Signorini frictionless contact problem

Dr.-Ing. Andreas Apostolatos

March 22, 2020

1 Theory

1.1 Linear elasticity

Given is a deformable body which is geometrically described by $\Omega \subset \mathbb{R}^d$ where $d = 2, 3$ stands for the number of the problem's spatial dimensions. Its deformation due to the applied boundary conditions can be uniquely defined by a displacement field $\mathbf{u} = u_i \, \mathbf{e}_i$, $\mathbf{e}_i$ being the Cartesian basis, which maps each each material point of the reference configuration $\mathbf{X} \in \Omega$ to a material point in the current configuration $\mathbf{x} \in \Omega_t$, that is, $\mathbf{x} = \mathbf{X} + \mathbf{u}$.

The strain is described by means of the *Green-Lagrange* (GL) strain second order tensor $\boldsymbol{\mathcal{E}} \in \mathfrak{S}^2$ given by,

$$\boldsymbol{\mathcal{E}} = \frac{1}{2} \left(\boldsymbol{\nabla} \mathbf{u} + (\boldsymbol{\nabla} \mathbf{u})^{\mathrm{t}} + \boldsymbol{\nabla} \mathbf{u} \cdot (\boldsymbol{\nabla} \mathbf{u})^{\mathrm{t}} \right) \, , \tag{1}$$

where the underlying symbols are understood as follows,

$$\boldsymbol{\nabla} \mathbf{u} = \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial u_i}{\partial X_j} \, \mathbf{e}_i \otimes \mathbf{e}_j \, , \tag{2a}$$

$$(\boldsymbol{\nabla} \mathbf{u})^{\mathrm{t}} = \sum_{k=1}^{d} \sum_{l=1}^{d} \frac{\partial u_l}{\partial X_k} \, \mathbf{e}_k \otimes \mathbf{e}_l \, , \tag{2b}$$

$$\boldsymbol{\nabla} \mathbf{u} \cdot (\boldsymbol{\nabla} \mathbf{u})^{\mathrm{t}} = \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} \frac{\partial u_i}{\partial X_j} \frac{\partial u_j}{\partial X_k} \, \mathbf{e}_i \otimes \mathbf{e}_k \, . \tag{2c}$$

Assumed is also a Saint-Venant material model which is governed by the material fourth order tensor $\boldsymbol{\mathcal{C}} \in \mathfrak{S}^2$

$$\boldsymbol{\mathcal{C}} = \frac{E}{2(1+\nu)} \left(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} + \frac{2\nu}{1-2\nu} \delta_{ij} \delta_{kl} \right) \, \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l \, , \tag{3}$$

where E and ν stand for the Young's modulus and Poisson ratio of the material, respectively. The stress state of the problem is described by means of the 2nd *Piola-Kirchhoff* (PK2) stress second order ternsor $\boldsymbol{\mathcal{S}} \in \mathfrak{S}^2$ which is defined as and given by the linear elastic isotropic law as,

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{C}} : \boldsymbol{\mathcal{E}} \, . \tag{4}$$

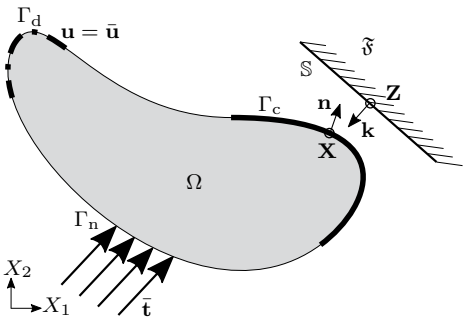

Figure 1: Theory: Signorini frictionless contact problem with boundary conditions.

Given the Dirichlet boundary conditions along Γ_d where the displacement is prescribed to a value $\bar{\mathbf{u}}$ which is assumed to be zero without loss of generality, the Neumann boundary conditions along Γ_n where external traction $\bar{\mathbf{t}}$ is applied and applied body forces $\mathbf{b}$, see Fig. 1, the strong form of the elasticity problem writes,

$$\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{S}} + \mathbf{b} = \mathbf{0} \quad \text{in } \Omega \,, \tag{5a}$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_\mathrm{d} \,, \tag{5b}$$

$$\boldsymbol{\mathcal{S}} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_\mathrm{n} \,, \tag{5c}$$

where $\mathbf{n}$ stands for the unit outward normal vector to Neumann boundary Γ_n.

