Efficient Numerical Evaluation of Singular Integrals in Volume Integral Equations

Cedric Münger, Kristof Cools

***Abstract*—We present a method for the numerical evaluation of 6D and 5D singular integrals appearing in Volume Integral Equations. It is an extension of the Sauter-Schwab/Taylor-Duffy strategy for singular triangle-triangle interaction integrals to singular tetrahedron-tetrahedron and triangle-tetrahedron interaction integrals. The general advantages of these kind of quadrature strategy is that they allow the use of different kinds of kernel and basis functions. They also work on curvilinear domains. They are all based on relative coordinates tranformation and splitting the integration domain into subdomains for which quadrature rules can be constructed. We show how to build these tensor-product quadrature rules in 6D and 5D and further show how to improve their efficiency by using quadrature rules defined over 2D, 3D and 4D simplices. Compared to the existing approach, which computes the integral over the subdomains as a sequence of 1D integrations, significant speedup can be achieved. The accuracy and convergence properties of the method are demonstrated by numerical experiments for 5D and 6D singular integrals. Additionally, we applied the new quadrature approach to the triangle-triangle interaction integrals appearing in Surface Integral Equations.**

***Index Terms*—volume integral equations, singular integrals, numerical quadrature**

I. Introduction

Volume Integral Equations (VIE) are used to solve EM scattering problems involving inhomogeneous domains, with application in geological prospecting or microwave imaging. There are different formulations for VIEs [1]. They all have in common that they require the accurate evaluation of 6D interaction integrals that may contain a singular kernel. Furthermore, some formulation involve the evaluation of boundary terms that require the evaluation of singular 5D interaction integrals between surface triangles and internal tetrahedrons. For VIE methods to be competitive with other well established numerical methods - such as finite element or finite difference methods - these integrals need an accurate and efficient evaluation. In recent years, several methods on how to compute these integrals have been proposed. There are semi-analytic approaches that evaluate a part of the integral analytically [2]. Alternatively, the divergence theorem can be used to convert the volume integrals into surface integrals [3].

In [4], our starting point was the approach from [5], where the integrand is expressed in relative coordinates and the integration domain is subdivided. The advantage of this method is that it moves the singularities to the boundaries of the subdomains. In [6], it is demonstrated how to evaluate the inner non-singular integrals analytically.

Instead of computing an analytic expression for the inner integrals we proposed to use lower dimensional quadrature rules to build 6D tensor-product quadrature rules and use the Duffy-Transformation [7] to remove the singularities and evaluate the full 6D integral numerically. This makes our approach suitable for use with a wide class of kernel functions and curvilinear parametrization of the integration domains.

In [8], triangle-triangle interactions are computed by a sequence of 1D integrals. For the 4D integrals encountered in computing triangle-triangle interactions, this results in acceptable performance. Unfortunately, in the 6D tetrahedron-tetrahedron case this approach leads to prohibitive computational costs. Here, we show in detail how to build customized quadrature methods that factorize the 6D or 5D integration in a sequence of lower dimensional integrals while maximizing the dimension for the individual integrals in the sequence. The numerical experiments will show this can reduce the computational cost up to an order of magnitude in 6D. New results for 5D and 4D integrals will show that this approach still can achieve an significant improved performance compared to the standard approach.

II. Formulation

We want to evaluate integrals of the form

$$I_{T\times T'} = \int_T \int_{T'} k(\vec{x},\vec{y}) d\vec{x} d\vec{y}, \tag{1}$$

where $k(\vec{x},\vec{y}) = b(\vec{y})g(\vec{x},\vec{y})\beta(\vec{x})$ and $g(\vec{x},\vec{y})$ is a scalar or dyadic Green's function and $b(\vec{y})$ and $\beta(\vec{y})$ are scalar or vector test function and trial function, respectively. T and T' are the integration domains, usually a pair of tetrahedrons, a pair of triangles or a tetrahedron and a triangle. Typically, the integrand $k(\vec{x},\vec{y})$ behaves like $1/r$ or $1/r^2$ with $r = |\vec{x}-\vec{y}|$. Thus, the numerical evaluation of (1) is not straightforward if the two integration domains overlap or touch.

A. Decomposition into subdomains

If the integration domains overlap or touch, we use the same steps as in [5] and [8] to subdivide the integration domain and move the singularity to a more tractable location. We assume that the tetrahedrons T and T' can be parametrized by the reference tetrahedron τ with corners $(0,0,0), (1,0,0), (1,1,0)$ and $(1,1,1)$ via the mappings $M_T : \tau \to T$ and $M_{T'} : \tau \to T'$. Further, we assume that the tetrahedrons overlap or touch

This paper is an expanded version of a conference paper presented at 2021 IEEE International Conference on Microwaves, Antennas, Biomedical Engineering & Electronic Systems (COMCAS), Tel Aviv, Israel.

C. Münger and K. Cools are with IDLab, Department of Information Technology at Ghent University.

in a very specific way. Namely, for the common face case we have

$$M_T(s,t,0) = M_{T'}(s,t,0) \quad \forall s,t \in [0,1],[0,s],$$

for the common edge case

$$M_T(s,0,0) = M_{T'}(s,0,0) \quad \forall s \in [0,1]$$

and for the common vertex

$$M_T(\vec{0}) = M_{T'}(\vec{0}).$$

This can be achieved by reordering the vertices and choosing an appropriate transformation to the reference tetrahedron. Similar assumptions are made if T or T' is a triangle.

With these assumptions in mind, all our subsequent manipulations are made on the reference element. We define the relative coordinate transformation $(\vec{x},\vec{y}) \mapsto (\hat{\vec{x}},\hat{\vec{z}})$ to move the singularity to $\hat{\vec{z}}=0$. This results in different coordinate transformations, when the two tetrahedrons are identical, share a face, edge or only a vertex. They are summarized in Table I for tetrahedron-tetrahedron interaction. Table II lists the transformations for triangle-tetrahedron interactions. For the sake of completeness, the relative coordinates for triangle-triangle interactions from [8] are listed in Table III.

