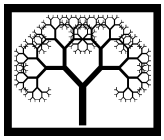
Paper 118



Proceedings of the Seventh International Conference
on Computational Structures Technology,
B.H.V. Topping and C.A. Mota Soares (Editors),
Civil-Comp Press, Stirling, Scotland.

Numerical Integration of Interpolation and Test Functions on any Convex Polyhedrons

T. Fröbel and P. Milbradt
Institute of Computer Science in Civil Engineering
University of Hannover, Germany

Abstract

The use of convex polyhedrons as basis of finite elements is enabled by the definition of a local coordinate system called the natural element coordinates [2]. With natural coordinates the formulation of interpolation and test functions is geometrically independent. In the case of finite element approximation it is necessary to differentiate and integrate these interpolation functions. In the paper especially the properties of the natural element coordinates are considered. The methods of the numeric implementation are introduced and advantages and disadvantages are described.

Keywords: convex polyhedrons, natural element coordinates, quadrature formulas.

1 Introduction

The numerical simulation of physical processes in science, nature and engineering is part of numerous engineering applications and constitutes the fundament of engineering decisions. Often the mathematical description of the physical model is based on partial differential equations. The analytical solution of these equations is predominantly not or only with extensive complexity computable. The use of numerical approximation procedures provides a tool to calculate an approximated solution for these equations.

The finite element method is such a numerical approximation procedure and allows an interpolation of given values or a numerical approximation of solutions of partial differential equations. The basis of the finite element method is the formulation of eligible finite elements and element decompositions. The considered investigation area is decomposed in subregions. As geometric fundament for the decomposition triangles and quadrangles in the plane as well as tetra- and hexahedrons in a three dimensional space are used. In addition to the geometrical

fundament finite elements have a set of interpolation functions and a set of degrees of freedom.

The extension of the geometrical fundament to any *n*-dimensional convex polyhedrons permits an easy and flexible decomposition of the investigation area. The definition of a local coordinate system called the natural element coordinates makes a description of interpolation functions possible.

2 Geometrical Basis

The geometrical basis of the traditional finite element method is build with edges in a one dimensional space, with triangles and quadrangles in the plane as well as with tetra- and hexahedrons in a three dimensional Euclidian space. The extension to convex polyhedrons can be understood as a generalisation of the geometrical elements and the finite element method.

2.1 Convex Polyhedrons

The description and definition of convex polyhedrons *Z* can be done by several ways.

As construction basis for instructions and algorithms it is mostly adequate to define a convex polyhedron with reference points of the convex hull $E = \{e^1, e^2, ..., e^N\}$ [1]. The set of vertices *E* can be visualised as an intersection of half spaces concerning a default set of reference points $P = \{p^1, p^2, ..., p^N\}$.

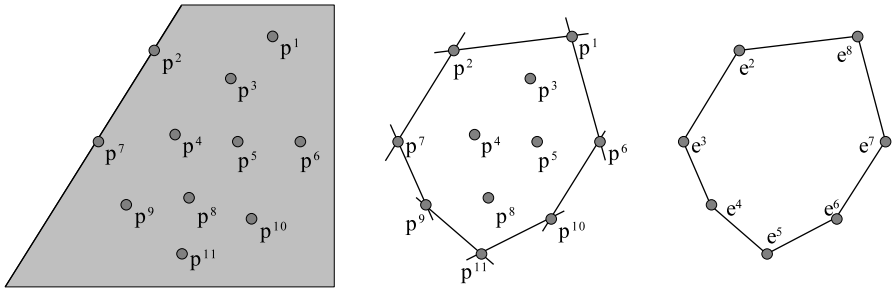


Figure 1: Half space, intersection of half spaces, convex polyhedron

As basis for finite elements a convex polyhedron can be defined conveniently via the Minkowsky-product:

$$Z := \{p : p = \lambda_1 e^1 + \lambda_2 e^2 + ... + \lambda_N e^N, \lambda_i \geq 0 \wedge \sum_i \lambda_i = 1\}$$

The dimension *m* of the convex polyhedron is defined by the number of *m* linearly independent points in the set of the vertices *E*. The *m*-dimensional convex polyhedron is composed by a finite set of regions. Every region itself is a convex polyhedron, too. The 0-dimensional regions are called vertices, the 1-dimensional regions edges an the (*m-1*)-dimensional regions facets.