1.2 Contact conditions

Assumed is also that there is a boundary $\Gamma_\mathrm{c} = \Gamma_\mathrm{c}(\mathbf{u})$ along which body Ω is expected to come into contact with a rigid body $\mathfrak{F}$ along a portion of its surface $\mathbb{S} \subset \partial\mathfrak{F}$. Assumed is that for each material particle $\mathbf{X} \in \Gamma_\mathrm{c}$ there exists a unique material particle $\mathbf{Z} \in \mathbb{S}$ which is closest to $\mathbf{X}$ in Euclidean sense,

$$\mathbf{Z}(\mathbf{X}) = \arg\min_{\mathbf{Y} \in \Gamma_\mathrm{c}} \|\mathbf{X} - \mathbf{Y}\|_2 \ . \tag{6}$$

The so-called gap function is then defined along Γ_c, namely $\mathbf{g} : \Gamma_\mathrm{c} \to \mathbb{R}$ and measures the normal distance between the deformable and the rigid body Ω along Γ_c and $\mathbb{S}$,

$$g = u_n - g_0 \quad \text{for all } \mathbf{X} \in \Gamma_\mathrm{c} \,, \tag{7}$$

where $u_n = \mathbf{u} \cdot \mathbf{n}$ is the normal component of the displacement field given the outward normal $\mathbf{n}$ to Γ_c at $\mathbf{X} \in \Gamma_\mathrm{c}$ and $g_0(\mathbf{X})$ is the initial gap given by,

$$g_0(\mathbf{X}) = -(\mathbf{X} - \mathbf{Z}(\mathbf{X})) \cdot \mathbf{n}(\mathbf{X}) \quad \text{for all } \mathbf{X} \in \Gamma_\mathrm{c} \,. \tag{8}$$

Herein it is assumed that neither the closest point $\mathbf{Z}$ nor the outward normal $\mathbf{z}$ depend on the displacement field $\mathbf{u}$ but solely on the point $\mathbf{X} \in \Gamma_\mathrm{c}$ and that $\mathbf{k}(\mathbf{X}) = -\mathbf{n}(\mathbf{X})$, where $\mathbf{k}(\mathbf{X})$ stands for the outward normal to $\mathbb{S}$ at $\mathbf{Z} \in \mathbb{S}$.

These assumptions are valid for sufficiently small gaps. Moreover, the following conditions are assumed,

i. No material particle can penetrate the body $\mathfrak{F}$, hence $g(\mathbf{X}) \leq 0$ for all $\mathbf{X} \in \Gamma_{\mathrm{c}}$,

ii. Surface $\mathbb{S}$ is sufficiently lubricated such that no shear tractions develop, that is, $t_t = \mathbf{n} \cdot \boldsymbol{\mathcal{S}} \cdot \mathbf{t} = 0$ where $\mathbf{t}$ stands for the normal tangent vector to $\mathbb{S}$,

iii. The normal tractions along Γ_{c} are compressive, that is, $t_n = \mathbf{n} \cdot \boldsymbol{\mathcal{S}} \cdot \mathbf{n} \geq 0$,

iv. If there is contact along Γ_{c} then $g = 0$ and $t_n \geq 0$ whereas is there is no contact along Γ_{c} then $g \leq 0$ and $t_n = 0$

In this way, the so-called complementarity conditions can be stated as follows,

$$g \leq 0 \,, \tag{9a}$$

$$t_n \geq 0 \,, \tag{9b}$$

$$t_n \, g = 0 \,. \tag{9c}$$

Condition in Eq. (9a) stands for the non-penetration condition whereas condition in Eq. (9b) states that the stresses along the contact boundary need to be compressive. Eq. (9c) is a direct consequence of condition iv.

1.3 Variational formulation

The weak formulation of the problem in Eq. (5) subject to the complementarity conditions in Eq. (9) can be written by means of the Langrange Multipliers method: Find $\mathbf{u} \in \boldsymbol{\mathcal{H}}^1(\Omega)$ with $\mathbf{u} = \mathbf{g}$ on Γ_{d} and $\lambda \in \mathcal{L}^2(\Gamma_{\mathrm{c}})$ with $\lambda \geq 0$ such that,

$$\int_{\Omega} \delta\boldsymbol{\mathcal{E}} : \boldsymbol{\mathcal{S}} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{c}}} \delta u_n \, \lambda \, \mathrm{d}\Gamma = \int_{\Omega} \delta\mathbf{u} \cdot \mathbf{b} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{n}}} \bar{\mathbf{t}} \cdot \mathbf{u} \, \mathrm{d}\Gamma \tag{10a}$$

