ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

BLACK-BOX SOLVER FOR ONE DIMENSIONAL MULTISCALE MODELLING USING THE QTT FORMAT

Ivan V. Oseledets[1,2], **Maxim V. Rakhuba**[1], **Andrei V. Chertkov**[1]

[1] Skolkovo Institute of Science and Technology
Skolkovo Innovation Center, Building 3, Moscow 143026, Russia
e-mail: {i.oseledets, m.rakhuba}@skoltech.ru, andrei.chertkov@skolkovotech.ru

[2] Institute of Numerical Mathematics, Russian Academy of Sciences
Gubkina St. 8, Moscow 119333, Russia
e-mail: ivan.oseledets@gmail.com

Keywords: Stable PDE discretization, TT-format, QTT-format, multiscale modelling, black-box solver.

Abstract. *In the present work we propose an efficient black-box solver for one-dimensional multiple scaled diffusion equation. For this problem it has been recently shown [1] that the solution can be represented in a certain low parametric representation, namely the quantized tensor train (QTT) format [2]. The key idea of the QTT format is to make the real space data multidimensional by introducing virtual dimensionalities. The next step is to apply the tensor train (TT) representation [3] to multidimensional data, which leads us to the logarithmic complexity. Hence very fine grids that describe the finest scale can be used.*

Since the solution of second order multi-scale problems can be represented in the QTT format, simple and efficient solvers can be developed using the existing software for the approximate solution of linear systems in the TT-format. However, if equations are discretized using standard finite element/difference methods, it is not possible to get to very fine meshes, say with 2^{50} *grid points due to the condition number. On the other hand, the theory guarantees the existence of a good QTT-FEM approximant of the continuous problem. Thus, another discretization should be used to compute it numerically.*

Our idea is to rewrite the initial formulation in a certain form without derivatives. After that we get an explicit formula, which consists of the inversion of a diagonal matrix and the multiplication by a dense matrix. The latter can be multiplied with logarithmic complexity in the QTT format due to a special structure. The numerical experiment show that this formula gives accurate results and can be used for 2^{50} *grid points with no problems with conditioning, while total computational time is around several seconds.*

1 INTRODUCTION

We consider a model 1D diffusion equation

$$-\frac{\partial}{\partial x}\left(k(x)\frac{\partial}{\partial x}u(x)\right) = f(x), \quad u(0) = u(1) = 0 \tag{1}$$

with coefficient $k(x)$ that has multiple scales. The problem to solve this equation directly is that very fine grid that describes the finest scale has to be introduced. Alternatively one can solve this problem using the approach based on analytical expansions or use specific multiscale finite element methods. Despite these approaches work very efficiently and are well-developed, still they do not have enough generality and rely on the knowledge of analytical behavior of the solution.

In [4] it was shown that the solution of (1) can be represented in a certain low parametric representation, namely the quantized tensor train (QTT) format [2]. The idea behind the QTT approach is as follows. First of all we introduce very fine grid that is able to describe the finest scale of the problem. To work efficiently with such grids we use low-parametric representations, namely tensor decompositions that deal with high-dimensional data. The key idea is to make the real space 1D data multidimensional by introducing virtual dimensionalities, which leads us to logarithmic complexity.

Although the solution can be approximated in the QTT-format [1], it is very diffucult to recover using standard finite difference approaches. Indeed, even if $k \equiv 1$ the simplest discretization scheme reads

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f(x_i), \quad i = 1, \ldots, n, \quad u_0 = u_{n+1} = 0, \quad h = \frac{1}{n+1}.$$

It is well known that $u_i - u(x_i) = \mathcal{O}(h^2)$, i.e. the smaller the h, the better is the approximation. However, in numerical computations we can not take h too small. Let ϵ be the accuracy of the computations. Then the approximation error of the action of discrete operator can be estimated as

$$\mathcal{O}\left(\frac{\epsilon}{h^2} + h^2\right),$$

which means that the minimal possible grid step is $h \sim \epsilon^{-1/4}$. For the double precision $\epsilon \approx 10^{-16}$ the grid step $h \sim 10^{-4}$ is the minimal possible. In principle, such small grid steps are rarely (not to say never) encountered in standard mathematical modelling, especially when we go from one-dimensional to 2D and 3D problems. However, as was already mentioned we have recently encountered a problem when we need h much beyond the number mentioned above, even $h \sim 2^{-d}$ where $d \geq 20$.

