Scientific Visualization, 2023, volume 15, number 4, pages 12 - 23, DOI: 10.26583/sv.15.4.02

On Application of Canonical Decomposition for the Visualization of Results of Multiparameter Computations

A.K. Alekseev[1,A,B], A.E. Bondarev[2,A], Yu. S. Pyatakova[3,B]

[A] Keldysh Institute of Applied Mathematics RAS
[B] RSC Energia, Korolev, Russia

[1] ORCID: 0000-0001-8317-8688, aleksey.k.alekseev@gmail.com
[2] ORCID: 0000-0003-3681-5212, bond@keldysh.ru
[3] ORCID: 0000-0002-8055-7807 , yuliya.pyatakova@rsce.ru

Abstract

The approximation of the tensor appearing at a discretization of the multidimensional function is considered from the viewpoint of storing and treating of the results of parametric computations obtained in computational aerogasdynamics. The new algorithm for the computation of the canonical decomposition using gradient descent and approximately decomposable goal functional is described.

This algorithm applies the random set of points on the hyperplane orthogonal to the computed core of the canonical decomposition ("umbrella") that ensures its flexible application for an approximation of the tensors with a priori unknown rank and may be naturally transferred on such tensor decomposition as the tensor train. The results of the numerical tests are presented for the model six-dimensional functions and for an ensemble of the numerical solutions for the two-dimensional Euler equations. These equations describe the flow of the compressible gas with two crossing shock waves. The Mach number and angles of the flow deflection serve as the flow parameters. The results are provided for the dimensionality 3 (simple numerical solution) and 4 (the ensemble of the numerical solutions in dependence on the Mach number).

Keywords: canonical decomposition, parametric calculations, computational aerogas dynamics, gradient optimization, sliding regularization.

Introduction

The approximation of the multidimensional functions is of interest in a lot of applications such as quantum mechanics, solving kinetic equations of the Boltzmann type, parametric solving of the aerogasdynamic problems.

Consider function *f(x,y,z,u,v,w)* determined in the domain $\Omega \subset R^6$ and corresponding the Boltzmann probability density as the example. It is necessary to store 10^{12} numbers for the grid containing 100 nodes along each coordinate. For the 64-bit number this statement requires the operative memory 64 TB that corresponds to parameters of the most powerful modern supercomputers. This circumstance hinders the practical applications of the Boltzmann equation both from the viewpoint cost and the time of computations.

We consider the set of numerical solutions for parameter CFD problems as another example. It is a set of functions $f_i(x, y, z, \vartheta_1 \ldots \vartheta_q)$, where $i = 1, \ldots, p$ corresponds to the flow variables (density, velocity components, energy), and $(\vartheta_1 \ldots \vartheta_q) \subset \Omega_q \subset R^q$ corresponds to the flow parameters (Mach, Reynolds numbers, angles of attack, etc.). These functions are defined in five-seven-dimensional spaces for the widespread practical applications that implies the use of the computational resources similar to above mentioned ones.

From the viewpoint of the practical applications, it is desirable to solve these problems with the computational costs close to resources of the standard personal computer.

The approximation of functions in the multidimensional space is very computationally difficult problem due to exponential growths of required operational memory at increase of dimensionality ("curse of the dimensionality"). The application of the tensor forms of the multidimensional problems and their approximation by the tensor decompositions [1,2,3] is one of the ways to overcome these difficulties. The present paper id addressed to this problem.

1. Canonical decomposition

Canonical decomposition $A = \sum_{1}^{r} Q_r^1 \otimes Q_r^2 \otimes ... \otimes Q_r^d$ [1,2] is the most fundamental tensor decomposition since it determines the rank of the tensor. Herein $\otimes$ is the outer (tensor) product of vectors having the index form $A_{ij} = a_i \otimes b_j = a_i b_j$. The canonical decomposition in the index form is written as

$$A_{i_1,i_2...i_d} = \sum_{\alpha=1}^{r} Q^1_{\alpha,i_1} Q^2_{\alpha,i_2} ... Q^d_{\alpha,i_d} . \quad (1)$$

This expression is unique with account the permutation and scaling [4,5].

The memory required in order to store this approximation of the tensor is about $\sim rnd$ (where d is the dimension of the space, n is the number of nodes over one of directions, r is the tensor rank) (instead the total memory necessary to store the tensor n^d). The time for the single node calculation $\sim rd$. The canonical decomposition provides the tremendous compression of the data (number of grid nodes used for function approximation in R^d), that is to say $n^d \to d \cdot n \cdot r$.

The problem for the determination of cores $\{Q^1_{\alpha,i},...,Q^d_{\alpha,k}\}$ in the variational form has the appearance

$$Q^n_\beta = \arg\min_{Q^n_\beta} \left\| A - \sum_{1}^{r} Q^1_\alpha \otimes Q^2_\alpha \otimes ... \otimes Q^d_\alpha \right\| . \quad (2)$$

The tensor rank r is the key value at application of the canonical decomposition. In accordance with [1,4] this value is not computable due to the ill-posedness of the following statement

$$r = \arg\min_{r} \left\| A - \sum_{\alpha=1}^{r} Q^1_\alpha \otimes Q^2_\alpha \otimes ... \otimes Q^d_\alpha \right\| . \quad (3)$$

The applications of the canonical decomposition suffers from instabilities and requires a regularization [1, 4].

2. Methods for Calculation of the Tensor Decompositions

Methods for the calculation of the tensor decompositions may be with certain tolerance subdivided by two subclasses: linear algebra based methods, for example [3] and variational methods [5] (which are subdivided by the direct and iterative ones).

