Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

EQP - A 2D/3D library for integration of polynomials times step function

Gregorio Mariggiò, Sebastiano Fichera, Mauro Corrado [*], Giulio Ventura

Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

A R T I C L E I N F O

Article history:
Received 6 October 2020
Received in revised form 14 November 2020
Accepted 23 November 2020

Keywords:
Numerical integration
Heaviside step function
Polynomial functions
Computer graphics
Library
Fortran
Mass properties
Subdomain quadrature

A B S T R A C T

The EQuivalent Polynomials library, *EQP*, herein provided is a powerful tool for the numerical integration, with classical quadrature rules (e.g. Gauss–Legendre), of a function given by the product of an arbitrary polynomial times a Heaviside step function. The library can handle a multiplicity of shapes for the integration domain in one, two and three dimensions. Originally developed by Ventura (Ventura, 2006) to overcome the long-standing problem of integrating discontinuous functions in the context of the eXtended Finite Element Method, *EQP* library has been recently generalized to meet the needs of very different fields, spanning from computational mechanics, to computer graphics, evaluation of geometric region (mass) properties and computer simulation in general.

Code metadata

Current code version	v 1.2
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-20-00060
Code Ocean compute capsule	None
Legal Code License	GNU General Public License (GPL)
Code versioning system used	None
Software code languages, tools, and services used	FORTRAN 90
Compilation requirements, operating environments & dependencies	None
If available Link to developer documentation/manual	www.equivalent-polynomials.net/reference-documentation
Support email for questions	giulio.ventura@polito.it

1. Motivation and significance

The numerical integration of polynomial functions is needed to solve physical models and calculate quantities in many physical and engineering fields as well as in computer graphics. Let us consider, as a few representative examples, the computation of the moments and products of inertia entering the inertia tensor of geometric shapes needed for physics-based animations of rigid bodies [1,2]; the computation of the stiffness matrix of a mechanical system in the framework of the finite element method for the prediction of the mechanical behavior of solids and structures [3, 4]; the calculation of mass, total energy, angular momentum, and entropy to impose the conservation laws governing the dynamics and thermodynamics of the atmosphere [5].

When the shape of the domain is an elementary geometry (triangle, parallelogram, parallelepiped) or can be brought back to an elementary geometry with a transformation, efficient numerical quadrature rules are available for polynomial integrands. However, when the polynomial shows a jump discontinuity or is to be integrated over a subdomain, a multiplying Heaviside step function is introduced. In this case, as numerical quadrature rules such as the Gauss–Legendre, implicitly introduce a polynomial approximation of the integrand function, this may lead to large errors in the computation of the integral. This condition can be encountered, for instance, in computer graphics, when computing the geometrical properties of complex bodies that can be seen as a partition of a regular geometric shape. Let us consider, for instance, the body Ω^+ in Fig. 1, which is obtained by splitting

* Corresponding author.
E-mail address: mauro.corrado@polito.it (M. Corrado).

https://doi.org/10.1016/j.softx.2020.100636

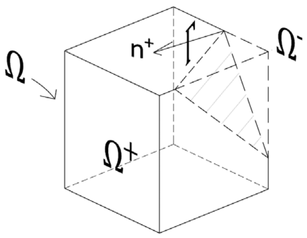


Fig. 1. 3D domain Ω crossed by a discontinuity surface Γ.

the cubic domain Ω in two parts by the discontinuity surface Γ. The geometrical properties of Ω^+, such as, for instance, volume, moments and products of inertia, can be computed through the integral:

$$I = \int_{\Omega^+} p_n(\boldsymbol{x})\,\mathrm{d}\Omega = \int_{\Omega} H(\boldsymbol{x}) p_n(\boldsymbol{x})\,\mathrm{d}\Omega \tag{1}$$

where $p_n(\boldsymbol{x})$ is the polynomial function of degree n with respect to the set of variables $\boldsymbol{x} = (x, y, z)$ needed to compute the desired geometrical property ($p_n(\boldsymbol{x}) = 1$ for the volume, $p_n(\boldsymbol{x}) = y^2 + z^2$ for the second moment of area with respect to the x-axis, etc.), and $H(\boldsymbol{x})$ is the standard Heaviside step function, given by Eq. (2).