In this paper we propose an explicit formula for finding solution of (1) that resolves the problem with accuracy on very fine grids. We describe how to apply this formula in the QTT format and provide bound estimates for the rank of solution. We also show the relation between the proposed formula and standard second order finite difference scheme. In numerical experiments we illustrate theoretical results and provide comparison with the homogenization approach.

2 ROBUST DISCRETIZATION SCHEME

Consider a one-dimensional diffusion equation (1). This problem is equivalent to the minimization of the functional

$$u = \arg\min_{v(0)=v(1)=0} F(v), \quad F(v) = \int_0^1 k\left(\frac{\partial v}{\partial x}\right)^2 dx - 2\int_0^1 vf dx.$$

Now we introduce additional variable

$$v_x = \frac{\partial v}{\partial x} = B(v). \tag{2}$$

From (2) and taking into account the boundary conditions, we can write

$$v(x) = \int_0^x v_x(t)dt.$$

To satisfy the boundary condition at $x = 1$ the function v_x has to satisfy

$$\int_0^1 v_x(t)dt = 0.$$

Finally we have the following optimization problem:

$$u_x = \arg\min \int_0^1 k v_x^2 dx - 2\int_0^1 \mathcal{B}(v_x) f dx, \quad \text{s.t.} \quad \int_0^1 v_x(t)dt = 0, \tag{3}$$

where $u_x = u'$ is the derivative of the solution of equation (1).

Now we replace the integrals in (3) by the rectangular rule on the uniform mesh with grid step h and have the following quadratic optimization problem:

$$F(v_x) = (Dv_x, v_x) - 2(Bv_x, f) = (Dv_x, v_x) - 2(v_x, B^\top f), \quad \text{s.t.} \quad e^\top u_x = 0,$$

where B is the discretization of the operator $\mathcal{B}$, D is a diagonal matrix with

$$d_i = k\left(x_{i-\frac{1}{2}}\right), \quad x_{i-\frac{1}{2}} = \left(i - \frac{1}{2}\right)h, \quad i = 1, \ldots, n$$

and e is the vector of all ones. The unknowns u_x are defined in the midpoints as well. Introducing Lagrange multiplier for the constraint, we have

$$Du_x = B^\top f + \alpha e, \quad e^\top u_x = 0,$$

therefore

$$\alpha = -\frac{e^\top D^{-1} B^\top f}{e^\top D^{-1} e}.$$

Finally the solution is given (not unexpectedly!) by the explicit formula

$$u = Bu_x = BD^{-1}B^\top f - \frac{e^\top D^{-1} B^\top f}{e^\top D^{-1} e} BD^{-1}e. \tag{4}$$

Matrix B plays a crucial role. Let u be defined on the grid points, and the centered second-order finite different scheme reads

$$(u_x)_{i-\frac{1}{2}} = \frac{u_i - u_{i-1}}{h},$$

and the matrix B is given as

$$B_{ij} = \begin{cases} h, & i \geq j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The formula (4) can be considered as a stable discretization of the original equation. It involves only elementwise operations and Volterra integral operations, and it is easy to see that errors decrease while h goes to zero. Moreover, these operations can be efficiently implemented in the QTT-format, and this is the main motivation why such discretizations are interesting in practice.

3 CONNECTION WITH FINITE DIFFERENCE SCHEME

The next theorem shows the relation between standard second order discretization scheme and the formula (4).

