NonLocal-FEM

David Arturo Rodriguez Herrera

Diciembre 2020

Índice

Índice de figuras

Índice de Tablas

Agradecimientos

Resumen

Introducción

Objetivos

Marco Teórico

Metodología

En orden de cumplir con los objetivos propuestos en el capítulo 4, se plantea una implementación de NL-FEM que permita realizar variaciones en los parámetros. Para ello, se parte de la metodología propuesta por Pisano (Pisano et al. 2009) que sigue la ecuación constitutiva planteada en el capítulo 5

Implementación Elasticidad No Local

Para llegar a la implementación se parte de la implementación clásica (local) de FEM. Para efectos de este estudio se asumen esfuerzos planos, por lo que:

$$\sigma_z = 0 \tag{1}$$

Método FEM

Se parte de la metodología planteada en (Reddy, 2005) donde se divide el método en 6 pasos:

1. Enmallado:

   Se divide el dominio en elementos y se enumeran sus nodos con grados de libertad. Para el caso de elasticidad, se deben tener dos grados de libertad por nodo, uno que pertenece al desplazamiento en x y otro que pertenece al desplazamiento en y. El número de nodos por elemento depende del orden de aproximación y de la geometría del elemento. Para este estudio se usarán elementos de cuatro lados con 8 nodos por elemento, comúnmente llamados *Serendipity Elements*.

2. Ecuaciones a nivel de elemento

   Mediante un modelo de elementos finitos (al que se puede llegar con la forma débil de la ecuación diferencial) se realiza el proceso de integración de las matrices de coeficientes, fuerzas nodales y flujos por elemento. Generalmente la integración se realiza usando la cuadratura de Gauss-Legendre.

3. Ensamblaje

   Se obliga a la solución a ser continua en todo el dominio y se asegura el equilibrio de flujos en los nodos. Esto termina en un sistema de ecuaciones el cual se ensambla mediante las matrices y vectores encontrados en el paso anterior.

4. Imposición de condiciones de borde

   Se imponen condiciones de borde operando matricialmente los valores conocidos para la solución o asignando directamente el flujo conocido al sistema de ecuaciones.

5. Solución

   Se soluciona el sistema de ecuaciones encontrando la solución en cada grado de libertad.

6. Post - Proceso

   Se pueden encontrar las soluciones para las variables secundarias (en el caso de elasticidad son las deformaciones en X y Y). El objetivo es poder mostrar gráficas de resultados.

Modificaciones No Locales

Para la inclusión de la no localidad se deben modificar los 3 primeros pasos del método local. Estos cambios se deben a la inclusión de una integral convolutiva en la ecuación constitutiva y a la elección de la función de atenuación.

1. Enmallado

   Uno de los cambios mas relevantes en el proceso de NL-FEM es la inclusión de los elementos no locales por cada elemento local. Para ello, la literatura se propone encontrar los elementos cuyos centroides se encuentren dentro de una distancia de influencia Lr (Zingales et al. 2011), sin embargo, parenciteFernandoayudaporfavor sugiere que el proceso de selección de elementos no locales debe tener en cuenta la distancia adicional que se tenga al punto mas alejado del elemento. En la **Figura (REF)** se muestran gráficamente ambos procesos.

   Dadas las propiedades de las funciones de atenuación descritas en el capítulo 5, se puede asegurar que si se toma una distancia $Lr \to \infty$ el resultado no cambia de manera significante. Esto simplificaría el proceso de enmallado en caso de que no se disponga de un algoritmo que reconozca los elementos no locales. En caso de que se tenga $Lr \to \infty$ el costo computacional aumenta severamente, por lo que se aconseja siempre identificar los elementos no locales.

   Para este estudio se tomara Lr variable, pero nunca menor de $6l$ como se expuso en el Capítulo 5

   En conclusión, para cada elemento n existe un numero M de elementos no locales m. M varía para cada elemento pues depende de su posición en el dominio.

2. Ecuaciones a nivel de elemento

   Como se expuso en el capítulo 5, se debe realizar una integral convolutiva en el proceso de integración. Esto se reduce a que por cada elemento se obtiene una matriz local y un numero específico de matrices no locales.

   La matriz local se obtiene usando la teoría clásica (local) de elasticidad.

   Las matrices no locales se encuentran usando el modelo propuesto por Pisano et al. 2009 donde:

$$\boldsymbol{k_{nm}^{nloc}} = \int_{V_n} \int_{V_m} A(|\boldsymbol{x} - \boldsymbol{x'}|/l) \boldsymbol{B_n^T}(\boldsymbol{x}) \boldsymbol{D} \boldsymbol{B_n}(\boldsymbol{x}) dV' dV \tag{2}$$

   Donde:

   n se refiere al elemento local.

   m se refiere al elemento no local.

