

NTS Contact Smoothing

Ahmed Elgailani - June, 24, 2020

Last update:

July 29, 2020

1 Introduction

The nts contact method we originally implemented has a discontinuity in the normal to the mesh surface as a natural result of the linear discretization we are using (Figure 1). The surface is discretized by the segment of the outer linear triangles of our meshes. This resulted in a convergence problem, especially with FIRE minimization method that we are using. One way to solve it is to use higher-order elements such as triangles with six degrees of freedom, three nodes per side. However, this will result in an unnecessary computational cost as we are only interested in the outer surface elements. Another method is to interpolate surface segments by C^2 polynomials that ensure the continuity of the normal and solve the problem. This could be achieved by multiple schemes, the one we adopt here is three points Hermitian interpolation.

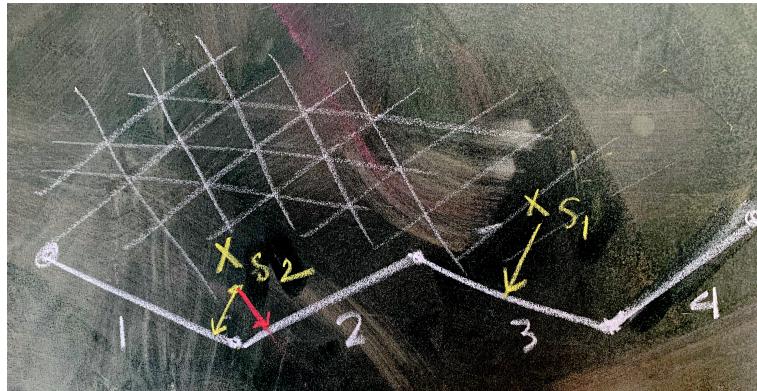


Figure 1: Discontinuity in the normal of linear discretization. If the slave node stays far from the corners, such as slave node X_{s1} , there will be no continuity issue for small deformations relative to the segment size. For nodes near the corners, such as slave node X_{s1} , the normal, which determine the direction of the penalty repulsion, will jump in its direction causing large residual forces.

2 Hermitian Interpolation

Consider the neighboring surface segments defined by nodes X_2 , X_3 , and X_4 (save X_1 for the slave node later) in figure 2. Without getting into the details, the goal is to find a polynomial $\mathbf{X}(\xi)$ that has slopes $\frac{d\mathbf{X}}{d\xi}|_{\xi=-1} = \mathbf{X}_3 - \mathbf{X}_2$, and $\frac{d\mathbf{X}}{d\xi}|_{\xi=1} = \mathbf{X}_4 - \mathbf{X}_3$. The answer is taken directly from Abaqus theory manual ¹ to be:

$$\mathbf{X} = \frac{\alpha}{4} (\xi^2 - 2\xi + 1) \mathbf{X}_2 + \frac{1}{4} (1 - 2\alpha(\xi^2 + 1)) \mathbf{X}_3 + \frac{\alpha}{4} (\xi^2 + 2\xi + 1) \mathbf{X}_4 \quad (1)$$

¹<http://ivt-abausdoc.ivt.ntnu.no:2080/v6.14/books/stm/default.htm> equation (5.1.2-32)

Where α is a user parameter that determines the location of the “slopes merging points” as follows:

$$\mathbf{X}_a = \mathbf{X}(\xi = -1) = \mathbf{X}_3 + \alpha(\mathbf{X}_2 - \mathbf{X}_3) \quad (2)$$

And

$$\mathbf{X}_b = \mathbf{X}(\xi = 1) = \mathbf{X}_3 + \alpha(\mathbf{X}_4 - \mathbf{X}_3) \quad (3)$$