$$H(\boldsymbol{x}) = \begin{cases} 1 & \mathit{if}\ \boldsymbol{x} \in \Omega^+ \\ 0 & \mathit{otherwise} \end{cases} \tag{2}$$

The positive part of the domain Ω is defined by the $\mathbf{n}^+$ vector, that points inwards Ω^+. The vector $\mathbf{n}^+$ is orthogonal to the internal discontinuity surface and, in the 3D case, its components are expressed by the coefficients a, b and c of the equation of the surface Γ (Eq. (3)).

$$ax + by + cz + d = 0 \tag{3}$$

The above stated problem of numerical integration is commonly solved by partitioning the integration domain to generate quadrature subcells where the integrands are polynomials. However, it is very difficult to identify and catalog all the possible shapes of the integration subdomains, especially for 3D geometries. Therefore, a methodology to eliminate the subdivision of the quadrature domain has been suggested in [6,7], in the context of the eXtended Finite Element Method (XFEM) [8,9]. The method has been actively used or discussed in other frameworks also and some non-exhaustive references are [10–27]. Although the method has been developed for polynomial integrals, in principle it can be applied/extended to the piecewise polynomial representation of splines for introducing discontinuities or trimmed domains. In this field a literature on the construction of specialized quadrature rules can be recalled [28–31], also addressing the case of trimmed domains [32]. In fact, the proposed approach is independent of the particular numerical quadrature employed in implementation.

The methodology is based on replacing the Heaviside function $H(\boldsymbol{x})$ with an equivalent polynomial function $\tilde{H}(\boldsymbol{x})$ such that:

$$\int_{\Omega} \tilde{H}(\boldsymbol{x}) p_n(\boldsymbol{x})\,\mathrm{d}\Omega = \int_{\Omega^-} H(\boldsymbol{x}) p_n(\boldsymbol{x})\,\mathrm{d}\Omega + \int_{\Omega^+} H(\boldsymbol{x}) p_n(\boldsymbol{x})\,\mathrm{d}\Omega \tag{4}$$

The equivalent polynomial $\tilde{H}(\boldsymbol{x})$ depends on Γ, has the same degree of $p_n(\boldsymbol{x})$ and is represented by polynomial functions:

$$\tilde{H}(\boldsymbol{x}) = \mathbf{c} \cdot \mathbf{m}(\boldsymbol{x}) \tag{5}$$

where $\mathbf{m}(\boldsymbol{x})$ collects a monomial basis, e.g. $\mathbf{m}(\boldsymbol{x}) = (1, x, y, z, x^2, \ldots)$, and $\mathbf{c}$ is a vector of coefficients.

Taking into account Eq. (4), Eq. (1) can be rewritten as:

$$I = \int_{\Omega} \tilde{H}(\boldsymbol{x}) p_n(\boldsymbol{x})\,\mathrm{d}\Omega \tag{6}$$

Since $\tilde{H}(\boldsymbol{x}) p_n(\boldsymbol{x})$ is a polynomial function continuous over the entire domain Ω, it can be exactly integrated with an appropriate numerical quadrature rule [33]. Note that, in general, the equivalent polynomial function $\tilde{H}(\boldsymbol{x})$ has the same degree as $p_n(\boldsymbol{x})$, so that the integrand in Eq. (6), compared to Eq. (1), has doubled its degree. Note that the introduction of the equivalent polynomial allows integration on the standard domain Ω instead of the non-standard partitioned subdomain Ω^+. This is the advantage of the equivalent polynomial approach.

The present paper describes a standalone library, *EQP*, that provides, for five different shapes of the integration domain Ω, namely triangle, parallelogram, circle, tetrahedron and parallelepiped, the expression of the equivalent polynomial function, $\tilde{H}(\boldsymbol{x})$, as a function of the position of the discontinuity line or surface, Γ. The details on the mathematical procedure adopted to determine the expression of the equivalent polynomials can be found in Refs. [6,7].

2. Software description

The library is built in Fortran, a multi-platform language that can be coupled with existing libraries. Nonetheless, the structure of the library is straightforward, so that it can be ported to any language with negligible effort.

