

PROGRAMA DE ELEMENTOS FINITOS DE CÓDIGO ABIERTO PARA LA RESOLUCIÓN DE PROBLEMAS MECÁNICOS EN ESTADO PLANO

OPEN SOURCE FINITE ELEMENT PROGRAM FOR THE SOLUTION OF MECHANICAL PROBLEMS IN PLANE STATE

Martín E. Almenar^a, Facundo Isla^{a,b}, Sergio Gutierrez^a y Mariela Luege^{a,b}

^a*Instituto de Estructuras “Arturo M. Guzmán”, FACET, UNT, Av. Independencia 1800, 4000 S.M. de Tucumán, Argentina, malmenar@herrera.unt.edu.ar, <http://www.herrera.unt.edu.ar/iest/>*

^b*CONICET*

Palabras clave: Método de elementos finitos, programa de código abierto, estados bidimensionales de tensiones, programación en Octave

Resumen. Se desarrolla una herramienta computacional de código abierto basada en el método de los elementos finitos (MEF) para la resolución de problemas mecánicos en estado plano. Los ejemplos presentados tienen como objetivo explicar de manera detallada la aplicación de las condiciones de borde e interpretación de los resultados obtenidos en el cálculo; aspectos esenciales para el correcto uso de un software comercial si se tiene en cuenta que el proceso de cálculo conforma de alguna manera una caja negra a la cual no se tiene acceso. El empleo de una herramienta computacional propia y de código abierto brinda además la posibilidad de que el estudiante implemente las ecuaciones que gobiernan el problema de elasticidad plana que han sido debidamente desarrolladas en teoría y ofrece también la posibilidad de modificar el código o tomarlo de base para desarrollos futuros que resulten de su interés.

Keywords: Finite element method, open source program, two-dimensional stress states, programming in Octave

Abstract. An open source computational tool based on the finite element method (FEM) is developed to solve mechanical problems in a plane state. The examples presented are intended to explain in detail the application of boundary conditions and interpretation of the results obtained in the calculation; essential aspects for the correct use of commercial software if one takes into account that the calculation process somehow forms a black box to which there is no access. The use of a proprietary and open source computational tool also provides the possibility for the student to implement the equations that govern the plane elasticity problem that have been duly developed in theory and also offers the possibility of modifying the code or taking it as a basis for future developments that result from your interest.

1. INTRODUCCIÓN

El método de elementos finitos (MEF) es actualmente uno de los métodos numéricos más usados para la resolución de problemas formulados en ecuaciones diferenciales. La física de distintos problemas puede ser descripta mediante el empleo de ecuaciones diferenciales, como los son puntualmente muchos de los problemas que se presentan en la Ingeniería Civil. Hoy en día resulta imposible pensar que un ingeniero civil no utilice el MEF para la resolución de problemas estructurales, hidráulicos, geotécnicos, etc. En la práctica profesional se utilizan generalmente programas comerciales que tienen implementado el MEF como una caja negra, no obstante, aquel ingeniero que posee conocimientos acerca del método posee también un espíritu crítico sobre los resultados obtenidos. En este sentido, a los estudiantes de la carrera de Ingeniería Civil de la Universidad Nacional de Tucumán (UNT), Facultad de Ciencias Exactas y Tecnología (FACET) se les imparte clases sobre la formulación del MEF en el análisis estructural durante su cursado cuatrimestral en la asignatura de Estabilidad IV perteneciente al módulo VII de la carrera.

Los conceptos del MEF que se brindan al alumno se aplican a la resolución de problemas de mecánica del continuo bidimensional, estáticos y con linealidad material y geométrica. La unidad temática que correspondiente al MEF consta de clases teóricas y prácticas. La teoría se desarrolla en base a la bibliografía recopilada de los libros de Oñate (1995) y Zienkiewicz y Taylor (1991). En lo que respecta a las clases prácticas, los alumnos desarrollan en forma grupal 4 trabajos prácticos. Dos de estos trabajos prácticos corresponden a problemas de estados planos de tensiones o deformaciones, mientras que los otros 2 corresponden a problemas de placas. Para el desarrollo de estos 4 trabajos prácticos, los estudiantes utilizan versiones académicas gratuitas de programas que permiten aplicar las diferentes etapas del MEF: introducción de datos, proceso y finalmente obtención de resultados para el análisis de los mismos.