Theorem 1. *In exact arithmetics solution obtained by the proposed formula* (4) *is equivalent to the solution obtained by the standard second order discretization scheme on uniform grid*

$$\begin{gathered} -\frac{k_{i+1/2}u_{i+1} - (k_{i+1/2} + k_{i-1/2})u_i + k_{i-1/2}u_{i-1}}{h^2} = f(x_i), \quad i = 1, \ldots, n, \\ u_0 = u_{n+1} = 0, \quad h = \frac{1}{n+1}. \end{gathered} \tag{6}$$

Proof.

$$\text{Let} \quad B = h \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & \ldots & \ldots & 1 \end{pmatrix}, \quad \text{then} \quad B^{-1} = \frac{1}{h} \begin{pmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix},$$

as it is easy to check that $BB^{-1} = I$. Let us denote by u_x vector of approximate derivatives

$$u_x \equiv \left(\frac{u_1 - u_0}{h}, \frac{u_2 - u_1}{h}, \ldots, \frac{u_{n+1} - u_n}{h} \right)^T,$$

and $u = (u_1, \ldots, u_{n+1})$. Due to the fact that $u_0 = 0$ we get

$$u_x = B^{-1}u.$$

Due to (6)

$$B^{-T}DB^{-1}u = \left(f(x_1), \ldots, f(x_n), k_{n+1/2}\frac{u_{n+1} - u_n}{h^2} \right)^T, \tag{7}$$

where

$$D = \text{diag}(k_{1/2}, \ldots, k_{n+1/2}).$$

Using additional information that $u_{n+1} = 0$ we get

$$\left(h^{-2}k_{n+1/2}e_{n+1}e_n^T + B^{-T}DB^{-1}\right)u = f,$$

where e_i is zero vector with only one 1 in the i-th position and $f = (f(x_1), \ldots, f(x_n), 0)^T$.

Let us apply Sherman-Woodbury-Morrison formula

$$u = BD^{-1}B^T f - h^{-2}k_{n+1/2}\frac{BD^{-1}B^T e_{n+1}e_n^T BD^{-1}B^T f}{1 + h^{-2}k_{n+1/2}e_n^T BD^{-1}B^T e_{n+1}}. \tag{8}$$

To get (4) let us simplify the latter expression. First of all,

$$B^T e_{n+1} = he, \quad e_n^T B = h(1, \ldots, 1, 0),$$

therefore,

$$BD^{-1}B^T e_{n+1}e_n^T BD^{-1}B^T f = h^{-2}BD^{-1}e\,(1, \ldots, 1, 0)D^{-1}B^T f = h^{-2}BD^{-1}ee^T D^{-1}B^T f$$

Finally the denominator in (8) can be written as

$$1 + h^{-2} k_{n+1/2} e_n^T B D^{-1} B^T e_{n+1} = k_{n+1/2} \left(\frac{1}{k_{n+1/2}} + \frac{1}{k_{1/2}} + \cdots + \frac{1}{k_{n-1/2}} \right) = k_{n+1/2}\, e^T D^{-1} e,$$

As a result, we get formula (4)

$$u = BD^{-1}B^\top f - \frac{e^\top D^{-1} B^\top f}{e^\top D^{-1} e} BD^{-1} e.$$

□

Remark 2. *The important consequence of Theorem 1 is that solution obtained by the formula* (4) *converges to the exact solution with the second order.*

Remark 3. *From* (7) *it follows that if Dirichlet-Neumann boudary conditions are used, i.e.* $u_0 = 0$ *and* $u_{n+1} = u_n$*, then the formula* (4) *reads* $u = BD^{-1}B^\top f$.

4 QTT REPRESENTATION FOR THE ONE DIMENSIONAL CASE

The concept of the QTT looks as follows. Let $n = 2^d$, then the vector has 2^d unknowns. We treat this one-dimensional vector as a d-dimensional tensor of size $2 \times \ldots \times 2$. This tensor V is then approximated in the *tensor-train* (TT) format. A tensor $V(i_1, \ldots, i_d)$ is said to be in the TT-format, if

$$V(i_1, \ldots, i_d) = G_1(i_1) G_2(i_2) \ldots G_d(i_d),$$

where $G_k(i_k)$ is an $r_{k-1} \times r_k$ matrix for each fixed i_k, and $r_0 = r_d = 1$.