2.1. Software functionalities

Although the purpose of *EQP* library is to provide the expression of the equivalent polynomial function $\tilde{H}(\boldsymbol{x})$, for a practical use the library needs being wrapped into an algorithm that evaluates the integral defined in Eq. (6) by applying, for example, the Gauss quadrature rule. Nonetheless, the library can be directly included in the algorithm of any quadrature rule able to integrate pure (non involving discontinuities) polynomial functions.

The functionality of *EQP* library is herein presented through a generic example. Let us assume a polynomial $p_n(\boldsymbol{X})$ is to be integrated over a subdomain $\bar{\Omega}^+$ obtained by splitting a parallelepiped $\bar{\Omega}$ with a flat surface $\bar{\Gamma}$, as shown in Fig. 2a. $\boldsymbol{X} = (X, Y, Z)$ is the global reference system where the problem is defined. The desired solution can be obtained by applying Eq. (6), once the equivalent polynomial function $\tilde{H}(\boldsymbol{X})$ is known:

$$I = \int_{\bar{\Omega}^+} p_n(\boldsymbol{X})\,\mathrm{d}\Omega = \int_{\bar{\Omega}} \tilde{H}(\boldsymbol{X}) p_n(\boldsymbol{X})\,\mathrm{d}\bar{\Omega} \tag{7}$$

First, the problem is to be transformed to a quadrature on a standard domain. Therefore, a change of variables from the global reference system *XYZ* to the parent coordinate system *xyz* is applied, so to evaluate the integral over a standard regular geometrical shape, as shown in Fig. 2b. Such a transformation, in fact, allows to treat a variety of cases with a single parent geometry. For instance, parallelepipeds having any size and position in the global reference system can be brought back to the cubic parent geometry defined in the local reference system $(x, y, z) \in [-1, +1]$ shown in Fig. 2b.

The mathematical approach adopted in the library to achieve the change of variables is the isoparametric mapping routinely used in the finite element method [34]. Let $P(x, y, z) \in \Omega$ be a point in the parent coordinate system corresponding to the point $\bar{P}(X, Y, Z) \in \bar{\Omega}$ in the global coordinate system. The affine mapping of any $P(x, y, z)$ onto $\bar{P}(X, Y, Z)$ is defined by:

$$X = \sum_{i=1}^{q} N_i(x, y, z) X_i \tag{8a}$$

$$Y = \sum_{i=1}^{q} N_i(x, y, z) Y_i \tag{8b}$$

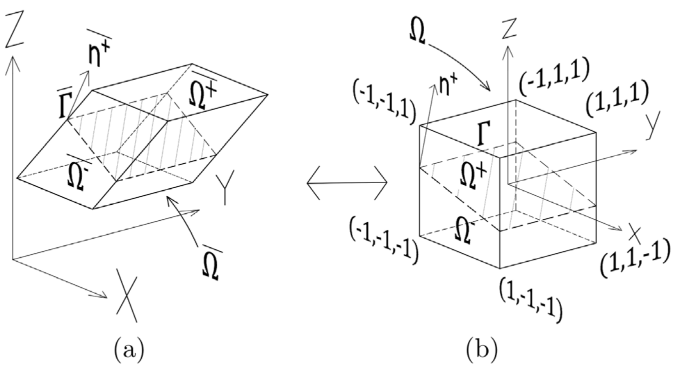


Fig. 2. Mapping of a parallelepiped element: (a) configuration in the global coordinate system; (b) configuration in the parent coordinate system.

$$Z = \sum_{i=1}^{q} N_i(x, y, z) Z_i \tag{8c}$$

where q denotes the number of nodes (vertices) of the geometric element having coordinates (X_i, Y_i, Z_i) in the global reference system, and $N_i(x, y, z)$ denotes the interpolation function in terms of the local coordinates for the i-th node of the parent element.