Como parte del preproceso, con estos programas se aprende a generar la geometría requerida, colocar condiciones de borde relacionadas a cargas y vínculos, definir las propiedades del material y finalmente realizar el mallado. Una vez que se realiza la corrida del programa se obtienen los resultados y se analizan los distintos mapas de colores como parte del posproceso. El proceso ejecutado por el programa se realiza totalmente en forma interna. Teniendo en cuenta la formación teórica brindada, el estudiante puede entender de manera global cuales son las operaciones que se efectúan internamente en el programa, pero no puede acceder a ellas. Por esta razón los docentes de la asignatura generaron un código de elementos finitos en el programa Octave para que los alumnos puedan observar cómo se lleva a cabo la etapa del proceso. Dicho código se utiliza en una primera instancia para realizar los trabajos prácticos correspondientes a estados planos. No es el objetivo principal que el estudiante ahonde en nociones de programación, sino que tenga conocimientos claros sobre el método y pueda ser crítico con respecto a los resultados obtenidos. En este sentido la herramienta de programación Octave, además de ser gratuita, posee un lenguaje bastante flexible, con cierto entrenamiento se consiguen distinguir la expresiones propias del MEF.

En este trabajo se expone la práctica correspondiente al MEF y su aplicación a la resolución de problemas en estado plano. Dicha práctica debe ser realizada por el alumno mediante el uso del programa generado por los docentes. Se presenta cómo cargar los datos del problema, acceder a las rutinas que resuelven las ecuaciones y finalmente analizar y expresar los resultados.

2. EJERCICIOS REALIZADOS EN EL TRABAJO PRÁCTICO

Se exponen a continuación los dos trabajos prácticos realizados por los estudiantes y relacionados a la aplicación del MEF a problemas en estado planos. En el primer trabajo práctico (TP) se propone una geometría sencilla para que el alumno se relacione con el programa y no tenga demasiadas dificultades durante el preproceso de cargar los datos (ver figura(1)).

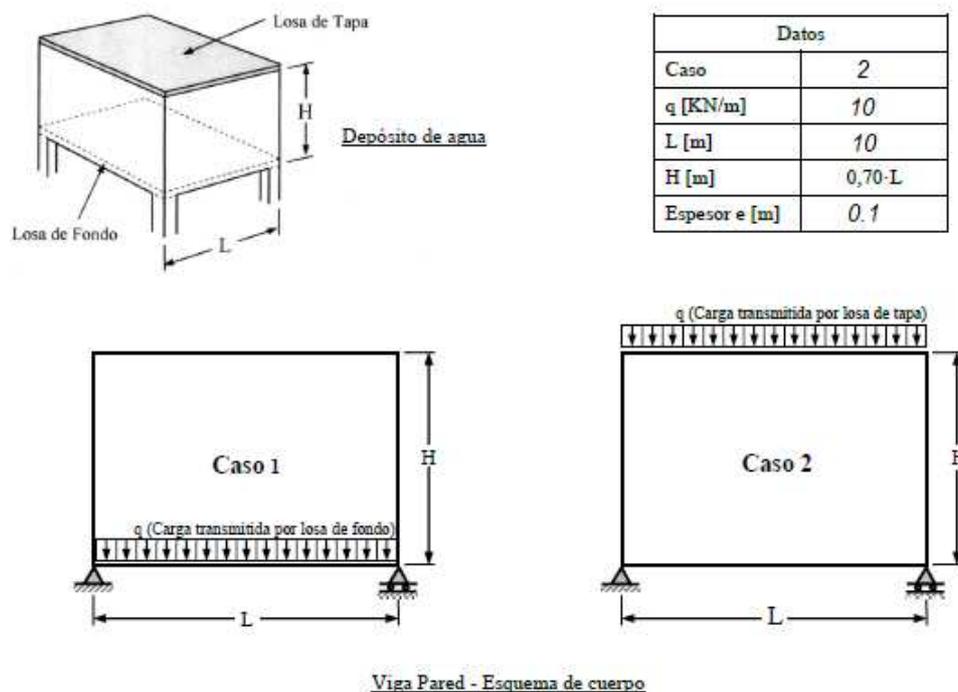


Figura 1: Ejercicio del primer TP de aplicación del MEF a problemas en estado plano.

