Symbolic-Numerical Algorithms for Solving Elliptic Boundary-Value Problems Using Multivariate Simplex Lagrange Elements

A. A. Gusev[1(✉)], V. P. Gerdt[1,2], O. Chuluunbaatar[1,3], G. Chuluunbaatar[1,2], S. I. Vinitsky[1,2], V. L. Derbov[4], A. Góźdź[5], and P. M. Krassovitskiy[1,6]

[1] Joint Institute for Nuclear Research, Dubna, Russia
gooseff@jinr.ru
[2] RUDN University, 6 Miklukho-Maklaya St., Moscow 117198, Russia
[3] Institute of Mathematics, National University of Mongolia, Ulaanbaatar, Mongolia
[4] N.G. Chernyshevsky Saratov National Research State University, Saratov, Russia
[5] Institute of Physics, University of M. Curie–Skłodowska, Lublin, Poland
[6] Institute of Nuclear Physics, Almaty, Kazakhstan

Abstract. We propose new symbolic-numerical algorithms implemented in Maple-Fortran environment for solving the self-adjoint elliptic boundary-value problem in a d-dimensional polyhedral finite domain, using the high-accuracy finite element method with multivariate Lagrange elements in the simplexes. The high-order fully symmetric PI-type Gaussian quadratures with positive weights and no points outside the simplex are calculated by means of the new symbolic-numerical algorithms implemented in Maple. Quadrature rules up to order 8 on the simplexes with dimension $d = 3 - 6$ are presented. We demonstrate the efficiency of algorithms and programs by benchmark calculations of a low part of spectra of exactly solvable Helmholtz problems for a cube and a hypercube.

Keywords: Elliptic boundary-value problem · Finite element method
Multivariate simplex lagrange elements
High-order fully symmetric Gaussian quadratures
Helmholtz equation for cube and hypercube

1 Introduction

The progress of modern computing power offers more possibilities for setting and numerical solution of multidimensional elliptic boundary-value problems (BVPs) with high accuracy. 3D BVPs have wide applications in such areas as vibrating membrane, electromagnetic radiation, motion of thermal neutrons in the reactor, seismology, and acoustics, see, e.g., [4], while multidimensional BVPs have applications in nuclear physics, see, e.g., [7]. For this purpose, novel numerical

V. P. Gerdt et al. (Eds.): CASC 2018, LNCS 11077, pp. 197–213, 2018.
https://doi.org/10.1007/978-3-319-99639-4_14

methods of high accuracy order are being developed. When reducing the boundary value problem to an algebraic one in the finite element method (FEM) of the order p, one of the problems is the calculation of integrals on a finite element (we consider only simplicial finite elements) containing the products of two basis functions of Lagrange or Hermite interpolation polynomials of the order p by the coefficients for the unknown functions [5,9]. There are three possible ways to calculate the integrals:

(i) using analytical calculation, which is possible for a limited number of cases;
(ii) using quadrature formulas with products of two basic functions used as a weight function;
(iii) using quadrature formulas with a single weight function.

It is well known [20] that as a result of applying the pth order FEM to the solution of the discrete spectrum problem for the elliptic (Schrödinger) equation, the eigenfunction and the eigenvalue are determined with an accuracy of the order $p+1$ and $2p$ provided that all intermediate quantities are calculated with sufficient accuracy. It follows that for the realization of the FEM of the order p in the third case, the integrals must be computed at least with an accuracy of the order $2p$, depending on the problem considered. The most economical calculation of such integrals is achieved using the quadratures of Gaussian type. In the one-dimensional case, the nodes and the quadrature Gaussian weights are expressed analytically; in the two-, three- and four-dimensional case, the high-order quadrature formulas are determined numerically [2,6,8,10,17–19,21]. Note that for multidimensional integrals, numerous quadrature formulas of the Newton–Cotes and third-order Gaussian type are known, too (see Ref. [1]).

The paper presents a new method for constructing fully symmetric multidimensional Gaussian-type quadratures on a standard simplex. The main idea of the method is replacing the coordinates of nodes with their symmetric combinations obtained using the Vieta theorem, which simplifies the system of nonlinear algebraic equations. The construction of the desired systems of equations is performed analytically using an original algorithm implemented in Maple [13]. The derived systems up to the sixth order are solved using the built-in procedure PolynomialSystem, implementing the technique of Gröbner bases, and the systems of higher order are solved using the developed symbolic-numerical algorithm based on numerical methods, implemented in Maple-Fortran environment. We demonstrate the efficiency of algorithms and programs by benchmark calculations of the lower part of spectra in exactly solvable Helmholtz problems for a cube and a hypercube.

The paper is structured as follows. In Sects. 2 and 3, the FEM schemes and algorithms for solving the d-dimensional BVP are presented. In Sect. 4, the algorithms for constructing the d-dimensional fully symmetric Gaussian quadratures are presented. In Sect. 5, the benchmark calculations of the exactly solvable Helmholtz problems for the cube and hypercube are presented. In Conclusion, we discuss the results and perspectives.

2 Setting of the Problem

Consider a self-adjoint boundary-value problem for the elliptic differential equation of the second order:

$$(D-E)\Phi(z) \equiv \left(-\frac{1}{g_0(z)} \sum_{ij=1}^{d} \frac{\partial}{\partial z_i} g_{ij}(z) \frac{\partial}{\partial z_j} + V(z) - E \right) \Phi(z) = 0. \quad (1)$$

For the principal part coefficients of Eq. (1), the condition of uniform ellipticity holds in the bounded domain $z = (z_1, \ldots, z_d) \in \Omega$ of the Euclidean space $\mathcal{R}^d$, i.e., the constants $\mu > 0$, $\nu > 0$ exist such that $\mu\xi^2 \le \sum_{ij=1}^{d} g_{ij}(z)\xi_i\xi_j \le \nu\xi^2$, $\xi^2 = \sum_{i=1}^{d} \xi_i^2$, $\forall \xi_i \in \mathcal{R}$. The left-hand side of this inequality expresses the requirement of ellipticity, while the right-hand side expresses the boundedness of the coefficients $g_{ij}(z)$. It is also assumed that $g_0(z) > 0$, $g_{ji}(z) = g_{ij}(z)$ and $V(z)$ are real-valued functions, continuous together with their generalized derivatives to a given order in the domain $z \in \bar{\Omega} = \Omega \cup \partial\Omega$ with the piecewise continuous boundary $S = \partial\Omega$, which provides the existence of nontrivial solutions obeying the boundary conditions [5] of the first kind

$$\Phi(z)|_S = 0, \quad (2)$$

or the second kind

$$\frac{\partial\Phi(z)}{\partial n_D}\Big|_S = 0, \quad \frac{\partial\Phi(z)}{\partial n_D} = \sum_{ij=1}^{d} (\hat{n}, \hat{e}_i) g_{ij}(z) \frac{\partial\Phi(z)}{\partial z_j}, \quad (3)$$

where $\frac{\partial\Phi_m(z)}{\partial n_D}$ is the derivative along the conformal direction, $\hat{n}$ is the outer normal to the boundary of the domain $S = \partial\Omega$, $\hat{e}_i$ is the unit vector of $z = \sum_{i=1}^{d} \hat{e}_i z_i$, and $(\hat{n}, \hat{e}_i)$ is the scalar product in $\mathcal{R}^d$.