TABLE I
Relative Coordinate Transformations for two tetrahedrons

Identical Tetrahedron	**Common Face**
$\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\\ \hat{x}_3\\ \hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ x_3\\ y_1-x_1\\ y_2-x_2\\ y_3-x_3\end{bmatrix}$	$\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\\ \hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\\ \hat{z}_4\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ y_1-x_1\\ y_2-x_2\\ y_3\\ x_3\end{bmatrix}$
Common Edge	**Common Vertex**
$\begin{bmatrix}\hat{x}_1\\ \hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\\ \hat{z}_4\\ \hat{z}_5\end{bmatrix} = \begin{bmatrix}x_1\\ y_1-x_1\\ y_2\\ y_3\\ x_2\\ x_3\end{bmatrix}$	$\begin{bmatrix}\hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\\ \hat{z}_4\\ \hat{z}_5\\ \hat{z}_6\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ x_3\\ y_1\\ y_2\\ y_3\end{bmatrix}$

TABLE II
Relative Coordinate Transformations for tetrahedron and triangle

Common Face	**Common Edge**	**Common Vertex**
$\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\\ \hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ y_1-x_1\\ y_2-x_2\\ x_3\end{bmatrix}$	$\begin{bmatrix}\hat{x}_1\\ \hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\\ \hat{z}_4\end{bmatrix} = \begin{bmatrix}x_1\\ y_1-x_1\\ y_2\\ x_2\\ x_3\end{bmatrix}$	$\begin{bmatrix}\hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\\ \hat{z}_4\\ \hat{z}_5\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ x_3\\ y_1\\ y_2\end{bmatrix}$

TABLE III
Relative Coordinate Transformations for two triangles

Identical Triangle	**Common Edge**	**Common Vertex**
$\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\\ \hat{z}_1\\ \hat{z}_2\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ y_1-x_1\\ y_2-x_2\end{bmatrix}$	$\begin{bmatrix}\hat{x}_1\\ \hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\end{bmatrix} = \begin{bmatrix}x_1\\ y_1-x_1\\ y_2\\ x_2\end{bmatrix}$	$\begin{bmatrix}\hat{z}_1\\ \hat{z}_2\\ \hat{z}_3\\ \hat{z}_4\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ y_1\\ y_2\end{bmatrix}$

In a second step, we change the order of integration such that the outer integration is over $\vec{z}$ and the inner integration variable is $\vec{x}$. The reordering of integration variables moves the singular behavior to the outer integrals, which will help with relocating the singularity to the boundary of the integration subdomains. Having the singularity at the boundary of the subdomains is necessary to be able to apply the Duffy-Transformation that we will discuss in the next section.

During the reordering, we split the integration domain D into multiple subdomains D_i. The reasoning behind this subdivision is twofold. First, it will allow for the singularity to be located on the boundaries of the subdomains. And second, it will simplify the description of the integration bounds. For a illustration of this concept in 1D, let us assume we have two identical unit segments (I and I') that completely overlap.

$$M_I(s) = M_{I'}(s) \quad \forall s \in [0,1]$$

Then the integration domain is $D = [0,1] \times [0,1]$, where the singularity $(x = y)$ is located along the diagonal in leftmost picture of Figure 1. With the coordinate transform $(\hat{x},\hat{z}) = (x, y-x)$, we can move the singular points to the more tractable location of $\hat{z}=0$ (middle Figure 1). In a final step, we switch axis of $\hat{x}$ and $\hat{z}$ and split the integration domain along the singularity $(\hat{z}=0)$ into two subdomains D_1 and D_2.

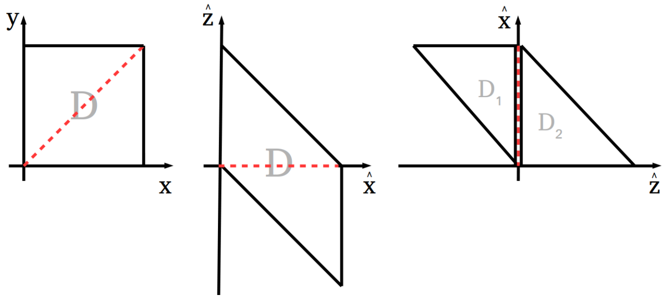

Fig. 1. Integration domain in Cartesian coordinates (left), same domain in relative coordinates (middle), and after swapping the relative coordinates and subdividing the integration domain. (right).

Written down algebraically this translates to the following constraints for the integration bounds.

$$\begin{aligned}\begin{Bmatrix}0 \le x \le 1\\ 0 \le y \le 1\end{Bmatrix} &= \begin{Bmatrix}0 \le \hat{x} \le 1\\ -\hat{x} \le \hat{z} \le 1-\hat{x}\end{Bmatrix} \\ &= \begin{Bmatrix}-1 \le \hat{z} \le 0\\ -\hat{z} \le \hat{x} \le 1\end{Bmatrix} \cap \begin{Bmatrix}0 \le \hat{z} \le 1\\ 0 \le \hat{x} \le 1-\hat{z}\end{Bmatrix}.\end{aligned} \tag{2}$$

Common Vertex: The splitting of the integration domain for the common vertex case is a bit different. We have only $\hat{z}$-variables after the relative coordinate transformation (see Table I, II and III). We can not simply change the order of the $\hat{z}$ and

$\hat{x}$ variables. The domain decomposition is again more easily illustrated in an 1D example. We again have two 1D segments but this time they only share one endpoint.

$$M_I(\vec{0}) = M_{I'}(\vec{0})$$

The integration domain forms a square with the singularity located in one corner. This square can be split along the diagonal going through the common (singular) vertex, see Figure 2. After we switch the order of integration in the second domain for a more symmetric result, the integration domain becomes

$$\begin{aligned}\begin{Bmatrix} 0 \le \hat{z}_1 \le 1 \\ 0 \le \hat{z}_2 \le 1 \end{Bmatrix} &= \begin{Bmatrix} 0 \le \hat{z}_1 \le 1 \\ 0 \le \hat{z}_2 \le \hat{z}_1 \end{Bmatrix} \cap \begin{Bmatrix} 0 \le \hat{z}_1 \le 1 \\ \hat{z}_1 \le \hat{z}_2 \le 1 \end{Bmatrix} \qquad (3) \\ &= \begin{Bmatrix} 0 \le \hat{z}_1 \le 1 \\ 0 \le \hat{z}_2 \le \hat{z}_1 \end{Bmatrix} \cap \begin{Bmatrix} 0 \le \hat{z}_2 \le 1 \\ 0 \le \hat{z}_1 \le \hat{z}_2 \end{Bmatrix}.\end{aligned}$$

This can be analogously extended to higher dimensions. The result is that we can divide the integration domain of the common vertex case in only two subdomains.