For the use of convex polyhedrons as a basis for finite elements a uniform description of all points of the finite element is necessary. This is achieved with the implementation of the natural element coordinates.

2.2 Natural Element Coordinates

The definition of a convex polyhedron Z via the Minkowsky product defined by the vertices E allows a description of local coordinates with the coefficients λ_i of the linear combination. A m-dimensional convex polyhedron with m+1 linearly independent vertices gets unique coefficients, known as barycentric coordinates.

If more than m+1 vertices exists the coefficients are not unique. A modification on natural environment coordinates of Sibson [6] achieved natural element coordinates which are in relationship with vertices of the convex polyhedron.

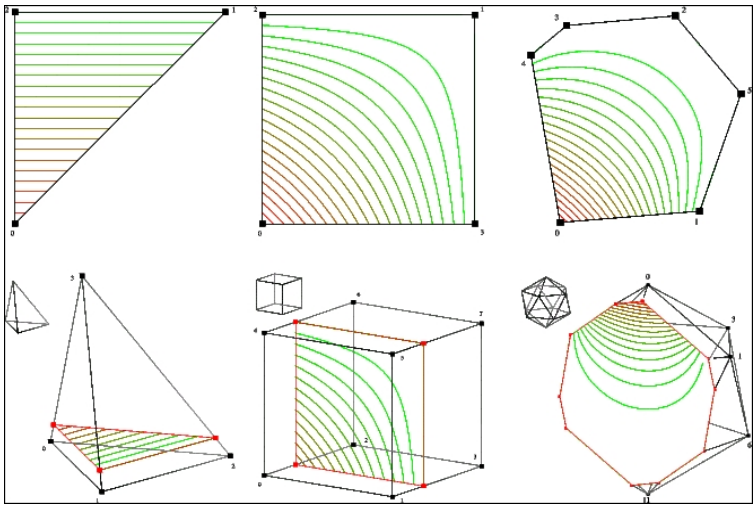


Figure 2: Natural element coordinates on convex polyhedrons

To assess natural element coordinates for a point x the Voronoi decomposition of first order is computed according to the vertices e^i of the convex polyhedron. Every vertex can be assigned to a Voronoi region $R(e^i)$. A Voronoi region in terms of a vertex is the set of all points where the distance d to the vertex is less than or equal the distance to all other vertices in E.

$$R(e^i) := \{p \in \Re^n : d(p,e^i) \leq d(p,e^j) \forall e^i, e^j \in \mathrm{E} \wedge j \neq i\}$$

A Voronoi region $R(x)$ of first order of a point x of the convex polyhedron can be expressed as follows:

$$R(x) := \{p \in \Re^n : d(p,x) \leq d(p,e^i) \forall e^i \in \mathrm{E}\}$$

Additionally, the Voronoi regions of second order are computed according to the vertices of the convex polyhedron and the point *x*. A Voronoi region of second order in terms of point *x* and vertex e^i is the set of all points where the distance *d* from point *x is* less than or equal the distance to each vertices e^i and where the distance *d* to this vertex is less than or equal the distance to the vertices e^j.

$$R(x,e^i) := \{p \in \Re^n : d(p,x) \leq d(p,e^i) \leq d(p,e^j) \forall e^i, e^j \in \mathrm{E} \wedge j \neq i\}$$

A Lebesgue measure $\mu(R(e^i))$ and $\mu(R(x,e^i))$ can be computed for Voronoi regions of first and second order. In the 2-dimensional space the value μ can be interpreted as area of the region.

