

FINITE ELEMENT IMPLEMENTATION OF GEOMETRICALLY NONLINEAR CONTACT

ONDŘEJ FALTUS*, MARTIN HORÁK

Czech Technical University in Prague, Faculty of Civil Engineering, Department of Mechanics, Thákurova 7, 166 29 Prague 6, Czech Republic

* corresponding author: ondrej.faltus@fsv.cvut.cz

ABSTRACT. The OOFEM finite element software has been recently updated to include contact algorithms for small strain applications. In this work, we attempt to extend the contact algorithms to large strain problems. Reviewing the current code and comparing it with approaches encountered in literature, we arrive at a specific algorithmic solution and integrate it into the current code base. The current code is explained, the necessary extensions are derived and documented, and the algorithmic changes are described. Tests confirm the functionality and quadratic rate of convergence of the proposed implementation.

KEYWORDS: Contact, large strains, node-to-segment, OOFEM, penalty method.

1. INTRODUCTION

Having originated with Heinrich Hertz in 1881 [1], contact mechanics has been a fringe discipline of structural mechanics for most of the 20th century, owing mainly to its limited options for analytical analysis. Only with the advent of numerical methods, such as the finite element method (FEM), has development in this field sprung forward [2]. This surge in new advances is only exacerbated by the ever more rapid evolution of computational hardware capabilities experienced today; ever more complicated contact problems necessitating ever more complicated methods previously unimaginable due to hardware constraints are coming into focus [2, 3].

The tool used for the purposes of this article is the finite element computational software OOFEM [4]. OOFEM is a free, open-source C++ code developed mainly at the Czech Technical University in Prague. It is based on object-oriented design with emphasis on variability and extendability. Currently, OOFEM's finite element code supports a number of modules for solution of various types of numerical problems. Of interest for our purposes is the `sm` module intended for structural mechanics.

Basic introduction of contact algorithms into the OOFEM software has been achieved in [5]. This implementation, however, has certain limitations, among them chiefly the lack of support for large strain, i.e., geometrically non-linear problems. In this work, we strive to remove this particular limitation and demonstrate a simple method of extending existing basic penalty-based formulation to handle large strain problems.

2. ALGORITHM

2.1. BASIC EQUATION

Without loss of generality, let us consider two deformable bodies \mathcal{B}^γ , $\gamma = 1, 2$. Moreover, let Ω_0^γ and Ω^γ represent the regions occupied by the body in the reference and deformed configuration, respectively. Position of a point in the reference configuration is denoted by \mathbf{X}^γ , and a position \mathbf{x}^γ of a point in the deformed configuration is obtained by means of the one-to-one deformation mapping φ^γ as

$$\mathbf{x}^\gamma = \varphi^\gamma(\mathbf{X}^\gamma) \quad (1)$$

The basic equations for each body under consideration are the following:

- Kinematic equations

$$\mathbf{F}^\gamma(\mathbf{X}^\gamma) = \frac{\partial \varphi^\gamma(\mathbf{X}^\gamma)}{\partial \mathbf{X}^\gamma} \quad \forall \mathbf{X}^\gamma \in \Omega^\gamma \quad (2)$$

- Equilibrium equations

$$\text{Div} \mathbf{P}^\gamma(\mathbf{X}^\gamma) + \bar{\mathbf{B}}^\gamma(\mathbf{X}^\gamma) = 0 \quad \forall \mathbf{X}^\gamma \in \Omega^\gamma \quad (3)$$

- Dirichlet boundary conditions

$$\mathbf{u}^\gamma(\mathbf{X}^\gamma) = \bar{\mathbf{u}}^\gamma(\mathbf{X}^\gamma) \quad \forall \mathbf{X}^\gamma \in \Gamma_u^\gamma \quad (4)$$

- Neumann boundary conditions

$$\bar{\mathbf{t}}^\gamma(\mathbf{X}^\gamma) = \mathbf{P}^\gamma(\mathbf{X}^\gamma) \cdot \bar{\mathbf{N}}^\gamma(\mathbf{X}^\gamma) \quad \forall \mathbf{X}^\gamma \in \Gamma_\sigma^\gamma \quad (5)$$

- Constitutive equation

$$\mathbf{P}^\gamma(\mathbf{F}) = \lambda^\gamma \text{Tr}(\mathbf{E}^\gamma) \mathbf{F}^\gamma + 2\mu^\gamma \mathbf{F}^\gamma \cdot \mathbf{E}^\gamma \quad (6)$$

Here, we use the so-called Saint-Venant Kirchhoff material model. Note that the contact implementation is not affected by the choice of constitutive law and extension to inelastic law is straightforward.

