EPFL

SEMESTER PROJECT

Internodes

Fabio Matti

supervised by
Dr Guillaume Anciaux
Raquel Dantas Batista

December 20, 2022

1 Introduction

2 The INTERNODES method for contact mechanics

2.1 Radial basis interpolation

Interpolant of $g : \mathbb{R}^d \to \mathbb{R}$ at interpolation nodes $\xi_1, \dots, \xi_M$ with radius parameter r

$$\Pi(\mathbf{x}) = \sum_{m=1}^{M} g_m \phi(\|\mathbf{x} - \xi_m\|, r) \tag{2.1}$$

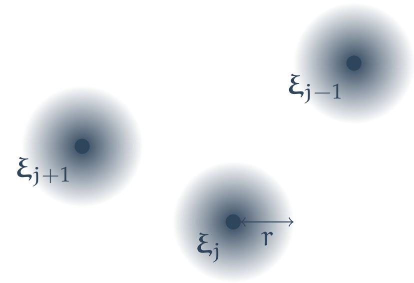


FIGURE 2.1 – Radial basis functions

Wendland C^2 radial basis function

$$\phi(\delta) = (1-\delta)_+^4 (1+4\delta) \tag{2.2}$$

with $\delta = \|\mathbf{x} - \xi_m\| / r$ are well suited.

Denoting $\mathbf{g}_\zeta = (g(\zeta_1), \dots, g(\zeta_N))^T$ and $\mathbf{g}_\xi = (g(\xi_1), \dots, g(\xi_M))^T$ we can write

$$\mathbf{g}_\zeta = \mathbf{D}_{NN}^{-1} \mathbf{\Phi}_{NM} \mathbf{\Phi}_{MM}^{-1} \mathbf{g}_\xi \tag{2.3}$$

with the radial basis matrices

$$(\mathbf{\Phi}_{MM})_{ij} = \phi(\|\xi_i - \xi_j\|, r_j) \quad i, j \in \{1, \dots, M\} \tag{2.4}$$

$$(\mathbf{\Phi}_{NM})_{ij} = \phi(\|\zeta_i - \xi_j\|, r_j) \quad i \in \{1, \dots, N\},\ j \in \{1, \dots, M\} \tag{2.5}$$

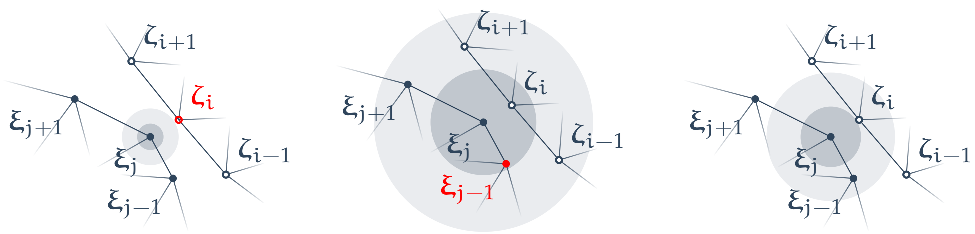
Deparis et. al. [?] proposed two modifications:

- Localized radius parameters for each node r_j, $j \in \{1, \dots, M\}$
- Rescaling with $\mathbf{D}_{NN}^{-1}$ to obtain exact interpolation of constant functions

2.2 Radius parameters

Conditions [?]:

$$
\begin{aligned}
&\text{Limited number of supported interpolation nodes:} \\
&\forall i : \#\{j \neq i : \left\|\xi_i - \xi_j\right\| < r_i\} < 1/\phi(c) && \text{(2.6a)} \\
&\text{Interpolation nodes not be too deep in another support:} \\
&\exists c \in (0,1), \forall i \neq j : \ \left\|\xi_i - \xi_j\right\| \geqslant c r_j && \text{(2.6b)} \\
&\text{All reference nodes in support of interpolation node:} \\
&\exists C \in (c,1), \forall i, \exists j : \ \left\|\zeta_i - \xi_j\right\| \leqslant C r_j && \text{(2.6c)}
\end{aligned}
$$