Analogously, the equation of the discontinuity $\bar{\Gamma}(\boldsymbol{X})$, defined by the user in the global reference system, has to be transformed into $\Gamma(\boldsymbol{x})$, defined in the parent coordinate system (see Fig. 2). In the library, this is done by:

- computing the signed distances D_i between $\bar{\Gamma}(\boldsymbol{X})$ and the nodes of the integration domain in the global reference system;
- writing the coefficients a, b, c and d of $\bar{\Gamma}(\boldsymbol{X})$ in terms of D_i by solving a system of linear equations;
- replacing the variables X, Y, and Z of $\bar{\Gamma}(\boldsymbol{X})$ with the expressions in Eq. (8), thus obtaining $\Gamma(\boldsymbol{x})$, defined by new coefficients a', b', c' and d' dependent on D_i.

For instance, in the triangle case it is obtained:

$$a' = D_2 - D_1 \tag{9a}$$

$$b' = D_3 - D_1 \tag{9b}$$

$$c' = D_1 \tag{9c}$$

After $\bar{\Gamma}(\boldsymbol{X})$ is transformed into $\Gamma(\boldsymbol{x})$, the proper expression of the equivalent polynomial function $\tilde{H}(\boldsymbol{x})$ is provided by the library with reference to the parent domain coordinate system.

Then, the coordinate and integration domain transformation is introduced in the quadrature through the Jacobian matrix, containing the partial derivatives of the interpolation functions N_i, differentiated with respect to the parent system variables x, y, z [35]:

$$I = \int_{\bar{\Omega}} \tilde{H}(\boldsymbol{X}) p_n(\boldsymbol{X}) \, \mathrm{d}\bar{\Omega} = \int_{\Omega} \tilde{H}(\boldsymbol{x}) p_n(\boldsymbol{x}) |\mathbf{J}| \mathrm{d}\Omega = \sum_{j=1}^{gp} w_j \tilde{H}(x_j, y_j, z_j) p_n(x_j, y_j, z_j) |\mathbf{J}(x_j, y_j, z_j)| \tag{10}$$

being $|\mathbf{J}|$ the determinant of the Jacobian. The integral of Eq. (7) is evaluated in Eq. (10) with the standard form of Gauss–Legendre numerical quadrature [36]. In Eq. (10), gp denotes the number of Gauss–Legendre quadrature points and w_j their weights.

It is worth underlining again that the library integrates over the entire domain $\bar{\Omega}$ and yields the result of the integral over the subdomain $\bar{\Omega}^+$. Therefore, the equation of the discontinuity has to be properly defined in order to have the unit vector $\mathbf{n}^+$ pointing in the desired subdomain direction. For instance, the evaluation over the domain $\bar{\Omega}^-$ can be done by simply changing sign to all the discontinuity coefficients. Moreover, note that it is not mandatory the discontinuity to intersect $\bar{\Omega}$: when this does not happen the result of the quadrature will be zero if $\bar{\Omega}^+ = \emptyset$ or will coincide with the integral over $\bar{\Omega}$ if $\bar{\Omega}^+ = \bar{\Omega}$.

As it can be inferred from Refs. [6,7], the degree and the composition of the polynomial function that can be exactly integrated with the proposed approach depend on some conditions imposed to determine the equivalent polynomial. The parent geometrical shapes that are included in the current version of the library and the monomials which the polynomial function to be integrated can be composed of are listed in Table 1. Note, however, that it is possible to extend the Library to any polynomial degree in each of the examined domains.

2.2. Software architecture

The software library architecture is straightforward. The main library file is *eqpol.f90*. It contains the algorithms to transform the equation of the discontinuity from the global to the parent coordinate system, to get the coefficients and evaluate the equivalent polynomial function $\tilde{H}(\boldsymbol{x})$. It is completed by other files containing the algebraic expressions of the coefficients needed to define the equivalent polynomial function.

For the sake of demonstration of use, the library is complemented by a main program file *main.f90* and a module file *mapping_module.f90* that are not part of the library, but allow to compute the integral in Eq. (10) in some example cases.