In the equations above, \mathbf{F} is the deformation gradient, \mathbf{E} is the Green-Lagrangian strain tensor, Div is divergence operator in the reference configuration, \mathbf{P} is the first Piola-Kirchhoff stress, $\bar{\mathbf{B}}$ are prescribed body forces in the reference configuration, $\bar{\mathbf{u}}$ are prescribed displacements, $\bar{\mathbf{N}}$ is a unit normal vector in the reference configuration, Γ_u and Γ_σ are the Dirichlet and Neumann boundaries which are mutually disjoint, i.e., $\Gamma_u^\gamma \cap \Gamma_\sigma^\gamma = \emptyset$, and $\Gamma_u^\gamma \cup \Gamma_\sigma^\gamma = \Gamma^\gamma$, λ^γ and μ^γ are Lamé's parameters.

Moreover, contact conditions introduce essentially a new set of boundary conditions into the task. Note that only normal contact is considered in what follows. The core issue of contact problems can be described by means of the contact constraints in the form of Karush-Kuhn-Tucker optimality conditions [6]:

$$t_c \leq 0 \quad g_c \geq 0 \quad g_c t_c = 0 \quad (7)$$

The first condition represents the fact that contact forces t_c are only compressive, the second condition is characterized by a gap function $g(\mathbf{u})$ and express a non-penetration condition. The last equation is simply complementarity between two distinct and mutually exclusive states for the system; either contact occurs, and therefore the gap function g_c is zero, while a non-zero compressive contact force is present, or the contact force is zero and the gap function is larger than zero.

2.2. PENALTY METHOD

Due to the constraint conditions in form of inequalities, see (7), weak form of the above introduced formulation is represented by a variational inequality, see [2]. Several methods how to handle the contact constraints are possible, e.g., the Lagrangian multiplier method, the augmented Lagrangian method, or the Nitsche method [3, 6]. Currently in OOFEM, the Lagrangian multiplier method and the penalty method are implemented [5]. It is the penalty method which we have selected for the purpose of large strain extension, as its formulation is more straightforward and the necessary code complexity is notably lower.

For the penalty method, the weak form can be written as

$$\sum_\gamma \int_{\Omega^\gamma} \delta \mathbf{F}^\gamma : \mathbf{P}^\gamma - \delta \mathbf{u}^\gamma \cdot \bar{\mathbf{B}}^\gamma \, dV - \int_{\Gamma_c^\gamma} \delta \mathbf{u}^\gamma \cdot \bar{\mathbf{t}}^\gamma \, d\Gamma + W^c = 0 \quad (8)$$

where δ represents variation and W^c is the penalty-based contact contribution along contact interface Γ_c defined as

$$W^c(\mathbf{u}) = \int_{\Gamma_c} \frac{1}{2} H(-g(\mathbf{u})) p g_c^2(\mathbf{u}) \, d\Gamma \quad (9)$$

where H is the Heaviside function and p is the penalty parameter. The main disadvantages of the penalty

method is the that the achieved solution is imprecise (it would only be precise for $p \rightarrow \infty$), but moreover, any improvement in precision acts directly against the stability of the computation, as larger penalty parameters introduce larger members into computational stiffness matrices, which disturbs most numerical solvers [2].

Those disadvantages are balanced by the relatively simple formulation of the method and, in comparison with the Lagrangian multiplier method, by simple resulting equation systems, as no new variables are introduced into the computation.