(A) Too small (B) Too big (C) Just right

FIGURE 2.2 – Finding radius parameters

Distance matrix between nodes $\{\xi_1, \xi_2, \dots\}$ and $\{\zeta_1, \zeta_2, \dots\}$ is defined as

$$\mathbf{D}^{\xi,\zeta}(i,j) = \left\|\xi_i - \zeta_j\right\| \tag{2.7}$$

Algorithm 1 Computation of radius parameters

Require: Positions of interpolation nodes $\{\xi_1, \xi_2, \dots\}$
Require: Radial basis function $\phi : \mathbb{R} \to \mathbb{R}_{\geqslant 0}$
Require: Constant $c \in (0, C)$

```
Compute distance matrix D^{ξ,ξ} defined in (2.7)
For each node i, compute distance to closest neighbor d_i = min_{j≠i} D^{ξ,ξ}(i,j)
while c ⩽ C do
    For each node i, let r_i ← d_i/c                          ▷ Condition (2.6b)
    For each node i, count n_i = #{j ≠ i : D^{ξ,ξ}(i,j) < r_i}
    if For all nodes i, n_i < 1/φ(c) then                     ▷ Condition (2.6a)
        break
    end if
    Increase c
end while
return Radius parameters {r_1, r_2, ...}
```

This algorithm has three main benefits over the previous implementation:

- For uniform meshes (constant mesh size) all radial basis parameters will be the exact same
- The computation of the distance matrix is done outside the `while`-loops
- The algorithm will find interface nodes for a wider class of examples

---

Algorithm 2 Search for interpolation nodes

Require: Positions of primary nodes $\{\xi_1, \xi_2, \dots\}$
Require: Positions of secondary nodes $\{\zeta_1, \zeta_2, \dots\}$
Require: Radial basis function $\phi : \mathbb{R}^d \to \mathbb{R}_{\geqslant 0}$
Require: Constant $C \in (0, 1)$

```
Let 𝓘 = {1, 2, . . .} and 𝓙 = {1, 2, . . .} denote an index set of active nodes
Compute distance matrix D^{ξ,ζ} defined in (2.7)
while 𝓘 or 𝓙 were modified in the previous iteration do
    Obtain radial basis parameters r_i^ξ, i ∈ 𝓘 and r_j^ζ, j ∈ 𝓙 using Algorithm 1
    Remove isolated nodes i ∈ 𝓘 with min_{j∈𝓙} D^{ξ,ζ}(i, j) ⩾ C r_j^ζ   ▷ Condition (2.6c)
    Remove isolated nodes j ∈ 𝓙 with min_{i∈𝓘} D^{ξ,ζ}(i, j) ⩾ C r_i^ξ   ▷ Condition (2.6c)
end while
return Sets of active nodes 𝓘 and 𝓙 with radius parameters r_i^ξ, i ∈ 𝓘 and r_j^ζ, j ∈ 𝓙
```

2.3 Strong form

The strong form of the problem is formulated for the displacement field $\mathbf{u}$:

$$\begin{cases} -\operatorname{div}(\boldsymbol{\sigma}(\mathbf{u})) = \mathbf{f} & \text{Differential equation (Cauchy stress tensor } \boldsymbol{\sigma}) \\ \mathbf{u} = \mathbf{g} & \text{Dirichlet boundary conditions (displacement field } \mathbf{g}) \\ \boldsymbol{\sigma}(\mathbf{u})\mathbf{n} = \mathbf{t} & \text{Neumann boundary conditions (surface traction } \mathbf{t}) \\ \boldsymbol{\sigma}(\mathbf{u})\mathbf{n} = \boldsymbol{\lambda} & \text{Lagrange multipliers } \boldsymbol{\lambda} \text{ defined along interface } \Gamma \\ \boldsymbol{\lambda} \cdot \mathbf{n} \leqslant 0 & \text{Hertz-Signorini-Moreau condition enforced along interface } \Gamma \end{cases} \tag{2.8}$$

