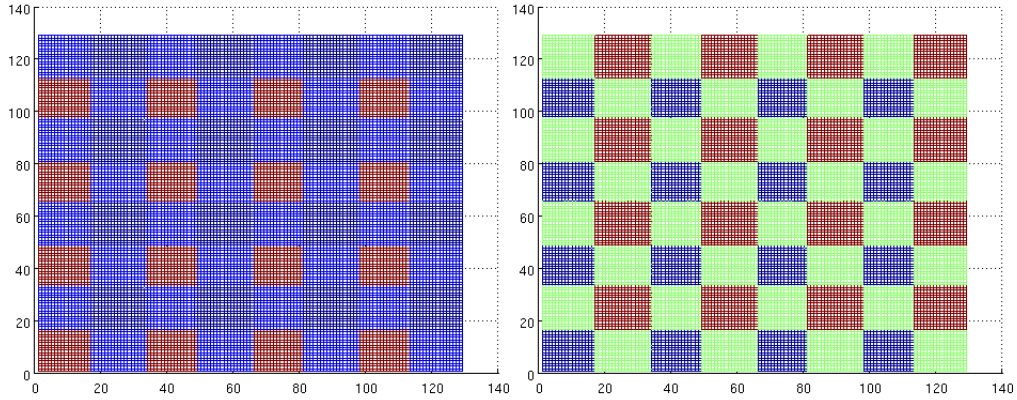

Figure 4.2: Diffusion coefficient $a(x, y)$ (left) and $1/a(x, y)$ (right) in the example 2

Table 4.1: Number of iterations to $||Au - f||/||f|| < 10^{-8}$, and CPU time of the solution versus the number of grid points in each direction n, $\alpha = 10$, example 2.

n^2	iterations	CPU time , s
32^2	11	0.015
64^2	11	0.036
128^2	12	0.116
256^2	12	0.433
512^2	12	1.935
1024^2	12	7.754
2048^2	12	29.53

Table 4.2: Number of iterations to $||Au - f||/||f|| < 10^{-8}$, versus jumps in the coefficient α, $n = 256$, example 2.

α	iterations
0.01	21
0.1	13
2	6
10	12
100	21
1000	27
10^4	31
10^5	64

We can also see, that, despite of the Neumann boundary conditions, the preconditioned problem is solvable by CG.

Example 4 (Table 4.5). The coefficient in this example was taken from [4]: 2D

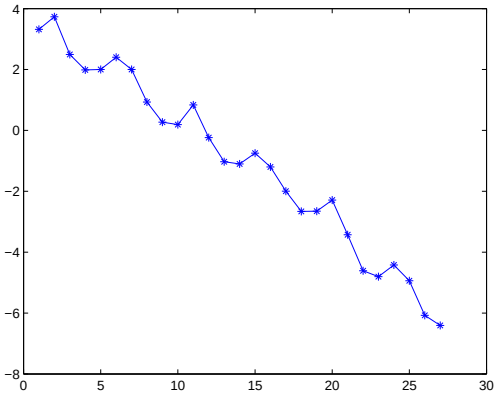

Figure 4.3: Convergence history in example 2: $\log_{10} ||Au - f||$ vs iteration, $\alpha = 10^3$, $n = 256$.

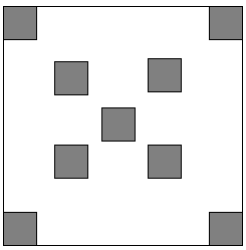

Figure 4.4: The projection of the coefficient $a(x, y, z)$ on each of planes xy,yz,zx in example 3

Table 4.3: Number of iterations to $||Au - f||/||f|| < 10^{-8}$ and CPU time of the solution versus the number of grid points in each direction n, $\alpha = 10^3$, example 3.

n^3	iterations	CPU Time, sec
64^3	15	2.877
128^3	16	32.6
256^3	16	325

Dirichlet problem,

$$a(x,y) = \begin{cases} 1, & 0.125 \le \sqrt{(x-0.5)^2 + (y-0.5)^2} \le 0.25, \\ 10, & otherwise. \end{cases}$$

We show here also the convergence results of AMG from the work [4].