2.3. NODE-TO-SEGMENT DISCRETIZATION

An implementation of contact problems into the framework of the finite element method necessitates selecting a distinctive approach at contact discretization. As the physical world is described by FEM in the form of finite elements and connecting nodes, contact can only be considered between those. Typical contact discretizations include node-to-node, node-to-segment, segment-to-segment and other approaches [3]. At current time, node-to-node and node-to-segment discretizations are implemented in OOFEM [5]. We are only extending the node-to-segment discretization for large strains, as the node-to-node discretization is by definition inadequate for this purpose.

In node-to-segment discretization, a single node comes into contact with a segment. Various objects may be understood under the term "segment", most prominently a set of element boundaries (element edges in 2D domains or element surfaces in 3D domains) or an arbitrarily defined rigid analytical surface.

Contact detection in node-to-segment discretization necessitates determining three things [3, 6]:

- (1.) The gap function g_c , essentially a projection of the node onto the contact segment, which determines whether there is contact or not; analogical to the penetration function in the analytical approach to contact problems.
- (2.) The normal vector in the deformed configuration \mathbf{n} , which determines the orientation of the contact surface at the contact point. Obtained for most segments in OOFEM as vector perpendicular to the contact segment's surface at the contact point [5].
- (3.) The extended N-matrix \mathbf{N}^* , which ensures allocation of the external forces and stiffness to the proper degrees of freedom in the FEM formulation. When the segment in question is an element boundary, the matrix takes the form [5]

$$\mathbf{N}^* = (\mathbf{I}_{(d \times d)} \quad -\mathbf{N}_e(\mathbf{x}_c)) \quad (10)$$

where $\mathbf{I}_{(d \times d)}$ is a square unit matrix of the appropriate dimension, which corresponds to degrees of freedom of the contacting node, and $\mathbf{N}_e(\mathbf{x}_c)$ is the

matrix of element boundary interpolation functions evaluated at the contact point.

2.4. FEM FORMULATION

To integrate contact constraints into the FEM framework, considerations have to be made for the contact forces and energy discussed above within the principles of FEM discretization.

For the penalty method, contact energy is expressed as

$$W^c = \frac{1}{2} p g_c^2 \quad (11)$$

Note that the energy contact contribution above is calculated only in the contact point, which can be interpreted as Lobatto integration of (9), see, e.g., [6] for more details.

Variation of this part of energy yields

$$\delta W^c = p g_c \delta g_c \quad (12)$$

It is worth noting that the term $p g(\mathbf{u})$ represents the contact force. Variation of the gap function can be further expressed as

$$\delta g_c = \left(\mathbf{N}^{*T} \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) \delta \mathbf{d}_c \quad (13)$$

where $\delta \mathbf{d}_c$ are the variational nodal displacements of relevant FEM nodes and \mathbf{n} is the normal vector in deformed configuration.

The external forces of contact are therefore

$$\mathbf{f}_c = -p g_c \left(\mathbf{N}^{*T} \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) \quad (14)$$

To obtain the associated tangential stiffness matrix, the internal forces have to be subjected to differentiation with respect to nodal displacements \mathbf{d}_c [5]. For small strains, it is generally assumed that the direction of contact does not change throughout the analysis, and therefore \mathbf{n} remains independent of \mathbf{d}_c [5, 6].

$$\mathbf{K}_c = -\frac{\partial \mathbf{f}_c}{\partial \mathbf{d}_c} = p \left(\mathbf{N}^{*T} \frac{\mathbf{n}}{\|\mathbf{n}\|} \right)^T \left(\mathbf{N}^{*T} \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) \quad (15)$$

2.5. EXTENSION TO LARGE STRAIN PROBLEMS

Now we stand before a question: which of the previously described considerations change if we consider large strains? The mechanics of introducing contact constraints to FEM remain the same, as well as the definition of contact energy.

Two critical differences appear. Firstly, it can no longer be assumed that contact follows the same direction throughout the whole computation. The normal vector has to be reassessed after each computational step and also the stiffness matrix has to reflect this. Secondly, large deformations result in change of contact point on the segment, which is reflected in the evaluation of matrix \mathbf{N}^* .