The linear algebra based methods significantly depend on the matrization of the tensors, singular decomposition and contain a lot of interesting and original algorithms enabling to perform operations directly over cores without using the approximated functions.

Variational statements usually are based on alternating least squares (ALS) [6,7], and, as a rule, also apply the matrization of tensors. Commonly, ALS is realized for the canonical decompositions via the Khatri Rao product ($\odot$) [6,7]. Unfortunately, the matrization of the tensor does not provide its compression and does not relieve the curse of dimensionality.

However, within the frames of the variational approach, one may construct algorithms that do not apply the matrization of tensors. Herein, we consider an optimization method including some elements of the alternating least squares and the stochastic gradient descent. The application of the special set ("umbrella"), which is used in the optimization, is the specific feature of the method. This set enables to decompose the goal functional in the sum of the practically independent functionals. Another feature is the application of the Tikhonov regularization of the zero order at the estimation of cores in the gliding over range option. At such approach to the regularization, the components of cores are affected by the damping that is proportional to the index of core layer rank.

3. The gradient optimization on the stochastic "umbrella"

Formally, the quality of the approximation of a function by the canonical decomposition at every step of the global iteration may be estimated using the following discrepancy (written for 6D).

$$\varepsilon = \sum_{i,j,k,l,m,p} \{\sum_{\alpha=1}^{r} Q^x(\alpha,i)\cdot Q^y(\alpha,j)\cdot Q^z(\alpha,k)\cdot Q^u(\alpha,l)\cdot Q^v(\alpha,m)\cdot Q^w(\alpha,p) - \tilde{J}_{\ldots} \quad . \quad (4)$$

Herein, all nodes of the tensor are used similarly to ALS with the application of the Khatri-Rao product. Unfortunately, for the considered dimensions, this functional can not be computed (at least, at usual personal computers) due to tremendous needs for computation time. We numerically estimated its value using Monte-Carlo method in the form

$$\varepsilon_{MC} = \frac{1}{2Mc}\sum_{s=1}^{s=MC} \{\sum_{\alpha} Q^x(\alpha,i)\cdot Q^y(\alpha,j)\cdot Q^z(\alpha,k)\cdot Q^u(\alpha,l)\cdot Q^v(\alpha,m)\cdot Q^w(\alpha,p) - \tilde{J}_{\ldots} \quad , \quad (5)$$

where at every step of summation s every index from i,j,k,l,m,p was chosen as the random uniformly distributed number. In the result, we compute the averaged over the ensemble sum of the squares of approximation error. The number of tries in the ensemble is in the range $MC = 1000 \div 100000$, where the result relatively weakly changes at the variation of the number of tries.

Some alternative is necessary since the functional (4) is practically noncomputable and functional (5) may be applied to the calculation of cores only in the frame of the Monte-Carlo method. As such alternative, herein, we apply the special form of the goal functional, which does not require the great computational efforts and enables to obtain expressions for the gradient of the goal functional over cores, which are suitable for the gradient based optimization. In the contrast to the standard approach based on the Khatri Rao product, the considered algorithm is not restricted by the dimensionality (does not applies tensor matrization or the total set of the tensor nodes) and is significantly simpler algorithmically.

The special choice of the ensemble of points used in the optimization is the basic feature of this algorithm. These points are randomly selected on the hyperplane that is orthogonal to the coordinate corresponding to the core (on "umbrella", Fig. 1).

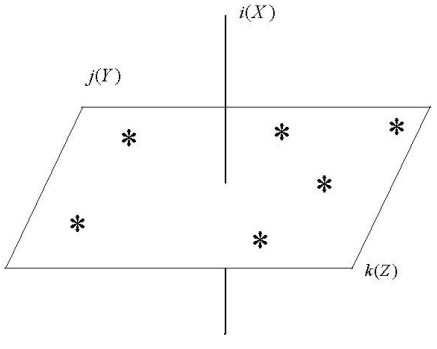

Fig. 1. The ensemble of points used in the optimization ("umbrella")

The cores $Q^x(\alpha,i), Q^y(\alpha,j), Q^z(\alpha,k)...$ are obtained by the consequent run.

Let's consider the algorithm on the example on the core over x at point i: $Q^x(\alpha,i), \alpha = 1...r$ and corresponding discrepancy $\varepsilon_{x,i}(Q^x(*,i))$, (other components of the core $Q^x(\alpha,i)$ are calculated consequently over i)

$$\varepsilon_{x,i}(Q^x) = 1/2 \sum_{e=1...Lens} \{(\sum_{\alpha} C(\alpha)\cdot Q^x(\alpha,i)\cdot Q^y(\alpha,j_e^i)\cdot Q^z(\alpha,k_e^i)... - \tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i} \cdot (\sum_{\beta} C(\beta)\cdot Q^x(\beta,i)\cdot Q^y(\beta,j_e^i)\cdot Q^z(\beta,k_e^i)... - \tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i} \quad . \tag{6}$$

The points of ensemble $j_e^i, k_e^i, l_e^i, m_e^i, p_e^i$ are numbered by the index $e = 1...Lens$ (i corresponds to the node where the "umbrella" opens. They are located on the hyperplane, which is orthogonal to x (at point i) and are selected randomly.

Herein $\tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i}$ is the exact value of the function at point $i, j_e^i, k_e^i, l_e^i, m_e^i, p_e^i$.