2.4 Weak form

The weak formulation results in a linear system for the displacements $\mathbf{u}$ and Lagrange multipliers $\boldsymbol{\lambda}$ [?]:

$$\underbrace{\begin{bmatrix} \mathbf{K}_{\Omega_1\Omega_1} & \mathbf{K}_{\Omega_1\Gamma_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\Gamma_1\Omega_1} & \mathbf{K}_{\Gamma_1\Gamma_1} & \mathbf{0} & \mathbf{0} & -\mathbf{M}_{\Gamma_1} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{\Omega_2\Omega_2} & \mathbf{K}_{\Omega_2\Gamma_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{\Gamma_2\Omega_2} & \mathbf{K}_{\Gamma_2\Gamma_2} & -\mathbf{M}_{\Gamma_2}\mathbf{R}_{\Gamma_2\Gamma_1} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{R}_{\Gamma_1\Gamma_2} & \mathbf{0} \end{bmatrix}}_{\mathbf{A} \text{ (INTERNODES matrix)}} \underbrace{\begin{bmatrix} \mathbf{u}_{\Omega_1} \\ \mathbf{u}_{\Gamma_1} \\ \mathbf{u}_{\Omega_2} \\ \mathbf{u}_{\Gamma_2} \\ \boldsymbol{\lambda} \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{f}_{\Omega_1} \\ \mathbf{f}_{\Gamma_1} \\ \mathbf{f}_{\Omega_2} \\ \mathbf{f}_{\Gamma_2} \\ \mathbf{d} \end{bmatrix}}_{\mathbf{b}} \tag{2.9}$$

$$\mathbf{M}: \text{ Interface mass matrices} \tag{2.10}$$
$$\mathbf{K}: \text{ Stiffness matrices} \tag{2.11}$$
$$\mathbf{f}: \text{ Body force} \tag{2.12}$$
$$\mathbf{d}: \text{ Nodal gaps in configuration} \tag{2.13}$$

2.5 Contact algorithm

Algorithm 3 Contact algorithm for interrnodes method

Require: Positions of primary nodes $\{\xi_1, \xi_2, \ldots\}$
Require: Positions of secondary nodes $\{\zeta_1, \zeta_2, \ldots\}$
Require: Interface candidate index sets $\mathcal{I}^C$ and $\mathcal{J}^C$

```
while 𝓘^C or 𝓙^C were modified in the previous iteration do
    Determine interface nodes 𝓘 and 𝓙 with radius parameters r_i^ξ, i ∈ 𝓘 and r_j^ζ, j ∈ 𝓙 using Algorithm 2 on the candidate index sets 𝓘^C and 𝓙^C
    Assemble the matrix A and right-hand side b of (2.9)
    Solve (2.9) to obtain displacements u and Lagarnge multipliers λ
    Update the interface candidate nodes 𝓘^C and 𝓙^C with 𝓘 and 𝓙, respectively, where all nodes in tension have been removed and all interpenetrating nodes have been added
end while
```