The normal vector is deformation dependent and can be expressed geometrically [2]

$$\mathbf{n} = \mathbf{t}_1 \times \mathbf{t}_2 \quad (16)$$

where \times is the vector cross product and \mathbf{t}_i are tangential vectors of the element boundary in the deformed configuration (in the case of an element edge, i.e., contact formulation in 2D, one is the tangent and the other is the \mathbf{e}_3 , that is, a unit vector perpendicular to the plane).

For the new expression of stiffness, we elect to utilize a formula presented in [2]. Specifically for the case of node-to-2D-segment penalty method, it reads

$$\mathbf{K} = p \left(\mathbf{N}_v \mathbf{N}_v^T - \frac{g_c}{l} (\mathbf{B}_v \mathbf{T}_v^T + \mathbf{T}_v \mathbf{B}_v^T + \frac{g_c}{l} \mathbf{B}_v \mathbf{B}_v^T) \right) \quad (17)$$

with

$$\mathbf{N}_v = \mathbf{N}^{*T} \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad \mathbf{B}_v = \mathbf{B}^{*T} \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad \mathbf{T}_v = \mathbf{N}^{*T} \mathbf{t} \quad (18)$$

where \mathbf{B}^* is the extended B-matrix. Note that the first member of the stiffness matrix corresponds with the previously derived term for the small strain case.

For a linear edge in 2D, the extended matrices are defined as

$$\mathbf{N}^* = \begin{bmatrix} -1 & 0 & \xi - 1 & 0 & -\xi & 0 \\ 0 & -1 & 0 & \xi - 1 & 0 & -\xi \end{bmatrix} \quad (19)$$

$$\mathbf{B}^* = -\frac{\partial \mathbf{N}^*}{\partial \xi} = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \quad (20)$$

where ξ is the parametric coordinate on the element edge.

3. IMPLEMENTATION

3.1. CLASS STRUCTURE

The OOFEM code has a strictly object-oriented structure [4]. The integration of contact into this structure has been mainly accomplished in [5] and only slight additions were necessary for the large strain extension.

Figure 1 displays the relevant parts of OOFEM class structure together with the classes used for contact.

The general idea of class structure centers around two main classes: `ElementEdgeContactSegment` and `Node2SegmentPenaltyContact`.

`ElementEdgeContactSegment` is derived from the general `ContactSegment` class and represents element edges in 2D. When initialized from an OOFEM input file, the class receives a set of element boundaries. Internal processes ensure that contact is considered always with the edge which is closest to the contacting node [5].

`Node2SegmentPenaltyContact` is a subclass of `ActiveBoundaryCondition`, an OOFEM feature