At computations, the dimension of the ensemble is much greater the preliminary estimate of the tensor rank $Lens >> r$. The enhancement of the ensemble dimension increases the computational costs for the discrepancy and its gradient estimation.

Let's disturb this core by $\Delta Q^x(\beta,i)$. Corresponding variation of discrepancy has the appearance:

$$\Delta\varepsilon_{x,i}(Q^x) = \sum_{e=1...Lens} \{(\sum_{\alpha} C(\alpha)\cdot Q^x(\alpha,i)\cdot Q^y(\alpha,j_e^i)\cdot Q^z(\alpha,k_e^i)... - \tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i} \cdot (\sum_{\beta} C(\beta)\cdot \Delta Q^x(\beta,i)\cdot Q^y(\beta,j_e^i)\cdot Q^z(\beta,k_e^i)... - \tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i} \quad . \tag{7}$$

Accordingly, the gradient of the discrepancy has the form:

$$\partial\varepsilon_{x,i} / \partial Q^x(\beta,i) = \sum_{e=1...Lens} \{(\sum_{\alpha} C(\alpha)\cdot Q^x(\alpha,i)\cdot Q^y(\alpha,j_e^i)\cdot Q^z(\alpha,k_e^i)... - \tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i} \cdot C(\beta)\cdot Q^y(\beta,j_e^i)\cdot Q^z(\beta,k_e^i)...\} . \tag{8}$$

For the cross derivatives (over other cores, herein, for example, over $Q^y(\beta,j)$) in ensemble points e, the corresponding expression has the appearance:

$$\partial\varepsilon_{x,i} / \partial Q^y(\beta,j) = \{(\sum_{\alpha} C(\alpha)\cdot Q^x(\alpha,i)\cdot Q^y(\alpha,j_e^i)\cdot Q^z(\alpha,k_e^i)... - \tilde{j}_{j_e^i,k_e^i,l_e^i,m_e^i,p_e^i} \cdot C(\beta)\cdot Q^x(\beta,i)\cdot Q^z(\beta,k_e^i)...\} . \tag{9}$$

at $j = j_e^i$ (otherwise- 0). It is important that the summation over the ensemble disappears. By this reason, such terms are approximately by $1/L_{ens}$ less the main one (8). Due to this circumstance we neglect these terms that enables to perform the optimization only over $Q^x(\beta,i)$.

Thus, instead single global discrepancy (used in computations for check)

$$\varepsilon_{\Sigma} = \{\sum_i \varepsilon_{x,i} + \sum_j \varepsilon_{y,j} + ...\} / (6N\cdot L_{ens}) \tag{10}$$

we optimize $6N$ separate discrepancies (for simplicity, we assume that there is N nodes along each coordinate)

$$\varepsilon_{x,i}; \varepsilon_{y,j} .... \tag{11}$$

The components of the cores are determined using consequent optimization of discrepancies over the number of the core and the number of the node on the core

$$Q^c(\beta,i) = \arg\min \varepsilon_{c,i}(Q^c(\beta,i)) \tag{12}$$

The gradient based steepest descent is used for the optimization, for $Q^x(\beta,i)$ it has the appearance:

$$\{Q^x(\beta,i)\}^{n+1} = \{Q^x(\beta,i)\}^n - \tau\nabla_{x,i,\beta}\varepsilon\,. \qquad (13)$$

The general structure of the algorithm has the form (circle over cores and their nodes):

Algorithm 1. The search for the cores of the canonical decomposition using steepest descent

Input
Initialize
for $c = 1...d$ (X,Y,Z,U,V,W) ***do ! circle over coordinates***
for i=$1...N_c$ ***do ! circle over nodes along core***
The search of the minimum is realized by the method of the steepest descent
for $\beta = 1...r$ ***do ! circle over index of rank***
$(Q^c(\beta,i))^{n+1} = (Q^c(\beta,i))^n - \tau_{c,i}\cdot\partial\varepsilon_{c,i}/\partial Q^c(\beta,i)$
end for
output

Such primitive structure of the algorithm may be realized because the gradients are approximately decomposable:

$$\nabla\varepsilon_\Sigma = \{\nabla_{Q_i^x}\varepsilon_{x,i} + \nabla_{Q_i^x}\varepsilon_{y,j} + ... + \nabla_{Q_j^y}\varepsilon_{x,i} + \nabla_{Q_j^y}\varepsilon_{y,j} + ...\} = \{\nabla_{Q_i^x}\varepsilon_{x,i} + \nabla_{Q_j^y}\varepsilon_{y,j} + ...\} + O(1/L_{ens})\,. \qquad (14)$$

with the tolerance of $\sim O(1/L_{ens})$. The choice of the ensemble of points that are used for the estimation of the discrepancy functional is the reason of the decomposability.

3.1 The gliding regularization

In the paper the regularization gliding over the index of rank is applied, that is based on the zero order Tikhonov regularization [9] with the regularizing coefficient that increases along the rank index β as

$$\varepsilon_{c,i} + \gamma Q^c(\beta,i)^2\cdot(\beta-1)^2/2 \qquad (15)$$

Herein γ is the basic regularizing coefficient.

Corresponding iterations has the form:

$$(Q^c(\beta,i))^{n+1} = (Q^c(\beta,i))^n - \tau_{c,i}\cdot\{\partial\varepsilon_{c,i}/\partial Q^c(\beta,i) + \gamma\cdot Q^c(\beta,i)\cdot(\beta-1)^2 \qquad (16)$$

The general optimization is reduced to local consequent optimization over $\sim 6N$ discrepancies. The values ε_{MC} (5) and ε_Σ (10) are used for the control of the convergence.