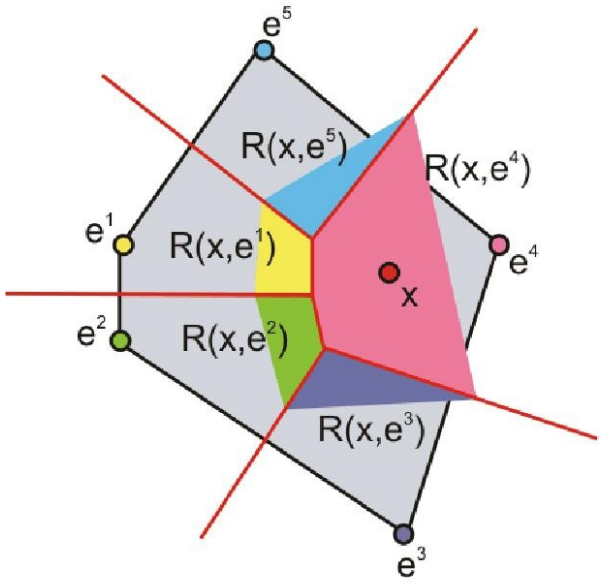


Figure 3: Computation of natural element coordinates

The natural element coordinates λ_i can be computed as the ratio of the Lebesgue measure of the Voronoi regions $R(x,e^i)$ and $R(x)$.

$$\lambda_i(x,e^i) := \frac{\mu(R(x,e^i))}{\mu(R(x))}$$

The natural element coordinate is undefined if point *x* lies outside the convex polyhedron.

Furthermore, if a point x is located on a facet of the convex polyhedron the measure of the Voronoi regions of second order become infinite. In this case the natural element coordinates depend only on the vertices of the facet.

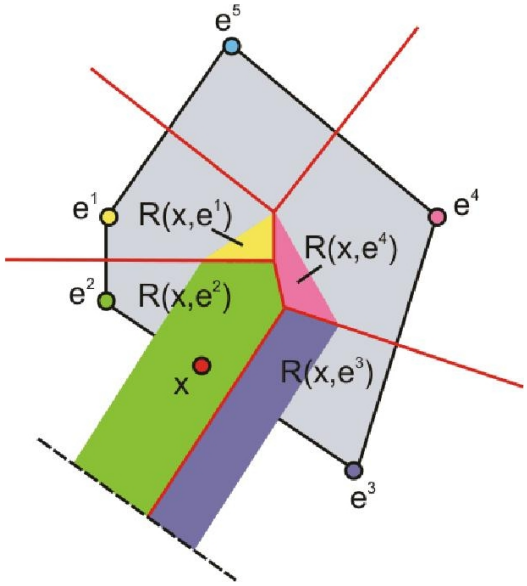


Figure 4: Computation of natural element coordinates on a facet

With the use of the Voronoi regions of second order the natural element coordinates do not necessarily depend on all vertices of the convex polyhedron. The coordinates of those vertices without influence become zero.

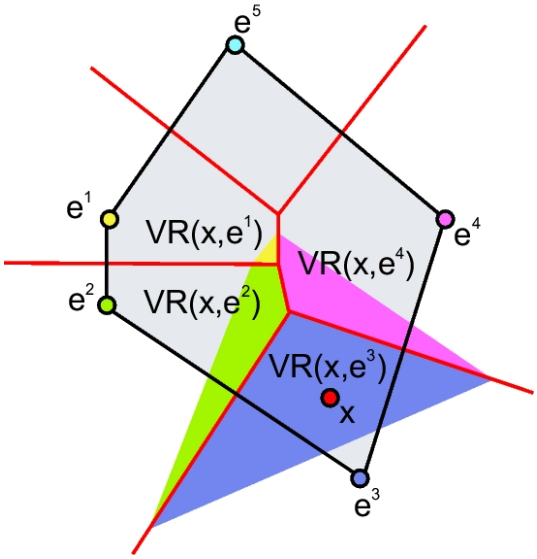


Figure 5: Effect of natural element coordinates

The natural element coordinates can be understood as generalised barycentric coordinates. The extension of convex polyhedrons to parametric cells [3] allows to handle curved borders.