Se asocia el ejemplo a un problema real de ingeniería, en este caso se plantea la pared de un recipiente que recibe carga a través de su losa de fondo y de su tapa (casos 1 y 2 respectivamente). No se analiza en este TP el efecto producido por la presión hidrostática del agua perpendicular a la viga pared. El estado de tensiones antes descripto hace trabajar a la estructura como una placa. Dicho estado se estudia posteriormente en la asignatura y puede superponerse con el estado de tensiones planas que se analiza en este TP. El estado de tensiones planas se produce en la estructura al actuar la carga de la losa de fondo o de la tapa (casos 1 y 2 respectivamente de la figura (1)). En este primer trabajo práctico se pide que se indiquen cuáles son los pasos a aplicar con el método de elementos finitos agrupándolos en preproceso, proceso y posproceso; que se haga un resumen de las ecuaciones a aplicar y que se indique el orden de las matrices y vectores globales del problema. Luego se debe resolver el problema con el programa brindado. El primer resultado que se sugiere chequear es la deformación de la estructura para corroborar que la misma es compatible con las condiciones de apoyo y carga. Una vez hecho esto se deben presentar los resultados de las tensiones como mapas de colores y como diagramas sobre determinadas líneas de la estructura (ver figura (2)).

En el segundo TP se pide la misma presentación de resultados pero con problemas en general de geometría más compleja (figura (3)) y que permiten aplicar en algunos casos condiciones de simetría. Se solicita densificar la malla en los puntos de concentración de tensión y se puede

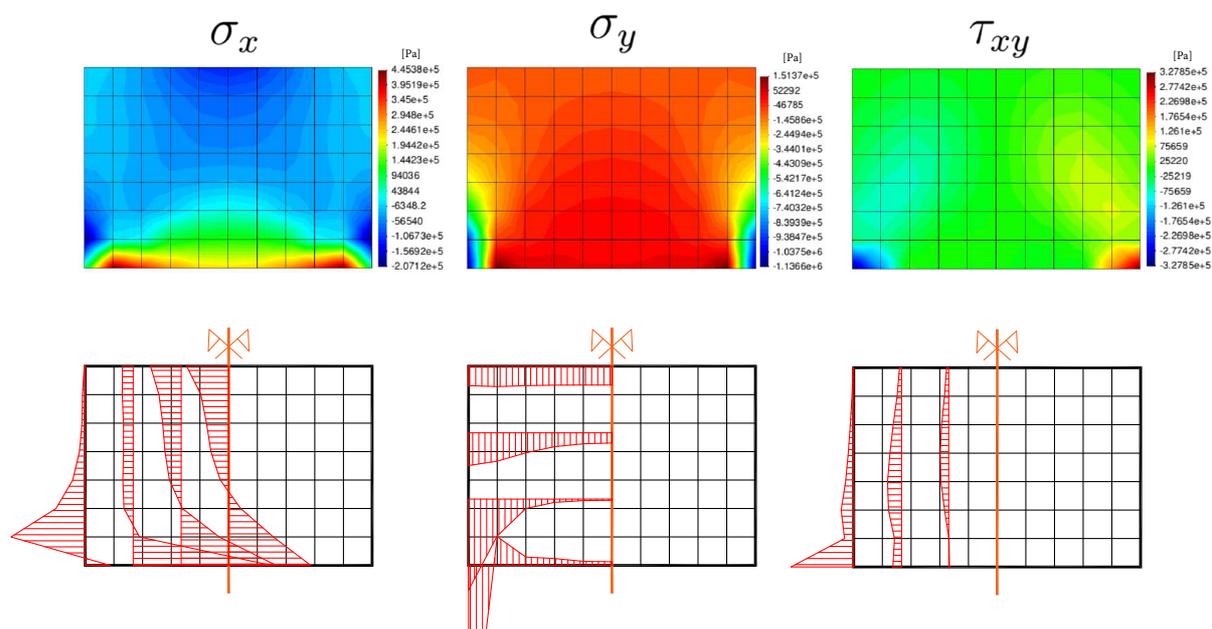


Figura 2: Diagramas de tensiones sobre estructura correspondiente al caso 2.