   A es la función de atenuación.

   $\boldsymbol{B_n}$ es la matriz de derivadas parciales de las funciones de forma.

   $\boldsymbol{D}$ tensor de rigidez.

   $\boldsymbol{x}$ representa un punto del elemento local n.

   $\boldsymbol{x'}$ representa un punto del elemento no local m.

   En la terminología de Reddy, 2005 se puede reescribir esta ecuación separando los desplazamientos en x y y en submatrices de la matriz del elemento. Como ejemplo se muestra la matriz $\boldsymbol{K_{uu}}$:

$$\boldsymbol{UU_{nm}^{nloc}}[i,j] = \int_{\Omega_n} \int_{\Omega_m} A(\rho) \left[C_{11} \frac{\partial \psi_i^l}{\partial x} \frac{\partial \psi_j^{nl}}{\partial x} + C_{66} \frac{\partial \psi_i^l}{\partial y} \frac{\partial \psi_j^{nl}}{\partial y} \right] d\Omega_m d\Omega_n \tag{3}$$

   Donde:

   ρ es la relación $|\boldsymbol{x} - \boldsymbol{x'}|/l$.

   ψ_i es la i-ésima función de forma.

   C_{11}, C_{66} son los componentes del tensor elástico para esfuerzos planos.

   El super índice ^{l} significa que la función pertenece al elemento local.

   El super índice ^{nl} significa que la función pertenece al elemento no local.

Las funciones estan evaluadas en los puntos correspondientes a su elemento, es decir, la función ψ_i^{nl} esta evaluada en los puntos x_{nl}, y_{nl}

En los anexos (REF) se encuentra el desarrollo de las otras submatrices.

3. Ensamblaje

   Dado que los elementos no locales tienen aportaciones a la matriz general del sistema, el ensablaje de dichos elementos debe tenerse en cuenta usando los grados de libertad de los elementos correspondientes. Por ejemplo, si se evalúa la matriz no local del elemento local 3 con respecto al elemento 4, en la matriz del sistema se debe incluir este aporte el las filas que correspondan a los grados de libertad del elemento 3 con las columnas que representen los grados de libertad del elemento 4. Esta generalización hace posible que se puedan tener combinación de diferentes tipos de elementos (como elementos triangulares y rectangulares).

Estructura del Programa

La implementación computacional propuesta consta de 2 módulos, el módulo procesador y el módulo integrador. La implementación completa se encuentra en GitHUB

1. Módulo Procesador

   El módulo procesador esta escrito en Python y requiere de las librerías *triangle*, *NumPy* y *Matplotilib*. Este módulo se puede encargar de la solución completa de un problema de NL-FEM. Para el enmallado se usa el *wrapper* de *triangle* (Shewchuk, 1996), que lleva el mismo nombre. Fue desarrollado por (REF) y consta de una serie de funciones que realiza triangulaciones Delauney. El enmallado por *triangle* no suele ser uniforme, ya que en el proceso los elementos son refinados para obligar a que los ángulos de los triángulos no sean menores a 35 grados. Adicionalmente Fernando (REF) desarrolló un algoritmo para realizar el proceso de enmallado en dominios rectangulares identificando elementos no locales a una distancia Lr. Esto facilitó el proceso de enmallado para los casos de estudio propuestos.

   Para las ecuaciones a nivel de elemento se usa la ecuación 3 y sus variaciones de los anexos (REF).

   Para la integración se usa la cuadratura de Gauss Legendre. Se puede integrar con cualquier número de puntos. Los puntos se extraen de la librería NumPy.

   Para el ensamblaje de matrices y vectores se usan las propiedades de los arreglos de *NumPy*.

   Para las condiciones de borde se crearon métodos auxiliares que permiten generar las condiciones en los nodos que pertenezcan a segmentos. Los segmentos pueden ser definidos por el usuario o ser generados automáticamente por el programa. Esto se realizó ya que las condiciones de borde generalmente se asignan en los contornos del dominio.

   Las condiciones de borde se toman como listas nativas de Python, lo que permite que se puedan manejar de manera sencilla. Un ejemplo es tener condiciones de borde en dos segmentos distintos. Para poder concatenarlas y que el programa reconzca y aplique ambas condiciones adecuadamente basta con usar el símbolo $+$ entre ambas condiciones de borde y automáticamente el programa será capaz de aplicarlas correctamente. En el caso en el que se aplique una condición de borde natural y esencial al mismo nodo el programa reconocerá la condición de borde esencial.