For a discrete spectrum problem, the functions $\Phi_m(z)$ from the Sobolev space $H_2^{s\ge1}(\Omega)$, $\Phi_m(z) \in H_2^{s\ge1}(\Omega)$, corresponding to the real eigenvalues E: $E_1 \le E_2 \le \ldots \le E_m \le \ldots$ satisfy the conditions of normalization and orthogonality

$$\langle \Phi_m(z) | \Phi_{m'}(z) \rangle = \int_\Omega dz g_0(z) \Phi_m(z) \Phi_{m'}(z) = \delta_{mm'}, \quad dz = dz_1 \ldots dz_d. \quad (4)$$

The FEM solution of the boundary-value problems (1)–(4) is reduced to the determination of stationary points of the variational functional [3,5]

$$\Xi(\Phi_m, E_m) \equiv \int_\Omega dz g_0(z) \Phi_m(z) \left(D - E_m\right) \Phi(z) = \Pi(\Phi_m, E_m), \quad (5)$$

where $\Pi(\Phi, E)$ is the symmetric quadratic functional

$$\Pi(\Phi, E) = \int_\Omega dz \left[\sum_{ij=1}^{d} g_{ij}(z) \frac{\partial\Phi(z)}{\partial z_i} \frac{\partial\Phi(z)}{\partial z_j} + g_0(z)\Phi(z)(V(z) - E)\Phi(z) \right].$$

3 FEM Calculation Scheme

In FEM, the domain $\Omega = \Omega_h(z) = \bigcup_{q=1}^{Q} \Delta_q$, specified as a polyhedral domain, is covered with finite elements, in the present case, the simplexes Δ_q with $d+1$ vertices $\hat{z}_i = (\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{id})$, $i = 0, \ldots, d$. Each edge of the simplex Δ_q is divided into p equal parts, and the families of parallel hyperplanes $H(i,k)$ are drawn, numbered with the integers $k = 0, \ldots, p$, starting from the corresponding face (see also [5]). The equation of the hyperplane is $H(i,k)$: $H(i;z) - k/p = 0$, where $H(i;z)$ is a linear function of z.

The node points of hyperplanes crossing A_r are enumerated with the sets of integers $[n_0, \ldots, n_d]$, $n_i \geq 0$, $n_0 + \ldots + n_d = p$, where n_i, $i = 0, 1, \ldots, d$ are the numbers of hyperplanes, parallel to the simplex face, not containing the ith vertex $\hat{z}_i = (\hat{z}_{i1}, \ldots \hat{z}_{id})$. The coordinates $\xi_r = (\xi_{r1}, \ldots, \xi_{rd})$ of the node point $A_r \in \Delta_q$ are calculated using the formula

$$(\xi_{r1}, \ldots, \xi_{rd}) = (\hat{z}_{01}, \ldots, \hat{z}_{0d})\frac{n_0}{p} + (\hat{z}_{11}, \ldots, \hat{z}_{1d})\frac{n_1}{p} + \ldots + (\hat{z}_{d1}, \ldots, \hat{z}_{dd})\frac{n_d}{p} \quad (6)$$

from the coordinates of the vertices $\hat{z}_j = (\hat{z}_{j1}, \ldots, \hat{z}_{jd})$. Then the Lagrange interpolation polynomials (LIP) $\varphi_r^p(z)$ are equal to one at the point A_r with the coordinates $\xi_r = (\xi_{r1}, \ldots, \xi_{rd})$, characterized by the numbers $[n_0, n_1, \ldots, n_d]$, and equal to zero at the remaining points $\xi_{r'}$, i.e., $\varphi_r^p(\xi_{r'}) = \delta_{rr'}$, have the form

$$\varphi_r^p(z) = \prod_{i=0}^{d} \prod_{n_i'=0}^{n_i - 1} \frac{H(i;z) - n_i'/p}{H(i;\xi_r) - n_i'/p}. \quad (7)$$

As shape functions in the simplex Δ_q we use the multivariate Lagrange interpolation polynomials $\varphi_l^p(z)$ of the order p that satisfy the condition $\varphi_l^p(x_{1l'}, x_{2l'}) = \delta_{ll'}$, i.e., equal 1 at one of the points A_l and zero at the other points. In this method, the piecewise polynomial functions $N_l^p(z)$ in the domain Ω are constructed by joining the shape functions $\varphi_l^p(z)$ in the simplex Δ_q:

$$N_l^p(z) = \{\varphi_l^p(z), A_l \in \Delta_q; 0, A_l \notin \Delta_q\}$$

and possess the following properties: the functions $N_l^p(z)$ are continuous in the domain Ω; the functions $N_l^p(z)$ equal 1 at one of the points A_l and zero at the rest of the points; $N_l^p(z_{l'}) = \delta_{ll'}$ in the entire domain Ω. Here l takes the values $l = 1, \ldots, N$.

The functions $N_l^p(z)$ form a basis in the space of polynomials of the pth order. Now, the function $\Phi(z) \in \mathcal{H}^1(\Omega)$ is approximated by a finite sum of piecewise basis functions $N_l^p(z)$:

$$\Phi^h(z) = \sum_{l=1}^{N} \Phi_l^h N_l^p(z). \quad (8)$$

Table 1. The orbits and their number of permutations for $d = 3, 4, 5, 6$.

$d=3$		$d=4$		$d=5$				$d=6$			
Orbits	Perm.	Orbits	Perm.	Orbits	Perm.	Orbits	Perm.	Orbits	Perm.	Orbits	Perm.
S_4	1	S_5	1	S_6	1	S_{3111}	120	S_7	1	S_{4111}	210
S_{31}	4	S_{41}	5	S_{51}	6	S_{2211}	180	S_{61}	7	S_{3211}	420
S_{22}	6	S_{32}	10	S_{42}	15	S_{21111}	360	S_{52}	21	S_{2221}	630
S_{211}	12	S_{311}	20	S_{33}	20	S_{111111}	720	S_{43}	35	S_{31111}	840
S_{1111}	24	S_{221}	30	S_{411}	30			S_{511}	42	S_{22111}	1260
		S_{2111}	60	S_{321}	60			S_{421}	105	S_{211111}	2520
		S_{11111}	120	S_{222}	90			S_{331}	140	$S_{1111111}$	5040
								S_{322}	210		

After substituting expansion (8) into the variational functional (5) and minimizing it [3,20], we obtain the generalized eigenvalue problem

$$\mathbf{A}^p \boldsymbol{\Phi}^h = \varepsilon^h \mathbf{B}^p \boldsymbol{\Phi}^h. \tag{9}$$