The practical use of the library foresees the following steps:

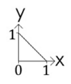
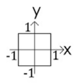
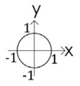
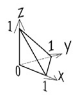
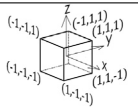
1. preliminary data preparation:
   - selection of the domain of integration (see Table 1);
   - individuation of the domain nodal coordinates in the global reference system, or center and radius for the circular domain;
   - individuation of the discontinuity plane coefficients in the global cartesian reference;
2. transformation to the parent (standard) domain and evaluation of the equivalent polynomial coefficient vector by the subroutine *Heqpol_coefficients*
3. quadrature through any selected rule, e.g. Eq. (10), with the values of the equivalent polynomial at quadrature points provided by function *HeqPol* and the value of the transformation determinant of the Jacobian given by the function *det_J*.

The current version of the library allows to evaluate the exact quadrature result under the following hypotheses:

- the transformation Jacobian in Eq. (10) is constant (this always applies to the domains: triangle, circle, parallelogram, tetrahedron, parallelepiped);
- the polynomial p_n is a linear combination of the monomials listed in Table 1.

Note that, in case a user wants to integrate the core of the library directly in his own algorithm of quadrature, two calls have to be foreseen: a call to the subroutine *Heqpol_coefficients* for each integration domain (step 2 above), and another to the function *HeqPol* for each quadrature point (step 3 above). The function *det_J* is not included in the core of the library since it should be part of the quadrature algorithm.

Table 1
Domain of integration, domain type, parent domain, monomial basis.

Domain of integration	Triangle	Parallelogram	Circle	Tetrahedron	Parallelepiped
etype	20	21	22	30	31
Parent domain					
Monomial basis	1	$1, x, x^2, y, xy, y^2$	$1, x, x^2, y, xy, y^2$	1	$1, x, x^2, y, xy, x^2y, y^2, xy^2, x^2y^2, z, xz, x^2z, yz, xyz, x^2yz, y^2z, xy^2z, z^2, xz^2, x^2z^2, yz^2, xyz^2, y^2z^2$

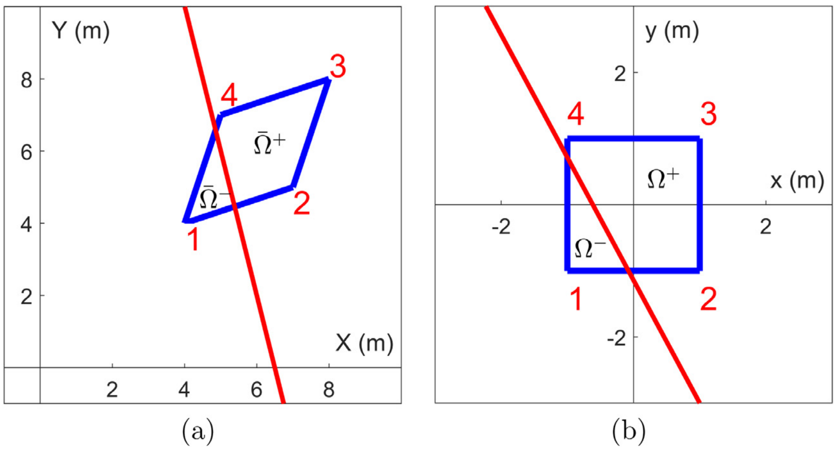


Fig. 3. Parallelogram (etype 21): (a) configuration in global coordinate system; (b) configuration in parent coordinate system.

3. Illustrative examples

The following examples are based on the Library usage example file *main.f90*. For the presented examples the Library provides exact results up to machine precision.

3.1. Parallelogram

In this example, the parallelogram shown in Fig. 3(a) is considered. Dimensions are meters. The element is splitted by the line of equation $4X + Y - 26 = 0$ into $\bar{\Omega}^+$ and $\bar{\Omega}^-$. The goal is to compute the area and the inertia tensor of $\bar{\Omega}^+$. The parallelogram is mapped by the software onto the parent coordinate system, as shown in Fig. 3(b), where $\tilde{H}(\boldsymbol{x})$ is computed and the integration is performed.