Example 5 (Table 4.6, 4.7). In the last example, we test ranks of the QTT tensor approximation (see [15, 10]) of the reciprocal coefficient $1/a$ and the operator $\Gamma(1/a)$. As Δ_h^{-1} is known to have good QTT-compression properties, we are to study $\Gamma(1/a)$. In future work we are going to formulate our preconditioner in QTT format. As here we use the compression from the full representation of vector and matrix, we are limited in the grid size. Below we use 32 and (in 2D) 64 grid points in each direction. As the QTT-rank we

Table 4.4: Number of iterations to $||Au - f||/||f|| < 10^{-8}$ versus jumps in the coefficient α, $n = 128$, example 3.

α	iterations
0.01	15
0.1	10
10	10
100	15
1000	16
10^4	18
10^5	24

Table 4.5: Number of iterations to $||Au - f||/||f|| < 10^{-8}$ and CPU time of the solution versus the number of grid points in each direction n, example 4.

n^2	iters(AMG)	iters($\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}$)	CPU Time($\Delta_h^{-1}\Gamma(1/a)\Delta_h^{-1}$), s
128^2	10	6	0.089
256^2	11	6	0.337

denote the maximum rank of QTT cores.

Table 4.6: QTT ranks of $1/a$ and $\Gamma(1/a)$ in previous examples. Results are the same for approximation tolerances $\varepsilon = 10^{-2}, ..., 10^{-12}$.

example, n	QTT-rank($1/a$)	QTT-rank($\Gamma(1/a)$)
1, $n = 32$	2	4
1, $n = 64$	2	4
2, $n = 32$	5	9
2, $n = 64$	5	9
3, $n = 32$	9	22
4, $n = 32$	14	30
4, $n = 64$	26	54

We can see, that the QTT-ranks of matrix $\Gamma(1/a)$ is proportional to the ranks of $1/a$. Moreover, if the pattern of a has the rectangle structure, the ranks do not depend on grid size n. In Example 4 the interface has the circle shape, so, its QTT-compression properties are worse.

Example 6 (Table 4.8). Now we test the preconditioner for the coefficient a which tends to zero near the boundary (degenerate coefficients):

$$a(x, y) = x(1 - x)y(1 - y),$$

Table 4.7: QTT ranks of discrete solution u_h in previous examples vs approx. tolerance ε.

example, n	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-10}$
1, $n = 32$	4	8	11	13	16
1, $n = 64$	4	8	12	15	26
2, $n = 32$	7	16	24	30	32
2, $n = 64$	7	22	34	49	59
3, $n = 32$	12	40	85	120	128
4, $n = 32$	12	16	22	26	29
4, $n = 64$	14	30	33	43	48

and random right-hand side. As the condition number of the preconditioned matrix is limited by $\max a / \min a$, this preconditioner works sufficiently worse, than in the previous examples. We see, that the number of iterations is sufficiently large, and depends on a grid

Table 4.8: Number of iterations to $||Au - f||/||f|| < 10^{-8}$, CPU time of the solution and QTT ranks versus the number of grid points in each direction n, example 6.

n^2	iterations	CPU time , s	QTT-rank($1/a$)	QTT-rank(u_h)
32^2	16	0.04	10	17
64^2	24	0.126	14	21
128^2	35	0.597	16	25
256^2	51	6.87	17	29
512^2	73	23.96	19	33
1024^2	105	282.12	19	36

size. Nevertheless, this preconditioner can be used for moderate grid sizes. Another way to achieve good convergence (and, also, approximation of the discretization scheme) is to use the adaptive grids, which increase the grid-points density near the boundaries (boundary layer).

QTT ranks of the diffusion coefficient are about 3 for any compression accuracy ε and grid size n. But one can see, that the ranks of the reciprocal coefficient and the solution (in this example we set the compression accuracy to 10^{-10}) are quite large and depend on a grid size.

5 Conclusion

We studied the schemes for the solution of multidimensional elliptic problem. We described the quasi-optimal preconditioner for the elliptic equation in operator and discrete forms. The preconditioned matrix of the discrete problem is spectrally equivalent to the identity matrix with constants depending only on jumps in the diffusion coefficient. We tested the proposed

preconditioner on 2D and 3D problems with Dirichlet and Neumann boundary conditions. In the case of non-degenerate coefficient the preconditioner provides the convergence of the PCG or GMRES type methods in at most of several tens iterations independently on the grid size. In many cases, the proposed method can be applied as a black-box solver and it provides better convergence and timings, than multigrid methods [4]. Our approach is suitable for a wide class of the coefficients, in comparison with more special preconditioners [1].