3 Finite Elements

A finite element FE can be understood as a triple consisting of a geometrical basis GE , a set of degrees of freedom Θ and a set of interpolation functions Φ.

$$FE := (GE, \Theta, \Phi)$$

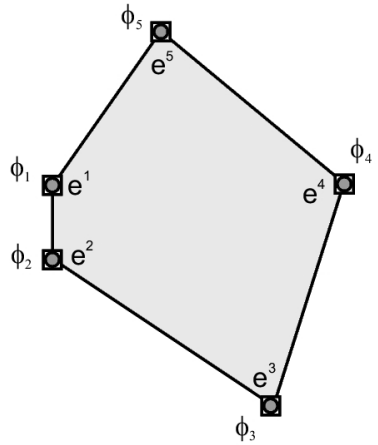
The complete description of complex problems will be realised with a set of simple interpolation functions with unknown parameters for subregions (finite elements) of an element decomposition. The solution of a differential equation can be approximated with the solution of corresponding algebraic system of equations. A degree of freedom is normally composed of a point (element of the geometrical basis), an interpolation function and a value. The corresponding interpolation function is described by natural element coordinates of the geometrical basis and the value of one is assigned to the corresponding degree of freedom.

3.1 Interpolation Functions

The computation of an approximated solution u_h of an exact solution u of any partial differential equation can be achieved via the linear combination of the interpolation functions. Generally, the variable u_i is a placeholder for the values of the degrees of freedom.

$$u(x) \approx u_h(x) = \sum_{i=1}^{n} \Phi_i(x) u_i$$

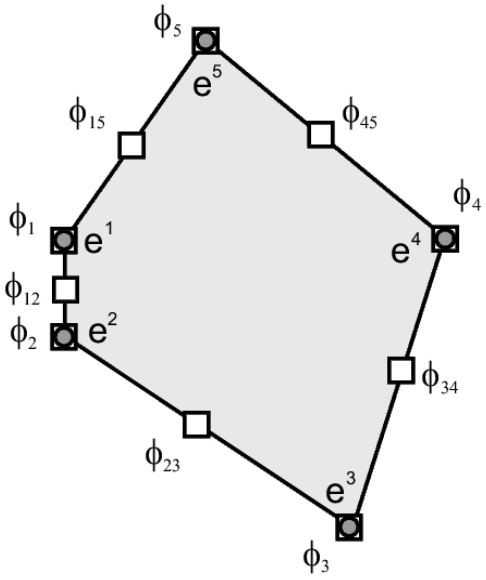
A simple formulation of interpolation functions is defined on the edges of the convex polyhedron [4]. In the following, Lagrangian functions will be considered and defined in natural element coordinates of the convex polyhedron. Due to edge-linear considerations the interpolation function is based exclusively on natural element coordinates of the associated vertices.



$$\Phi_1(\lambda) := \lambda_1$$
$$\Phi_2(\lambda) := \lambda_2$$
$$\Phi_3(\lambda) := \lambda_3$$
$$\Phi_4(\lambda) := \lambda_4$$
$$\Phi_5(\lambda) := \lambda_5$$

Figure 6: Edge-linear Lagrangian function

For edge-quadratic considerations a new degree of freedom is introduced in middle of the edges. Within the formulation of interpolation functions at the vertices those middle degrees of freedom are considered half on each incident edge. For the interpolation functions at these nodes the vertices and a coefficient are considered.



$$\Phi_1(\lambda) := \lambda_1 - 1/2\Phi_{12} - 1/2\Phi_{15}$$
$$\Phi_2(\lambda) := \lambda_2 - 1/2\Phi_{23} - 1/2\Phi_{12}$$
$$\Phi_3(\lambda) := \lambda_3 - 1/2\Phi_{34} - 1/2\Phi_{23}$$
$$\Phi_4(\lambda) := \lambda_4 - 1/2\Phi_{45} - 1/2\Phi_{34}$$
$$\Phi_5(\lambda) := \lambda_5 - 1/2\Phi_{15} - 1/2\Phi_{45}$$

$$\Phi_{12}(\lambda) := 4\lambda_1\lambda_2$$
$$\Phi_{23}(\lambda) := 4\lambda_2\lambda_3$$
$$\Phi_{34}(\lambda) := 4\lambda_3\lambda_4$$
$$\Phi_{45}(\lambda) := 4\lambda_4\lambda_5$$
$$\Phi_{15}(\lambda) := 4\lambda_1\lambda_5$$

Figure 7: Edge-quadratic Lagrangian function

The considered interpolation functions allow a c^0-continuous interpolation on decompositions consisting of polyhedrons and parametric cells.