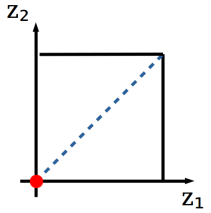

Fig. 2. Domain splitting for common vertex. Common vertex is the point at the origin and the integration domain is split along the diagonal.

Performing the relative coordinate transformation and the reordering of the integration variables on (1), gives us

$$I_{T\times T'} = \iint_{T\times T'} k'(\hat{\vec{x}}, \hat{\vec{z}}) d\hat{z} d\hat{x} = \sum_{i=1}^{N} \iint_{D_i} k'(\hat{\vec{x}}, \hat{\vec{z}}) d\hat{x} d\hat{z}, \qquad (4)$$

where the integral splits into N integrals over the subdomains D_i. For singular tetrahedron-tetrahedron interaction integrals we have in the case of 'Identical Tetrahedrons' that the change of integration variables results in a total of 18 subdomains ($N = 18$). For the 'Common Face' case we have $N = 15$, for the 'Common Edge' case we have $N = 5$ and for the 'Common Vertex' we have $N = 2$. The corresponding number of subdomains for the different cases in 5D and 4D can be found in Table IV.

TABLE IV
OVERVIEW OF NUMBER OF SUBDOMAINS AFTER SPLITTING

	Tet x Tet	**Tet x Tria**	**Tria x Tria**
Common Vertex	2	2	2
Common Edge	5	5	5
Common Face	15	9	6
Identical Tetrahedron	18		

Note that in (4) the inner integrals (integrals over $\hat{x}$) could be evaluated explicitly as demonstrated in [6]. Due to the relative coordinate transformation the singular behavior only happens in the $\hat{z}$ variable. We apply a numerical quadrature scheme on the full 6D or 5D integral to approximate it. This allows our method to be kernel independent, as opposed to an semi-analytic approach. In a semi-analytic algorithm the analytic part changes for each different kernel function. In addition our method is applicable with minimal modifications to integrals over curvilinear parametrized domains. In a semi-analytic algorithm, it gets soon too difficult and complex to compute the analytic part of the solution if one has to deal with curved integration domains.

B. Duffy-Transformation and Simplex tensor-product

After the transformation and change of integration order we map each integration domain to a reference domain. Many integration domains can be mapped to the same reference domains. When integration domains map to the same reference domain, we have to ensure that the singularities from all domains are mapped to the same vertex, edge or face of the reference domain. This allows us to only define quadrature rules on the reference domains and reuse them for multiple integration domains. For example, in the case of 'Identical Tetrahedrons' ($T = T'$) we will have a total of 18 integration domains. For the first 16 (out of 18) domains there exist linear mappings to the 6D simplex and for the last two domains there are linear mappings to a polytope with 8 corners. Instead of defining quadrature rules for 18 different integration domains we only need to define two rules, one on the 6D simplex and one on the polytope.

The integrals still have a singularity at $\hat{\vec{z}} = 0$. This singularity can be reduced by applying a Duffy-Transformation [7]. In the standard approach all the reference domains are transformed to the 6D hypercube. The determinant of the transformation's Jacobian cancels the singularity in the integrand. To evaluate the integrand over the hypercube, the most simple and straight-forward way is to build a 6D tensor-product quadrature from the 1D Gauss-Legendre quadrature rule (G_{1D}). Transforming a reference domain to the 6D hypercube and using the 6D Gauss tensor-product ($G_{1D} \times G_{1D} \times G_{1D} \times G_{1D} \times G_{1D} \times G_{1D}$) is very inefficient. If a 10-point 1D Gauss-Legendre quadrature rule is used to construct a 6D tensor-product one ends up with 10^6 quadrature points.

Thus, we are looking for better transformations that still remove the singularity but are less expensive. As it turns out, it is sufficient to inflate a domain only along some dimensions to remove the singularity with the Jacobian. But since the reference domain is no longer the hypercube, we can no longer use the tensor-product of 1D Gauss quadrature rules. Therefore, we look for other quadrature rules or combination of quadrature rules to cover the inflated domains. For an illustration of this concept in 3D see Fig. 3, where we can either inflate a tetrahedron to a cube or a prism. For the cube, we use a standard tensor-product of three 1D rules ($G_{1D} \times G_{1D} \times G_{1D}$) and for the prism we can combine a 1D rule with any 2D quadrature rule defined over a triangle ($G_{1D} \times S_{2D}$). For triangles, there are dedicated quadrature rules available that are significantly more efficient in terms of number of quadrature nodes for a given order when compared to quadrature rules that are built by pullback of tensor rules for the square to the triangle. The computation time required to compute the integrals decreases along with the number of quadrature points.

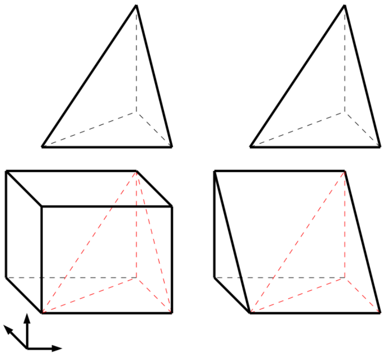

Fig. 3. Domain inflation in 3D to a cube (left) and only along 'necessary' dimension to a prism (right).