The optimization of the cores components is terminated at $\varepsilon \le \varepsilon_1$, $\varepsilon_1 = 10^{-7} \div 10^{-9}$. The process of the determination of the canonical decomposition was broken at $\varepsilon_{MC} \le \varepsilon_2$, $\varepsilon_2 = 10^{-5} \div 10^{-7}$.

4. The results of the numerical experiments on the interpolation of the multidimensional functions

Let's consider the approximation of the six-dimensional function $\tilde{}$ (more strictly speaking the tensor $\tilde{}$, which corresponds the values of the function at nodes of the regular grid) by the canonical decomposition and the corresponding set of cores $Q^x(\alpha,i)\cdot Q^y(\alpha,j)\cdot Q^z(\alpha,k)\cdot Q^u(\alpha,l)\cdot Q^v(\alpha,m)\cdot Q^w(\alpha,p)$.

The grid containing 100 nodes along each coordinate was used in the numerical tests. Formally, the storage of $\tilde{}$ requires 10^{12} numbers on this grid that is closely nonrealistic both from the necessary memory and the computing time. As the illustration one should mark that, at the considered case, the memory necessary to store cores with the rank 100 (ra-

ther great) is $6 \cdot 100 \cdot 100 = 60000$ numbers. This illustrates the superhigh data compression ($\sim 10^7$) at the canonical decomposition.

At debugging the comparison of the numerical (obtained by the direct numerical differentiation) and analytical (obtained using the expression (8)) gradients was performed that demonstrates their practically perfect coincidence.

The results of computations provide enough stable and reproducible estimates of errors.

At the first stage the testing of the approximation of different functions using the canonical decomposition was performed. The following multidimensional functions were considered that are arranged in the order of the complexity increasing.

$$f = x + y \tag{17}$$

It is utterly simple function, however its rank is not known a priori. 30 iterations were used and the rank was varied from 1 to 10. The dependence of the discrepancy value (5) on the rank magnitude is presented in Table 1.

Table 1. The dependence of the discrepancy value (5) on the rank for the function (17).

rank	1	2	3	4	5	10
discrepancy	$1.46 \cdot 10^{-2}$	$7.13 \cdot 10^{-8}$	$1.68 \cdot 10^{-7}$	$2.22 \cdot 10^{-7}$	$4.39 \cdot 10^{-7}$	$2.00 \cdot 10^{-5}$

The rapid decreasing at reaching true value of the rank (herein, 2) and slow increasing at further growth of the rank is specific for the discrepancy. These results demonstrate that, in general, the discrepancy magnitude may serve as an indicator of the true rank of the tensor. As the rank increases the noise in the results grows. We consider this circumstance as an evidence of instabilities occurring due to the ill-posedness of the canonic decomposition [1,4].

Fig. 2 and 3 provide the true function (17) and its approximation at rank 2.

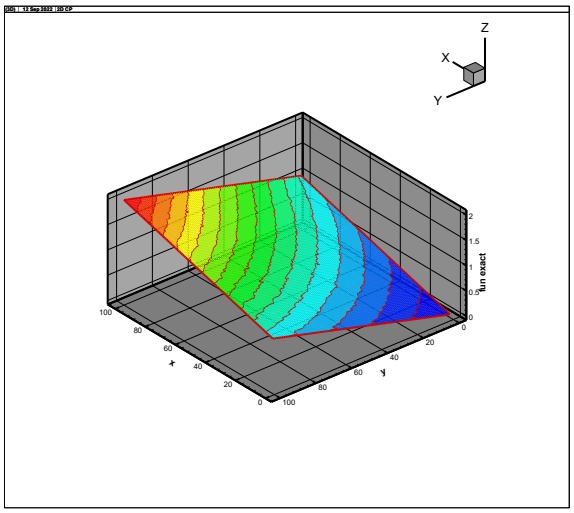

Fig. 2. True function

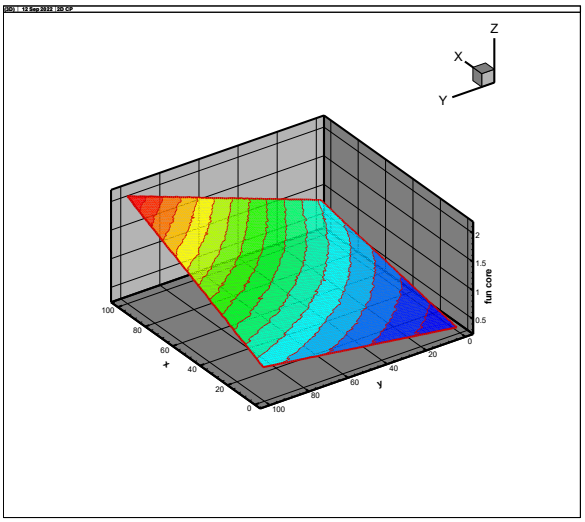

Fig. 3. Approximation via the canonical decomposition

Function (17) is two-dimensional, let's consider more complex true six-dimensional event assigned by a Gaussian and a sum of sines along all directions in a form

$$f = 5 \cdot \exp(-rad^2) + \sin(x/5) + \sin(y/5) + \sin(z/5) + \sin(u/5) + \sin(v/5) + \sin(w/5)$$
$$(rad = 0.001 \cdot ((ix - 50) + (iy - 50) + (iz - 50) + (iu - 50) + (iv - 50) + (iw - 50))). \tag{18}$$

Figs. 4 and 5 present results for the rank 10 and the ensemble of 1000 points in the plane x, y, other variables correspond to the middles of the intervals at grid dimension 100. The results of approximation practically can't be distinguished from the true function. At decreasing of the ensemble dimension the quality of the approximation deteriorates. In general, at approximation of enough complex functions, the ensemble should be significantly greater the

a priori estimate of the rank. Fortunately, the application of the zero order Tikhonov regularization in the gliding option enables to minimize the current rank and leaves the necessarity of a priori knowledge of the rank (or running over the ranks).).

