Nocal

Version 0.1.0

Developed by
John A. Moore, PhD
Caitlin Martinez
Ayushi Chandel
. . .
Support
`john.a.moore@marquette.edu`

1 Introduction

Nocal is python-based nonlocal pre/post processing software for nonlocal analysis of finite element analysis results.

2 Input Deck

Nocal is controlled by a python dictionary where all the user inputs (i.e., *cards*) are stored. This file must be called dictionary.py . Each subsection describes a different card and the data type of each input is shown in italics next to the card

2.1 deckName

String
If the user chooses to build the connectivity matrix from an Abaqus input deck, this is the name of the deck minus the .inp.
Example:
For an abaqus input deck named paraFipMesh_11.inp
`'deckName' :' paraFipMesh_11'`
Module Location

- createAllNlElsetsInRegionParaNConf.py,
- getElemCent.py,
- nonlocalFIP.py,
- nonlocalFIPWeight.py

2.2 dx

Float
The is the length of an edge of square nonlocal volume, for spherical nonconformal volumes Nocal will convert this value into a sphere with an equivalent volume to the cube with edges *dx*
Example:
For $\Delta x = 0.215443469$
`'dx' : 0.215443469` Module Location

- createAllNlElsetsInRegionParaNConf.py,

2.3 numProc

Integer
Number of processors to use when preprocessing nonlocal volumes
Example:
If you wanted to use 4 processors

$'\mathtt{numProc}' : \mathtt{4}$ Module Location

- createAllNlElsetsInRegionParaNConf.py,

2.4 isWeight

boolean
True if you want a weighted average over the nonlocal volume; False if you want a standard mean
The weighted average for a value Γ is given by

$$\Gamma^{\mathrm{nl}}(\boldsymbol{x}) = \frac{1}{A(\boldsymbol{x})} \int_{\Omega} \phi(\boldsymbol{x} - \boldsymbol{y}) \Gamma^{\mathrm{loc}}(\boldsymbol{y}) d\Omega, \tag{1}$$

where Γ^{loc} and Γ^{nl} are the local and nonlocal values of some finite element output Γ respectively. The position where the nonlocal Γ is calculated is $\boldsymbol{x}$ and $\boldsymbol{y}$ is positions around $\boldsymbol{x}$ over which Γ^{loc} is integrated. The nonlocal volume is Ω, and ϕ is a Gaussian distributed weighting function. The normalizing value of A is given by

$$A(\boldsymbol{x}) = \int_{\Omega} \phi(\boldsymbol{x} - \boldsymbol{y}) d\Omega, \tag{2}$$

and

$$\phi(\boldsymbol{x}) = \exp\left(-||\boldsymbol{x}||^2 / l^2\right), \tag{3}$$

where l is the nonlocal length scale.
Example:
if you want a weighted average
$'\mathtt{isWeight}' : \mathtt{True}$ Module Location

- runNonlocalFip.py

2.5 resultsFileName

String
This is the name of the results file where you want to postprocess over the predetermined nonlocal values, you should include the extention (although what extension it is does not matter)
Example:
For a results file named paraFipMesh-FIP_1_11.txt
$'\mathtt{resultsFileName}' :' \mathtt{paraFipMesh - FIP\_1\_11.txt}'$
Module Location

- nonlocalFIP.py,
- nonlocalFIPWeight.py

2.6 L

Float
Length scale l for Equation 3, It is only used if isWeight=True
Example:
You want to weight over a region of $l = 0.13365046175$
`'L' : 0.13365046175`
Module Location

- nonlocalFIPWeight.py

2.7 isNonConf

boolean
Flag to tell if you should use conformal or nonconformal volumes If you want conformal volumes isNonConf=True If you want nonconformal volumes sNonConf=False
Example:
If you want conformal volumes
`'isNonConf' : True`
Module Location

- nonlocalFIP.py

2.8 plotVolumes

boolean
Flag to tell if Nocal should produce plots of specific nonlocal volumes If you want plots (stored in results) plotVolumes=True If you do not want plots plotVolumes=False
Example:
If you want plots
`'plotVolumes : True`
Module Location

- plot.py

2.9 vol2Plot

float or *array*
Only used if plotVolumes=True This tells Nocal what number of volumes you want to plot. Note the plots are stored in the results directory and not actually pushed to the screen
Example:
If you want to plot volume 1, 3 and 50
`'vol2Plot' : [1,3,50]`
If you want to plot only volume 100
`'vol2Plot' : 100`
Module Location

- plot.py

2.10 runOnly

string
This allows you to only run various parts of the preprocessing, plotting, or post processing. For example if you only want to plot some nonlocal volumes you can do this without having to re-determine all the nonlocal volumes and post processes, etc. The options are

- `getNlNodesElem`, will only extract the nodes and elements from the mesh
- `getElemCent`, will only extract the centroids on nodes and elements (which have to be already extracted from the mesh)
- `createVolume`, will only create the nonlocal volumes
- `plotNl`, will only create plots and will not pre/post process anything
- `runNonlocalFip`, will only post process the results

Example:
If you only want to plot and not pre/post process then
`'runOnly' :' plotNl'`
If you want to run the whole code
`'runOnly' :' none'`
Module Location

- main.py

3 Running Nocal

If all finite element input decks and all results files are in ./data (or .\data)
then run from your command line

```
python main.py
```

The results of any postprocessing are stored in ./results (or .\results)