4 Numerical Integration

Within the approximation of partial differential equations based on finite elements it is possible to generate integral equations whose integrals generally are not analytical computable.

Methods for numerical integration also referred as quadrature are essential for the numerical solution of differential equations. The main problem is the computation of the integral

$$\int_{\Omega} f(x)d\Omega$$

over a region Ω. Depending on the used method the complexity of the numerical integration is variable. The methods are defined on given integration regions. The integration of any functions over an integration region (finite element) are the basis of finite element interpolation and approximation methods [5].

4.1 Riemann Integral

The concept of the Riemann integral is to compute the integral of a function defined over a complex region via an approximation of the complex region by easier subregions and local function values. Easier subregions are intervals in the 1-dimensional case, rectangles in the 2-dimensional case, cuboids in the 3-dimensional case and "*n*-dimensional" cuboids in the *n*-dimensional case.

In the **1-dimensional** case a function is defined by $f:[a,b]\subset\Re\to\Re$. A set of points of an interval [a , b] with $a=x_1<x_2<...x_N<x_{N+1}=b$ can referred as a decomposition $Z:=\{x_1,x_2,...,x_{N+1},\}$. The decomposition Z and the function f enables to compute a lower- and upper sum as follows:

$$U(Z,f):=\sum_{j=1}^{N}(x_{j+1}-x_j)\cdot\min_{x\in[x_j,x_{j+1}]}\{f(x)\}$$

and

$$O(Z,f):=\sum_{j=1}^{N}(x_{j+1}-x_j)\cdot\max_{x\in[x_j,x_{j+1}]}\{f(x)\}$$

The combined limit *R* for Δx towards zero named as Riemann integral

$$R=\int_a^b f(x)dx \text{ with } R:=U(Z,f)=O(Z,f).$$

A Function is called Riemann-integrable if the combined limit above exists.

In the ***n*-dimensional** case a scalar function over a "*n*-dimensional" cuboid is defined by $f:[a,b]\subset\Re^n\to\Re$. The design of the cuboid *B* can be realised via intervals on the *N*-axes $B=[a_1,b_1]\times...\times[a_N,b_N]$. The partition of intervals in subintervals

$$a_k=x_1^{(k)}<x_2^{(k)}<...<x_{n_k+1}^{(k)}=b_k \text{ with } k=1,...,N$$

allows the definition of subcuboids as follows:

$$B_{sub\ l_1,...,l_n}=[x_{l_1}^{(1)},x_{l_1+1}^{(1)}]\times[x_{l_2}^{(2)},x_{l_2+1}^{(2)}]\times...\times[x_{l_N}^{(N)},x_{l_N+1}^{(N)}] \text{ with } k=1,...,N_K$$

The distance between the single subintervals can be variegate. The set of all subcuboids S_q build the cuboid B. The lower- and upper sum can be computed as follows:

$$U(B,f) := \sum_{B_{sub\,l_1,\ldots,l_n} \in S_q} vol(B_{sub\,l_1,\ldots,l_n}) \cdot \min_{B_{sub} \in B_{sub\,l_1,\ldots,l_n}} \{f(B_{sub})\}$$

$$O(B,f) := \sum_{B_{sub\,l_1,\ldots,l_n} \in S_q} vol(B_{sub\,l_1,\ldots,l_n}) \cdot \max_{B_{sub} \in B_{sub\,l_1,\ldots,l_n}} \{f(B_{sub})\}$$

4.2 Numerical Integration via Refinement

The definition of the Riemann integral shows a possible numerical approximation. With a gradually refinement of the subregions the accuracy level of the approximation can be increased.

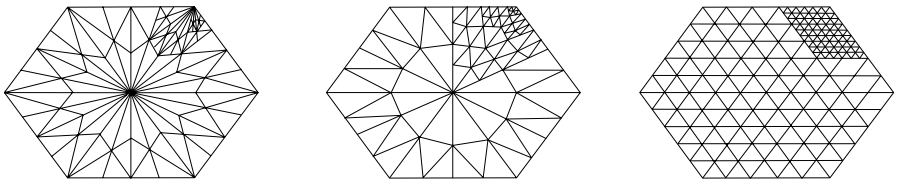


Figure 8: Different refinement methods for triangles (simplexes)

The right choice of geometrical shape for these subregions allows both the exacting boundary description of the convex polyhedron and further refinements. The description of subregions via simplexes has proved to be advantageous.