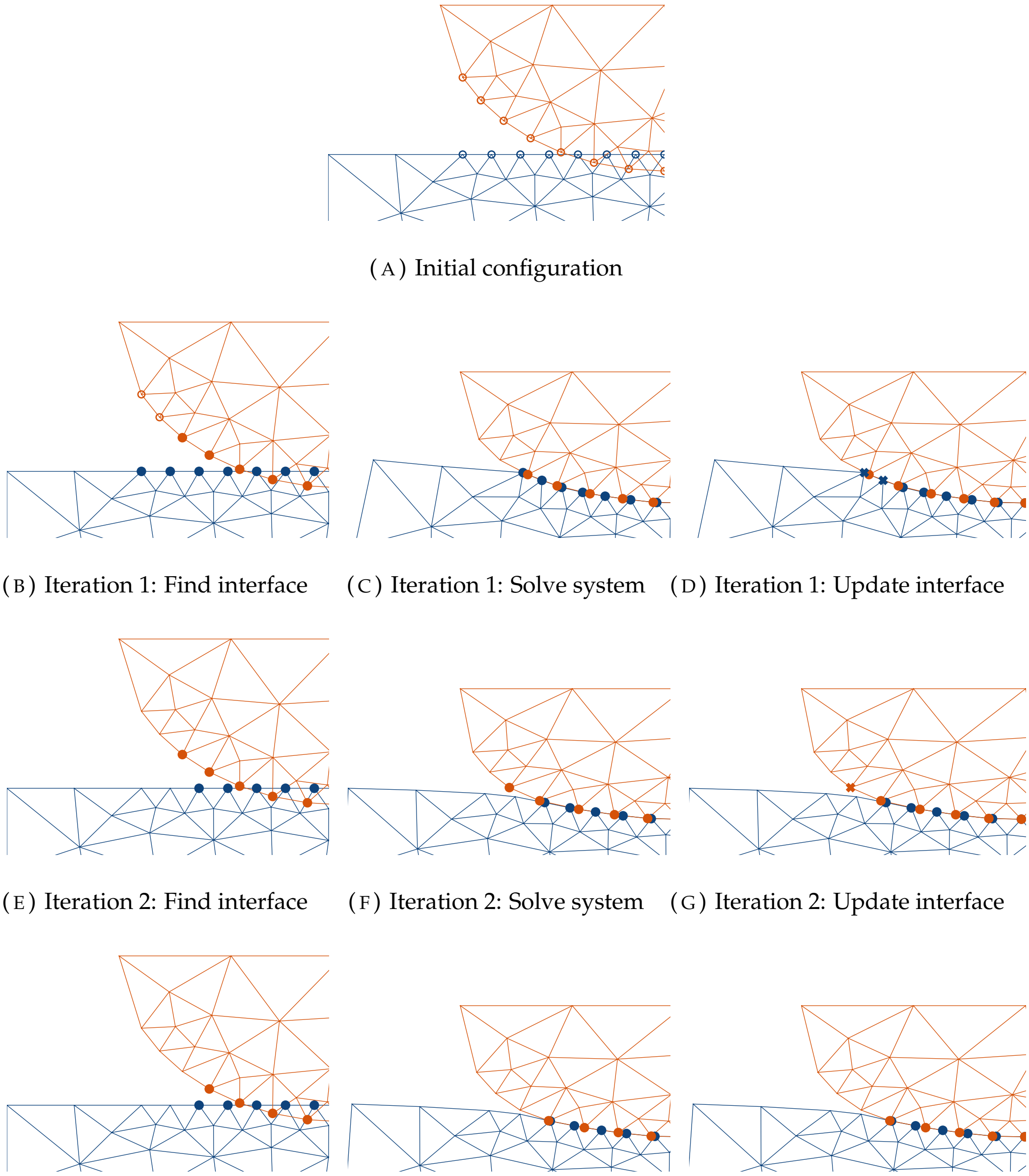
3 Experiments

4 Adjacent work

5 Discussion

Flaws when same nodes that were dumped are added back in (no convergence)

Usually interpenetrating nodes won't be considered part of interface in next iteration



(A) Initial configuration

(B) Iteration 1: Find interface (C) Iteration 1: Solve system (D) Iteration 1: Update interface

(E) Iteration 2: Find interface (F) Iteration 2: Solve system (G) Iteration 2: Update interface

(H) Iteration 3: Find interface (I) Iteration 3: Solve system (J) Iteration 3: Update interface