The main benefit of the QTT-format is that it leads to logarithmic complexity to represent the vector of unknowns, if the ranks r_k are bounded: we only need to store $\mathcal{O}(dr^2)$ parameters. For elliptic problems, the upper bounds of QTT-ranks were provided in [5] and extended to the highly oscillating case in [4]. The last case is the most practically interesting, since it is exactly the case when astronomically large grids are needed. In order to turn (4) into a computational formula, we need a tensor representation of the matrices and vectors involved. Linear operator acting on tensors from $R^{\otimes_{i=1}^d n_i}$ to $R^{\otimes_{i=1}^d n_i}$, and is naturally represented as a $2d$ tensor

$$A(i_1, \ldots, i_d; j_1, \ldots, j_d).$$

Such linear operator is said to be in the *TT-matrix format*, if

$$A(i_1, \ldots, i_d; j_1, \ldots, j_d) = A_1(i_1, j_1) \ldots A_d(i_d, j_d).$$

For $r_k = 1$ this boils down to the Kronecker product of 2×2 matrices. The product of two TT-matrices is also a TT-matrix with ranks bounded by the product of the ranks of the terms, thus it is only necessary to put the matrices B and D^{-1} into the QTT-format.

Lemma 4. *The matrix* B *defined by* (5) *can be exactly represented in the QTT-format with QTT-ranks equal to* 2.

Lemma 5. *Let* d *be a vector with* 2^d *elements in the QTT-format with QTT-ranks* r_k*. Then, the matrix*

$$D = \mathrm{diag}(d),$$

can be represented in the QTT-format with QTT-ranks r_k.

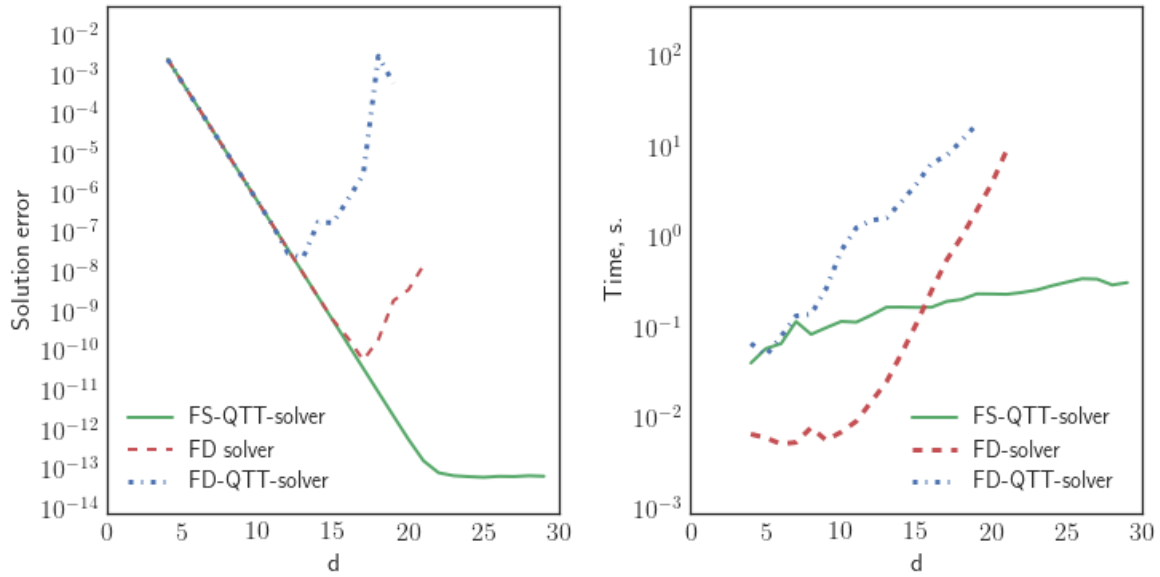

Figure 1: Error of calculated solution u (on the left plot) and total calculation time (on the right plot) w.r.t. the mesh size factor d (total number of grid nodes is 2^d) for the model PDE with known analytic solution. Results are presented for three different solvers, that are described in the text.

Thus, the most difficult task is to put the vector of values of the function k^{-1} to the QTT-format. For many practically interesting cases, the QTT-ranks are bounded [6, 7, 8]. To get such approximation the cross approximation algorithm is the method of choice [9], which allows to recover the approximation by adaptively sampling $\mathcal{O}(dnr^2)$ points. The right-hand also has to be put into the QTT-format using the same cross approximation procedure.

Provided that both f and k^{-1} are represented in the QTT format, it is easy to find bounds on rank of the solution.