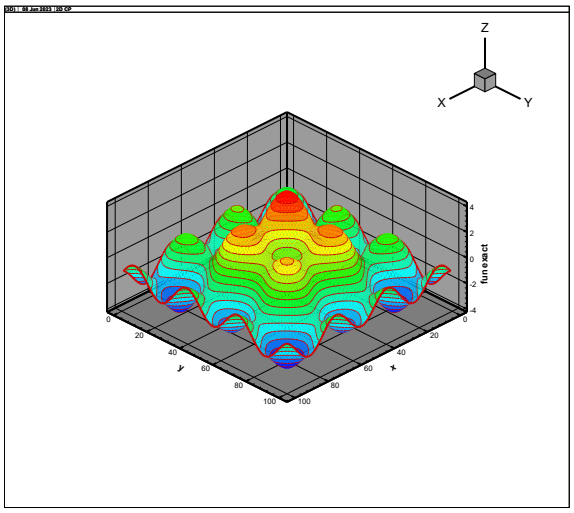

Fig. 4. Exact function (18)

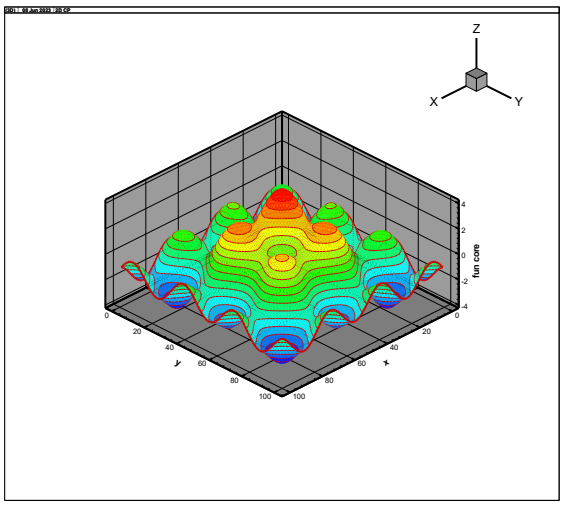

Fig. 5. Approximation of (18) by the canonical decomposition

The Fig. 6 provides the behaviour of the different convergence criteria in the dependence on the number of iterations. The discrepancies ε_{Σ} (10) (Eps), the global discrepancy estimated using Monte-Carlo method ε_{MC} (5) (Eps_MC) and the norm of the discrepancy gradient (grad norm) are provided in the logarithmic form.

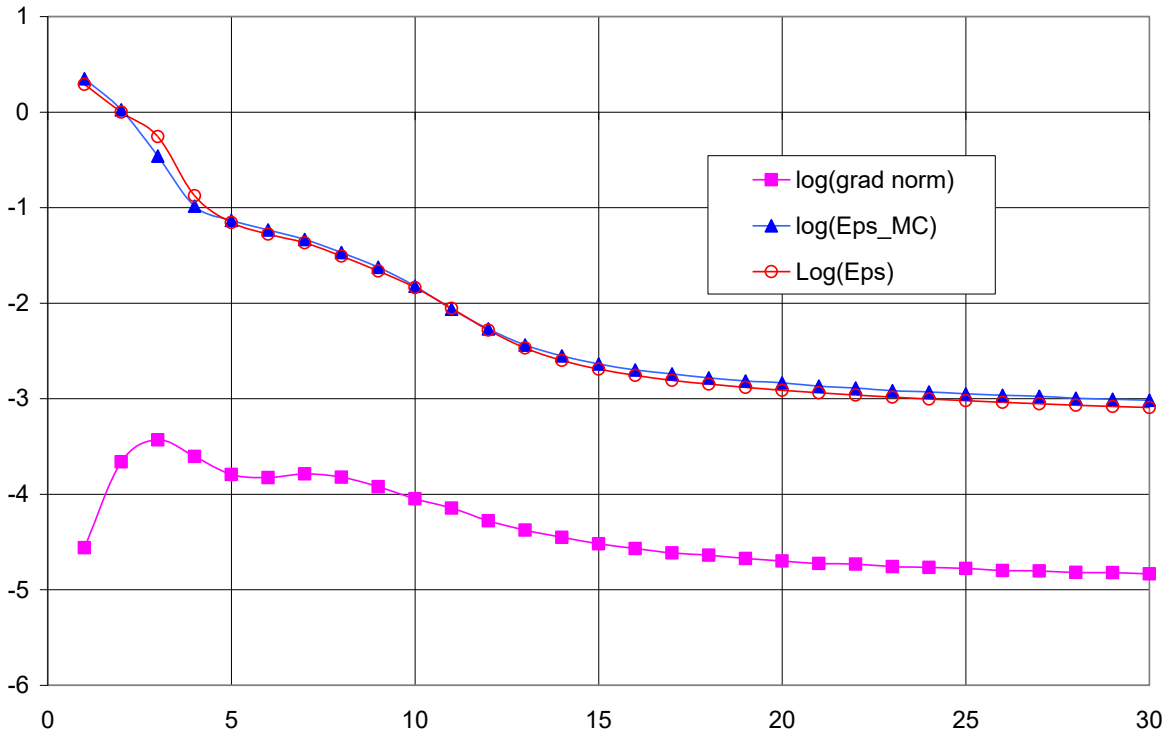

Fig. 6. Different criteria of the convergence in the dependence on the number of iterations