   Para aplicar las condiciones de borde se hace uso del método matricial que cambia las filas y columnas de los grados de libertad con condición de borde a 0 exceptuando la diagonal. En caso de que se apliquen dos condiciones de borde distintas al mismo nodo, el programa reconocerá la segunda condición de borde como la correcta.

   Para la solución del sistema de ecuaciones se usa la librería NumPy

   Para las gráficas se usa la librería *Matplotlib*. Se pueden graficar las soluciones así como las variables secundarias. Para encontrar perfiles en la solución se creó un algoritmo que extrae el perfil interpolando la solución con las funciones de forma. Dado que las funciones de forma se

evalúan en el dominio natural, se implementó un algoritmo de mapeo inverso usando el método de Newton.

2. Módulo Integrador

   Debido a que Python es un lenguaje interpretado (REF), se conoce con anterioridad que el lenguaje será lento para procesos iterativos complejos. Para el caso de NL-FEM el paso mas costoso computacional mente es la integración de las matrices no locales, por tal razon, se decidió realizar un segundo módulo que se encargue unicamente de realizar el proceso de integración. Para ello se usó C++ que es un lenguaje compilado. En este programa se siguen exactamente los mismos paradigmas que se introdujeron en el módulo procesador con la única diferencia que al ser un programa ejecutable, se deben proporcionar los parámetros por medio de la consola en lugar de modificarlos en el código fuente.

   Gracias al módulo *subprocess* de Python se pudo automatizar el proceso de llamar al programa de C++ desde Python, es decir, el usuario no debe salir de Python para realizar el proceso de integración.

   El módulo integrador usa la cuadratura de Gauss Legendre y obtiene los puntos de una implementación propia adaptada de Rosseta Code

3. Manejo de datos y recurrencia

   Para garantizar la comunicación correcta de los dos módulos se manejan archivos que guardan la información sobre el enmallado y la integración.

   a) Archivo enmallado: Se parte como base del archivo de enmallado del programa de Fernando (REF) que sigue la estructura de la tabla 1. Este archivo es usado por el módulo procesador y el módulo integrador.

   Al ser el archivo principal, lo ideal es generar los archivos de enmallado previamente a realizar el proceso NL-FEM. Por ello, con el enmallado interno de *triangle* se provee una funcionalidad que exporta el archivo de enmallado con el objetivo que el modulo integrador funcione incluso si no se usa un enmallado previamente definido.

   b) Archivos integración: Dado que generar las matrices de los elementos es un proceso computacional mente costoso, el módulo integrador generará una serie de carpetas que guardan la información de las matrices. Para cada elemento del enmallado se guarda un archivo KL_i.csv que contiene la matriz local. Para cada elemento del enmallado se guarda un archivo llamado KNLS.csv donde cada fila del archivo corresponde a la j-ésima matriz no local. Estas matrices se aplanan por columnas y filas por lo que al cargarlas se deben transponer. La estructura del directorio de salida se encuentra en la figura 1

Tabla 1. Estructura del archivo de enmallado

Linea 1	#GDL	#E
GDL_i	X	Y
E_i	GDL_1	GDL_j
ENL_i, j	#ENL	E_j^{nl}

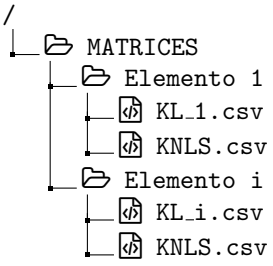

Figura 1. Estructura de guardado de archivos para las matrices

4. Estructura final El programa completo se orientó a objetos y su estructura se muestra en el diagrama UML de la figura (REF)

Casos de Estudio

Para poder evaluar el funcionamiento del programa y poder validar los resultados se parte de los casos de estudio propuestos por Pisano et al. 2009.

1. **Barra a tensión:**

   La barra a tensión se puede modelar como se muestra en la figura 2

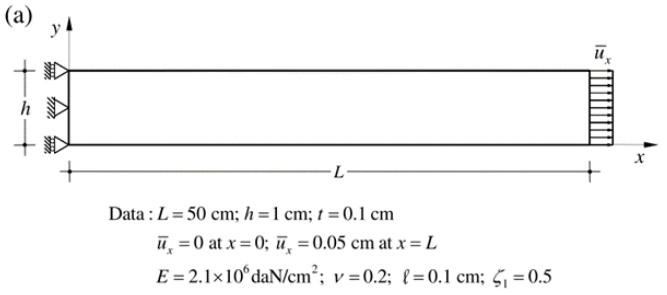

   Figura 2. Modelo de barra unidimensional (FIGURA PARCIAL, SE CAMBIARA POR UNA PROPIA)

   Se tienen las siguientes características:

   $L = 50\ cm$

   $h = 1\ cm$

   $t = 0.1\ cm$

   $u_0 = 0.05\ cm$

   $E = 2.1 * 10^6\ daN/cm^2$

   $\nu = 2.1 * 10^6\ daN/cm^2$

   Bajo las siguientes condiciones de borde:

   $U = 0$ en $x = 0$

   $V = 0$ en el punto $x = 0, y = h/2$

   $U = u_0$ en $x = L$

   Usando el programa, este problema puede modelarse como:

```python
import NLFEM
import numpy as np
import matplotlib.pyplot as plt
from NLFEM.Mesh import Rect
import os
import copy
import subprocess

RUTA_M = 'MATRICES'
E = 2.1*10**6
V = 0.2
u = 0.05
t = 0.1
l = 0.1
NEX = 200
NEY = 4
subprocess.run("index.exe input.txt 3 "+format(E)+" "+format(V)+"
  "+format(t)+" "+format(l)+" "+RUTA_M+" 1", shell=True, check=True)

PATH = os.getcwd()
nombre = "\\input.txt"
GEOMETRIA = Rect.Rect(PATH+nombre,NEX,NEY)
GEOMETRIA.dibujarse()
plt.show()
```

Figura 3. Paso 1. Imports

Se importa la librería NLFEM que contiene el programa de NL-FEM completo. Se importan las librerías *NumPy* y *Matplotlib*. Se definen las variables iniciales del problema, por ejemplo, el módulo de Young, grosor, etc. Se llama al módulo integrador con el comando *subprocess* y se le pasan por parámetro los datos del enmallado. Note que el archivo index.txt contiene los datos del enmallado y sigue los lineamientos propuestos en la tabla 1. Por último, se carga el enmallado como una geometría del programa. En las líneas 22 y 23 se mostrará gráficamente el enmallado, por lo tanto, estas dos últimas líneas son opcionales.

2. **Placa a tensión:**

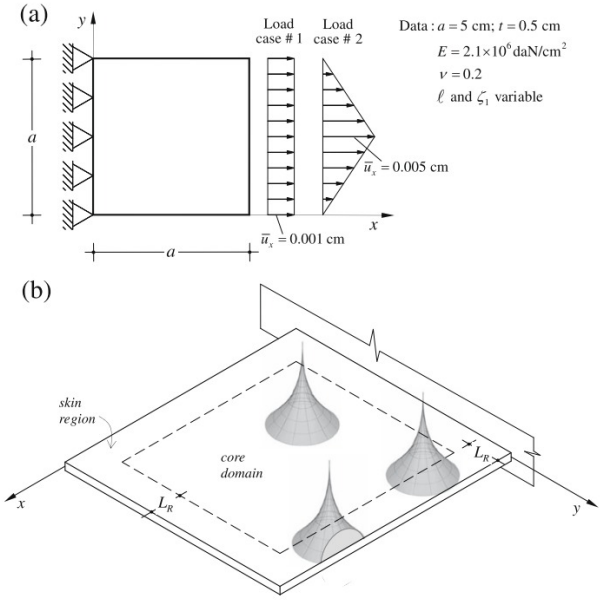

Figura 4. Modelo de placa a tensión (FIGURA PARCIAL, SE CAMBIARA POR UNA PROPIA)

Se tienen las siguientes características:

$a = 5\ cm$

$t = 0.5\ cm$

$u_0 = 0.001\ cm$

$E = 2.1 * 10^6\ daN/cm^2$

$\nu = 2.1 * 10^6\ daN/cm^2$

Bajo las siguientes condiciones de borde:

$U = 0, V = 0$ en $x = 0$

$U = u_0$ en $x = a$

Validación

Variación de Parámetros

Lista de Parámetros

1. Parámetro i

Variaciones Planteadas

1. Variación en funciones de atenuación
2. Z's

Resultados

Conclusiones

Referencias

Pisano, A., Sofi, A. & Fuschi, P. (2009). Nonlocal integral elasticity: 2D finite element based solutions. *International Journal of Solids and Structures*, *46*(21), 3836-3849. https://doi.org/10.1016/j.ijsolstr.2009.07.009

Reddy, J. N. D. (2005). *An Introduction to the Finite Element Method*. McGraw-Hill Education. https://books.google.com.co/books?id=8gqnRwAACAAJ

Shewchuk, J. R. (1996). Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator [From the First ACM Workshop on Applied Computational Geometry]. En M. C. Lin & D. Manocha (Eds.), *Applied Computational Geometry: Towards Geometric Engineering* (pp. 203-222). Springer-Verlag.

Zingales, M., Di Paola, M. & Inzerillo, G. (2011). The finite element method for the mechanically based model of non-local continuum. *International Journal for Numerical Methods in Engineering*, *86*, 1558-1576. https://doi.org/10.1002/nme.3118