Lemma 6. *Let f and k^{-1} has maximal ranks r_f and $r_{k^{-1}}$ correspondingly. In this case maximal rank r_u of the solution satisfies*

$$r_u \leq 2r_{k^{-1}}(2r_f + 1)$$

Proof. The proof immediately follows from the fact that the bound on rank of matrix-vector product is product of ranks. In our case

$$\text{rank}(BD^{-1}B^T u) \leq 2 \cdot r_{k^{-1}} \cdot 2 \cdot r_f, \quad \text{rank}(BD^{-1}e) \leq 2 \cdot r_{k^{-1}}.$$

□

5 NUMERICAL EXPERIMENTS

In this section we illustrate the theoretical results presented above with numerical experiments. Firstly, we consider a PDE with known analytic solution for validation of the developed solver (denoted hereinafter as finite sum QTT-solver or FS-QTT-solver). After that we consider a more complicated case of multiscale PDE. Special analytic form of multiscale PDE coefficients makes it possible to construct exact homogenized solution and first order correction, hence we can check the accuracy of numerical computations result in the terms of energy.

We compared calculation results obtained by FS-QTT-solver with the results of two solvers based on a finite difference discretization scheme. The first of them (denoted hereinafter as

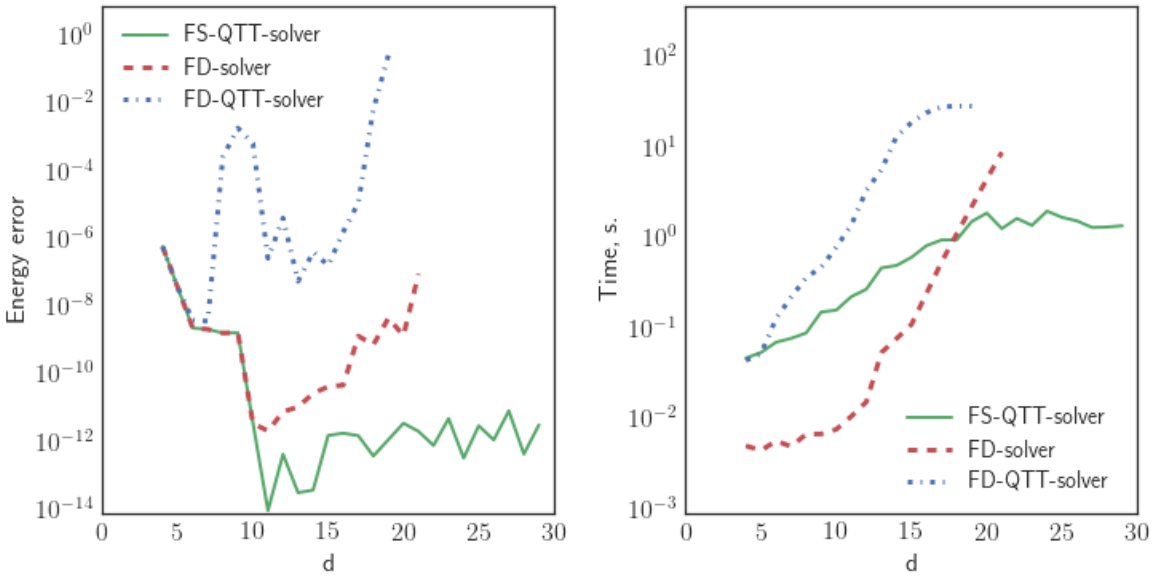

Figure 2: Error of the energy $(D\nabla u, \nabla u)$ (on the left plot) and total calculation time (on the right plot) w.r.t. the mesh size factor d (total number of grid nodes is 2^d) for the multiscale PDE with scale parameter $\epsilon = 10^{-4}$. Results are presented for three different solvers, that are described in the text.

FD-solver) is constructed in sparse format, and hence can operate only with moderate grids. We also construct a QTT version of the FD-solver (denoted hereinafter as FD-QTT-solver).