Discrepancy ε_{Σ} (10) (Eps) and the global discrepancy estimated using Monte-Carlo ε_{MC} (5) (Eps_MC) demonstrate similar behaviour suitable for the convergence check. The norm of

the gradient (grad norm) is not monotonic at the start iterations that restrict its applicability as the stopping criterion.

In general, the considered approach does not provide the monotonic minimization of the global criteria since the minimization is performed by points. However, the numerical tests demonstrate monotonic convergence for the global criteria. The considered algorithm provides significantly more rapid convergence if compare with one used by [10].

Really, in the numerical tests we calculate not the true rank of the function (tensor), but the approximation of the rank r_ε, providing the discrepancy ε. Fig. 7 demonstrates that the variant of the Tikhonov regularization, used in the paper, enables to decrease the effective rank by increasing regularizing parameter, however, the regularization qualitatively changes cores.

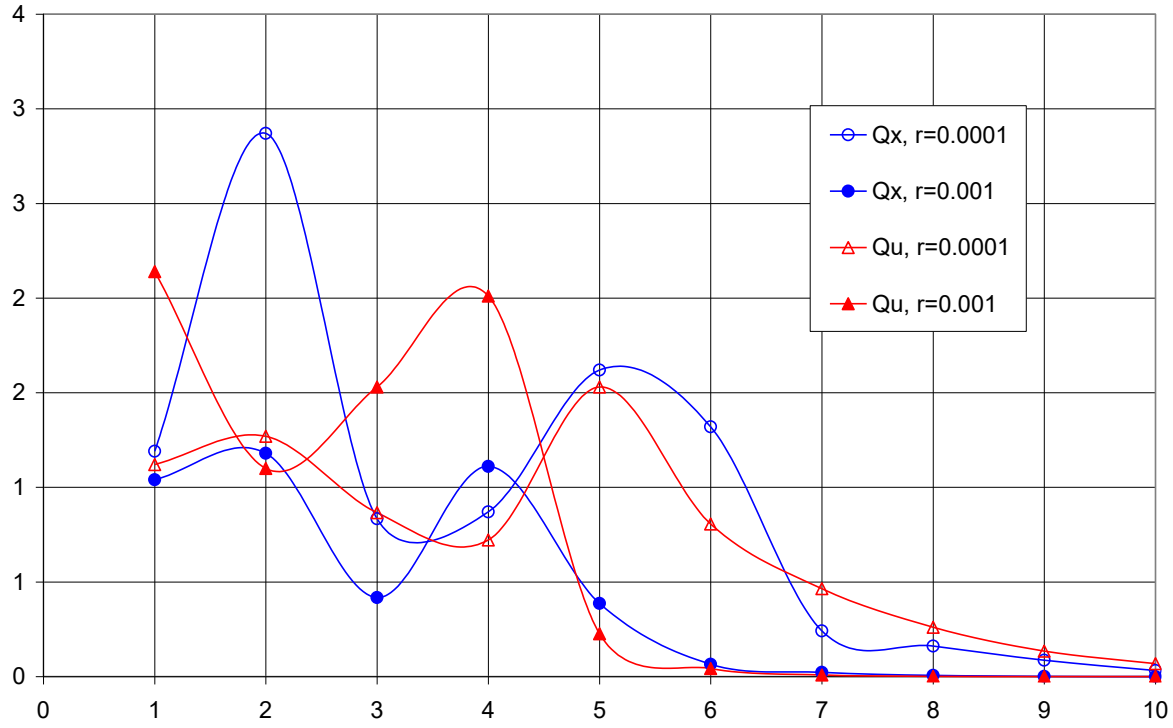

Fig. 7. The dependence of the discrepancy on the rank for the function (18)

5. The results of the numerical tests for the approximation of the results of the parametric calculations.

As an example, we consider the two dimensional steady flowfield in the parameter space of the dimension from one to three. It corresponds to the case of the interaction of two supersonic air flow occurring at the flow around two symmetric edges. The flow structure is described by the steady Mach and regular modes of the shock waves interaction. The Mach number $M \in (2,6)$ and flow deflection angles $(\theta_1, \theta_2) \in (15,30)$ are the parameters.

The ensemble of the solutions on the same spatial grid with different parameters (θ_1, θ_2, M) may be considered as the tensor of the order 6: $T_{p;i,j;k,n,m} = (\rho_{i,j;k,n,m}, u_{i,j;k,n,m}, v_{i,j;k,n,m}, e_{i,j;k,n,m})$ (p is the number of the gasdynamical variables, i, j are the indices of the coordinate nodes, k, n, m are the numbers of the parameters (θ_1, θ_2, M)).

Let's consider the formation of the canonical decomposition on such ensemble of solutions that enables the radical compression of the data.

We construct the canonical decomposition using the structure “umbrella”. One calculation of the flowfield (elementary “building block”) $T_{p;i,j;1,1,1}$ - it is complete completed tensor of the order 3.