$$\int_{\Gamma_{\mathrm{c}}} \delta\lambda \, u_n \, \mathrm{d}\Gamma = \int_{\Gamma_{\mathrm{c}}} \delta\lambda \, g_0 \, \mathrm{d}\Gamma \,, \tag{10b}$$

for all $\delta\mathbf{u} \in \boldsymbol{\mathcal{H}}^1(\Omega)$ and for all $\delta\lambda \in \mathcal{L}^2(\Gamma_{\mathrm{c}})$ with $\delta\lambda \geq 0$. In fact, λ in this case represents the normal traction along the contact boundary Γ_{c} and therefore it must be positive, see Eq. (9b). Eq. (10a) is nothing else but the internal virtual work in Ω, whereas Eq. (10b) accounts for the variation of complementarity condition in Eq. (9c) and renders the problem symmetric as it is also demonstrated in its discrete form.

Important is to note that the variational problem in Eq. (10) is valid only when $\Gamma_{\mathrm{c}} \neq \emptyset$. The latter means that there has to be a non-empty set Γ_{c} along which the complementarity conditions in Eq. (9) are valid. Otherwise, in case that $\Gamma_{\mathrm{c}} = \emptyset$ variational problem in Eq. (10) reduces to: Find $\mathbf{u} \in \boldsymbol{\mathcal{H}}^1(\Omega)$ with $\mathbf{u} = \mathbf{g}$ on Γ_{d} such that,

$$\int_\Omega \delta\boldsymbol{\mathcal{E}} : \boldsymbol{\mathcal{S}}\, \mathrm{d}\Omega = \int_\Omega \delta\mathbf{u}\cdot\mathbf{b}\, \mathrm{d}\Omega + \int_{\Gamma_\mathrm{n}} \bar{\mathbf{t}}\cdot\mathbf{u}\, \mathrm{d}\Gamma \quad \text{for all } \delta\mathbf{u}\in\boldsymbol{\mathcal{H}}^1(\Omega)\ . \tag{11}$$

2 Discretization

Assumed is that the displacement field is discretized with the standard linear *Finite Element Method* (FEM) on triangles or quadrilaterals. Employing the Buvnon-Galerkin FEM then the unknown displacement field $\mathbf{u}$ and its variation $\delta\mathbf{u}$ are discretized using the same basis functions $\boldsymbol{\varphi}_i$, that is,

$$\mathbf{u}_h = \sum_{i=1}^{n} \boldsymbol{\varphi}_i \hat{u}_i\ , \tag{12a}$$

$$\delta\mathbf{u}_h = \sum_{i=1}^{n} \boldsymbol{\varphi}_i \delta\hat{u}_i\ , \tag{12b}$$

where the linear/bilinear basis functions $\boldsymbol{\varphi}_i$ are constructed as a dual product of the linear/bilinear shape functions N_i at the element's parametric space and the Cartesian basis $\mathbf{e}_i$. Accordingly, $\hat{u}_i$ and $\delta\hat{u}_i$ stand for the *Degrees of Freedom* (DOFs) of the unknown and the test fields, respectively, and $n\in\mathbb{N}$ is the number of nodes in the mesh. Subscript h indicates then the smallest element length size within the employed mesh.

Within the node-based contact method, the shape functions for the discretization of the Lagrange Multipliers fields λ and $\delta\lambda$ are chosen the diract delta distributions supported on the finite element nodes $\mathbf{X}_i$, that is,

$$\lambda_h = \sum_{i=1}^{n_\mathrm{c}} \delta(\mathbf{X}_i)\hat{\lambda}_i\ , \tag{13a}$$

$$\delta\lambda_h = \sum_{i=1}^{n_\mathrm{c}} \delta(\mathbf{X}_i)\delta\hat{\lambda}_i\ , \tag{13b}$$

where as before $\hat{\lambda}_i$ and $\delta\hat{\lambda}_i$ stand for the Lagrange Multipliers DOFs of the unknown and test fields, respectively. Note that altough there are two DOFs per node for the displacement field, there is only one DOF per node for the Lagrange Multipliers field. This is because the displacement field is a vector field in the $\mathbb{R}^2$ space in two-dimensional elasticity whereas the Lagrange Multipliers field is a scalar field due to the complementarity constraint in Eq. (9c) which is a scalar-valued constraint. In case of a vector-valued constraint, then the Lagrange Multipliers field would also be a vector field.