efectuar la comparación de los resultados obtenidos mediante el MEF con la solución exacta de cada caso que se plantea (ver figura (3)) y que se encuentra publicada en la bibliografía sobre el tema. Al igual que en el primer TP, el primer resultado que se sugiere chequear es la deformación de la estructura para corroborar que la misma es compatible con las condiciones de apoyo y carga. Luego, una vez hecho esto se debe presentar los resultados de la tensiones como mapas de colores y como diagramas sobre determinadas líneas de la estructura. Considerando el Ejercicio 1 correspondiente a la figura (3), se presentan en la figura (4) las geometrías a modelar y mallas tomando uno o dos ejes de simetría. En la figura (5) se presentan los resultados obtenidos en mapas de colores y líneas de interés para el caso de dos ejes de simetría considerando $P = 10\text{KN}$ y $H = 0,60\text{m}$.

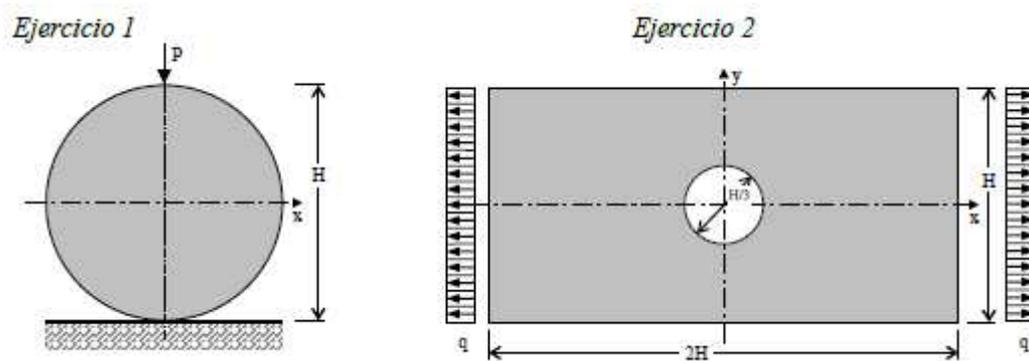


Figura 3: Ejercicios del segundo TP de aplicación del MEF a problemas en estado plano.

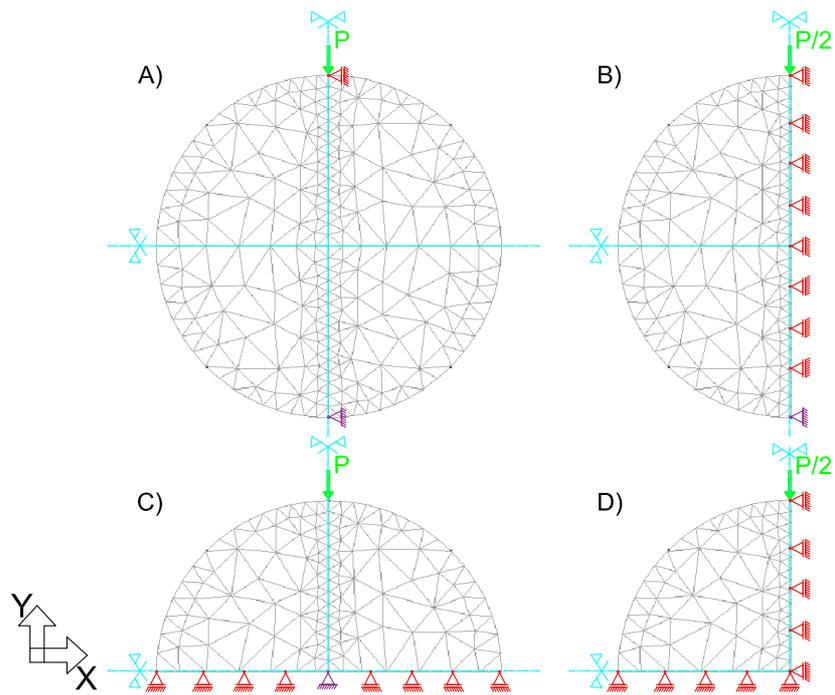


Figura 4: Geometría, apoyos, carga y malla para probeta cilíndrica. A) Sin ejes de simetría. B) Con eje de simetría vertical. C) Con eje de simetría horizontal. D) Con dos ejes de simetría.