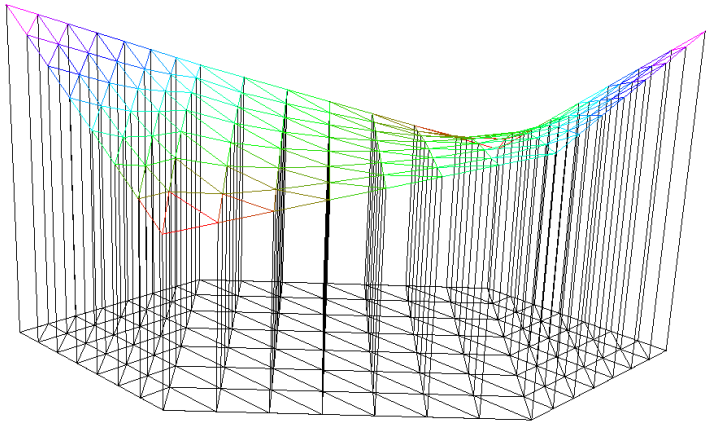


Figure 9: Integration on a hexagon via prisms

According to the definition of Riemann integral the approximated solution relating to an adequate level of refinement will present the solution of reference.

4.3 Quadrature Formulas

According to the geometry, the quadrature is interpreted as a graphical meta–morphosis of surfaces to coextensive squares. Regarding the mathematics, the quadrature is interpreted as a computation of surfaces via integral calculus. The numerical quadrature enabled an approximated computation of Riemann integrals.

$$\int_a^b f(x)dx$$

The introduction of the numerical quadrature is shown below. An extension to the *n*-dimensional case is possible in an analogue way. At first simple considerations specify a polynomial or a piecewise polynomial approximation function of *f*. Afterwards, the integral of the polynomial or the spline can be computed exactly. In addition to simple considerations more effective methods and possibilities for error estimation and failure recording exists.

For an approximated computation of Riemann integrals via the quadrature the quadrature formula *Q*

$$Q_{N+1}(f; w^{(N)}, x^{(N)}) = \sum_{j=0}^{N} w_j^{(N)} f(x_j^{(N)})$$

is used. The coefficients $w_j^{(N)}$ describe the weights and the coefficients $x_j^{(N)}$ describe the supporting points with $j = 0,...,N$. A sequence of such quadrature formulas define a quadrature method.

An error estimation makes an introduction of an quadrature error *R* necessary. The term *R* allows to give statements about convergence criterions of quadrature methods.

$$R_{N+1}(f; w^{(N)}, x^{(N)}) := \int_a^b f(x)dx - \sum_{j=0}^{N} w_j^{(N)} f(x_j^{(N)})$$

and

$$w^{(N)} := (w_0^{(N)},...,w_N^{(N)}) \ , \ x^{(N)} := (x_0^{(N)},...,x_N^{(N)})$$

The formulation of the quadrature formula depends on the choice of supporting points and whose weights.

Newton-Cotes ascertaines the weights on given supporting points, Chebyshev computes the supporting points on given weights and Gauss calculates both supporting points and weights.

4.3.1 Newton-Cotes Formulas

The concept of Newton-Cotes formulas is to find an approximated interpolation polynomial *P* for the supporting points and afterwards an exact integration of *P*. The description of *P* effected by Lagrange polynomials.

$$P(x) = \sum_{j=0}^{N} L_j(x) f(x_j) \quad \text{and} \quad L_j(x) = \prod_{i=0}^{N} \frac{x - x_i}{x_j - x_i} \quad , i \neq j$$

The integral can be described as follows:

$$\int_a^b P(x)dx = \sum_{j=0}^{N} f(x_j) \int_a^b L_j(x)dx \tag{1}$$

The supporting points $x_j = a + t(b-a)$ on an interval [a , b] can be predetermined with optional values. The parameter t allows a consideration of the ends of an interval.