Substituting the latter expressions in Eq. (10) the following discrete system of equations is obtained in terms of a Newton-Raphson formulation,

$$\begin{bmatrix} \mathbf{K}(\hat{\mathbf{u}}_{\hat{i}}, \hat{\boldsymbol{\lambda}}_{\hat{i}}) & \mathbf{C}^\mathrm{t} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \Delta_{\hat{i}} \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}(\hat{\mathbf{u}}_{\hat{i}}, \hat{\boldsymbol{\lambda}}_{\hat{i}}) \\ \boldsymbol{\mathcal{R}}(\hat{\mathbf{u}}_{\hat{i}}) \end{bmatrix}\ , \tag{14}$$

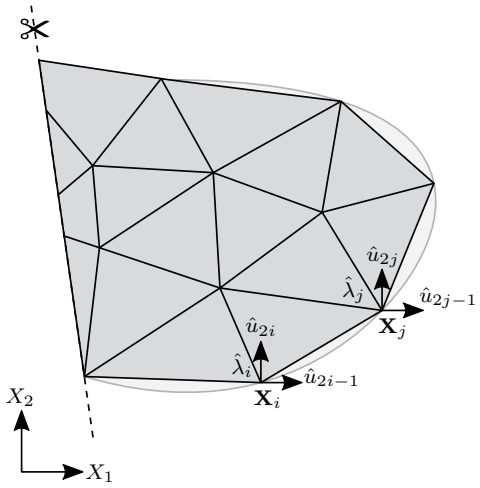

Figure 2: Discretization: Finite element mesh and degrees of freedom.

over the so-called active set of Lagrange Multipliers DOFs. The i-th component of the residual vector $\mathbf{R}$ derived from Eq. (10a) is defined as follows,

$$R_i(\hat{\mathbf{u}}_{\hat{i}}, \hat{\boldsymbol{\lambda}}_{\hat{i}}) = \int_\Omega \frac{\partial \boldsymbol{\mathcal{E}}}{\partial \hat{u}_i} : \boldsymbol{\mathcal{S}} \, \mathrm{d}\Omega + \sum_{j=1}^{n_\mathrm{c}} \bar{n}(\mathbf{X}_j) \, \hat{\lambda}_j - \int_\Omega \boldsymbol{\varphi}_i \cdot \mathbf{b} \, \mathrm{d}\Omega - \int_\Omega \boldsymbol{\varphi}_i \cdot \bar{\mathbf{t}} \, \mathrm{d}\Gamma \ . \quad (15)$$

The second term in Eq. (15) is derived by inserting Eq. (13a) in Eq. (10a) and by using the well-known identity of the Dirac delta distribution, that is,

$$\int_{\Gamma_\mathrm{c}} \delta u_n \, \lambda \, \mathrm{d}\Gamma \approx \int_{\Gamma_\mathrm{c}} \delta \mathbf{u}_h \cdot \mathbf{n} \, \lambda_h \, \mathrm{d}\Gamma = \sum_{i=1}^{n} \sum_{j=1}^{n_\mathrm{c}} \delta \hat{u}_i \int_{\Gamma_\mathrm{c}} \boldsymbol{\varphi}_i \cdot \mathbf{n} \, \delta(\mathbf{X}_j) \hat{\lambda}_j \, \mathrm{d}\Gamma =$$
$$\sum_{i=1}^{n} \sum_{j=1}^{n_\mathrm{c}} \delta \hat{u}_i \, \boldsymbol{\varphi}_i(\mathbf{X}_j) \cdot \mathbf{n} \, \hat{\lambda}_j = \sum_{j=1}^{n_\mathrm{c}} \delta \hat{u}_j \, \mathbf{e}_{\mathrm{mod}\left(\frac{j-1}{2}\right)+1} \cdot \mathbf{n} \, \hat{\lambda}_j = \sum_{j=1}^{n_\mathrm{c}} \delta \hat{u}_j \, \bar{n}(\mathbf{X}_j) \, \hat{\lambda}_j \ ,$$

where $\bar{n}(\mathbf{X}_j) = n_1(\mathbf{X}_j) + n_2(\mathbf{X}_j)$ is nothing else but the sum of the components of the outward normal vector $\mathbf{n}$ to the boundary Γ_c at node $\mathbf{X}_j \in \Gamma_\mathrm{c}$.