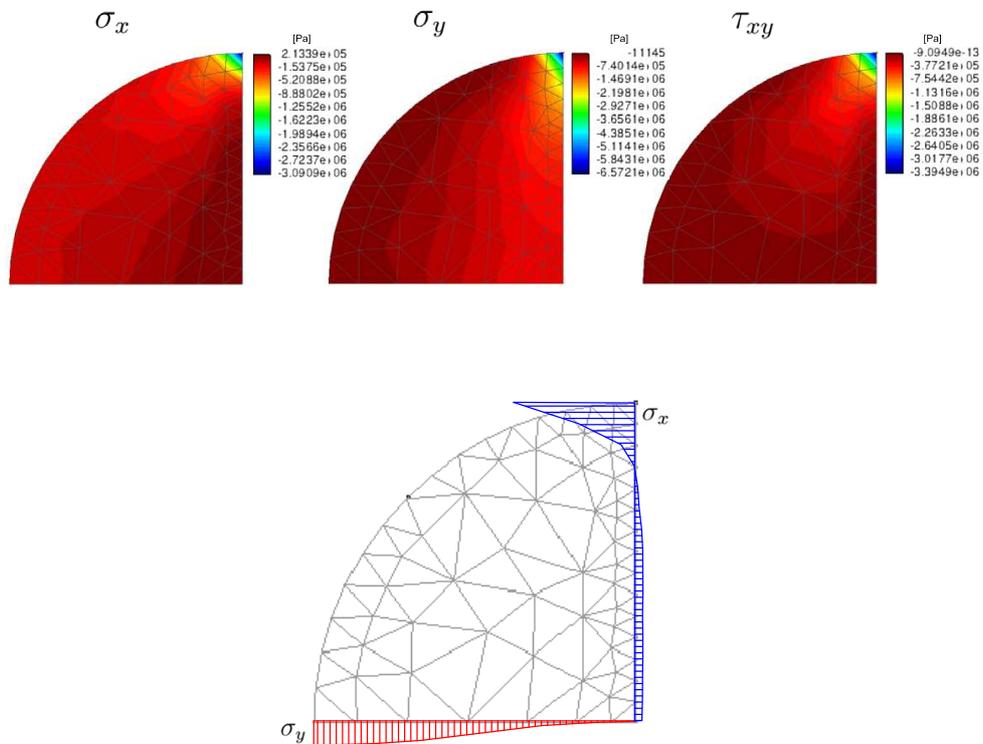


Figura 5: Diagramas de tensiones sobre cilindro.

3. PROGRAMA DE ELEMENTOS FINITOS

Para la resolución de los trabajos prácticos explicados en la sección anterior se brindan al alumno dos herramientas. Por un lado la generación de la geometría y malla se debe realizar con la versión educativa del programa GiD¹ (<https://www.gidhome.com/download/official-versions/>). Una vez generada la malla se debe extraer las coordenadas de nodos, tabla de conectividades y numeración de nodos de vínculos y aplicación de acciones (carga o desplazamiento). Luego, se debe utilizar el programa realizado en GNU Octave², el cual se encuentra disponible por medio del siguiente link ([CLICK AQUI](#)) o solicitando por mail al primer autor del trabajo. La descarga de GNU Octave (<https://www.gnu.org/software/octave/#install>) puede realizarse de la web. El código responde a la nomenclatura de las ecuaciones utilizadas en la teoría de la asignatura. En la codificación se siguieron las recomendaciones de (Rahman y Valdman, 2013)). En este artículo se propone un modo eficiente de programación en el que se eliminan los lazos realizados comúnmente para operaciones asociadas al número de elementos finitos y número de puntos de integración. Aparte de un funcionamiento más eficiente, esto presenta una mejor claridad a la lectura del código lo cual es necesario para estudiantes poco experimentados en programación.