The same principle also applies for 6D integrals, but it is more difficult and requires more work. What is required, are dedicated quadrature rules for the 6D inflated domains. Due to their beneficial properties (positive weights, symmetry and locality), we implemented the family of quadrature methods that were developed and studied by Shunn et al. in [9], [10] and [11] for 2D, 3D and 4D simplices, respectively. The general idea behind these type of quadrature rules is that they are the optimal solution of a constraint optimization problem that minimizes the quadrature error of a polynomial of given order n while the constraints guarantee that the quadrature rule is exact for lower degree polynomials ($< n$). This means that the quadrature rule will exact up to degree $n-1$. We implemented a framework that allows us to generate these kind of rules. First we derive the symbolic expressions for the quadrature error of a given degree and all necessary constraints. Of course, these expressions additionally depend on the dimension of the simplex and the number of quadrature points in the simplex. Afterwards, we use the analytic expressions in combination with an optimization algorithm to find the optimal quadrature rules for the given set of parameters (dimension, number of quadrature points, etc.). Finding an optimal solution for this problem becomes harder and more expensive with increasing order and dimension. For more details on these kind of quadrature method and the detailed nature of the quadrature error expression and the constraints we refer to [11].

Now, we demonstrate how we find the right combination of quadrature rules that span the inflated domain. Here, we only cover the case of 'Identical Tetrahedrons'. The other cases are analogous. First, we assume that the integrand from (4) has a $1/r^2$ singularity and hence can be written in relative coordinates as

$$k'(\vec{x},\vec{z}) = \frac{f(\vec{x},\vec{z})}{z_1^2+z_2^2+z_3^2} \tag{5}$$

where f is an analytic function and the singular behavior in the denominator only depends on $\vec{z}$. Recall, that the reference domain D_{ref} for the domains D_{1-16} is the 6D simplex

$$\begin{aligned} D_{ref} = \{&0 \le x_1 \le 1; 0 \le x_2 \le x_1; 0 \le x_3 \le x_2; \\ &0 \le z_1 \le x_3; 0 \le z_2 \le z_1; 0 \le z_3 \le z_2\}^{.} \end{aligned} \tag{6}$$

We can inflate D_{ref} to the 6D hypercube and get the Duffy-Transformation between the D_{ref} and the 6D hypercube

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_1\rho_2 \\ \rho_1\rho_2\rho_3 \\ \rho_1\rho_2\rho_3\rho_4 \\ \rho_1\rho_2\rho_3\rho_4\rho_5 \\ \rho_1\rho_2\rho_3\rho_4\rho_5\rho_6 \end{bmatrix} \tag{7}$$

with $0 \le \rho_i \le 1$. Its Jacobian is

$$\rho_1^5\rho_2^4\rho_3^3\rho_4^2\rho_5. \tag{8}$$

Using (7) together with (5) results in the integral

$$\int \frac{f'(\rho_1,\rho_2,\rho_3,\rho_4,\rho_5,\rho_6)\rho_1^5\rho_2^4\rho_3^3\rho_4^2\rho_5}{(\rho_1\rho_2\rho_3\rho_4)^2+(\rho_1\rho_2\rho_3\rho_4\rho_5)^2+(\rho_1\rho_2\rho_3\rho_4\rho_5\rho_6)^2} d\vec{\rho} \tag{9}$$

We simplify the denominator with the determinant and see that the integrand is now analytic:

$$\int \frac{f'(\rho_1,\rho_2,\rho_3,\rho_4,\rho_5,\rho_6)\rho_1^3\rho_2^2\rho_3\rho_5}{1+(\rho_5)^2+(\rho_5\rho_6)^2} d\vec{\rho}. \tag{10}$$

In this case, it is more efficient if we only inflate along the forth dimension and use the transformation

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_4\rho_5 \\ \rho_4\rho_6 \end{bmatrix} \tag{11}$$

with $0 \le \rho_1 \le 1, 0 \le \rho_2 \le \rho_1, 0 \le \rho_3 \le \rho_2, 0 \le \rho_4 \le \rho_3$ and $0 \le \rho_5 \le 1, 0 \le \rho_6 \le \rho_5$. Its Jacobian is ρ_4^2. Using (11) together with (5) results in the integral

$$\int \frac{f((\rho_1,\rho_2,\rho_3),(\rho_4,\rho_4\rho_5,\rho_4\rho_6))}{\rho_4^2+(\rho_4\rho_5)^2+(\rho_4\rho_6)^2}\rho_4^2 d\vec{\rho} \tag{12}$$

which is again analytic.

We observe that ρ_1-ρ_4 span a pentatope (4D simplex) and ρ_5 and ρ_6 span a triangle (2D simplex). Thus, we use the combination of a 4D and a 2D simplex quadrature rule (S_{4D} and S_{2D}) on these simplices respectively to build a simplex tensor-product rule ($S_{4D} \times S_{2D}$) that spans the whole inflated domain. For other domains and other cases, similar Duffy-Transformations and combinations of quadrature rules can be defined and found.

If we combine all above steps, we will receive quadrature rules for singular interaction integrals. As an example for one of these quadrature rules, the quadrature points in case of two tetrahedron with a common edge are given in Table V, the tables for other rules can can be found at [12]. We recall that in the case of an common edge, we need to subdivide the integration domain in five parts, thus we have five Duffy-Transformations, one for each subdomain $D_1 - D_5$. For each subdomain we give the combination of the quadrature rules used and the Duffy-Transformation. The last column in Table V contains the coordinates of the quadrature point on the reference tetrahedron after reversing the relative coordinate

TABLE V
SIMPLEX TENSOR-PRODUCT QUADRATURE FOR TWO TETRAHEDRON WITH A COMMON EDGE. S^{Y}_{XD} IS AN X-DIMENSIONAL SIMPLEX QUADRATURE RULE WITH QUADRATURE POINTS $(p^{Y}_{1}, ..., p^{Y}_{X})$.