5.1 Method validation for PDE with known analytic solution

First we consider a PDE with homogeneous Dirichlet boundary conditions:

$$-\left(k(x)u'(x)\right)' = f(x), \quad x \in [0,1], \quad u(0) = u(1) = 0, \tag{9}$$

with $k(x) = 1 + x$ and $f(x) = \pi^2(1+x)\sin(\pi x) - \pi\cos(\pi x)$. This problem has exact analytic solution of the form

$$u(x) = \sin(\pi x). \tag{10}$$

Then for the numerical solution $\hat{u_h}$ on a uniform grid with step h we can calculate an error:

$$E_h^{(1)} = \frac{||\hat{u_h} - u_h||_2}{||u_h||_2},$$

where u_h is the exact solution (10) discretized on the same grid.

The dependence of $E_h^{(1)}$ and the total calculation time on grid size h for FS-QTT, FD and FD-QTT-solver is presented in Figure 1. As follows from the results, all solvers have the same accuracy for small grids ($d < 10$). FD-QTT-solver and FD-solver as anticipated become unstable for finer grids and the second order convergence for larger d is remained only for FS-QTT-solver. At the same time FS-QTT-solver works faster than FD-solver for grids with $d > 15$ as time scales linearly with d.

5.2 Multiscale problem

Here we consider the multiscale case of equation (9) with $f = -1$ and two-scale coefficient k^ϵ of the form

$$k(x,y) = k_0(x)k_1(y), \quad k_0(x) = 1 + x, \quad k_1(y) = \frac{2}{3}(1 + cos^2(2\pi y)).$$

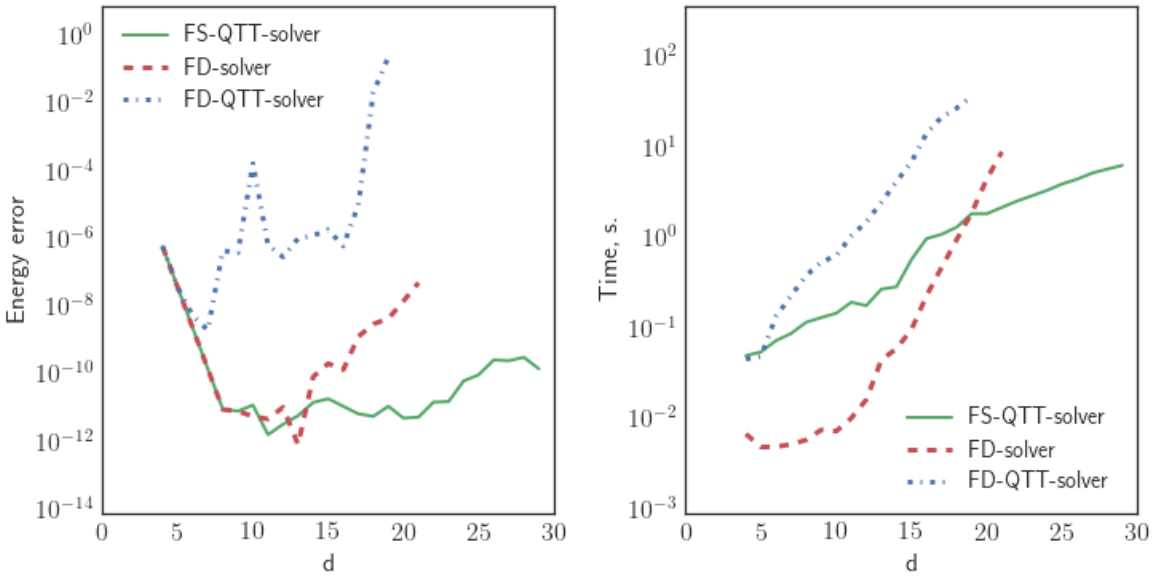

Figure 3: Error of the energy $(D\nabla u, \nabla u)$ (on the left plot) and total calculation time (on the right plot) w.r.t. the mesh size factor d (total number of grid nodes is 2^d) for the multiscale PDE with scale parameter $\epsilon = 10^{-6}$. Results are presented for three different solvers, that are described in the text.

that is Y-periodic function for

$$y = \frac{x}{\epsilon} \in Y = (0, 1).$$

The parameter $\epsilon > 0$ stands for some small scale in the problem.