Los datos extraídos del programa GiD deben pegarse en la rutina *entradas*, la cual está debidamente comentada para la introducción de dichos datos. En la rutina *entradas*, se define también el tipo de problema: estado plano de tensiones o de deformaciones; el tipo de elemento finito lineal: triangular o rectangular; la cantidad de puntos de integración y el espesor (en caso de problema de estado plano de tensiones). A continuación se transcriben las partes de la rutina *entradas* donde deben cargarse los datos antes indicados.

```
function [coord, conect, vinc, desp, fuer, n, t_element, ...
pg, fc, mat, C, estado, X, Y]=entradas();
%% números y cantidades importantes
estado.tipo=1; %1ept, 2epd, 3axial simetria
t_element=1;% 1:triángulos de 3 nodos, 2: cuadriláteros de 4 nodos
n.NPG=6;%puntos de integración
%% conectividades(cargar archivo en directorio del programa)
conect.elemento=load('conectividad.txt');
%% coordenadas, espesor y factor de escala
%(cargar archivo en directorio del programa)
coord.espesor=1*ones(n.NPG,n.ne);
flong=1;
coord.xy=load('coordenadas.txt');
%% vinculos
%cargar vectores de vinculo en x e y
vincx=[];
vincy=[];
```

¹ GiD: es un pre y post procesador universal, adaptable y fácil de usar para simulaciones numéricas en ciencia e ingeniería. Ha sido diseñado para cubrir todas las necesidades comunes en el campo de simulaciones numéricas desde el preproceso hasta el postprocesado: modelado geométrico (CAD), generación de malla, definición de datos de análisis, transferencia de datos al software de análisis, operaciones de posprocesamiento, visualización de resultados

² GNU Octave: es un lenguaje de alto nivel, destinado principalmente a cálculos numéricos. Proporciona una interfaz de línea de comando conveniente para resolver problemas lineales y no lineales numéricamente, y para realizar otros experimentos numéricos utilizando un lenguaje que es principalmente compatible con Matlab. También se puede usar como un lenguaje orientado a objetos.

```

%% fuerza
%cargar vectores de fuerza en x e y
fuex=[ ];
fuery=[];
%% Materiales
mat1_elas=[2.6*10^6 0];
%[E ni] se ingresa solo E(modulo elástico)
%y ni(coeficiente de poisson)
end

```

Con todos los datos cargados se corre la rutina *problema* y se obtiene la solución. La solución puede visualizarse con rutinas gráficas que permiten visualizar la estructura deformada, o sin deformar, con algún mapa de color. Los valores del mapa de colores se pueden consultar con las herramientas gráficas del Octave (ver figura (6)). A continuación se transcribe la rutina *problema* la cual se debe hacer correr una vez cargados los datos de entrada para obtener la solución.

```

%% preproceso
[coord,conect,vinc,desp,fuer,n,t_element,...
pg,fc,mat,C,estado,X,Y]=entradas();
%calculo de matriz B y coeficientes de integracion CI
[N B CI]=prop_element(pg,coord,n,t_element,estado);
%% Proceso
[DESP,TENS,DEF]=...
solver_lineal(fc.vincdesp,n,B,C,desp,fuer,X,Y,CI);