Domain	**Quad. Rule combination**	**Duffy-TransformationRef. Tet.**	**Quad. Points on**
D_1	$S^1_{2D} \times S^2_{2D} \times S^3_{2D}$	$\xi_1 = p^1_1$ $\xi_2 = p^1_2$ $\xi_3 = p^1_2 p^2_1$ $\eta_1 = p^1_2 p^2_2$ $\eta_2 = p^1_2 p^3_1$ $\eta_3 = p^1_2 p^3_2$	$x_1 = \xi_1$ $x_2 = \xi_2 - \eta_1$ $x_3 = \xi_3 - \eta_1$ $y_1 = \xi_1 - \xi_2 + \eta_2$ $y_2 = \eta_2$ $y_3 = \eta_2 - \eta_3$
D_2	$S^1_{2D} \times S^2_{3D} \times S^3_{1D}$	$\xi_1 = p^1_1$ $\xi_2 = p^1_2$ $\xi_3 = p^1_2 p^2_1$ $\eta_1 = p^1_2 p^2_2$ $\eta_2 = p^1_2 p^2_3$ $\eta_3 = p^1_2 p^3_1$	$x_1 = \xi_1$ $x_2 = \xi_2$ $x_3 = \xi_2 - \eta_3$ $y_1 = \xi_1 - \xi_2 + \xi_3$ $y_2 = \eta_1$ $y_3 = \eta_2$
D_3	$S^1_{2D} \times S^2_{4D}$	$\xi_1 = p^1_1$ $\xi_2 = p^1_2$ $\xi_3 = p^1_2 p^2_1$ $\eta_1 = p^1_2 p^2_2$ $\eta_2 = p^1_2 p^2_3$ $\eta_3 = p^1_2 p^2_4$	$x_1 = \xi_1 - \eta_1$ $x_2 = \xi_2 - \eta_1$ $x_3 = \xi_3 - \eta_1$ $y_1 = \xi_1$ $y_2 = \eta_2$ $y_3 = \eta_3$
D_4	$S^1_{2D} \times S^2_{1D} \times S^3_{3D}$	$\xi_1 = p^1_1$ $\xi_2 = p^1_2$ $\xi_3 = p^1_2 p^2_1$ $\eta_1 = p^1_2 p^3_1$ $\eta_2 = p^1_2 p^3_2$ $\eta_3 = p^1_2 p^3_3$	$x_1 = \xi_1 - \eta_3$ $x_2 = \eta_1 - \eta_3$ $x_3 = \eta_2 - \eta_3$ $y_1 = \xi_1$ $y_2 = \xi_2$ $y_3 = \xi_3$
D_5	$S^1_{2D} \times S^2_{1D} \times S^3_{2D} \times S^4_{1D}$	$\xi_1 = p^1_1$ $\xi_2 = p^1_2$ $\xi_3 = p^1_2 p^2_1$ $\eta_1 = p^1_2 p^3_1$ $\eta_2 = p^1_2 p^2_1 p^3_2$ $\eta_3 = p^1_2 p^3_1 p^4_1$	$x_1 = \xi_1 - \eta_2$ $x_2 = \xi_2 - \eta_2$ $x_3 = \xi_2 - \xi_3$ $y_1 = \xi_1$ $y_2 = \eta_1$ $y_3 = \eta_3$

transformation and the mapping to the reference subdomain for the Duffy-Transformation.

The complete implementation of the quadrature rules for the all cases can be found here [12].

III. NUMERICAL RESULTS

In this section, we use the quadrature method developed in the previous section to compute singular integrals and compare the accuracy of the Gauss tensor-product quadrature rule (standard) against the simplex tensor-product quadrature rule (new). We further analyze the efficiency of the simplex tensor-product compared to the Gauss tensor-product quadrature. In our experiments, we choose the generic $k(\vec{x}, \vec{y})$ in (1) to be

$$k(\vec{x}, \vec{y}) = (\vec{x} - \vec{P})^T \frac{e^{-ik|\vec{x}-\vec{y}|}}{4\pi|\vec{x} - \vec{y}|}(\vec{y} - \vec{Q}) \qquad (13)$$

where $\vec{P}$ and $\vec{Q}$ are corners of T or T', respectively. The reader might recognize the Green's function for the Helmholtz equation in (13). The terms $(\vec{x} - \vec{P})^T$ and $(\vec{y} - \vec{Q})$ can be thought as placeholders for test and trial functions that would arising in a Galerkin discretization and their particular shape has no major influence on the accuracy.

We consider different cases for the overlapping of the two tetrahedrons T and T'. The exact definitions used for T and T' in the experiments reported here can be found in the appendix as well as in [12]. We compute the relative error compared to a high accuracy version of the Gauss tensor-product method that uses a 15-point 1D Gauss rule as basis for the tensor-product. With 15 point in each dimension the standard Gauss tensor-product method reaches machine precision in 4D. In 6D the Gauss tensor-product method based on a 15-point 1D rule has precision of about 10^{-11}.

First, we consider again the case of 'Identical Tetrahedrons' ($T = T'$). Fig. 4 shows on the vertical axis the relative error. On the horizontal axis the number of quadrature points is shown. Here, the number of quadrature points corresponds to the number of evaluations of the integral kernel. The Gauss tensor-product quadrature rule and the simplex tensor-product quadrature rule are compared against the high accuracy version of the Gauss tensor-product quadrature rule. The errors of both methods behave similar, while the simplex tensor-product approach requires significantly fewer function evaluations to achieve the same level of accuracy. The similar behavior of the error can be expected, since both methods use the same technique (splitting into subdomains and Duffy-Transformation) to deal with the singularity in the kernel. The difference of number of quadrature points for the same accuracy can be explained by the following reasoning: The Gauss tensor-product method always inflates the integration subdomains to the 6D hypercube. In contrast, the Simplex tensor-product method inflates the subdomains to a 6D polytope (see Fig. 3 for visualization in 3D). Since this polytope usually only correspond to a significant smaller part of the hypercube, we need fewer quadrature points to cover the polytope with quadrature points than we need to cover the whole hypercube.

At the time of writing, the quadrature rules for 2D, 3D and 4D simplices have only been computed up to order 10, thus there are fewer measurements for the newer method depending on these simplex quadrature rules. Fig. 5 shows the relative error in the case where two tetrahedrons share a face. Again, the behavior of the errors are similar while the simplex tensor-product rules need fewer function evaluations for the same accuracy.