Once the library example program is launched, the input data are provided by typing on the screen:

```
EQUIVALENT POLYNOMIAL LIBRARY
etype (20,21,22,30,31): 21
a,b,c: 4,1,-26
```

where the number 21 identifies the parallelogram, and the numbers 4, 1, −26 are the coefficients of the straight line in the global coordinate system. After that, the library asks to input the nodal coordinates of the integration domain in the global coordinate system:

```
Insert element coordinates following
the scheme shown below:
4----------3
|          |
|          |
|          |
1----------2
X(1), Y(1): 4,4
X(2), Y(2): 7,5
X(3), Y(3): 8,8
X(4), Y(4): 5,7
```

At this stage the user can choose to integrate all the monomials listed in Table 1 for the selected geometry type, a monomial of those that will appear on the screen, or a polynomial defined in the *user_fun.txt* file as linear combination of the monomials listed in Table 1. For instance, if the user wants to compute the area A of the $\bar{\Omega}^+$ region, the monomial $p = 1$ should be entered in the *user_fun.txt* file, whereas, the monomials y^2, x^2 and xy have to be entered to obtain the tensor of inertia $\boldsymbol{I}$. The area and the tensor of inertia of $\bar{\Omega}^+$, calculated with respect to the XY axes and reported by the Library output, are:

$$A = 6.418\ \text{m}^2 \tag{11}$$

$$\boldsymbol{I} = \rho_s \begin{bmatrix} 254.436 & -254.281 \\ -254.281 & 258.876 \end{bmatrix} (\text{units: kg m}^2) \tag{12}$$

where ρ_s represents the surface density (kg/m^2) of the material.

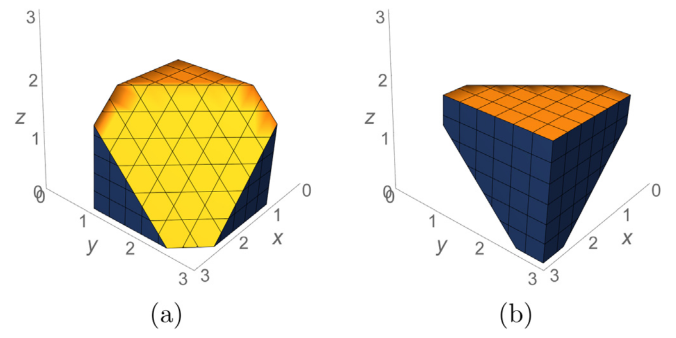


Fig. 4. Cube splitted by a plane surface: (a) positive part, $\bar{\Omega}^+$, domain of integration; (b) negative part, $\bar{\Omega}^-$, excluded from integration.

3.2. Parallelepiped

A 3D example is herein analyzed: a cube of side 2 m is cut by a surface of equation $-X-Y-Z+5.5=0$, as shown in Fig. 4.

The input procedure is similar to the previous example and is presented below.

```
EQUIVALENT POLYNOMIAL LIBRARY
etype (20,21,22,30,31) : 31
a,b,c,d : -1,-1,-1,+5.5

Insert element coordinates
following the scheme shown below:
   5-----------8
  /|          /|
 / |         / |
6-----------7  |
|  1--------|--4
| /         | /
|/          |/
2-----------3
X(1), Y(1), Z(1): 3,1,0
X(2), Y(2), Z(2): 3,3,0
X(3), Y(3), Z(3): 1,3,0
X(4), Y(4), Z(4): 1,1,0
X(5), Y(5), Z(5): 3,1,2
X(6), Y(6), Z(6): 3,3,2
X(7), Y(7), Z(7): 1,3,2
X(8), Y(8), Z(8): 1,1,2
```

The volume V and tensor of inertia $\boldsymbol{I}$ of $\bar{\Omega}^+$ (Fig. 4(a)) calculated with respect to the XYZ axes and reported by the Library output are:

$$V = 5.458 \text{ m}^3 \tag{13}$$

$$\boldsymbol{I} = \rho \begin{bmatrix} 25.008 & -17.791 & -7.838 \\ -17.791 & 25.008 & -7.838 \\ -7.838 & -7.838 & 39.456 \end{bmatrix} (\text{units: kg m}^2) \tag{14}$$

where ρ represents the volumetric density (kg/m^3) of the material.