It is worth noting that within linear FEM, the boundary Γ_c is only piecewise continuous which means that the the outward normal vector $\mathbf{n}$ to Γ_c is not uniquely defined at the finite element nodes $\mathbf{X}_i$. Therefore, $\mathbf{n}(\mathbf{X}_i)$ is constructed by averaging the outward normal vectors from the neighbouring elements along Γ_c.

The i-th component of the residual vector $\boldsymbol{\mathcal{R}}$ derived from Eq. (10b) is defined as follows,

$$\mathcal{R}_i(\hat{u}_i) = \bar{n}(\mathbf{X}_i) \, \hat{u}_i - g_0(\mathbf{X}_i) \ . \quad (16)$$

Then, the tangent stiffness matrix $\mathbf{K}$ has components,

$$K_{ij}(\hat{\mathbf{u}}_{\hat{i}}) = \frac{\partial R_i}{\partial \hat{u}_j} = \int_\Omega \frac{\partial \boldsymbol{\mathcal{E}}}{\partial \hat{u}_i} : \frac{\partial \boldsymbol{\mathcal{S}}}{\partial \hat{u}_j} + \frac{\partial^2 \boldsymbol{\mathcal{E}}}{\partial \hat{u}_i \partial \hat{u}_j} : \boldsymbol{\mathcal{S}} \, \mathrm{d}\Omega \, . \tag{17}$$

The Lagrange Multipliers matrix $\mathbf{C}$ is then a diagonal matrix due to the choice of the discretization for the Lagrange Multipliers field and has entries,

$$C_{ii} = \bar{n}(\mathbf{X}_i) \, , \tag{18}$$

at each active node $\mathbf{X}_i \in \Gamma_c$. Vectors $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\lambda}}$ stand for the collection of all the displacement and Lagrange Multipliers DOFs as follows,

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_n \end{bmatrix} \, , \tag{19a}$$

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{\lambda}_1 & \cdots & \hat{\lambda}_{n_c} \end{bmatrix} \, . \tag{19b}$$

The index $\hat{i}$ on $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\lambda}}$ indicates the Newton-Raphson iteration and $\Delta_{\hat{i}}(\bullet) = (\bullet)_{\hat{i}+1} - (\bullet)_{\hat{i}}$.

3 Realization

As aforementioned, the actual contact surface Γ_c is not known in advance, since it depends on the solution $\mathbf{u}$. Given the variational inequality nature of the problem, an iterative approach is employed. There are two possible procedures currently implemented to solve the problem, illustrated in Algorithm 1 and Algorithm 2.

1. Remove fully constrained nodes and compute the initial gap function in Eq. (8) ;
2. Compute the tangential stiffness matrix of the structure in Eq. (17) ;
3. Create the expanded system of equations ;
4. Solve the system iteratively ;

while *inactive nodes change* **and** *Lagrange Multipliers are not valid* **do**

   4.1. Find the *inactive nodes* from expanded displacement (Lagrange Multiplier) vector ;

   4.2. Reduce the system of equations according to the *inactive nodes* ;

   4.3. Solve the reduced system of equations in Eq. (14) ;

   4.4. Assemble the expanded displacement (Lagrange Multiplier) vector in Eq. (19) ;

   4.5. Evaluate convergence conditions ;

end

5. Get the values for displacements and Lagrange Multipliers ;

Algorithm 1: Contact algorithm 1

1. Remove fully constrained nodes and compute the initial gap function in Eq. (8) ;
2. Compute the tangential stiffness matrix of the structure in Eq. (17) ;
3. Reduce the system according to given constraints ;
4. Solve the system iteratively ;

while *active nodes change* **do**

   4.1. Build the expanded (complete) displacement vector ;

   4.2. Detect *active nodes* according to Eq. (9a) ;

   4.3. Rebuild the equation system in Eq. (14) if new *active nodes* are found ;

   4.4. Relax the system until only valid Lagrange multipliers are computed ;

   while *Lagrange Multipliers are not valid* **do**

      4.4.1. Compute displacements and Lagrange multipliers ;

      4.4.2. Detect and delete non-valid Lagrange Multipliers according to Eq. (9b) and Eq. (9c) ;

   end

end

5. Get the values for displacements and Lagrange Multipliers ;

Algorithm 2: Contact algorithm 2

Both algorithms work in a similar manner. At first we need to compute the initial gap function in Eq. (8) and tangential stiffness matrix of the structure in Eq. (17). Now we iteratively solve the system of equations.