TABLE VI
TIMING AND NUMBER OF FUNCTION EVALUATIONS FOR A SINGLE INTEGRAL USING GAUSS TENSOR-PRODUCT QUADRATURE

Case	**Time (ms)**	**#Eval.**	**Rel. Error**
Identical Tetrahedron	13.282	$281'250$	8.91×10^{-6}
Common Face	11.456	$234'375$	4.26×10^{-6}
Common Edge	4.516	$78'125$	5.30×10^{-7}
Common Vertex	2.516	$31'250$	3.39×10^{-6}
Positive Distance	0.211	$4'096$	3.22×10^{-7}

To obtain a clearer picture of the efficiency gain for the simplex tensor-product rule in the case of tetrahedron-tetrahedron

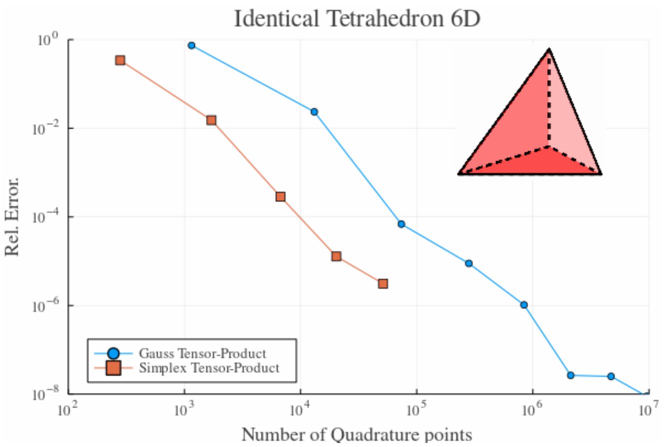

Fig. 4. Relative error convergence for overlapping tetrahedron using Gauss tensor-product and simplex tensor-product quadrature rules. Both methods converge at the same rate, while the simplex tensor-product method needs fewer function evaluations to achieve the same accuracy.

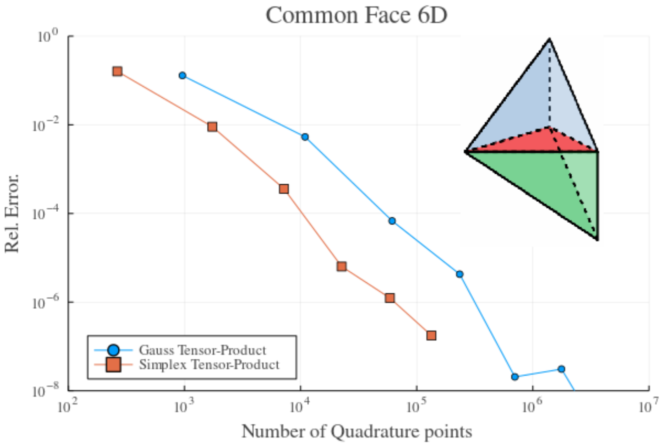

Fig. 5. Relative error convergence for overlapping face of tetrahedrons using Gauss tensor-product and simplex tensor-product quadrature rules. Both methods converge at the same rate, while the simplex tensor-product method needs fewer function evaluations to achieve the same accuracy.

TABLE VII
TIMING AND NUMBER OF FUNCTION EVALUATIONS FOR A SINGLE INTEGRAL USING SIMPLEX TENSOR-PRODUCT QUADRATURE

Case	Time (ms)	#Eval.	Rel. Error
Identical Tetrahedron	0.897	20′300	1.28×10^{-5}
Common Face	1.092	22′575	6.36×10^{-6}
Common Edge	2.107	41′895	4.28×10^{-7}
Common Vertex	0.302	6′272	4.94×10^{-6}
Positive Distance	0.020	400	9.38×10^{-7}

TABLE VIII
EFFICIENCY OF SIMPLEX TENSOR-PRODUCT QUADRATURE COMPARED TO GAUSS TENSOR-PRODUCT QUADRATURE FOR TWO TETRAHEDRONS

Case	Time (%)	#Eval (%)
Identical Tetrahedron	6.75	7.22
Common Face	9.53	9.632
Common Edge	46.66	53.63
Common Vertex	12.01	20.07
Positive Distance	9.47	9.77

interactions, we measured the time and the number of function evaluations for roughly the same relative error, namely 10^{-6}. The results for the different cases are summarized in Table VI for Gauss tensor-product and in Table VII for the simplex tensor-product method. For all time measurements an Intel® Core™ i7-1185G7 @ 3.00GHz CPU with 12MB of cache was used. Table VIII shows the relative efficiency in time and function evaluations of the simplex tensor-product over the Gauss tensor-product. In most cases, the number of function evaluations can be directly linked to the computation times. The speedup in time and the reduction of function evaluation ranges from a factor of 2.1 to 14.8. The largest speedups are gained in the case of 'Identical Tetrahedrons'.

The speedup for a 'Common Edge' is much lower. In this case, the subdomains are more complicated, and therefore the reference domains too. They can not be spanned efficiently by simplex tensor-product rules like in the other cases. The simplex tensor-product contains more factors.

The positive distance case is included for the sake of completeness, but there is no need to inflate the domain and thus it makes no sense to use a 6D tensor-product quadrature rule here. One can use a simple quadrature rule for each tetrahedron separately.

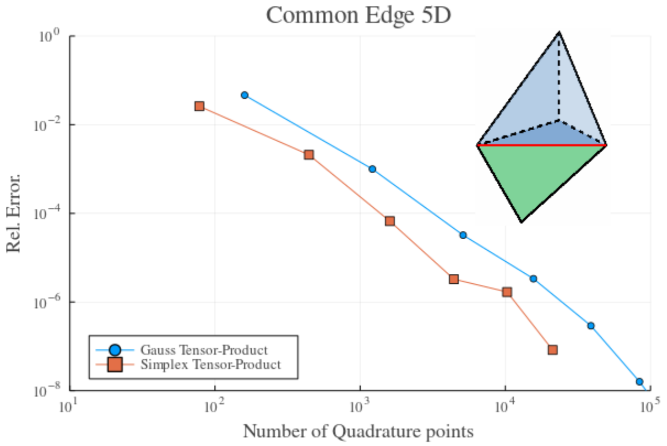

Fig. 6. Relative error convergence for tetrahedron (blue) and triangle (green) with common edge (red) using Gauss tensor-product and simplex tensor-product quadrature rules. Both methods converge very similar, while the simplex tensor-product method again needs fewer function evaluations to achieve the same accuracy.