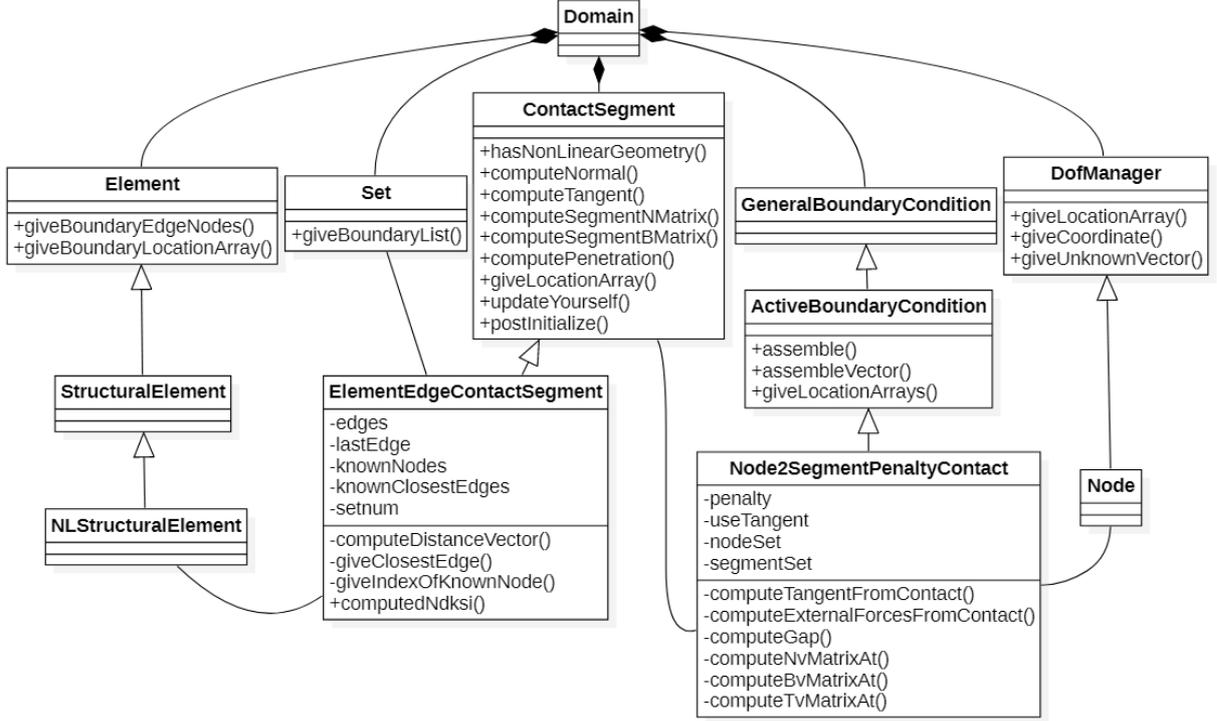


FIGURE 1. A diagram of OOFEM class structure, focused on contact-relevant parts, created with the help of [7].

which allows boundary conditions to contribute by themselves to the internal forces, external forces, and stiffness matrix. The utilization of this feature avoids creating any form of a "contact element", otherwise prevalent among FEM contact implementations [3]. The contact boundary condition is likewise initialized from the OOFEM input file with information on the size of the penalty parameter, a set of nodes and a set of one or more contact segments. Any node is always checked for contact with any segment [5].

3.2. OPERATIONS

The assembly of external forces and stiffness matrix takes place in the `Node2SegmentPenaltyContact` class, specifically in its overridden methods `assembleVector()` and `assemble()`. For the determination of gap g_c , the contact segment method `computePenetration()` is invoked, which computes it from a purely geometrical consideration, considering the node to have penetrated the segment when its deformed coordinates are on a different side of the segment than the undeformed ones.

The contact segment is also responsible for the computation of the tangent and normal vectors. The tangent vector is determined as

$$\begin{aligned} \mathbf{t} &= \frac{\partial \mathbf{x}}{\partial \xi} = -\mathbf{B}_{seg}^* \mathbf{x}_N = \\ &= \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{Bmatrix} \end{aligned} \quad (21)$$

where \mathbf{B}_{seg}^* is the segment part of the extended B-matrix and x_i, y_i coordinates of the segment nodes. Meanwhile, the normal vector is determined as

$$\mathbf{n} = \mathbf{e}_3 \times \mathbf{t} \quad (22)$$

for large strains and from the `Element::edgeEvalNormal()` function for small strains (which returns the undeformed normal vector).

Neither of those vectors is normalized. The normalization of the normal vector happens in the contact condition, when the \mathbf{N}_v , \mathbf{B}_v and \mathbf{T}_v matrices are determined according to (18). In the assembly of stiffness, the contact conditions asks the contact segment whether the large strain part shall be assembled. This is accomplished by the `ContactSegment::hasNonLinearGeometry()` function and based on the setting of the element in contact.

4. DEMONSTRATION

To demonstrate the functionality of our approach, we select a simple computational example in a 2D plane strain idealization. The mesh used can be seen in Figure 2a.