```

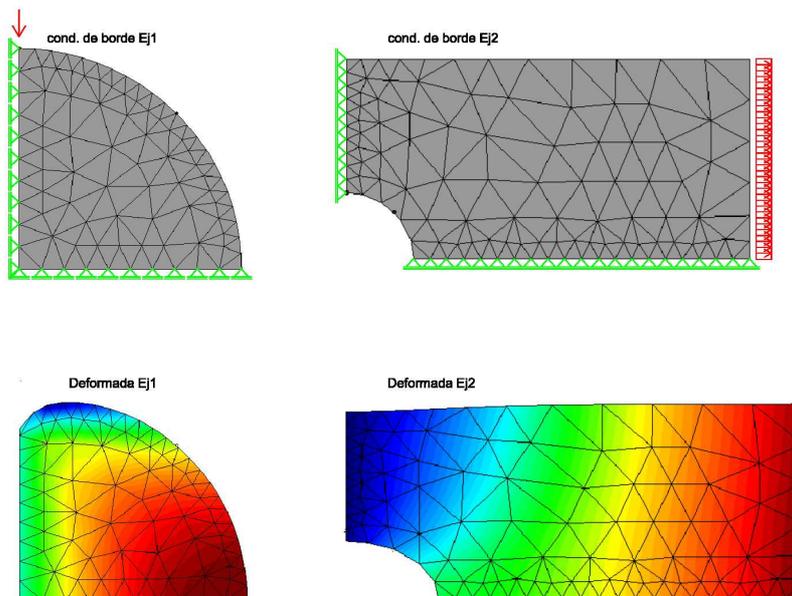


Figura 6: Condiciones de bordes, deformada y mapa de colores de desplazamientos para ejercicios de TP.

En cuanto al proceso de cálculo, existen 3 rutinas fundamentales que operan con los datos de *entradas* y que son importantes para ser inspeccionadas por los alumnos. Por un lado la rutina *Cep1* es la encargada de generar las matrices constitutivas de acuerdo al tipo de estado. La Rutina *prop element* es la encargada de generar las funciones de forma y sus derivadas evaluadas en los puntos de integración, y los coeficientes de integración numérica para el cálculo de las matrices de rigidez de elementos. Por último la rutina *solver lineal* toma todos los datos generados y resuelve el problema como se transcribe a continuación

```
function [U,TENS,def,FFF]=solver_lineal(fcl,n,B,C,U,R,X,Y,CI)
%calculo de desplazamientos
Dpe=vec2Dpe(U,n,X);
def=mmat(B,Dpe);
TENS=mmat(C,def);
fint=mmat(tr(B),TENS).*CI.ru;
kk=mmat3(tr(B),C,B).*CI.kuu;
FINT=sparse(X.resu,Y.resu,fint(:));
KKK=sparse(X.kuu,Y.kuu,kk(:));
FFF=sparse(FINT-R);
U(fcl,1)=U(fcl,1)-KKK(fcl,fcl)\(FFF(fcl));
%actualización de tens, def y reacciones
Dpe=vec2Dpe(U,n,X);
def=mmat(B,Dpe);
TENS=mmat(C,def);
fint=mmat(tr(B),TENS).*CI.ru;
FINT=sparse(X.resu,Y.resu,fint(:));
FFF=sparse(FINT-R);

%-----
%U=desplazamientos globales
%TENS=tensiones por elemento
%def=deformaciones por elemento
%FFF=Vector de Fuerza Global
%fcl=vector de grados de libertad incognita
%n=numero elemento, numero de nodos ...
%B=derivada de funciones de forma
%C=matrices constitutivas
%R=Vector de fuerzas aplicadas
%X,Y=posiciones para ensamblaje
%CI=coeficientes de integración
end
```

Nótese que esta rutina carece de lazos de elementos y puntos de integración como se indicó antes.

En el siguiente canal de Youtube se encuentra un video tutorial donde se explica el modo del empleo del programa mediante el desarrollo de un ejemplo ([CLICK AQUI](#)).

4. COMENTARIOS FINALES Y TRABAJOS FUTUROS

Se describió la práctica realizada para la resolución de problemas en estado plano con el MEF para los estudiantes de Ingeniería Civil de la FACET - UNT. Dicha práctica está orientada a fijar nociones básicas del funcionamiento del MEF y de la interpretación de sus resultados. La herramienta numérica, generada en el programa Octave, con la que los alumnos realizan la práctica brinda no sólo la posibilidad de asumir las tareas como un simple usuario, sino también de verificar el código, intervenirlo, y quizá en un futuro tomarlo de base para otros desarrollos. La idea de los docentes es expandir el código en el corto plazo para incluir la resolución de placas y posteriormente elementos de barras para que el programa pueda ser utilizado en otras asignaturas del área de la teoría de las estructuras y dimensionado.

REFERENCIAS

- Oñate E. *Cálculo de Estructuras por el método de elementos finitos*. CIMNE, Barcelona, 1995.
- Rahman T. y Valdman J. Fast matlab assembly of fem matrices in 2d and 3d: Nodal elements. *Applied Mathematics and Computation*, 219(13):7151 – 7158, 2013.
- Zienkiewicz O. y Taylor R. *The finite element method*, volumen II. McGraw Hill, 1991.