In Algorithm 1 we first create the complete expanded system of equations considering all possible combinations of contact nodes and rigid boundary. Now we check the displacement vector for nodes that are non-penetrating and have non-valid Lagrange Multipliers to detect the so-called *inactive nodes*. This happens at the same time. By using *inactive nodes* we then reduce the system, re-calculate the displacement vector and repeat the previous step until solution converges. The focus here is on the reduction of the equation system.

Algorithm 2 is a bit more complex. Here we first check the displacement vector for penetrating nodes or *active nodes*. We then build the equation system in Eq. (14) in each iteration step for *active nodes* only. At this point we don't know if all corresponding Lagrange Multipliers will be valid, so another nested loop is needed to compute the displacement vector and continuously check for non-valid Lagrange Multipliers. This loop runs until only valid multipliers remain. Displacement vector is updated and the cycle repeats until solution converges. In contrast to Algorithm 1, the focus now is on the system expansion.

By judging the speed of both algorithms only on the number of processed con-

tact nodes, it is clear that Algorithm 2 would preform faster. This is due to the fact, that here the equation system is build only for currently active nodes while in Algorithm 1 we build the whole system from the start. But in Algorithm 1 the system only has to be build once (at the beginning) and not in each iteration step as in Algorithm 2. Therefore the actual computation time depends on the implementation details and varies from case to case.

Before computing the gap function we can remove fully constrained nodes. Nodes that have both DOFs (x,y) fully constrained do not need to be tested for possible contact with the rigid boundary since they will not move. This is not necessary, but it can reduce the computation time.

4 Appendix

This section focuses on the numerical implementation of the Signorini frictionless contact problem in MATLAB computing environment. All data types and functions used in the script are described fully in detail and reflect the algorithms discussed previously.

4.1 Structure description

4.1.1 contactSegments

Is a struct that contains the information about rigid boundary. It consists out of three fields. Field *.points* is a 2x2xN matrix of coordinates structured as follows,

$$\begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \end{bmatrix}. \tag{20}$$

Coordinates x_0 and y_0 represent the first point **A**, while coordinates x_1 and y_1 represent the second point **B** of the segment. During the calculation, struct **contactSegments** is further expanded by the field *.normals* which is a Nx2 matrix. Where N is the number of segments defined in the field *.number*.

When defining the rigid boundary user only has to define the field *.points*. Orientation of each segment has to be respected by the user so that the normal vectors are orientated towards the body of interest. Other two fields are constructed in the function **buildSegmentsData**. Individual segments are independent from each other and do not have to be connected. It is recommended for individual segments to be longer than the size of mesh elements.

4.1.2 propContact

Is a struct that contains the information about the possible contact nodes defined by the user in GiD. It consists out of four fields. Field *.nodeIDs* is a vector that contains global numbering of nodes. Size of that vector is stored in field *.numberOfNodes*. Both fields are constructed during the parsing from GiD input file. Maximum number of solver iterations is stored in the field *.maxIter*. Field *.gap* contains perpendicular distances from each segment to each contact node and it is constructed during calculation time. It is a MxN matrix where M is

the number of contact nodes and N is the number of segments. Each column is linked to one segment.

User must define the possible contact nodes in GiD. In case of large applied forces, specifying the whole body as a boundary is usually a safe bet since large displacements and deformations in the first step of solver iteration might cause the contact nodes to move far away from the rigid boundary and in turn violating the linear gap function.

4.1.3 lagrange

Is a struct containing the information about Lagrange Multipliers. It consists out of two fields. Field *.active_nodes* contains the global numbering of the mesh nodes where Lagrange Multipliers apply. So only the active contact nodes. While field *.multipliers* stores the values of Lagrange Multipliers of that nodes. This enables us to visualize the contact nodes and see the results.

4.2 Function description

4.2.1 buildSegmentsData

Is a function that expands a data structure **segments** by the field *.normals*. Normal vectors are perpendicular to the segment and normalized to have an unit length. Segment that starts in point **A** and ends in point **B** has its normal pointing in the counterclockwise direction (Figure 3).