Here $\mathbf{A}^p$ is the symmetric stiffness matrix; $\mathbf{B}^p$ is the symmetric positive definite mass matrix; $\boldsymbol{\Phi}^h$ is the vector approximating the solution on the finite-element grid; and ε^h is the corresponding eigenvalue. The matrices $\mathbf{A}^p$ and $\mathbf{B}^p$ have the form:

$$\mathbf{A}^p = \{a^p_{ll'}\}^N_{ll'=1}, \mathbf{B}^p = \{b^p_{ll'}\}^N_{ll'=1}, \tag{10}$$

where the matrix elements $a^p_{ll'}$ and $b^p_{ll'}$ are calculated for simplex elements as

$$a^p_{ll'} = \sum_{ij=1}^{d} \int_{\Delta_q} g_{ij}(z) \frac{\partial \varphi^p_l(z)}{\partial z_i} \frac{\partial \varphi^p_{l'}(z)}{\partial z_j}\, dz + \int_{\Delta_q} g_0(z) \varphi^p_l(z) \varphi^p_{l'}(z) V(z)\, dz,$$

$$b^p_{ll'} = \int_{\Delta_q} g_0(z) \varphi^p_l(z) \varphi^p_{l'}(z) dz. \tag{11}$$

The economical implementation of FEM is the following.

The calculations, including those of FEM integrals for mass and stiffness matrices at each subdomain Δ_q are performed in the local (reference) system of coordinates x, in which the coordinates of the simplex vertices are the following: $\hat{x}_j = (\hat{x}_{j1}, \ldots, \hat{x}_{jd})$, $\hat{x}_{jk} = \delta_{jk}$, $j = 0, \ldots, d$, $k = 1, \ldots, d$.

Let us construct the Lagrange interpolation polynomial (LIP) on an arbitrary d-dimensional simplex Δ_q with vertices $\hat{z}_i = (\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{id})$, $i = 0, \ldots, d$. For this purpose, we introduce the local system of coordinates $x = (x_1, x_2, \ldots, x_d) \in \mathcal{R}^d$, in which the coordinates of the simplex vertices are $\hat{x}_i$. The relation between the coordinates is given by the formula:

$$z_i = \hat{z}_{0i} + \sum_{j=1}^{d} \hat{J}_{ij} x_j, \quad \hat{J}_{ij} = \hat{z}_{ji} - \hat{z}_{0i}, \quad i = 1, \ldots, d. \tag{12}$$

Table 2. Quadrature rule on tetrahedra.

Orbit	Weight	Abscissas
14-points 4-order rule		
S_{31}	0.080118675895755121455796780619 1	0.096372107615282718067986798 2109
S_{31}	0.124367442494243131747125119393 7	0.312306421813294126114726543 7508
S_{22}	0.030342587740001164531385399991 5	0.027470788685334495775013295 4191
14-points 5-order rule		
S_{31}	0.073493043116361949543710205486 3	0.092735250310891226402323913 7370
S_{31}	0.112687925718015850799185652333 3	0.310885919263300609797345733 7635
S_{22}	0.042546020777081466438069428120 3	0.045503704125649649491880526 2793
24-points 6-order rule		
S_{31}	0.039922750258167492099690627557 5	0.214602871259152029288839219 3863
S_{31}	0.010077211055320642948013237445 9	0.040673958534611353115579448956 4
S_{31}	0.055357181543654722095153277853 7	0.322337890142275510343994470 7625
S_{211}	0.048214285714285714285714285714 3	0.063661001875017525299235527605 7 0.603005664791649141367431139060 9
35-points 7-order rule		
S_4	0.095485289464130848860578436117 2	0.250000000000000000000000000000 0
S_{31}	0.042329581209967029076286170798 6	0.315701149778202799423429999593 3
S_{22}	0.031896927832857579934274824082 9	0.050489822598396368763053822986 6
S_{211}	0.037207130728334621369615561191 5	0.188833831026001047736431103854 6 0.575171637587000023483241577022 3
S_{211}	0.008110770829903341566103433491 1	0.021265472541483245988836101499 8 0.810830241098548561118105379848 2
46-points 8-order rule		
S_{31}	0.006397277740665617651504973876 4	0.039675751858211122527707893629 8
S_{31}	0.040190621438228806703869816180 2	0.314487768658878967238651688800 7
S_{31}	0.024308169212176077079539636319 2	0.101987346901070274803893756534 6
S_{31}	0.054858627763726492846425425358 4	0.184203769722815477118606567187 4
S_{22}	0.035719674756330901357934814982 9	0.063436395166279031838503537529 5
S_{211}	0.007183186265240405724897376933 2	0.021690128812349402198200121865 8 0.719931653005748253202189279620 3
S_{211}	0.016372077638328478835688598330 6	0.204480036267872801810154362979 9 0.580577556874088675978195089567 3

The inverse transformation and the relation between the differentiation operators are given by the formulas

$$x_i = \sum_{j=1}^{d} (\hat{J}^{-1})_{ij}(z_j - \hat{z}_{0j}), \frac{\partial}{\partial x_i} = \sum_{j=1}^{d} \hat{J}_{ji} \frac{\partial}{\partial z_j}, \frac{\partial}{\partial z_i} = \sum_{j=1}^{d} (\hat{J}^{-1})_{ji} \frac{\partial}{\partial x_j}.$$

Table 3. Quadrature rule on $d = 4$ dimensional simplex.

Orbit	Weight	Abscissas
20-points 4-order rule		
S_{41}	0.037953922420653961083151176063 4	0.078422464532008441270186009537 2
S_{41}	0.068138449514096507307237418942 1	0.244992500251650682974726724199 8
S_{32}	0.046953814032624765804805702497 3	0.065780705401760442932665992362 7
30-points 5-order rule		
S_{41}	0.049251680175315740938395667283 3	0.085346630830859408251632945252 6
S_{41}	0.032511460658739364936949373887 8	0.236960011661460705646083216339 8
S_{32}	0.017532710995800450876663590892 7	0.041298014131848401048205215945 0
S_{32}	0.041585718587171996185663888521 8	0.299744338479035286296335489564 9
56-points 6-order rule		
S_5	0.073279236743554772188440808855 0	0.200000000000000000000000000000 0
S_{41}	0.004742912171318373911790594179 8	0.041703381748481614470367973524 3
S_{32}	0.037167112402533006986944882925 5	0.295622797147098049191196334346 2
S_{311}	0.013336248018481771716654774405 6	0.154394924873116842736992119567 3 0.522750646227696832515158469571 2
S_{311}	0.013230505900244392702503095144 0	0.047815675137827492151514862425 5 0.281973941992880602871627877781 1
76-points 7-order rule		
S_5	0.028272766759793510146165467413 7	0.200000000000000000000000000000 0
S_{41}	0.017163792015553795559126596836 5	0.249402089309377969567400055747 0
S_{32}	0.008426290417736873748764156645 8	0.039027995660106969047822346802 8
S_{32}	0.015163362756045314580986291487 9	0.128311404463812192159465856927 9
S_{311}	0.004109934841481556020447802548 6	0.033847470986564263527996961838 6 0.746262428681339061102062480377 5
S_{221}	0.018927101486499483611724700536 5	0.044833796455796184976390008452 7 0.209871085716232476426298177816 2
110-points 8-order rule		
S_{41}	0.020988963106203348828447185874 1	0.106416063260142058846827434852 4
S_{41}	0.002556930429961908711113352905 4	0.040543282412661311354934088265 7
S_{32}	0.015336414023745230822528153201 3	0.055320520485979115777864856400 0
S_{32}	0.014341370355404557767971236158 7	0.132984924720748876527117239830 5
S_{32}	0.021983906357169179701387411959 0	0.292164962367903993351239086340 8
S_{311}	0.003699835117610442071728496938 3	0.033339878866874728719032798603 3 0.696028477914025484511728247325 7
S_{311}	0.010287515395496733244605083680 3	0.174905546599082503418947240638 8 0.471358339480343408015545132262 7
S_{221}	0.002863553823128017435221922684 7	0.213995556297885214765130285694 7 0.005579447145523524409701578704 0