A double-clamped beam is positioned above a cantilever. The length of the beam is 5 m and length of the cantilever is 3 m. Both are 0.25 m high with a gap of 0.25 m between them. The top of the beam is loaded in every step by an uniform distributed load of 8000 kN/m, causing vertical deformation and thus

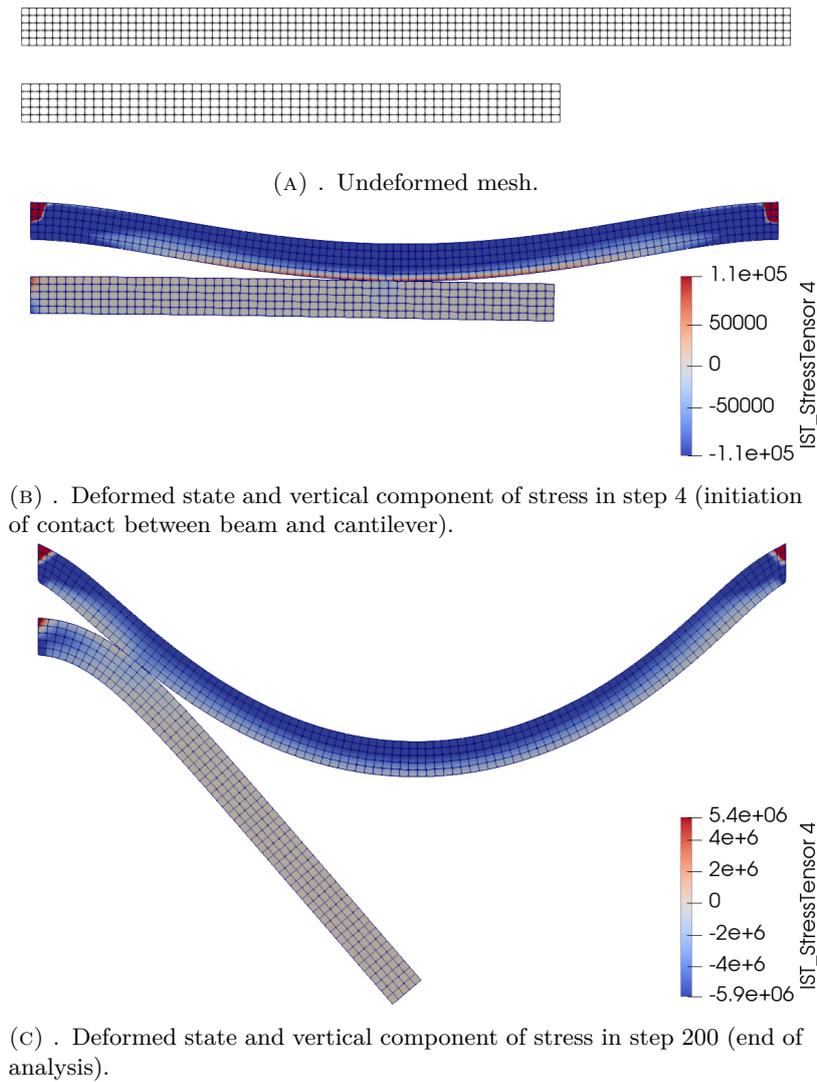


FIGURE 2. A computational example demonstrating the code and its convergence.

Iteration	Force err. x	Disp. err. x	Force err. y	Disp. err. y
1	5.074e-05	0.000e+00	5.450e-03	0.000e+00
2	7.311e-03	8.044e-02	1.583e-01	4.515e-02
3	1.879e-04	2.920e-03	4.120e-03	2.014e-03
4	1.465e-04	2.844e-03	3.202e-03	1.062e-03
5	2.041e-07	5.071e-04	1.488e-06	1.809e-04
6	1.418e-12	1.863e-06	3.298e-11	7.549e-07

TABLE 1. Evolution of global force and displacement errors in iterations of loading step 14.

subsequently contact between the two bodies. Loading is performed in 200 uniform loading steps.

Both beam and cantilever are modelled by rectangular plane strain elements (OOFEM element `Quad1PlaneStrain`). In total, the task mesh consists of 805 elements. Refinement of the mesh has been accomplished with the help of a custom script in the MATLAB software [8]. All elements are made of the same Saint-Venant Kirchhoff material with the parameters $\lambda = 100$ MPa and $\mu = 150$ MPa.

Contact conditions are introduced between the

lower edge of the beam and the upper edge of the cantilever below. On the latter interface, two boundary conditions are defined, one pairing the beam's nodes with the cantilever's element edges and the other vice versa. This ensures that no clipping caused by crude or otherwise imperfect meshing can take place.