One may relatively easy to form the tensor of the order 4 using the sequence of the tensors of the order 3 (by the transition from $T_{p;i,j;1,1,1}$ to $T_{p;i,j;2,1,1}$ and so on). It is enough laborious ($I_{ens} = 10 \div 100$ of the separate computations), but realistic task. However, already at the generation of the tensor of the order 5 the troubles appear. At the every layer one should to determine the tensor of the order 4 from the scratch. It is highly laborious from the computational standpoint and practically impossible for the tensor of the following rank. So, the additional problem of the tensor completion using incomplete data arises, which can be considered in the above described optimization statement with the special selection of the support ensemble of calculations.

Fig. 8 and 9 provides the results of the calculation of the flowfield (density) for $\theta_1 = 15^\circ$ (the deflection of the upper flow), $\theta_2 = 20^\circ$ (the deflection of the down flow), $M = 4$ and corresponding approximation by the canonical decomposition (rank 15, the number of point on the ensemble 2000). The tensor $T_{p;i,j}$ of the order 3 is used. The flow is directed from the right side. The zone of the high density past crossing shock waves is presented in the foreground.

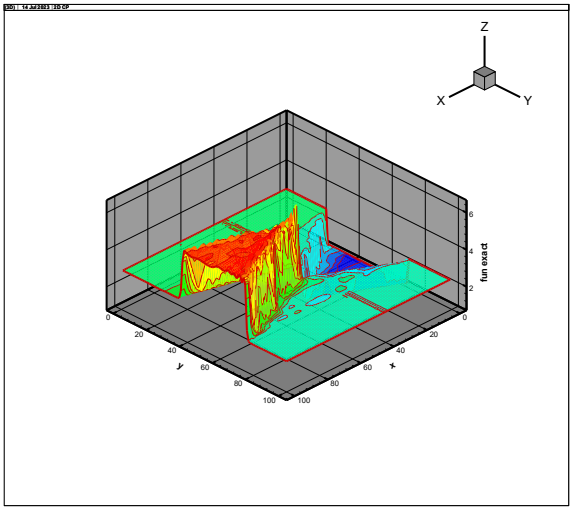

Fig. 8. Density field calculation

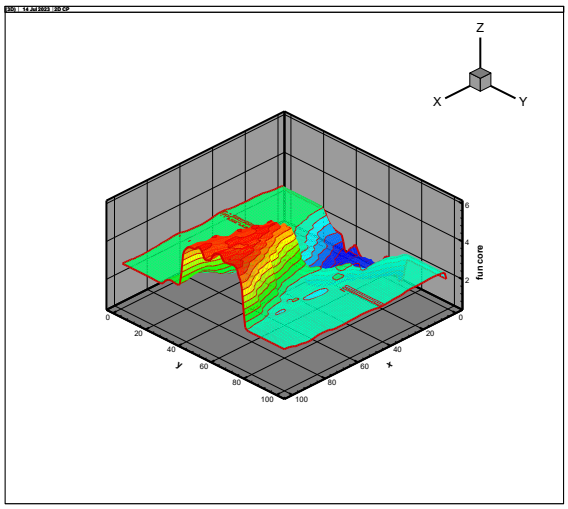

Fig. 9. The approximation of the density field using the canonical decomposition (third order tensor)

The results presented by Figs. 8 and 9 demonstrate the operationability of the approximation (compression) of the flowfields using the canonical decomposition.

Fig. 10 and 11 present the results of the flow-field computations (density) ($\theta_1 = 15^\circ$, $\theta_2 = 20^\circ$, $M = 3$) and corresponding approximation using canonical decomposition (rank 3, the number of the points of the ensemble -200). The tensor of the order 4: $T_{p;i,j,M}$ was used, with five layers on $M = (3.0; 3.5; 4.0; 4.5; 5.0)$.

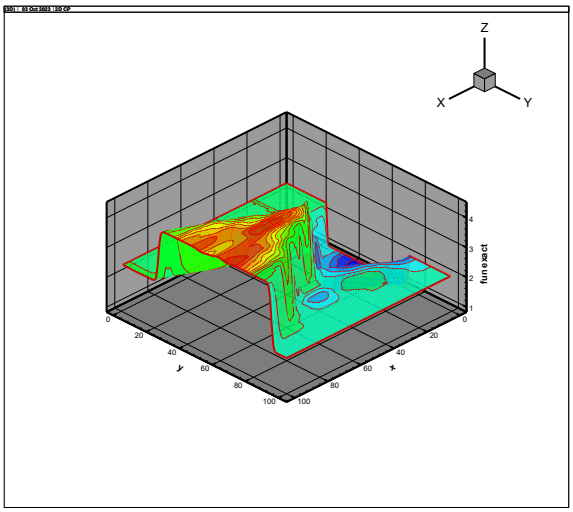

Fig. 10. Density field calculation

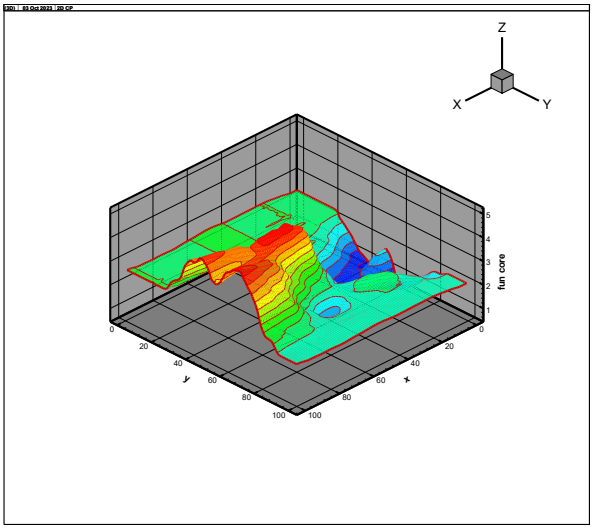

Fig. 11. The approximation of the density field using the canonical decomposition (fourth order tensor)

The results presented in Figs. 10 and 11 demonstrate the correctness of the approximation (compression) of the one parameter flowfield using the canonical decomposition. The observed oscillations of the approximation are possibly related with the ill-posedness of the determination of the canonical decomposition. There exists some hope that the quality of the solution may improve after the transition to the tensor train format.