4.2.2 computeGapFunction

Is a function that expands the data structure **propContact** by the field *.gap*. Gap function in our case is nothing more than a perpendicular distance from each node to each segment. Gap is shown on Figure 3 and is calculated in the following way. Point **P** represents a node in undeformed configuration, while point **R** is the position of that node after applied displacement **u** and can be represented as follows,

$$\mathbf{R} = \mathbf{P} + \mathbf{u}\,. \tag{21}$$

Initial gap function is only calculated for undeformed configuration at $\mathbf{u} = 0$, therefore $\mathbf{R} = \mathbf{P}$ and $\mathbf{g} = \mathbf{g_0}$. Point $\mathbf{R}^*$ is the projection of the point **R** on the segment **AB** and can be represented by a linear combination of points **A** and **B** by parameter α.

$$\mathbf{R}^* = (1-\alpha)\mathbf{A} + \alpha\mathbf{B}\,. \tag{22}$$

As vectors **AB** and $\mathbf{R}^*\mathbf{R}$ are perpendicular, their dot product must be equal to zero,

$$(\mathbf{B} - \mathbf{A}) \cdot (\mathbf{R} - \mathbf{R}^*) = 0\,. \tag{23}$$

If we substitute Eq. (22) into Eq. (23) we get a closed form for parametric distance α along the segment **AB**,

$$\alpha = \frac{(\mathbf{A} - \mathbf{B}) \cdot (\mathbf{R} - \mathbf{A})}{(\mathbf{A} - \mathbf{B}) \cdot (\mathbf{B} - \mathbf{A})} \,, \tag{24}$$

$$\mathbf{g} = \mathbf{n} \cdot (\mathbf{R} - \mathbf{R}^*) \,. \tag{25}$$

Now we can easily compute point $\mathbf{R}^*$. Gap is then the dot product between normal and vector $\mathbf{R}^*\mathbf{R}$ in Eq. (25) In case the node is penetrating the distance is negative, otherwise it is positive.

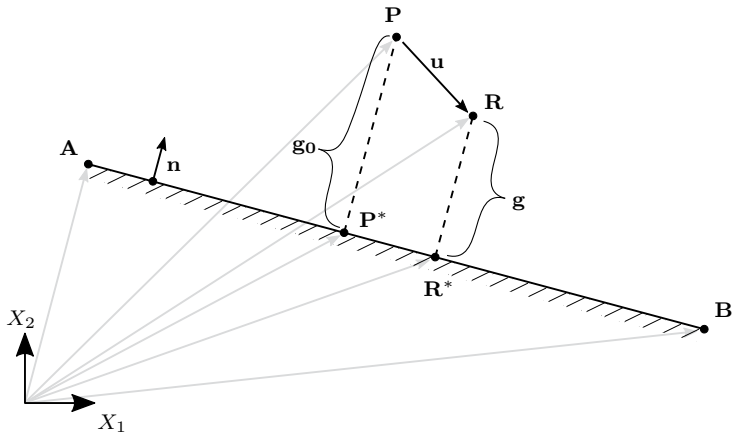

Figure 3: Calculation of the gap function (Euclidean distance).

4.2.3 buildConstraintMatrix

Is a function that builds the constraint (Lagrange Multipliers) matrix **C** in Eq. (18) to be appended to the tangent stiffness matrix **K** in Eq. (17). The constraint matrix is built with dimensions MxN, where M is the number of DOFs and N is the number of active Lagrange Multipliers. The matrix is filled with the normal vectors of the segments.

In case that the function is used in Algorithm 1, the **C** matrix is constructed only once at the beginning. Number of active Lagrange Multipliers is unknown and the dimension N is replaced by the number of segments multiplied by the number of contact nodes.

4.2.4 detectInactiveNodes

Is a function used in Algorithm 1 that detects inactive degrees of freedom in each step of iteration. Inactive DOFs are the ones that are non-penetrating and have non-compressive Lagrange Multipliers. To check if a node is non-penetrating, we have to evaluate its gap function to each segment after applied displacement $\mathbf{u} \neq 0$. Node **R** must not lie within the gray area marked on the Figure 4. Calculation is the same as in **computeGapFunction**, but now three extra conditions are evaluated,

$$g > 0 \,, \tag{26}$$

$$\alpha < 0\,, \tag{27a}$$
$$\alpha \geq 1\,. \tag{27b}$$

In addition to that, the node must have a non-compressive Lagrange Multiplier which is checked by Eq. (28). If any of aforementioned conditions holds, the node is inactive.