Parameter t	Interval	Name
$t = \frac{j}{N} \quad , j = 0,..., N$	interval is included	Newton-Cotes formulas of the closed type
$t = \frac{j+1}{N+2} \quad , j = 0,..., N$	interval is not included	Newton-Cotes formulas of the open type

The computation of the integral is described via a sum as follows:

$$\int_a^b f(x)dx \approx (b-a) \sum_{j=0}^{N} w_j^{(N)} f(x_j^{(N)}) \tag{2}$$

A formula to compute the weights $w_j^{(N)}$ can be obtained by equate (1) and (2) as follows:

$$w_j^{(N)} = \frac{1}{b-a} \int_a^b L_j(x)dx$$

If it is difficult to integrate boundary points, Newton-Cotes formulas of the open type are used. For Newton-Cotes formulas of the closed type ($N \geq 8$) and for the open type ($N \geq 2$) alternating algebraic signs occur on the weights. The alternating algebraic signs are the reason for rounding errors. Therefore Newton-Cotes formulas are used only for small *N* on fragmented intervals of [a , b].

4.3.2 Gauss Quadrature

Until now, numerical integration formulas have been considered to determine suitable weights on given supporting points. Instead of this, the Gauss quadrature tries to determine both supporting points and weights optimal. The choice of a nonnegative weight function $\lambda(x)$ on an interval [a , b] allows to extend an integration on infinite integration intervals.

$$\int_a^b \lambda(x) f(x) dx \approx \sum_{j=0}^{N} w_j^{(N)} f(x_j^{(N)})$$

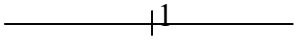
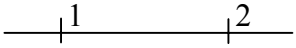
The weights w_j already include the coefficient $(b-a)$ of Newton-Cotes formulas. In the literature can be found following weight functions:

Interval [a , b]	Weight Function $\lambda(x)$	Rules
$[-1,1]$	1	Gauss-Legendre
$[-1,1]$	$(1-x^2)^{-1/2}$	Gauss-Chebyshev
$[-1,1]$	$(1-x)^{\alpha}(1+x)^{\beta} \quad \alpha, \beta > -1$	Gauss-Jacobi
$[0,\infty]$	$\exp(-x)$	Gauss-Laguerre
$[-\infty,\infty]$	$\exp(-x^2)$	Gauss-Hermite

Analogue to the Newton-Cotes formulas, the weights defined by Lagrange polynomial can be computed:

$$w_j^{(N)} = \int_a^b \lambda(x) L_j(x) dx$$

Legendre polynomials are the result of a chosen weight function $\lambda(x) = 1$ on an interval [-1,1]. The dedicated integration formula is called Gauss-Legendre rule and can be used like a composed formula such as Newton-Cotes formulas. In the 1-dimensional space Gaussian points can be computed on the edges.

Gaussian Points [-1,1]	i	Supporting Points $x_i^{(N)}$	Weights $w_i^{(N)}$
	1	0	2
	1 2	-0.577350269189626 0.577350269189626	1 1

An extension of Gauss quadrature for the *n*-dimensional space is enabled by the 1-dimensional Gaussian points. In the literature the Gaussian points are given for triangles and quadrangles in the 2-dimensional space and tetra- and hexahedrons in the 3-dimensional space.

4.3.3 Implementation on Convex Polyhedrons

On basis of known polyhedrons in the broader named as quadrature polyhedrons the computation of supporting points on an interval [a , b] is enabled. Quadrature polyhedrons are edges in the 1-dimensional, triangles or quadrangles in the 2-dimensional and tetra- or hexahedrons in the 3-dimensional space.

Therefore to use quadrature formulas a decomposition of the convex polyhedrons in quadrature polyhedrons is necessary.