6. Discussion

Due to the ill-posedness of the problem of the estimation of the cores of the canonical decomposition [4] the feasibility of the quadratic regularization of the zero order (Tikhonov [9]) with the gliding regularization is provided. It enables to compress the approximation of the rank r_ε, ensuring the discrepancy ε (to obtain the upper estimate of the rank).

Tensor decompositions are enough actively used for the visualization [11], since it rather easy enable to engender a computable model that approximate the difficult set of the data in the parameter space. Tensor decompositions (canonical decomposition, tensor train, hierarchical Tucker) are used for economic solving of the multidimensional problems such as the Boltzmann equation [5,13,14]. The tensor train format and the cross-approximation [13] are rather often used for these purposes. The above considered approach may be easily expanded to the tensor train format and has no restrictions on the computer resources specific for the cross-approximation (related with the matrization of the tensor). The applied herein canonical decomposition enables efficiently approximate and store the multidimensional functions.

The computational time for the operations with the functions in the six-dimensional space (at 100 nodes along every coordinates that formally requires to store and operate with 10^{12} numbers) is about 1-3 minutes on the personal computer (processor Intel I5, 2.66 GHz) at the required memory for cores store (maximally) about 10^5 numbers. Some hope exists that the transition to the tensor train format may enable to overcome part of the canonical decomposition drawbacks related to ill-posedness and instability.

Conclusion

The optimization algorithm for the determination of the cores in canonical decomposition is offered, which requires much less memory if compare with the standard methods using the matrization of the tensor and the Khatri-Rao product. The gliding over rank regularization is applied in the algorithm, which enables to reduce the effective rank of the approxi-

mation for the sake of the reduction of the norm of core layers with the higher indexes of ranks.

Numerical tests demonstrate that the application of the above described algorithm for the realization of the canonical decomposition enables to store and visualize functions in the multidimensional space with moderate needs for the memory and the computational time. This circumstance provides a special interest from the standpoint of analyzing and treating results of the calculations for the multiparametric problems of computational fluid dynamics.

References

1. W. Hackbusch. Tensor Spaces and Numerical Tensor Calculus. Springer, 2012.
2. H. Yorick, S. Willi-Hans, Matrix Calculus, Kronecker Product and Tensor Product: A Practical Approach to Linear Algebra, Multilinear Algebra and Tensor Calculus with Software I, WOS 2019
3. Oseledets I. V., Tensor- train decomposition, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317
4. V. D. Silva and L -H. Lim, Tensor rank and the ill-posedness of the best low rank approximation problem, SIAM J. Matrix Anal. Appl., 30(3) 1084-1127, 2008.
5. A. M. P. Boelens, D. Venturi, D. M. Tartakovsky, Parallel tensor methods for high-dimensional linear PDEs, J. Computat. Phys. 375 (2018) 519-539.
6. Shuangzhe Liu, Gaotz Trenkler, Hadamard, Khatri-Rao, Kronecker and other matrix products, Int. J. Information and system Scienses, 2008, Volume 4, Number 1, Pages 160-177
7. P. Comon, X. Luciani, and A.L.F. De Almeida, Tensor decompositions, alternating least squares and other tales, Journal of Chemometrics, vol. 23, no. 7-8, pp. 393–405, 2009. A. Uschmajew, Local convergence of the alternating least squares algorithm for canonical tensor approximation, SIAM J. Matrix Anal. Appl. 33 (2) (2012) 639–652,
8. S. Rabanser, O. Shchur, S. Gunnemann, Introduction to Tensor Decompositions and their Applications in Machine Learning, arXiv:1711.10781v1, 2017
9. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-Posed Problems. Winston and Sons, Washington DC (1977).
10. A.K. Alekseev, A.E. Bondarev, Yu.S. Pyatakova, On the visualization of multidimensional functions using canonical decomposition, Scientific Visualization, 2022, volume 14, number 3, pages 73 - 91
11. Yannis Panagakis, Jean Kossaifi, Grigorios G. Chrysos, James Oldfield, Mihalis A. Nicolaou, Anima Anandkumar and Stefanos Zafeiriou, Tensor Methods in Computer Vision and Deep Learning, arXiv:2107.03436v1, 2021
12. Oseledets I., Tyrtyshnikov E., TT-cross approximation for multidimensional arrays, Linear Algebra Appl., 432 (2010), pp. 70–88.
13. Arnout M. P. Boelens, Daniele Venturi, Daniel M. Tartakovsky, Tensor methods for the Boltzmann-BGK equation, arXiv:1911.04904v2 2020
14. A.V. Chikitkin, E.K. Kornev, V.A. Titarev, Numerical solution of the Boltzmann equation with S-model collision integral using tensor decompositions, arXiv:1912.04582v1 2019