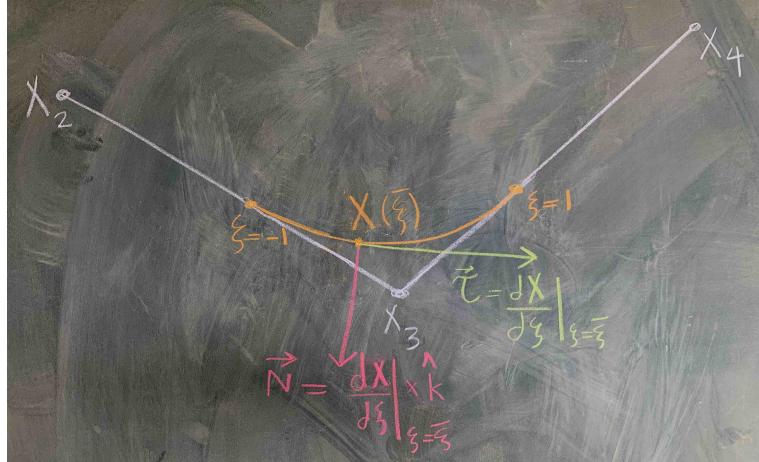


Figure 2: Hermittian three nodes interpolation.

The tangential vector to the interpolation curve is $\tau = \frac{d\mathbf{X}}{d\xi}$. And the outward normal is given by the cross product of τ with the unit vector $\hat{\mathbf{k}}$ orthogonal to the paper pointing toward your face! :

$$\mathbf{N} = \tau \times \hat{\mathbf{k}} \quad (4)$$

Notice that the way we defined the normal \mathbf{N} makes it **always** pointing in the outward direction of the master mesh interior, **regardless** of the convexity of the interpolation at any point (compare this to the valid definition of the normal as $\frac{d\tau}{d\xi}$ which points toward the center of curvature which is inward for nonconvex parts of the surface, requiring additional treatment of its direction).

3 Determining the point of closest approach (PCA)

A key point of contact computation is to find the closest approach of a potential “contactor” slave node to a master mesh (particle). An outward penalty repulsion will then be applied if the signed gap g between the slave node \mathbf{X}_1 and the PCA on the master mesh surface $\bar{\mathbf{X}}$ is negative. Finding $\bar{\mathbf{X}}$ is the hardest part. Consider figure 3. A slave node could be in a position where its PCA, $\bar{\mathbf{X}}$, lies in the portion of the segment that is not included in the interpolation. Example of this are slave nodes \mathbf{X}_6 , \mathbf{X}_7 , \mathbf{X}_9 , and \mathbf{X}_{11} . Or it could be in a position where its normal lies in the interpolation, such as slave nodes \mathbf{X}_1 , \mathbf{X}_5 , \mathbf{X}_8 , and \mathbf{X}_{10} . A robust algorithm is required to consider all the possibilities.

3.1 Contact global searching algorithm

The searching algorithm has two parts, global and local. In the global search, the code divides the simulation region into smaller ones called buckets or Verlet cells. Then it associates all surface nodes with the cell that contains them. To locate segments, the convention used in the code is to associate segments with the cells of their left nodes. So,

the summary of the global algorithm is to find all master nodes that live in the same Verlet stencil of a slave node. To find the closest approach by the local searching.