4. Impact

EQP library is a useful tool to integrate discontinuous functions with any numerical quadrature approach without splitting the integration domain. Its efficiency has been already proven in the field of computational mechanics (XFEM/GFEM) [6,7]. However, its impact is much wider, since the numerical integration of polynomial functions is a common problem in many fields. The applications herein envisaged concern the context of computational geometry, where *EQP* library can be used to calculate the geometrical properties of complex figures, obtained by cutting elementary shapes, such as squares or cubes, with plane surfaces and, eventually, combining them together. Besides, the peculiarities of the library can be exploited in simulations that involve a dynamic change of shapes and position of the objects such as, for instance, the breaking of an object or discretization cells into several pieces. In this context, for instance, there are already experiences of using XFEM to simulate brittle fracture and surface crack patterns in 3D elements without re-meshing [37], optimizing the approach proposed in [38].

Therefore, *EQP* library might be integrated in many modern computational tools for a wide range of application areas.

5. Conclusions

The *EQP* library presented in this paper is a ready-to-use tool that simplifies considerably the numerical computation of integrals of polynomial functions over subdomains obtained by splitting a standard quadrature domain, by reverting them to integrals of an equivalent polynomial over the entire domain. The strength of *EQP* resides in the computational speed, the elimination of complex subdomains computation, and the absolute generality and extendibility of the mathematical approach behind it. Due to all these advantages, it is expected that *EQP* may become a common tool for a broad range of applications. Future developments of the software are envisaged by adding new improved algorithms for automatically filtering, calibrating, mapping and combining the existing geometric shapes to easily create multi-body complex geometries.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Research Grant "XFAST-SIMS: Extra fast and accurate simulation of complex structural systems" by the Italian Ministry of Education, University and Research is gratefully acknowledged. Research support by Politecnico di Torino, Italy with the diffuse research grant initiative is gratefully acknowledged.

References

[1] Hahn JK. Realistic animation of rigid bodies. Comp Graph 1988;22:299–308.
[2] Bender J, Erleben K, Trinkle J. Interactive simulation of rigid body dynamics in computer graphics. Comput Graph Forum 2014;33:246–70.
[3] Bathe K-J. Finite element procedures. second ed.. Watertown, MA; 2014.
[4] Belytschko T, Liu WK, Moran B, Elkhodary K. Nonlinear finite elements for continua and structures. second ed.. Wiley; 2014.
[5] Machenhauer B, Kaas E, Lauritzen PH. Finite-volume methods in meteorology. In: Temam RM, Tribbia JJ, editors. Computational Methods for the Atmosphere and the Oceans. Elsevier Science; 2008, p. 1–120.
[6] Ventura G. On the elimination of quadrature subcells for discontinuous functions in the extended finite-element method. Internat J Numer Methods Engrg 2006;66:761–95.
[7] Ventura G, Benvenuti E. Equivalent polynomials for quadrature in Heaviside function enriched elements. Internat J Numer Methods Engrg 2015;102:688–710.
[8] Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. Internat J Numer Methods Engrg 1999;45(5):601–20.
[9] Belytschko T, Gracie R, Ventura G. A review of extended/generalized finite element methods for material modeling. Model Simul Mater Sci Eng 2009;17:1–24.

[10] Lv J-H, Jiao Y-Y, Rabczuk T, Zhuang X-Y, Feng X-T, Tan F. A general algorithm for numerical integration of three-dimensional crack singularities in PU-based numerical methods. Comput Methods Appl Mech Engrg 2020;363. 112908–1.

[11] Düster A, Allix O. Selective enrichment of moment fitting and application to cut finite elements and cells. Comput Mech 2020;65(2):429–50.

[12] Ali T, Mostefa B, Abdelkader D, Abdelkrim A, Habibe K. Experimental and numerical fracture modeling using XFEM of aluminum plates. Int J Eng Res Africa 2020;46:45–52.

[13] Surendran M, Natarajan S, Palani G, Bordas S. Linear smoothed extended finite element method for fatigue crack growth simulations. Eng Fract Mech 2019;206:551–64.

[14] Müller B, Krämer-Eis S, Kummer F, Oberlack M. A high-order discontinuous Galerkin method for compressible flows with immersed boundaries. Internat J Numer Methods Engrg 2017;110(1):3–30.