Table 4. Quadrature rule on $d = 5$ dimensional simplex.

Orbit	Weight	Abscissas
27-points 4-order rule		
S_6	0.23809523809523809523809523809523	0.16666666666666666666666666666667
S_{51}	0.04761904761904761904761904761904	0.08333333333333333333333333333333
S_{33}	0.02380952380952380952380952380952	0.31100423396407310779395386179222
37-points 5-order rule		
S_6	0.15372022030842936177271263672470	0.16666666666666666666666666666667
S_{51}	0.02891062244931516156159281628850	0.07500000000000000000000000000000
S_{42}	0.02723010532985785470252391583960	0.06209311779376804482624364735120
S_{42}	0.01762429766985412322132478186340	0.24941130698499301712065900751610
102-points 6-order rule		
S_{51}	0.02206097776999184163851718092160	0.09367847966579071795078831844940
S_{51}	0.00102889398402937477520011926020	0.02705664343407666257135586985700
S_{42}	0.01562641726187194574183800806100	0.06539509860373391797226924048050
S_{42}	0.02782824944458255462663419240310	0.22988441816266589010512133393900
S_{321}	0.00349401281465091993317688653240	0.01828680369243056677082032857110 0.19634263926151388664583592828580
137-points 7-order rule		
S_{51}	0.02510799129958512466905683799320	0.19625059980272023863027848359160
S_{51}	0.02681817730725463256882485941400	0.10730645294947929488891128334150
S_{42}	0.00888561063973810080037487732556	0.04996934657341685485161306607590
S_{33}	0.01559651055376095685964964090740	0.28122940505766557254493416595150
S_{411}	0.00132151302526338812734926405670	0.02873565824924136838125559693690 0.72430257945347491879697167732940
S_{321}	0.00339305378216281939171679128120	0.15732708623261516767689860126299 0.00365482861157487691470712917650
257-points 8-order rule		
S_{51}	0.01763037118952217983596151708290	0.10620792694405314278518218182300
S_{51}	0.00222612121038703660355631698450	0.04451287539385467473539305403018
S_{42}	0.01667473057972161270294936710850	0.22152716544879219455564360760780
S_{33}	0.00396602046262096545162702793650	0.02873624397023822982735213543050
S_{411}	0.00137127612890241935051020306700	0.03028073166281611842455123272460 0.57426252407471011190619642227320
S_{321}	0.00092619717524639362929412577410	0.01786537424100418243433166171320 0.15994850355465960507680998566760
S_{321}	0.00483119210977606932266212050330	0.09711754642246895375861977478710 0.35091359200390255665982196429990
S_{321}	0.00274730061139801406922384442740	0.15425984178365369044578798189590 0.01753019026610634957896259957140

Table 5. Quadrature rule on $d = 6$ dimensional simplex.

Orbit	Weight	Abscissas
43-points 4-order rule		
S_7	0.16689962424064264240465553487802	0.14285714285714285714285714285714
S_{61}	0.02716619815142700760903673620086	0.07120154347010901732552543625O4
S_{43}	0.01836962824855338010745351763 31	0.03787627104219600219620536572 98
64-points 5-order rule		
S_7	0.10556089403200693223264178793 46	0.14285714285714285714285714285714
S_{61}	0.02429904195320186500137946120 51	0.07155392508439903058574731017 07
S_{52}	0.01171346168792031576174415885 91	0.05067728321030771781231841506 43
S_{43}	0.01366751762426438233603070421 68	0.23043585212440365120245662379 56
175-points 6-order rule		
S_{61}	0.00046104931565255285484082283 37	0.02509909604870815447009085165 34
S_{61}	0.01301991674586050465013068956 16	0.16408824850302388029905815038 86
S_{52}	0.00203064971090217995679119523 05	0.02784407850016651933540912122 51
S_{43}	0.01622209262634312729009527370 70	0.05427117388472234767215445663 26
S_{421}	0.00281158430208050822113571174 90	0.12031965897287419105268484181 55 0.00375498171181802169768851192 86
266-points 7-order rule		
S_{61}	0.01035837264537888252615510306 59	0.16555370691703407135736243874 30
S_{52}	0.01279465427717344053399913268 92	0.08004169174138494538281587908 68
S_{52}	0.00386657976915606846805402497 46	0.04620602073726548357076393562 06
S_{43}	0.00684822737381594150629804039 42	0.22516267723705716736524194439 13
S_{43}	0.00130065466676527607925405064 06	0.01402083836117134817473437605 62
S_{511}	0.00053218990985704857284890002 18	0.02466780636399904904470747767 34 0.17596361300651512394911832179 36
S_{421}	0.00257183456071513788304591409 97	0.12428318118671194564818424084 70 0.00637231310142874735591924906 77
553-points 8-order rule		
S_{61}	0.01195769984391890953221406683 80	0.16467687533234213409428704255 51
S_{61}	0.01700338552088890217399887775 38	0.10107026106277182500519132582 75
S_{61}	0.00157632710208893572203094203 00	0.04450133014588455711806772835 28
S_{43}	0.00299601348511639014786666776 98	0.04442595335054347436540693296 55
S_{43}	0.00578102644320970733099508033 59	0.22110512716074526607395676536 53
S_{511}	0.00070969810729333061947960575 18	0.03038422111823568037998492356 50 0.25759784196158417691648228708 09
S_{421}	0.00031727721601467282707436680 40	0.01266863837585566447361723432 55 0.21017701247934510298958115975 03
S_{421}	0.00152765862898539069491639528 51	0.12326753489923003279547226294 36 0.00503160098647695485919297306 62
S_{322}	0.00121674348099515619245218166 20	0.09558682973748164107782263108 66 0.33778856869063836579701555683 62