FIGURE 2.3 – First iteration

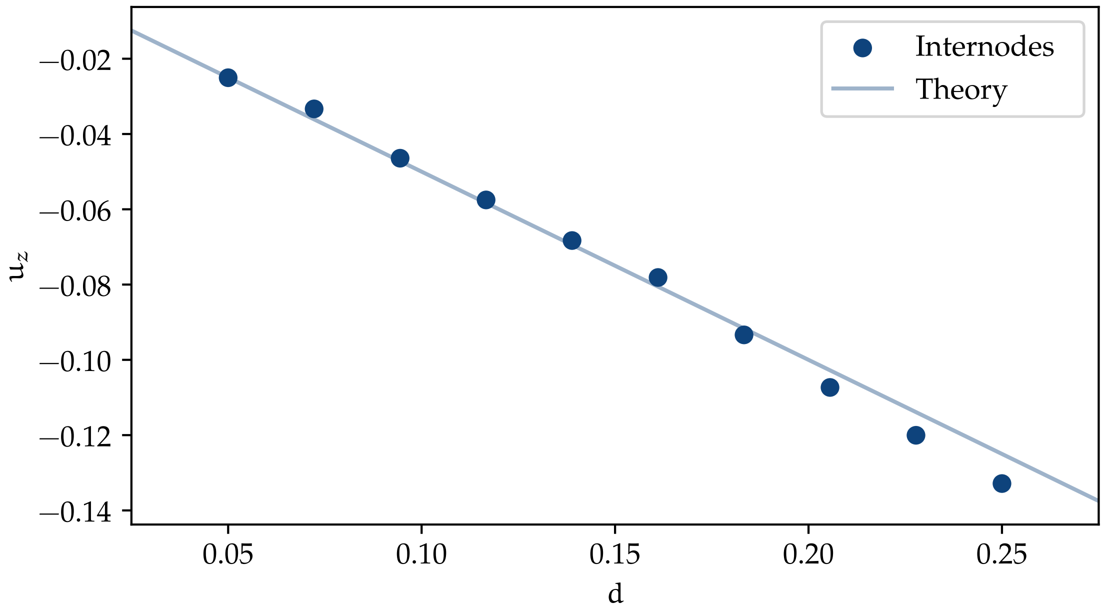


FIGURE 3.1 – Normal displacements

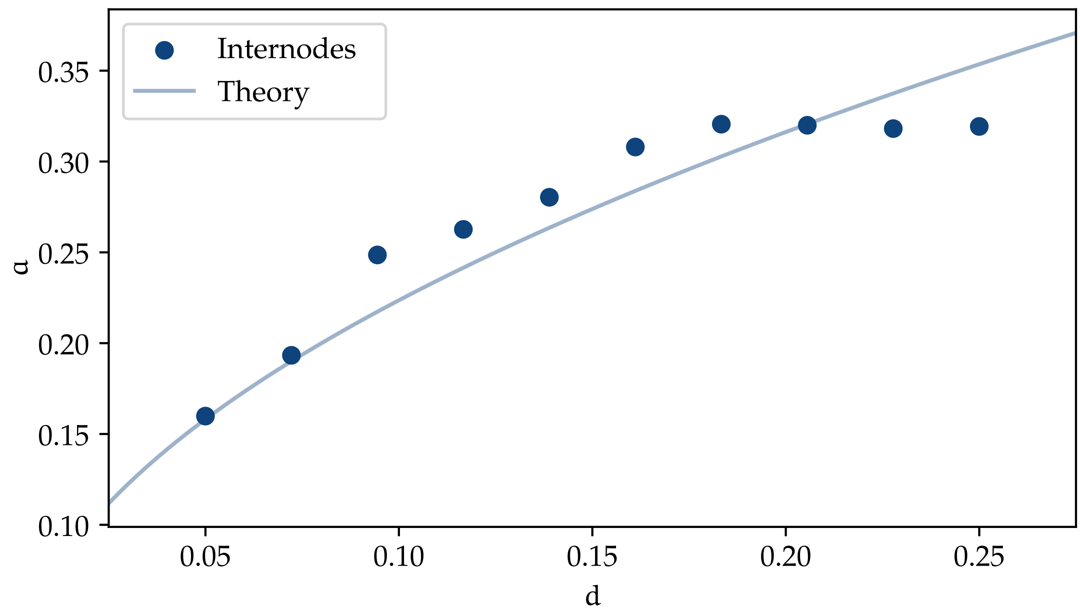


FIGURE 3.2 – Contact radius

(A) Initial class diagram of resolutions

(B) Refactored class diagram of resolutions

FIGURE 4.1 – Combine linear and quadratic penalty resolution by templating the classes with a penalty function.