Another important part of the work is the study of low-rank tensor approximations to the solutions of elliptic problems and of the proposed preconditioner. We obtained, that the finite element/finite difference matrix and the respective preconditioner can be approximated with low-rank tensor structures such as canonical or TT/QTT tensor formats. Although, most statements are proved in 2D case, their generalization to the higher dimensional case is straightforward. In this paper we tested the tensor properties of full solutions, obtained without tensor approximations. The implementation of the proposed preconditioned iteration in the compressed tensor formats will be considered in the forthcoming paper.

References

[1] B. Aksoylu, I. G. Graham, H. Klie and R. Scheichl: *Towards a rigorously justified algebraic preconditioner for high-contrast diffusion problems*, Computing and Visualization in Science, v. 11, no. 4-6 (2008), pp. 319-331.

[2] G. B. Arfken, H. J. Weber: *Mathematical Methods for Physicists, 6th edition*, Academic Press, San Diego (2005), pp. 95101.

[3] G. Beylkin, M. M. Mohlenkamp: *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., v. 26, no. 6 (2005), 2133–2159.

[4] A. J. Cleary, R. D. Falgout, V. E. Henson, and others: *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Comput., v. 21, no. 5 (2000), pp. 1886-1908.

[5] I.P. Gavrilyuk, W. Hackbusch, and B.N. Khoromskij: *Tensor-product approximation to the inverse and related operators in high-dimensional elliptic problems*. Computing **74** (2005), 131-157.

[6] L. Grasedyck: *Existence and computation of a low Kronecker-rank approximation to the solution of a tensor system with tensor right-hand side*. Computing **72** (2004), 247–265.

[7] W. Hackbusch, B.N. Khoromskij, S. Sauter and E. Tyrtyshnikov: *Use of Tensor Formats in Elliptic Eigenvalue Problems.* Preprint 78, MPI MiS, Leipzig 2008 (submitted).

[8] W. Hackbusch, B.N. Khoromskij and E.E. Tyrtyshnikov: *Hierarchical Kronecker tensor-product approximations*. J. Numer. Math. **13** (2005), 119–156.

[9] B.N. Khoromskij: *Tensor-Structured Preconditioners and Approximate Inverse of Elliptic Operators in* $\mathbb{R}^d$. J. Constructive Approx., **30**: 599-620 (2009).

[10] B. N. Khoromskij: *$O(d \log N)$-Quantics Approximation of $N - d$ Tensors in High-Dimensional Numerical Modeling*. Preprint 55/2009. Max-Plank-Institut für Mathematik in den Naturwissenschaften. Leipzig 2009 (submitted).

[11] B.N. Khoromskij: *Tensors-structured Numerical Methods in Scientific Computing: Survey on Recent Advances*. Preprint 21/2010, MPI MiS Leipzig 2010 (submitted).

[12] B. N. Khoromskij, and I. V. Oseledets: *Quantics-TT Approximation of Elliptic Solution Operators in Higher Dimensions*. Preprint 79/2009. Max-Plank-Institut für Mathematik in den Naturwissenschaften. Leipzig 2009 (submitted).

[13] I.V. Oseledets: *A new tensor decomposition*. Doklady Mathematics, Vol. 80, No. 1 (2009), pp. 495-496.

[14] I.V. Oseledets, E.E. Tyrtyshnikov: *Breaking the curse of dimensionality, or how to use SVD in many dimensions*. SIAM J. Sci. Comput. Vol. 31, No. 5 (2009), pp. 3744-3759.

[15] I.V. Oseledets, E.E.: *Approximation of $2^d \times 2^d$ matrices using tensor decomposition*. SIAM J. Matrix Anal. Appl., Vol. 31, No. 4 (2010), pp. 2130-2145.

[16] E.E. Tyrtyshnikov: *Tensor approximations of matrices generated by asymptotically smooth functions*. Sbornik: Mathematics **194**, No. 5-6 (2003), 941–954 (translated from *Mat. Sb.* **194**, No. 6 (2003), 146–160).

[17] E.E. Tyrtyshnikov: *Kronecker-product approximations for some function-related matrices*. Linear Algebra Appl. **379** (2004), 423–437.