The integrals that enter the variational functional (5) on the domain $\Omega_h(z) = \bigcup_{q=1}^{Q} \Delta_q$, are expressed via the integrals, calculated on the element Δ_q, and recalculated to the local coordinates x on the element Δ,

$$\int_{\Delta_q} dz g_0(z)\varphi_r^p(z)\varphi_{r'}^p(z)V(z) = J\int_{\Delta} dx g_0(z(x))\varphi_r^p(x)\varphi_{r'}^p(x)V(z(x)), \quad (13)$$

$$\int_{\Delta_q} dz g_{s_1s_2}(z)\frac{\partial\varphi_r^p(z)}{\partial z_{s_1}}\frac{\partial\varphi_{r'}^p(z)}{\partial z_{s_2}}$$

$$= J\sum_{t_1,t_2=1}^{d} \hat{J}^{-1}_{s_1s_2;t_1t_2}\int_{\Delta} dx g_{s_1s_2}(z(x))\frac{\partial\varphi_r^p(x)}{\partial x_{t_1}}\frac{\partial\varphi_{r'}^p(x)}{\partial x_{t_2}},$$

where $J = \det\hat{J} > 0$ is the determinant of the matrix $\hat{J}$ from Eq. (12), $\hat{J}^{-1}_{s_1s_2;t_1t_2} = (\hat{J}^{-1})_{t_1s_1}(\hat{J}^{-1})_{t_2s_2}$, $dx = dx_1 \ldots dx_d$.

In the local coordinates, the LIP $\varphi_r^p(x)$ is equal to one at the node point ξ_r characterized by the numbers $[n_0, n_1, \ldots, n_d]$, and zero at the remaining node points ξ_r', i.e., $\varphi_r(\xi_r') = \delta_{rr}$ are determined by Eq. (7) at $H(0;x) = 1 - x_1 - \ldots - x_d$, $H(i;z) = x_i$, $i = 1, \ldots, d$:

$$\varphi_r(x) = \prod_{i=1}^{d}\prod_{n_i=0}^{n_i-1}\frac{x_i - n_i/p}{n_i/p - n_i/p}\prod_{n_0=0}^{n_0-1}\frac{1 - x_1 - \ldots - x_d - n_0/p}{n_0/p - n_0/p}. \quad (14)$$

Integrals (13) are evaluated using the Gaussian quadrature of the order $2p$.

Let ε_m and $\Phi_m(z)$ be exact solutions of Eq. (9) and ε_m^h and Φ_m^h be the corresponding numerical solutions. Then the following estimations are valid [20]

$$|\varepsilon_m - \varepsilon_m^h| \le c_1|\varepsilon_m|h^{2p}, \quad \|\Phi_m(z) - \Phi_m^h\|_0 \le c_2|E_m|h^{p+1}, \quad (15)$$

where $\|a(z)\|_0^2 = \langle a(z)|a(z)\rangle$, h is the maximal step of the finite-element grid, m is the number of the corresponding solution, and the positive constants c_1 and c_2 do not depend on the step h.

To solve the generalized eigenvalue problem (9), we choose the subspace iteration method [3,20] elaborated by Bathe [3] for the solution of large symmetric banded-matrix eigenvalue problems. This method uses the skyline storage mode which stores the components of the matrix column vectors within the banded region of the matrix, and is ideally suited for banded finite-element matrices.

4 Construction of the *d*-dimensional Quadrature Formulas

Let us construct the d-dimensional p-ordered quadrature formula

$$\int_{\Delta_q} dz V(z) = |\Delta_q|\sum_{j=1}^{n_t} w_j V(z_j), \quad z = (z_1, \ldots, z_d), \quad dz = dz_1 \ldots dz_d, \quad (16)$$

for integration over the d-dimensional simplex Δ_q with vertices $\hat{z}_i = (\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{id})$, $i = 0, \ldots, d$, which is exact for all polynomials of the variables $z_1, \ldots, z_d$ of degree not exceeding p, where n_t is the number of nodes that is determined during the calculation process. In Eq. (16), w_j, $j = 1, \ldots, n_t$ are the weights and $z_j = (z_{j1}, z_{j2}, \ldots, z_{jd})$ are the coordinates of nodes. $|\Delta_q|$ denotes the volume of Δ_q. For each node z_j, instead of sets of d coordinates we use the sets of $d+1$ barycentric coordinates (BC) $(x_{j0}, x_{j1}, \ldots, x_{jd})$:

$$z_j = x_{j0}\hat{z}_0 + \ldots + x_{jd}\hat{z}_d, \quad x_{j0} + \ldots + x_{jd} = 1. \tag{17}$$

For this purpose, we introduce the local coordinate system $x = (x_1, x_2, \ldots, x_d)$ and (12). Therefore, without loss of generality, we construct the d-dimensional p-ordered quadrature formula (16) on the standard simplex Δ with vertices $\hat{x}_j = (\hat{x}_{j1}, \ldots, \hat{x}_{jd})$, $\hat{x}_{jk} = \delta_{jk}$, $j = 0, \ldots, d$, $k = 1, \ldots, d$, which is exact for all polynomials of the variables $x_1, \ldots, x_d$ of degree not exceeding p:

$$\int_{\Delta} dx V(x) = \frac{1}{d!} \sum_{j=1}^{n_t} w_j V(x_{j0}, \ldots, x_{jd}). \tag{18}$$

Since the following formula is valid for all permutations $(l_0, \ldots, l_d)$ of $(k_0, \ldots, k_d)$:

$$\int_{\Delta} dx x_1^{l_1} \ldots x_d^{l_d} (1 - x_1 - \ldots - x_d)^{l_0} = \frac{\prod_{i=0}^{d} k_i!}{\left(d + \sum_{i=0}^{d} k_i\right)!},$$

we consider the fully symmetric Gaussian quadratures

$$\int_{\Delta} dx V(x) = \frac{1}{d!} \sum_{j=1}^{a} w_j \sum_{j_0, \ldots, j_d} V(x_{j_0 0}, x_{j_1 1}, \ldots, x_{j_d d}), \tag{19}$$

where the internal summation by $j_0, \ldots, j_d$ is carried out over the different permutations of $(x_{j0}, x_{j1}, \ldots, x_{jd})$. Table 1 presents the orbits and the corresponding number of different permutations for $d = 3, 4, 5, 6$. Here, for example, the orbit S_{331} at $d = 6$ contains BC $(\alpha, \alpha, \alpha, \beta, \beta, \beta, \gamma)$, $\alpha \neq \beta \neq \gamma$, $\alpha \neq \gamma$, $3\alpha + 3\beta + \gamma = 1$ and their different 140 permutations.