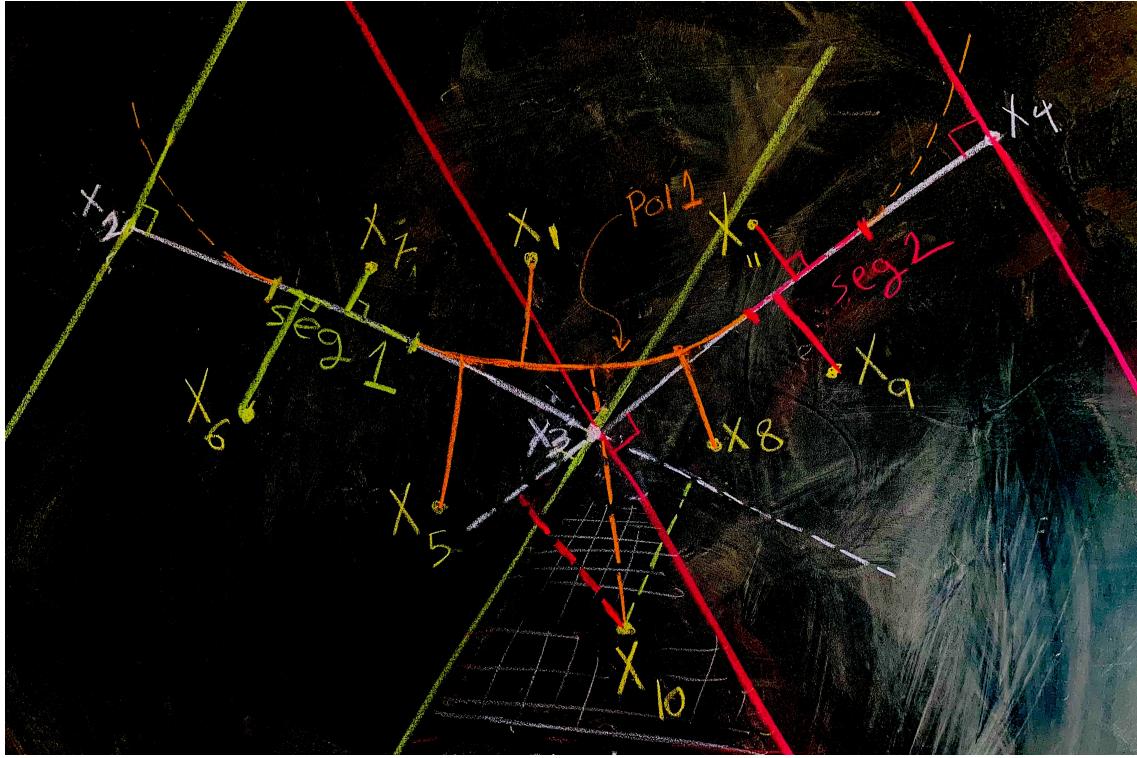


Figure 3: Point of closest approach (PCA) cases.

3.2 Contact local searching algorithm (finding \bar{X})

For all the nodes in a given stencil, we loop over all associated segments and find the normal projection point of the slave node on the segment line (meaning we consider the segment extention in both directions). Consider the segment in figure 4, let s be a convective parameter on the segment such that $s = 0$ at X_0 , $s = 1$ at X_1 , $s < 0$ for point $\mathbf{X} \in (-\infty, X_0)$, and $s > 1$ for $\mathbf{X} \in (X_1, \infty)$ where $(-\infty, X_0)$, (X_1, ∞) are the left and right extsions of the segment, respectively. Now consider the porjectios of nodes X_2 and X_4 on the line containing the segment. They both are on the actual segment, hence they can be written as $\bar{\mathbf{X}} = \mathbf{X}_0 + s(\mathbf{X}_1 - \mathbf{X}_0)$ where $0 \leq s \leq 1$. On the other hand, the projection points of nodes X_3 and X_5 lie on the right and left extensions where $s \geq 1$ and $s \leq 0$, respectively. Also notice that the signed gap g is given by:

$$g = (\mathbf{X}_i - \bar{\mathbf{X}}) \cdot \hat{\mathbf{n}} \quad (5)$$

Where i is the i -th slave node.