[15] Saye R. High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. SIAM J Sci Comput 2015;37(2):A993–1019.

[16] Martin A, Esnault J-B, Massin P. About the use of standard integration schemes for X-FEM in solid mechanics plasticity. Comput Methods Appl Mech Engrg 2015;283:551–72.

[17] Sudhakar Y, Moitinho de Almeida J, Wall W. An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: Application to embedded interface methods. J Comput Phys 2014;273:393–415.

[18] Benvenuti E. XFEM with equivalent eigenstrain for matrix-inclusion interfaces. Comput Mech 2014;53(5):893–908.

[19] Alves P, Barros F, Pitangueira R. An object-oriented approach to the generalized finite element method. Adv Eng Softw 2013;59:1–18.

[20] Loehnert S, Mueller-Hoeppe D, Wriggers P. 3D corrected XFEM approach and extension to finite deformation theory. Internat J Numer Methods Engrg 2011;86(4–5):431–52.

[21] Mousavi S, Sukumar N. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. Comput Mech 2011;47(5):535–54.

[22] Antonietti P, Verani M, Vergara C, Zonca S. Numerical solution of fluid-structure interaction problems by means of a high order discontinuous Galerkin method on polygonal grids. Finite Elem Anal Des 2019;159:1–14.

[23] Egger A, Pillai U, Agathos K, Kakouris E, Chatzi E, Aschroft I, Triantafyllou S. Discrete and phase field methods for linear elastic fracture mechanics: A comparative study and state-of-the-art review. Appl Sci (Switzerland) 2019;9(12).

[24] Wu J-Y, Qiu J-F, Nguyen V, Mandal T, Zhuang L-J. Computational modeling of localized failure in solids: XFEM vs PF-CZM. Comput Methods Appl Mech Engrg 2019;345:618–43.

[25] Formaggia L, Vergara C, Zonca S. Unfitted extended finite elements for composite grids. Comput Math Appl 2018;76(4):893–904.

[26] Chin E, Lasserre J, Sukumar N. Modeling crack discontinuities without element-partitioning in the extended finite element method. Internat J Numer Methods Engrg 2017;110(11):1021–48.

[27] Kudela L, Zander N, Kollmannsberger S, Rank E. Smart octrees: Accurately integrating discontinuous functions in 3D. Comput Methods Appl Mech Engrg 2016;306:406–26.

[28] Hiemstra RR, Calabro F, Schillinger D, Hughes TJ. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. Comput Methods Appl Mech Engrg 2017;316:966–1004.

[29] Bartoň M, Calo VM. Gauss–Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. Comput Aided Des 2017;82:57–67.

[30] Bartoň M, Calo VM. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. Comput Methods Appl Mech Engrg 2016;305:217–40.

[31] Calabro F, Sangalli G, Tani M. Fast formation of isogeometric Galerkin matrices by weighted quadrature. Comput Methods Appl Mech Engrg 2017;316:606–22.

[32] Scholz F, Mantzaflaris A, Jüttler B. First order error correction for trimmed quadrature in isogeometric analysis. In: Chemnitz Fine Element Symposium. Springer; 2017, p. 297–321.

[33] Stroud A. Approximate calculation of multiple integrals. Englewood Cliffs, N.J.: Prentice Hall; 1971.

[34] Zienkiewicz OC, Taylor RL, Zhu JZ. The finite element method: Its basis and fundamentals. sixth ed.. Butterworth-Heinemann; 2005.

[35] Amazigo J, Rubenfeld L. Advanced calculus and its application to the engineering and physical science. John Wiley & Sons Inc; 1980.

[36] Davis PJ, Rabinowitz P. Methods of numerical integration. second ed.. Academic Press; 1984.

[37] Song C, Zhang H, Wu Y, Bao H. Cutting and fracturing models without remeshing. In: Chen F, Jüttler B, editors. Advances in Geometric Modeling and Processing. Springer Berlin Heidelberg; 2008, p. 107–18.

[38] Iben HN, O'Brien JF. Generating surface crack patterns. Graph Models 2009;71(6):198–208. http://dx.doi.org/10.1016/j.gmod.2008.12.005.