Substituting a monomial of the order not exceeding p in Eq. (19) instead of $V(x)$, we arrive at a system of nonlinear algebraic equations, that using the Vieta theorem reduces to the form:

$$\int_{\Delta} dx s_2^{l_2} s_3^{l_3} \times \ldots \times s_{d+1}^{l_{d+1}} = \frac{1}{d!} \sum_{j=1}^{a} w_j Q_j s_{j2}^{l_2} s_{j3}^{l_3} \times \ldots \times s_{jd+1}^{l_{d+1}}, \tag{20}$$

$$2l_2 + 3l_3 + \ldots + (d+1)l_{d+1} \leq p, \tag{21}$$

where

$$s_2 = \sum_{i=0, j \neq i}^{d} x_i x_j, \quad \ldots, \quad s_{d+1} = \prod_{i=0}^{d} x_i, \tag{22}$$

s_{ji}, $i = 2, \ldots, d+1$, are their values in the BC $(x_{j0}, x_{j1}, \ldots, x_{jd})$, and Q_j is the number of different permutation of the BC. As in Ref. [15], instead of Eq. (22), we can use

$$s_j = \sum_{i=0}^{d} x_i^j, \quad j = 2, \ldots, d+1. \tag{23}$$

The number of all $l_j \geq 0$ solutions of Eq. (21) provides the minimal number of independent nonlinear equations for the quadrature formula of the order p. It means that we can obtain a set of independent polynomials by adding new polynomials when increasing the order p. Below the first few independent polynomials of the order not exceeding $p \leq 6$ for $d \geq 5$ are presented:

$$\begin{aligned} V_1(x) &= s_1, && \text{for } p = 1, \\ V_2(x) &= s_2, && \text{for } p = 2, \\ V_3(x) &= s_3, && \text{for } p = 3, \\ V_4(x) = s_2^2, V_5(x) &= s_4, && \text{for } p = 4, \\ V_6(x) = s_2 s_3, V_7(x) &= s_5, && \text{for } p = 5, \\ V_8(x) = s_2^3, V_9(x) = s_3^2, V_{10}(x) &= s_2 s_4, V_{11}(x) = s_6, && \text{for } p = 6. \end{aligned} \tag{24}$$

We consider fully symmetric rules with positive weights, and no points are outside the simplex (the so-called PI-type).

The n_p-points p-order quadrature rules are constructed with Algorithm 1 [21] implemented by us in Maple and Fortran:

– **for** each decomposition n_p **do**

repeat

1. Randomly choose an initial guess for the unknowns n_t.

2. Find a least square solution to Eqs. (20), (21) using a quasi-Newton algorithm.

3. If a PI-type solution is found satisfying Eqs. (20), (21), with sufficient accuracy, go to Step 4.

until maximum number of initial guesses tried.

– **end for**

– **Stop.**

– **4.** Minimize the nonlinear equation for unknowns n_t using the Levenberg–Marquardt algorithm with high accuracy [12,14].

The Levenberg–Marquardt Algorithm 2:

Let $f(\mathbf{x})$ be twice differentiable with respect to the variable $\mathbf{x} = (x_1, \ldots, x_n)$. We consider the minimization

$$\min_{\mathbf{x} \in R^n} f(\mathbf{x}). \tag{25}$$

1. Start with an initial value $\mathbf{x}_0$, in S, an initial damping parameter λ_0, and a scaling parameter ρ. For $k \geq 0$ do the following:

2. Determine a trial iterate $\mathbf{y}$, using

$$\mathbf{y} = \mathbf{x}_k - \left(H_f(\mathbf{x}_k) + \lambda \operatorname{diag}(H_f(\mathbf{x}_k))\right)^{-1} \nabla f(\mathbf{x}_k), \tag{26}$$

with $\lambda = \lambda_k \rho^{-1}$.
3. If $f(\mathbf{y}) < f(\mathbf{x}_k)$, where $\mathbf{y}$ is determined in Step 2, then set $\mathbf{x}_{k+1} = \mathbf{y}$ and $\lambda_{k+1} = \lambda_k \rho^{-1}$. Return to Step 2, replace k with $k+1$, and compute a new trial iterate.
4. If $f(\mathbf{y}) \geq f(\mathbf{x}_k)$ in Step 3, determine a new trial iterate, $\mathbf{y}$, using (26) with $\lambda = \lambda_k$.
5. If $f(\mathbf{y}) < f(\mathbf{x}_k)$, where $\mathbf{y}$ is determined in Step 4, then set $\mathbf{x}_{k+1} = \mathbf{y}$ and $\lambda_{k+1} = \lambda_k$. Return to Step 2, replace k with $k+1$, and compute a new trial iterate.
6. If $f(\mathbf{y}) \geq f(\mathbf{x}_k)$ in Step 5, then determine the smallest value of m so that when a trial iterate $\mathbf{y}$ is computed using (26) with $\lambda = \lambda_k \rho^m$, then $f(\mathbf{y}) < f(\mathbf{x}_k)$. Set $\mathbf{x}_{k+1} = \mathbf{y}$ and $\lambda_{k+1} = \lambda_k \rho^m$. Return to Step 2, replace k with $k+1$, and compute a new trial iterate.
7. Terminate the algorithm when $\|\nabla f(\mathbf{x}_k)\| < \epsilon$, where ϵ is the specified tolerance.

In the above Algorithm 2, $\nabla f(\mathbf{x})$, $H_f(\mathbf{x})$ are the gradient vector and the Hessian matrix functions of $f(\mathbf{x})$, respectively. $\operatorname{diag}(H_f(\mathbf{x}))$ is the diagonal matrix of the Hessian matrix function $H_f(\mathbf{x})$.

The weights (W) and the BC of PI-type rules of order p are presented in Tables 2, 3, 4 and 5. Here, for example, for the orbit S_{421} at $d = 6$ contains the BC $(\alpha, \alpha, \alpha, \alpha, \beta, \beta, \gamma)$, $\alpha \neq \beta \neq \gamma$, $\alpha \neq \gamma$ and their different 105 permutations. We present α in the first line and β in the second line, since γ is expressed in terms of α, β, i.e., $\gamma = 1 - 4\alpha - 2\beta$. The rules of the fifth and sixth order on tetrahedra coincide with the results of Ref. [2]. We believe that at least some of the rules presented in this paper are new. But we can not guarantee that the presented numbers of points of high-order quadrature rules are minimal. Note that up to the order $p = 6$ W and BC were calculated using Maple with 32 significant digits. For $p > 6$, W and BC were calculated using Fortran with 10 significant digits (the first three steps of Algorithm 1). These calculations were performed using the Central Information and Computer Complex, and HybriLIT heterogeneous computing cluster at JINR. Starting from the approximate values found with the Fortran code, W and BC were then calculated in Maple with 32 significant digits.