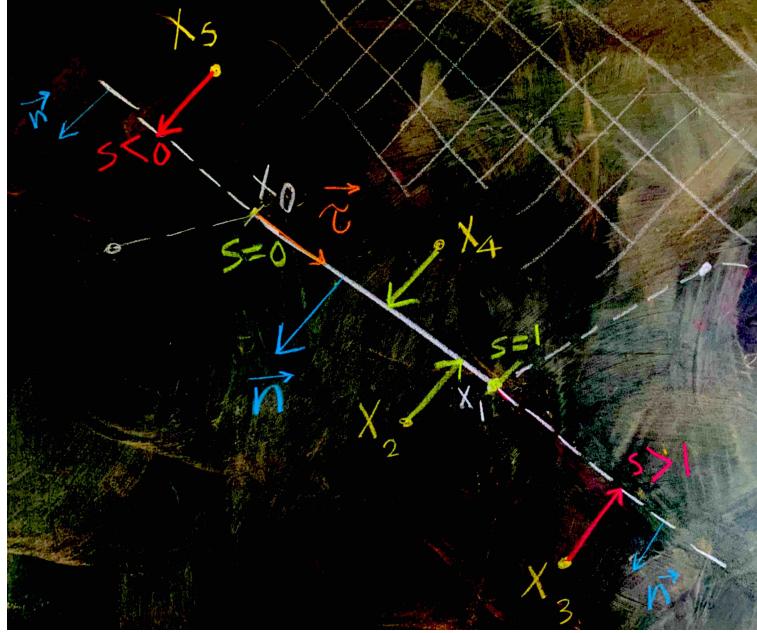


Figure 4: Projection of a slave node on a surface segment line.

We notice that $g \leq 0$ for nodes X_5 and X_4 , and $g \geq 0$ for X_2 and X_3 . The necessary condition for a segment to be an active master segment (forget about interpolations now) is that $g < 0$ and $0 \leq s \leq 1$. It is not sufficient, however, because it also must be the one with the minimum $|g|$ in the stencil. For $s < 0$ and $s > 1$, the PCA to a segment would be nodes X_0 and X_1 , respectively (again, this is before interpolating the corners). So the interaction will be a pair repulsion.

Now going back to figure 3. Because parts of the segments are "replaced" by a polynomial, a slave node projection $\bar{\mathbf{X}}$ will have two cases:

$$\bar{\mathbf{X}} \begin{cases} \in \text{segment} & \text{if } \alpha < s < (1 - \alpha) \\ \notin \text{segment} & \text{if } s < \alpha \text{ or } s > (1 - \alpha) \end{cases} \quad (6)$$

$$(7)$$

For example, X_6 , X_7 , X_9 , and X_{11} in figure 3 all have a normal within the non interpolated parts of a segment. In this case, the algorithm will not look for the normal on the nearby interpolations as indicated by (6). For the second case, the algorithm will search for the PCA within one of the nearby polynomials. To avoid duplicity in searching, as each segment is associated with two polynomials, we constructed the code to look only in the right polynomial to the segment that satisfied the condition in (7).

3.2.1 Finding the PCA to a three points Hermitian polynomial

To find PCA to our smoothed curves, we need to find $\xi = \bar{\xi}$ that minimizes the unsigned gap value:

$$|g| = |\mathbf{X}_i - \mathbf{X}(\xi)| \rightarrow MIN \quad (8)$$

or, equivalently

$$g^2 = (\mathbf{X}_i - \mathbf{X}(\xi)) \cdot (\mathbf{X}_i - \mathbf{X}(\xi)) \rightarrow MIN \quad (9)$$

This requires that

$$\partial_\xi(g^2)|_{\xi=\bar{\xi}} = 0 \quad (10)$$

This leads to

$$(\mathbf{X}_i - \mathbf{X}(\xi)) \cdot \partial_\xi \mathbf{X}(\xi) = 0 \quad (11)$$

But $\partial_\xi \mathbf{X}(\xi)$ is the tangent vector τ to the polynomial $\mathbf{X}(\xi)$. So we are looking for a point on the polynomial at which $(\mathbf{X}_i - \mathbf{X}(\xi)) \perp \tau$. We solve equation 11 by Newton method in our code. But notice that there are points where the solution does not exist! This leads the newton method to go in an infinite loop because it fails to satisfy the solution within the required tolerance (I used $\text{Tol} = 1E-12$ and it usually converges within ~ 7 steps).