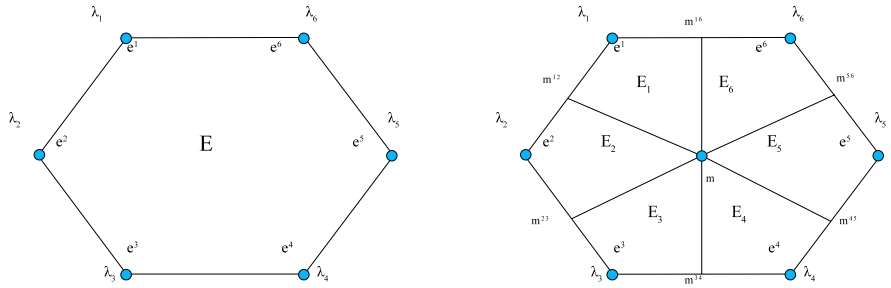


Figure 10: Decomposition of a hexagon in quadrangles

An integration over the convex polyhedron E can be conceive as sum of all subregions E_i.

$$\int_{\Omega_E} f(\lambda)d\Omega = \sum_{E_i} \int_{\Omega_{E_i}} f(\lambda)d\Omega$$

In order to sum up all subregions, the function value on supporting points regarding the convex polyhedron must be computed at first. This makes a transformation of the convex polyhedron to a unit polyhedron necessary.

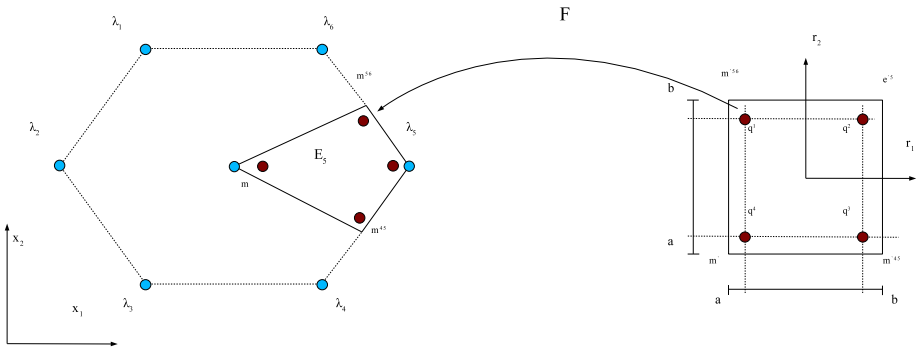


Figure 11: Transformation of a unit polyhedron to the convex polyhedron

The transformation accomplished via Jacobian matrix. The Jacobian matrix allows to establish a relationship between the locale coordinates r of the unit polyhedron and the locale coordinates λ of the convex polyhedron. The integration over the convex polyhedron can be computed as follows:

$$\int_{\Omega_E} f(\lambda) d\Omega = \sum_{E_i} \int_{\Omega_{E_i}} f(\lambda) d\Omega = \sum_{E_i} \sum_{j=0}^{N} w_j^{(N)} f(\lambda_j)$$

The numerical integration via classical quadrature formulas is exact for polynomials. In contrast to this the natural element coordinates are rational functions. Therefore a better approximation of integrals will be enforce as well as a reduction of the approximation error.

5 Conclusion

The fundamentals for the formulation of a finite element on arbitrary convex polyhedrons were created. Furthermore, investigations should be the extension of the geometry of the convex finite cells by using shape functions as it is common on parametric elements. This will then lead to the description of even warped lined or warped planed elements. Regarding the interpolation thoughts must be done about possibilities to optimize the differentiation and the integration. The extended finite element method allows a better solution for a description of complex engineering problems.

References

[1] S. Abramowski, H. Müller, „Geometrisches Modellieren“, BI-Wissenschafts-verlag, 1991

[2] P. Milbradt, “Algorithmische Geometrie in der Bauinformatik”, Habili-tationsschrift, Universität Hannover, 2001

[3] J. Schierbaum, M. Hain, “Parametrische Finite Elemente auf der Basis natürlicher Koordinaten”, Forum Bauinformatik 2002

[4] J. Schierbaum, A. Schwöppe, “Ansatzfunktionen höherer Ordnung für Finite konvexe Zellen”, Forum Bauinformatik 2003

[5] P. Milbradt, A. Schwöppe, “Finite Element Approximation auf der Basis geometrischer Zellen“, IKM, Weimar 2003

[6] R. Sibson, „A brief description of the natural neighbour interpolant“, Technical Report, Math. Department, Univ. of Barth, 1980

[7] E. Stein, “Error-controlled Adaptive Finite Elements in Solid Mechanics”, John Wiley&Sons, 2003