5 BVP for Helmholtz Equation in a *d*-dimensional Hypercube

For benchmark calculations, we use the BVP for the Helmholtz equation (HEQ) with the boundary condition (II) in a d-dimensional hypercube with the edge length π. Since the variables are separated, the eigenvalues $E_m = E_{m_1,\ldots,m_d}$ are sums of squared integers, $E_m = E_{m_1,\ldots,m_d} = m_1^2 + \ldots + m_d^2$, $m_k = 0, 1, \ldots$, $k = 1, \ldots, d$.

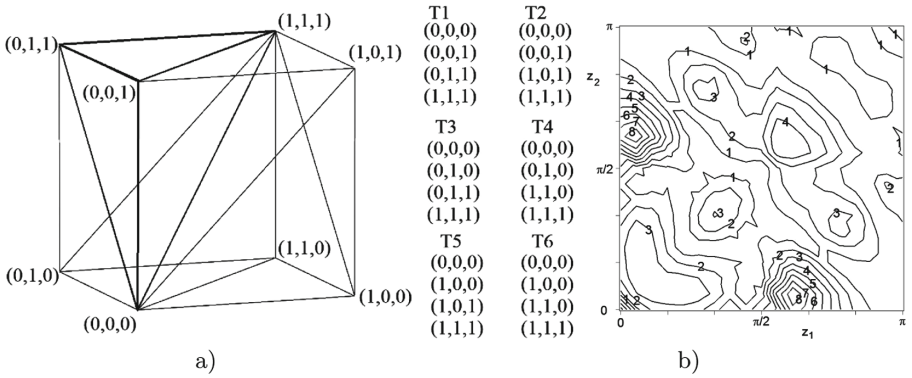

Fig. 1. (a) Division of a 3D cube into 3! = 6 equal tetrahedrons (T1,. . . ,T6). (b) The error $\Delta\Phi_8(z_1, z_2, z_3) = |\Phi_8^h(z_1, z_2, z_3) - \Phi_8(z_1, z_2, z_3)|$ for the eighth eigenfunction $\Phi_8^h(z_1, z_2, z_3)$ at fixed $z_3 = \pi/9$, calculated using FEM with third-order LIPs versus the exact eigenfunction $\Phi_8(z_1, z_2, z_3)$ corresponding to the eigenvalue $E_8 = 3$. Here the cube is divided into 2^3 cubes, each comprised of 6 tetrahedrons. The isolines marked 1 correspond to the values of $\Delta\Phi_8(z_1, z_2, z_3) = \Delta\Phi_8^{\max}/10$, the isolines marked 2 correspond to the values of $\Delta\Phi_8(z_1, z_2, z_3) = 2\Delta\Phi_8^{\max}/10$,. . . , at $\Delta\Phi_8^{\max} \approx 0.018$.

Assertion (see also [16])**.** The hypercube is divided into $d!$ equal simplices. The vertices of each simplex are located on broken lines composed of d mutually perpendicular edges, and the extreme vertices of all polygons are located on one of the diagonals of the hypercube (for $d = 3$ see Fig. 1a).

Algorithm 3.

Input. A single d-dimensional hypercube with vertices the coordinates of which are either 0 or 1 in the Euclidean space $\mathcal{R}^d$. The chosen diagonal of the hypercube connects the vertices with the coordinates $(0, \ldots, 0)$ and $(1, \ldots, 1)$.

Output. $z_k^{(i)} = (z_{k1}^{(i)}, \ldots, z_{kd}^{(i)})$, the coordinates of the ith simplex.

Local. The coordinates of the vertices of the polygonal line are $z_k = (z_{k1}, \ldots, z_{kd})$, $k = 0, \ldots, d$.

1. For all $i = (i_1, \ldots, i_d)$, the permutations of the numbers $(1, \ldots, d)$:

1.1. For all $k = 0, \ldots, d$ and $s = 1, \ldots, d$: $z_{k,s}^{(i)} = \{1, i_s \le k, ; 0, i_s > k\}$

1.2. If $\det(z_{ks}^{(i)})_{ks=1}^{d} = -1$ then $z_{kd}^{(i)} \leftrightarrow z_{kd-1}^{(i)}$.

3D HEQ for the cube. In Fig. 1b, we show the error $\Delta\Phi_8(z_1, z_2, z_3)$ for the eighth eigenfunction $\Phi_8^h(z_1, z_2, z_3)$ at fixed $z_3 = \pi/9$, calculated using FEM with third-order LIPs versus the exact eigenfunction $\Phi_8(z_1, z_2, z_3)$ corresponding to the eigenvalue $E_8 = 3$. In Fig. 2a, we also show the maximal error $\Delta\Phi_8^{\max}$ for the exact eighth eigenfunction $\Phi_8(z_1, z_2, z_3)$ calculated using FEM with LIPs of the orders $p = 3, 4, 5$ versus the number N of piecewise basis functions $N_l^p(z)$ in the expansion (8). In Fig. 2b, we show the error of eigenvalues of the 3D BVP for the HEQ at $d = 3$ with the boundary condition (II) using the FEM scheme with 3D LIP of the order $p = 6$. As seen from Fig. 2, the errors of the eigenfunctions and eigenvalues lie on parallel lines in the double logarithmic scale

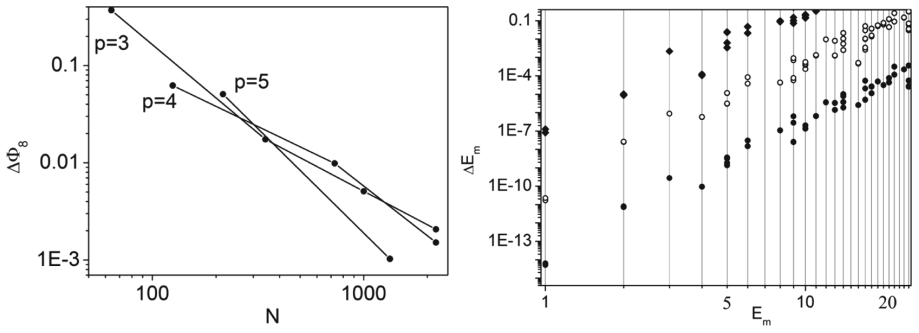

Fig. 2. (a) The maximal error $\Delta\Phi_8^{\max} = \max_{z_1} \in (0,\pi), z_2 \in (0,\pi), z_3 \in (0,\pi)|\Phi_8^h(z_1,z_2,z_3) - \Phi_8(z_1,z_2,z_3)|$ for the exact eighth eigenfunction $\Phi_8(z_1,z_2,z_3)$ calculated using FEM with LIPs of the orders $p = 3,4,5$ versus the number N of piecewise basis functions $N_l^p(z)$ in the expansion (8). (b)The error $\Delta E_m = E_m^h - E_m$ calculated using FEM with sixth-order LIPs versus the exact eigenvalue E_m. Squares: the cube divided into 6 tetrahedrons. Circles: the cube divided into 2^3 cubes, each comprised of 6 tetrahedrons. Solid circles: the cube divided into 4^3 cubes, each comprised of 6 tetrahedrons.