To understand this, consider figure 5. Suppose that the algorithm is looking for the PCA on segment 2. Clearly for node X_5 the PCA $\in (\mathbf{X}_3 - \mathbf{X}_2)$ because it satisfies the condition of (6). But what about nodes X_6 and X_7 ? They both satisfy the condition (7). We can see that for node X_6 there exists a solution PCA on polynomial 2 and can be obtained by solving equation (11). But X_7 has no solution on polynomial 2 and Newton method will go on an infinite loop trying to solve (11) for node X_7 . Examining the figure again, notice that any slave node within the region between the two red lines has a solution PCA on polynomial 2. So the additional condition for a slave node with $s_1 > 1$ to have a solution on polynomial 2 is to have $s_2 \leq \alpha$, where s_1 and s_2 is the normalized parametrization of segment 2 and segment 3, respectively. Also, notice that this condition is valid for both the convex and the nonconvex sides of the surface.



Figure 5: Solution region for PCA on a polynomial.

So the summary of the local PCA searching algorithm is as follows:

- Loop over all surface nodes X_i on each Verlet cell (as slave nodes).
- For each slave node X_i in a given Verlet cell, loop over all surface nodes in the current Verlet stencil that lives in a different mesh than X_i (we are not allowing for self contact because it is unexpected in our current study) and then check for PCA in the associated **right** segments (as potential master segments).
- Calculate s_1 value. $s_1 = \frac{(\mathbf{X}_1 - \mathbf{X}_0)}{|\mathbf{X}_1 - \mathbf{X}_0|} \cdot (\mathbf{X}_i - \mathbf{X}_0)$. (Remember that X_0 is the left node of the associated segment on a right handed walk around the surface).
- If $\alpha \leq s_1 \leq (1 - \alpha)$ then $\bar{\mathbf{X}} = \mathbf{X}_0 + s_1(\mathbf{X}_1 - \mathbf{X}_0)$. Continue.
- Else if $s_1 < (\alpha)$ then find s_2 value for the **left** associated segment in a similar manner. If $s_2 \geq (1 - \alpha)$ then find $\bar{\mathbf{X}}$ by solving equation (11) using Newton iterative method.

- If the magnitude of the gap $|g| = |\mathbf{X}_i - \bar{\mathbf{X}}|$ is less than the previously stored gap between \mathbf{X}_i and the current master mesh, then update the gap to the current one. Else, continue.
- Continue.

4 Contact nodal forces

The contact forces are derived from the potential

$$\phi = \sum_k \frac{1}{2} \epsilon g_k^2 \quad (12)$$

Where g_k is the magnitude of the k-th interference (negative gap). ϵ is the penalty stiffness.

The contact force on the i-th surface node is given by

$$\mathbf{f}_i = -\frac{\partial \phi}{\partial \mathbf{X}_i} = \sum_k -\epsilon g_k \frac{\partial g_k}{\partial \mathbf{X}_i} \quad (13)$$

To make our life easier, let's define g another way to get rid of $\hat{\mathbf{n}}$ in equation (5)

$$g = [(\mathbf{X}_i - \mathbf{X}(\xi = \bar{\xi})) \cdot (\mathbf{X}_i - \mathbf{X}(\xi = \bar{\xi}))]^{\frac{1}{2}} \quad (14)$$

Then we have

$$\frac{\partial g_k}{\partial \mathbf{X}_i} = \frac{1}{g_k} (\mathbf{X}_j - \mathbf{X}) \cdot \left(\delta_{ji} - \frac{\partial \mathbf{X}}{\partial \mathbf{X}_i} \right) = -\left(\delta_{ji} - \frac{\partial \mathbf{X}}{\partial \mathbf{X}_i} \right) \cdot \hat{\mathbf{n}} \quad (15)$$