$$\lambda \leq 0\,. \tag{28}$$

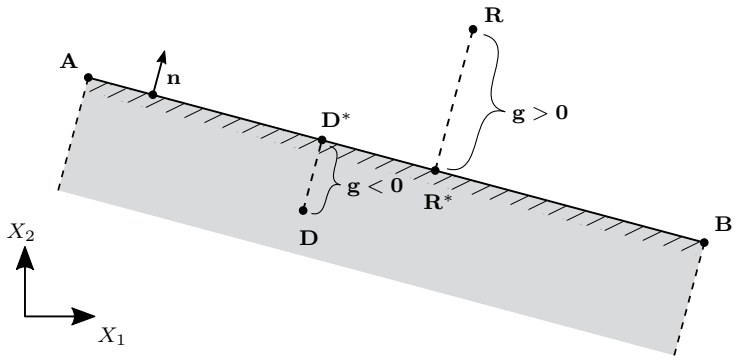

Figure 4: Node penetration area.

4.2.5 detectActiveNodes

Is a function used in Algorithm 2 that detects active degrees of freedom in each step of iteration. The function is almost the same as **detectInactiveNodes** with the only difference being that the conditions in Eq. (28) and Eq. (27) are now reversed and Lagrange Multipliers are not checked. Node must penetrate the rigid boundary therefore all of the following conditions must hold,

$$g < 0\,, \tag{29}$$

$$\alpha > 0\,, \tag{30a}$$
$$\alpha \leq 1\,. \tag{30b}$$

4.3 Equation systems

4.3.1 Expanded system of equations

Correspond to discrete system of equations in terms of a Newton-Raphson formulation in Eq. (14). If used in Algorithm 1 the expanded system of equations is build only once at the beginning with full degrees of freedom of the system,

as we would assume that all contact nodes can come with contact to any segment. Which nodes come into contact is of course not known in advance and it is computed during iterations. If used in Algorithm 2 the system is only build for the active nodes during each iteration step.

Size of the expanded system of equations is know in advance and it is build with dimensions depicted in Figure 5.

i. $\mathbf{K_{exp}}$ is an expanded global stiffness matrix. It consists of tangent stiffness matrix $\mathbf{K}$ and constraint (Lagrange Multiplier) matrix $\mathbf{C}$.

ii. $\mathbf{u_{exp}}$ is an expanded displacement vector. It contains displacements $\mathbf{u}$ for each DOF and Lagrange Multipliers λ for each node to each segment.

iii. $\mathbf{F_{exp}}$ is an expanded force vector. It contains applied forces for each DOF and the initial gap functions g_0 from each node to each segment.

iv. Algorithm 1: **NUM** = number of segments · number of contact nodes

v. Algorithm 2: **NUM** = number of active nodes (Lagrange Multipliers)

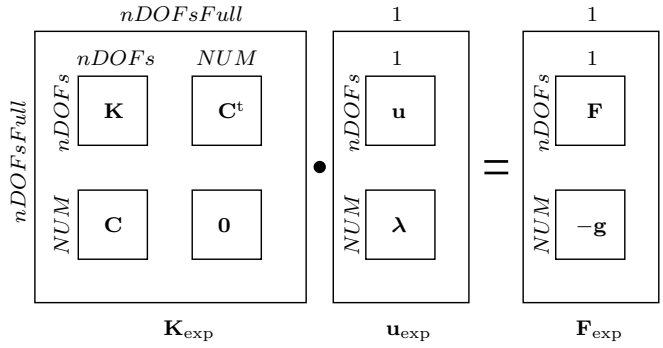

Figure 5: Expanded system of equations.

4.3.2 Reduced system of equations

After finding the inactive nodes in each iteration of Algorithm 1 we can merge them together with homDBC and remove corresponding entries in expanded system. We get the reduced system of equations that we can be solved by,

$$\mathbf{u_{red}} = \mathbf{K_{red}}^{-1}\mathbf{F_{red}} \,, \tag{31}$$

to obtain displacements. From reduced displacement vector the full (expanded) displacement vector is then build. If we on the other hand use Algorithm 2, then we simply solve the expanded system directly and relax it until only valid Lagrange Multipliers remain.

$$\mathbf{u_{exp}} = \mathbf{K_{exp}}^{-1}\mathbf{F_{exp}} \,. \tag{32}$$

4.3.3 Convergence conditions

There are two main convergence conditions that need to be meet before exiting either of aforementioned algorithms. Set of inactive nodes must stay the same as in previous iteration and all Lagrange Multipliers have to be valid. In addition, we specify the maximum number of iterations in case our solution does not converge.