TABLE IX
TIMING AND NUMBER OF FUNCTION EVALUATIONS FOR A SINGLE INTEGRAL USING GAUSS TENSOR-PRODUCT QUADRATURE FOR TETRAHEDRON AND TRIANGLE

Case	Time (ms)	#Eval.	Rel. Error
Common Face	1.253	28′125	6.35×10^{-6}
Common Edge	0.710	15′625	6.56×10^{-6}
Common Vertex	0.298	6′250	4.10×10^{-6}
Positive Distance	0.051	1′024	1.72×10^{-7}

Of course, the simplex tensor-product method can also be used in the case of tetrahedron-triangle interaction. We did the same measurements as in the previous experiment in 6D. Fig. 6 and 7 show the relative error using an increasing number

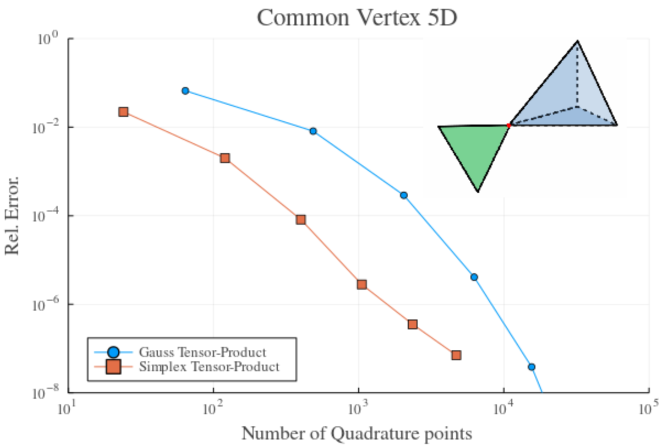

Fig. 7. Relative error convergence for tetrahedron and triangle with common vertex using Gauss tensor-product and simplex tensor-product quadrature rules. Both methods converge at the similar rate, while the simplex tensor-product method again needs fewer function evaluations to achieve the same accuracy.

TABLE X
TIMING AND NUMBER OF FUNCTION EVALUATIONS FOR A SINGLE INTEGRAL USING SIMPLEX TENSOR-PRODUCT QUADRATURE FOR TETRAHEDRON AND TRIANGLE

Case	Time (ms)	#Eval.	Rel. Error
Common Face	0.231	5′325	8.87×10^{-6}
Common Edge	0.203	4′425	2.72×10^{-6}
Common Vertex	0.055	1′050	2.81×10^{-6}
Positive Distance	0.009	200	2.96×10^{-5}

of quadrature points for a Gauss tensor-product quadrature rule and the simplex tensor-product quadrature rule in the case of a triangle and tetrahedron sharing an edge or an vertex, respectively. The error of both methods behaves again similar, while the simplex tensor-product approach requires significantly fewer function evaluations to achieve the same level of accuracy. The relative speedup is not as not as large as in 6D but the speed up still ranges between 3.5 and 5.7, see Table XI. We observe the best improvements in the case of 'Common Face' and a 'Common Vertex'. The reduced efficiency in 5D can be explained by that we still need to inflated the same number of dimension to remove the singularity with the Duffy-Transformation as in the 6D case, thus leaving fewer dimension that aren't inflated. This lowers the ratio between inflated and uninflated dimension in 5D compared to 6D. In the end, this leads to a lower efficiency of the new method in 5D. An even lower efficiency of the new method can be observed when we look at the results from 4D next.

TABLE XI
EFFICIENCY OF SIMPLEX TENSOR-PRODUCT QUADRATURE COMPARED TO GAUSS TENSOR-PRODUCT QUADRATURE FOR TRIANGLE AND TETRAHEDRON

Case	Time (%)	#Eval (%)
Common Face	18.41	18.93
Common Edge	28.53	28.32
Common Vertex	18.35	16.80
Positive Distance	17.57	19.53

TABLE XII
TIMING AND NUMBER OF FUNCTION EVALUATIONS FOR A SINGLE INTEGRAL USING GAUSS TENSOR-PRODUCT QUADRATURE FOR TWO TRIANGLES

Case	Time (ms)	#Eval.	Rel. Error
Identical Triangle	0.335	7′776	3.92×10^{-6}
Common Edge	0.278	6′480	2.51×10^{-7}
Common Vertex	0.115	1′296	1.73×10^{-7}
Positive Distance	0.011	256	6.20×10^{-9}

TABLE XIII
TIMING AND NUMBER OF FUNCTION EVALUATIONS FOR A SINGLE INTEGRAL USING SIMPLEX TENSOR-PRODUCT QUADRATURE FOR TWO TRIANGLES

Case	Time (ms)	#Eval.	Rel. Error
Identical Triangle	0.089	2′016	3.92×10^{-6}
Common Edge	0.117	2′520	3.28×10^{-7}
Common Vertex	0.039	441	2.57×10^{-7}
Positive Distance	0.005	100	3.26×10^{-8}

TABLE XIV
EFFICIENCY OF SIMPLEX TENSOR-PRODUCT QUADRATURE COMPARED TO GAUSS TENSOR-PRODUCT QUADRATURE FOR TWO TRIANGLES

Case	Time (%)	#Eval (%)
Identical Triangle	26.44	25.93
Common Edge	42.03	38.89
Common Vertex	33.75	34.03
Positive Distance	45.45	39.06

If we applied the simplex tensor-product in the triangle-triangle interaction integrals that are very common in Surface Integral Equations, we can still see an improvement between 2.2 up to 3.8 times faster (Table XIV) than if we employ standard Gauss tensor-product quadrature.