Table 6. The lower part of the exact spectrum E_m and the calculated spectrum E_m^h for the 6D hypercube.

E_m	E_m^h
0	0.183360983479286 e−10
1	1.00023, 1.00034, 1.00034, 1.00034, 1.00034, 1.00034
2	2.04760, 2.04760, 2.04760, 2.04760, 2.04760, 2.04760, 2.04760, 2.04760, 2.04760, 2.07391, 2.08478, 2.08478, 2.08478, 2.08478, 2.08478
3	3.15060, 3.15196, 3.15196, 3.15196, 3.15196, 3.15196, 3.15780, 3.15780, 3.15780, 3.15780, 3.15780, 3.16319, 3.16319, 3.16319, 3.16319, 3.16319, 3.16319, 3.16319, 3.16319, 3.16319, 3.16319

which agrees with the theoretical error estimates (15) for the eigenfunctions and eigenvalues depending on the maximal size of the finite element. For a cube with the edge π divided into 4^3 cubes, each of them comprising 6 tetrahedrons, the matrices $\mathbf{A}$ and $\mathbf{B}$ had the dimension 15625×15625. The matrices $\mathbf{A}$ and $\mathbf{B}$ were calculated in two ways: analytically or with Gaussian quadratures from Sect. 4 using Maple 2015, 2x 8-core Xeon E5-2667 v2 3.3 GHz, 512 GB RAM, GPU Tesla 2075. For the considered task, the values of matrix elements agree with Gaussian quadratures up to the order 10 with given accuracy. The generalized algebraic eigenvalue problem (9) was solved during 20 min using Intel Fortran.

6D HEQ for the hypercube. We solved HEQ at $d = 6$ with the boundary condition (II) using FEM scheme with 6D LIP of the order $p = 3$. The 6D hypercube having the edge π was divided into $n = d! = 6! = 720$ simplexes

(the size of the finite element being equal to π). On each of them $N_1(p) = (p+d)!/(d!p!) = 84$ third-order LIPs were used. The matrices $\mathbf{A}$ and $\mathbf{B}$ had the dimension 4096×4096. The lower part of the spectrum E_m is shown in Table 6. The errors of the second, the third, and the fourth degenerate eigenvalue are equal to 0.0003, 0.05, and 0.15, respectively. Note that applying the third-order scheme for solving the BVPs of smaller dimension d, we obtained errors of the same order. The calculation time was 9234.46 s using Maple 2015.

6 Conclusion

We have elaborated new calculation schemes, algorithms, and programs for solving the multidimensional elliptic BVP using the high-accuracy FEM with simplex elements. The elaborated symbolic-numerical algorithms and programs implemented in Maple-Fortran environment calculate multivariate finite elements in the simplex and the fully symmetric PI Gaussian quadrature rules. We demonstrated the efficiency of the proposed finite element schemes, algorithms, and codes by benchmark calculations of BVPs for Helmholtz equation of cube and hypercube. The developed approach is aimed at calculations of the spectral characteristics of nuclei models and electromagnetic transitions [7,11]. This will be done in our next publications.

Acknowledgment. The work was partially supported by the RFBR (grant No. 16-01-00080 and 18-51-18005), the MES RK (Grant No. 0333/GF4), the Bogoliubov-Infeld program, the Hulubei–Meshcheryakov program, the RUDN University Program 5-100 and grant of Plenipotentiary of the Republic of Kazakhstan in JINR. The authors are grateful to prof. R. Enkhbat for useful discussions.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions. Dover, New York (1965)
2. Akishin, P.G., Zhidkov, E.P.: Some symmetrical numerical integration formuas for simplexes. Communications of the JINR 11–81-395, Dubna (1981). (in Russian)
3. Bathe, K.J.: Finite Element Procedures in Engineering Analysis. Prentice Hall, Englewood Cliffs (1982)
4. Bériot, H., Prinn, A., Gabard, G.: Efficient implementation of high-order finite elements for Helmholtz problems. Int. J. Numer. Meth. Eng. **106**, 213–240 (2016)
5. Ciarlet, P.: The Finite Element Method for Elliptic Problems. North-Holland Publishing Company, Amsterdam (1978)
6. Cui, T., Leng, W., Lin, D., Ma, S., Zhang, L.: High order mass-lumping finite elements on simplexes. Numer. Math. Theor. Meth. Appl. **10**(2), 331–350 (2017)
7. Dobrowolski, A., Mazurek, K., Góźdź, A.: Consistent quadrupole-octupole collective model. Phys. Rev. C **94**, 054322-1–054322-20 (2017)
8. Dunavant, D.A.: High degree efficient symmetrical Gaussian quadrature rules for the triangle. Int. J. Numer. Meth. Eng. **21**, 1129–1148 (1985)

9. Gusev, A.A., et al.: Symbolic-numerical algorithm for generating interpolation multivariate hermite polynomials of high-accuracy finite element method. In: Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (eds.) CASC 2017. LNCS, vol. 10490, pp. 134–150. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66320-3_11
10. Gusev, A.A., et al.: Symbolic-numerical algorithms for solving the parametric self-adjoint 2D elliptic boundary-value problem using high-accuracy finite element method. In: Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (eds.) CASC 2017. LNCS, vol. 10490, pp. 151–166. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66320-3_12
11. Gusev, A.A., et al.: Symbolic algorithm for generating irreducible rotational-vibrational bases of point groups. In: Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (eds.) CASC 2016. LNCS, vol. 9890, pp. 228–242. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45641-6_15
12. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**, 164–168 (1944)
13. www.maplesoft.com
14. Marquardt, D.: An algorithm for least squares estimation of parameters. J. Soc. Ind. Appl. Math. **11**, 431–441 (1963)
15. Maeztu, J.I., Sainz de la Maza, E.: Consistent structures of invariant quadrature rules for the n-simplex. Math. Comput. **64**, 1171–1192 (1995)
16. Mead, D.G.: Dissection of the hypercube into simplexes. Proc. Am. Math. Soc. **76**, 302–304 (1979)
17. Mysovskikh, I.P.: Interpolation Cubature Formulas. Nauka, Moscow (1981). (in Russian)
18. Papanicolopulos, S.-A.: Analytical computation of moderate-degree fully-symmetric quadrature rules on the triangle. arXiv:1111.3827v1 [math.NA] (2011)
19. Sainz de la Maza, E.: Fórmulas de cuadratura invariantes de grado 8 para el simplex 4-dimensional. Revista internacional de métodos numéricos para cálculo y diseño en ingeniería **15**(3), 375–379 (1999)
20. Strang, G., Fix, G.J.: An Analysis of the Finite Element Method. Prentice-Hall, Englewood Cliffs (1973)
21. Zhang, L., Cui, T.: Liu. H.: A set of symmetric quadrature rules on triangles and tetrahedra. J. Comput. Math. **27**, 89–96 (2009)