Where we used that fact that $(\mathbf{X}_i - \mathbf{X}(\xi))$ is normal to the tangent vector of the interpolation curve at $\xi = \bar{\xi}$. Using equations (1) and (15), equation (13) can be written explicitly for a given interaction between a slave node X_1 and the three nodes X_2, X_3, X_4 of a master polynomial (figure 3) as:

$$\begin{aligned} \mathbf{f}_1 &= \epsilon g \hat{\mathbf{n}} \\ \mathbf{f}_2 &= -\epsilon g \frac{\alpha}{4} (\xi^2 - 2\xi + 1) \hat{\mathbf{n}} \\ \mathbf{f}_3 &= -\epsilon g \frac{1}{4} (1 - 2\alpha(\xi^2 + 1)) \hat{\mathbf{n}} \\ \mathbf{f}_4 &= -\epsilon g \frac{\alpha}{4} (\xi^2 + 2\xi + 1) \hat{\mathbf{n}} \end{aligned} \quad (16)$$

5 Contact contribution to stress

Suppose the total energy Π of a system is given by

$$\Pi = \sum_I \Phi_I + \sum_k \phi_k \quad (17)$$

Where Φ_I is the monolithic material free energy of the I-th element and ϕ_k is the free energy of the k-th contact interaction. We define the stress as

$$\sigma_{\alpha\beta} = \frac{\partial \Pi}{\partial \varepsilon_{\alpha\beta}} = \sum_{I\mu\nu} \frac{\partial \Phi_I}{\partial F_{I\mu\nu}} \frac{\partial F_{I\mu\nu}}{\partial \varepsilon_{\alpha\beta}} + \sum_{ki\mu} \frac{\partial \phi_k}{\partial g_k} \frac{\partial g_k}{\partial X_{i\mu}} \frac{X_{i\mu}}{\varepsilon_{\alpha\beta}} \quad (18)$$

Where ε is the strain tensor, \mathbf{F} is the deformation gradient tensor, and \mathbf{X}_i is the position vector of the i-th surface node in the **current** configuration.

Now lets work out the last term explicitly for polynomial-slavenode contact for ε_{xx} with the aid of equations (1), (13), (15) and (16):

$$\sum_{ki\mu} \frac{\partial \phi_k}{\partial g_k} \frac{\partial g_k}{\partial X_{i\mu}} \frac{X_{i\mu}}{\varepsilon_{xx}} = -\epsilon \sum_{ki\mu\gamma} g_k \left(\delta_{ij} \delta_{\gamma\mu} - \frac{\partial X_\gamma}{\partial X_{i\mu}} \delta_{\gamma\mu} \right) X_{i\mu} \delta_{\mu x} \hat{n}_\gamma = -\epsilon \sum_{ki} g_k \left(\delta_{ij} - \frac{\partial X_x}{\partial X_{ix}} \right) X_{ix} \hat{n}_x \quad (19)$$

And similarly for ε_{yy}

$$\sum_{ki\mu} \frac{\partial \phi_k}{\partial g_k} \frac{\partial g_k}{\partial X_{i\mu}} \frac{X_{i\mu}}{\varepsilon_{yy}} = -\epsilon \sum_{ki} g_k \left(\delta_{ij} - \frac{\partial X_y}{\partial X_{iy}} \right) X_{iy} \hat{n}_y \quad (20)$$

Where j is the index of the slave node associated with the k -th contact, and $\hat{\mathbf{n}}$ is the unit outward normal to the master polynomial at point \mathbf{X} . The minus sign is needed because for active contacts (when there is interference) the vector $(\mathbf{X}_i - \mathbf{X})$ points in the opposite direction of the outward normal. Let's call the expressions on (19) and (20) KirkWoodXX and KirkWoodYY, respectively, and let's write them down for the four nodes associated with a given contact k (again, consider the one in figure 3 with X_1 being the slave node). Notice that $\text{KirkWoodXX}=\text{KirkWoodYY}=0$ if the node i is not one of the four nodes involved in the contact.

$$\begin{aligned} \text{KirkWoodXX} &= -f_{1x}X_{1x} - f_{2x}X_{2x} - f_{3x}X_{3x} - f_{4x}X_{4x} \\ \text{KirkWoodYY} &= -f_{1y}X_{1y} - f_{2y}X_{2y} - f_{3y}X_{3y} - f_{4y}X_{4y} \end{aligned} \quad (21)$$