It is a classical result of homogenization theory [10] that for small ϵ, u^ϵ and ∇u^ϵ can be approximated by

$$u^\epsilon(x) \approx u_0(x) + \epsilon\xi(y)\nabla_x u_0,$$

and

$$\nabla u^\epsilon(x) \approx \nabla_x u_0(x) + \nabla_y \xi(y)\nabla_x u_0,$$

where $y = x/\epsilon$ and ξ is the Y-periodic solution of the so-called cell problem:

$$-\left(k^\epsilon(x, y)\xi'(y)\right)'_y = \left(k^\epsilon(x, y)\right)'_y. \tag{11}$$

The considered two-scale limiting equation has the exact homogenized solution

$$u_0(x) = \frac{3}{2\sqrt{2}}\left(x - \frac{\log(1 + x)}{\log 2}\right),$$

and ξ has a form

$$\xi = \left(\frac{1}{2\pi}\tan^{-1}\left(\frac{\tan(2\pi y)}{\sqrt{2}}\right) - y + C\right). \tag{12}$$

and is determined within an additive constant C for fixed x.

According to (4) FS-QTT-solver computes both the solution of the equation $\hat{u}$ and it's derivative $\hat{u_x}$, hence we can construct an energy functional: $(D\hat{u_x}, \hat{u_x})$, where $(\cdot, \cdot)$ is a scalar product, and compare its value with the same functional for ∇u^ϵ from (11). FD-solver and FD-QTT-solver calculate only solution $\hat{u}$ of the PDE, but for comparison purposes we also need to construct approximation of $\hat{u_x}$. We do it by applying finite difference operator for the obtained

solution $\hat{u}$. Then we can calculate an error:

$$E_h^2 = \frac{|(D\hat{u_x}, \hat{u_x}) - (D\nabla u^\epsilon, \nabla u^\epsilon)|}{(D\nabla u^\epsilon, \nabla u^\epsilon)}.$$

In the case of QTT-based solvers the calculations may be performed for a huge grid sizes and the reference functions (12) and (11) must be constructed on the same grid for comparison. We use a cross-approximation method for this purpose with accuracy two orders of magnitude greater than the one that was used for the numerical solution of the equation and calculate both error norms in the TT-format.

The dependence of $E_h^{(2)}$ and the total calculation time on grid size h for FS-QTT, FD and FD-QTT-solver for scale parameter values $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$ are presented in Figure 2 and Figure 3 respectively. As follows from the results, FS-QTT-solver outperforms both FD-QTT-solver and FD-solver in accuracy and in calculation time for fine grids.

6 CONCLUSIONS

In this paper we proposed explicit formula that resolves the problem with accuracy on very fine grids. We showed how to use it in the QTT format. We also proved that in exact arithmetics this formula is equivalent to the second-order finite discretization. Numerical experiments illustrated efficiency of the proposed formula.

ACKNOWLEDGEMENT

The work was supported by the Ministry of Education and Science of Russian Federation, Grant Agreement no. 14.618.21.0004, the unique project identifier RFMEFI61815X0004.

REFERENCES

[1] V. Kazeev, I. Oseledets, M. Rakhuba, and Ch. Schwab. Qtt-fe approximation for multiscale problems. *Research Report, Seminar for Applied Mathematics*, 2015.

[2] Ivan V Oseledets. Approximation of 2ˆd\times2ˆd matrices using tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2130–2145, 2010.

[3] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[4] Vladimir Kazeev, Ivan Oseledets, Maxim Rakhuba, and Christoph Schwab. Qtt-finite-element approximation for multiscale problems. Technical Report 2016-06, Seminar for Applied Mathematics, ETH Zürich, 2016.

[5] Vladimir Kazeev and Ch Schwab. Quantized tensor-structured finite elements for second-order elliptic pdes in two dimensions. Technical report, SAM research report 2015-24, ETH Zürich, 2015.

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

[7] B. N. Khoromskij. $\mathcal{O}(d \log n)$–Quantics approximation of N–d tensors in high-dimensional numerical modeling. *Constr. Approx.*, 34(2):257–280, 2011.

[8] L. Grasedyck. Polynomial approximation in hierarchical Tucker format by vector-tensorization. DFG-SPP1324 Preprint 43, Philipps-Univ., Marburg, 2010.

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

[10] G Papanicolau, A Bensoussan, and J-L Lions. *Asymptotic analysis for periodic structures*. Elsevier, 1978.