The exact speed up depends highly on the geometric arrangement of the involved elements. In general, the simplex tensor-product can increase the evaluation speed of singular integrals in 6D by a factor of 10, in 5D by a factor of 5 and in 4D by a factor of 3 while achieving the same accuracy as the method based on Gauss tensor-product quadrature.

IV. Conclusion

In this work, we extended the approach for singular triangle-triangle interaction integrals for Surface Integral Equations to singular tetrahedron-tetrahedron and tetrahedron-triangle interaction integrals that appear in Volume Integral Equations. Finally, we showed that with careful selection, which dimensions to inflate using a Duffy-Transformation, we can reduce the number of function evaluations by a factor varying from 2.1 to 14.8 in 6D. Also singular integrals in surface Integral Equations are evaluated up to 3 times faster with this new approach.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 101001847).

Appendix

Coordinates used in the numerical experiments

TABLE XV
Corners of tetrahedrons for tetrahedron-tetrahedron interactions and choices of P, Q in the integral kernel.

Identical Tetrahedron	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_1$
$T' = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$Q = T_3'$
Common Face	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_1$
$T' = \{(0,0,0),(1,0,0),(0,1,0),(0,0,-1)\}$	$Q = T_3'$
Common Edge	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_1$
$T' = \{(0,0,0),(1,0,0),(0,-1,0),(0,0,-1)\}$	$Q = T_3'$
Common Vertex	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_1$
$T' = \{(0,0,0),(-1,0,0),(0,-1,0),(0,0,-1)\}$	$Q = T_3'$
Positive Distance	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_1$
$T' = \{(10,0,0),(9,0,0),(10,-1,0),(10,0,-1)\}$	$Q = T_3'$

TABLE XVI
Corners of tetrahedron and triangle for tetrahedron-triangle interaction and choices of P, Q in the integral kernel.

Common Face	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_4$
$T' = \{(0,0,0),(1,0,0),(0,1,0)\}$	$Q = T_1'$
Common Edge	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_4$
$T' = \{(0,0,0),(1,0,0),(0,-1,0)\}$	$Q = T_1'$
Common Vertex	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_4$
$T' = \{(0,0,0),(-1,-1,0),(0,-1,0)\}$	$Q = T_1'$
Positive Distance	
$T = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	$P = T_4$
$T' = \{(10,0,0),(9,0,0),(10,-1,0)\}$	$Q = T_1'$

TABLE XVII
Corners of triangles for triangle-triangle interaction and choices of P, Q in the integral kernel.

Identical triangle	
$T = \{(0,0,0),(1,0,0),(0,1,0)\}$	$P = T_2$
$T' = \{(0,0,0),(1,0,0),(0,1,0)\}$	$Q = T_3'$
Common Edge	
$T = \{(0,0,0),(1,0,0),(0,1,0)\}$	$P = T_1$
$T' = \{(0,0,0),(1,0,0),(0,-1,0)\}$	$Q = T_2'$
Common Vertex	
$T = \{(0,0,0),(1,0,0),(0,1,0)\}$	$P = T_2$
$T' = \{(0,0,0),(-1,0,0),(0,-1,0)\}$	$Q = T_3'$
Positive Distance	
$T = \{(0,0,0),(1,0,0),(0,1,0)\}$	$P = T_2$
$T' = \{(10,0,0),(9,0,0),(10,-1,0)\}$	$Q = T_3'$

References

[1] J. Markkanen, P. Yla-Oijala and A. Sihvola, “Discretization of Volume Integral Equation Formulations for Extremely Anisotropic Materials,“ in IEEE Transactions on Antennas and Propagation, vol. 60, no. 11, pp. 5195-5202, Nov. 2012.

[2] E. H. Bleszynski, M. K. Bleszynski and T. Jaroszewicz, “Reduction of Volume Integrals to Nonsingular Surface Integrals for Matrix Elements of Tensor and Vector Green Functions of Maxwell Equations,“ in IEEE Transactions on Antennas and Propagation, vol. 61, no. 7, pp. 3642-3647, July 2013.

[3] J. Rivero, F. Vipiana, D. R. Wilton and W. A. Johnson, “Evaluation of 6-D Reaction Integrals via Double Application of the Divergence Theorem,“ in IEEE Transactions on Antennas and Propagation, vol. 70, no. 5, pp. 3523-3537, May 2022.

[4] C. Münger and K. Cools, “Efficient and kernel-independent Evaluation of Singular Integrals in Volume Integral Equations,“ 2021 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), 2021, pp. 188-192.

[5] D. J. Taylor, “Accurate and efficient numerical integration of weakly singular integrals in Galerkin EFIE solutions,“ IEEE Transactions on Antennas and Propagation, vol. 51, no. 7, pp. 1630-1637, July 2003.

[6] M. T. H. Reid, “Taylor–Duffy Method for Singular Tetrahedron-Product Integrals: Efficient Evaluation of Galerkin Integrals for VIE Solvers,“ in IEEE Journal on Multiscale and Multiphysics Computational Techniques, vol. 3, pp. 121-128, 2018.

[7] M. Duffy, “Quadrature Over a Pyramid or Cube of Integrands with a Singularity at a Vertex,“ SIAM Journal on Numerical Analysis, vol. 19. no. 6, pp. 1260-1262, 1982.

[8] S. Sauter and C. Schwab, “Generating the Matrix Coefficients,“ Boundary Element Methods, pp 289-352, 2011

[9] L. Shunn and F. Ham, “‘Symmetric quadrature rules for tetrahedra based on a cubic close-packed lattice arrangement,“ in Journal of Computational and Applied Mathematics, vol. 236, Issue 17, pp. 4348-4364, 2012.

[10] D. Williams, L. Shunn and A. Jameson, “Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements,“ Journal of Computational and Applied Mathematics, Vol. 266, pp. 18-38, 2014.

[11] D. Williams, C. Frontin, E. Miller and D. Darmofal, “A family of symmetric, optimized quadrature rules for pentatopes,“ Computers & Mathematics with Applications, vol. 80, Issue 5, pp. 1405-1420. 2020.

[12] *https://github.com/cmuenger/SauterSchwab3D.jl*