Figure 2b shows the deformed state of the mesh and the vertical component of the stress tensor in step 4 of 200, when the beam and the cantilever first come into contact. Likewise, Figure 2c shows the same in step 200 at the end of the analysis. To render those

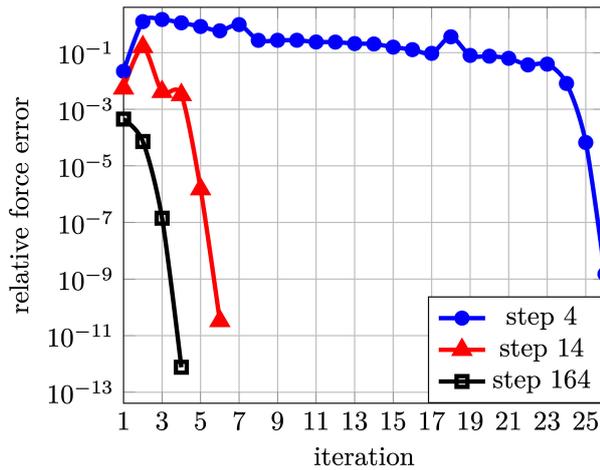


FIGURE 3. Evolution of error during the equilibrium iteration process.

figures, the ParaView open-source software was used [9]. Values of stress in the visualizations are given in Pa.

Note that between steps 4 and 200, the contact surface moves significantly to the left both on the beam and the cantilever. This was accomplished by alternating activation of the two opposite boundary conditions on the interface, which managed to subsequently introduce contact between neighboring node-segment pairs, while the contact gap between the original pairs opened up again.

The main objective of this test was to prove quadratic convergence of the new approach to large strain contact stiffness matrix.

The vast majority of the 200 steps computed managed to converge in 6 iterations or less. The only exception is step 4, in which the contact occurs for the first time. Table 1 shows error evolution in a typical computational step (step 14 was chosen). Moreover, the dependence of error (norm of residual forces) on the number of iterations during the equilibrium Newton-Raphson process for steps 4, 14, and 164 is visualized in Figure 3. Thanks to the semi-logarithmic scale used, it is apparent that approximate quadratic convergence has been achieved.

5. CONCLUSIONS

The paper describes an implementation of a two dimensional large strain node-to-segment penalty-based approach to contact mechanics into an open-source finite element code OOFEM. The emphasis is given on proper design of the hierarchy of classes with a focus on modularity and extensibility towards different methods for handling contact constraints, contact with friction, or three dimensional formulation. The functionality of the presented approach is successfully verified on a numerical example and a quadratic rate of convergence of the Newton-Raphson iteration procedure is demonstrated.

ACKNOWLEDGEMENTS

The authors would like to acknowledge support received for work on this project from a CTU grant no. SGS20/038/OHK1/1T/11.

REFERENCES

- [1] H. Hertz. On the contact of elastic solids. *Z Reine Angew Mathematik* **92**:156–171, 1881.
- [2] P. Wriggers. *Computational contact mechanics*. Springer, New York, 2nd edn., c2006.
- [3] V. A. Yastrebov. *Numerical methods in contact mechanics*. Wiley, Hoboken, NJ, 2013.
- [4] B. Patzák. OOFEM home page, 2000. [Http://www.oofem.org](http://www.oofem.org).
- [5] O. Faltus. *Object-Oriented Design and Implementation of Contact Mechanics into Finite Element Code OOFEM*. Master's thesis, Czech Technical University in Prague, 2020.
- [6] A. Konyukhov, R. Izi. *Introduction to computational contact mechanics*. Wiley, Chichester, West Sussex, 2015.
- [7] StarUML. *Version 3.2.2*. MKLabs Co.,Ltd., Seoul, Republic of Korea, 2020.
- [8] MATLAB. *Version 9.3.0 (R2017b)*. The MathWorks Inc., Natick, Massachusetts, 2017.
- [9] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, 1st edn., 2015.