

UCUENCA

Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería Civil

Implementación y validación de un programa basado en el método de elementos finitos para la solución de problemas cuasi-estáticos en cuerpos bidimensionales

Trabajo de titulación previo a la obtención del título de Ingeniero Civil

Autores:

Pedro Miguel Asmal Rodas

Leonel Octavio Yumbra Cadme

Director:

Juan Fernando Zalamea León

ORCID:  0000-0003-4756-9387

Co-director:

Esteban Patricio Samaniego Alvarado

ORCID:  0000-0002-8728-491X

Cuenca, Ecuador

2023-06-05

Resumen

El método de elementos finitos (MEF) es una herramienta utilizada para la discretización y análisis de medios continuos. No obstante, muchos de los programas basados en este método requieren licencias costosas y no proporcionan acceso al código fuente. Por tanto, este trabajo de titulación se enfoca en implementar y validar un programa basado en el método de elementos finitos para resolver problemas en cuerpos bidimensionales bajo condiciones cuasi-estáticas. Para lograr este objetivo, se adaptó y modificó un software existente llamado FSIPY a través de procesos de depuración y programación, en base a la teoría de la elasticidad plana. Posteriormente, se verificó que el comportamiento de los resultados del software modificado sea similar en términos de desplazamientos y esfuerzos mediante benchmarking con SAP2000 y ABAQUS. La modificación del software FSIPY demostró ser compatible con el análisis cuasi-estático y mostró convergencia al aumentar el número de elementos, donde se evidenció diferencias relativas en el orden del 1% en términos de desplazamientos. Este estudio validó con éxito un programa de código abierto y de acceso gratuito, lo que lo convierte en una herramienta útil para el desarrollo de nuevos módulos aplicables en diversas áreas de la ingeniería como Ingeniería Estructural, Geotecnia, Hidráulica, entre otras.

Palabras clave: método de elementos finitos, benchmarking, elasticidad plana, FSIPY

Abstract

The finite element method (FEM) is a tool used for the discretization and analysis of continuous media. However, many of the programs based on this method require expensive licenses and do not provide access to the source code. Therefore, this degree work focuses on implementing and validating a program based on the finite element method to solve two-dimensional body problems under quasi-static conditions. To achieve this goal, an existing software called FSIPY was adapted and modified through debugging and programming processes, based on the plane elasticity theory. Subsequently, the behavior of the results of the modified software was verified to be similar in terms of displacements and stresses by benchmarking with SAP2000 and ABAQUS. The FSIPY software modification proved to be compatible with the quasi-static analysis and showed convergence when increasing number of elements, where relative differences in the order of 1% in terms of displacements were evidenced. This study successfully validated an open-source and free access program, making it a useful tool for the development of new modules applicable in various engineering areas such as Structural Engineering, Geotechnics, Hydraulics, among others.

Keywords: finite element method, benchmarking, plane elasticity, FSIPY

Índice de contenido

Resumen	2
Abstract	3
Índice de contenido	4
Índice de figuras	8
Índice de tablas	11
Dedicatoria	13
Agradecimientos	14
Simbología y Nomenclatura	15
Capítulos	17
1 Introducción	17
1.1 Antecedentes y justificación	17
1.2 Objetivos	18
1.2.1 Objetivo general	18
1.2.2 Objetivos específicos	18
1.3 Alcance	18
1.4 Organización de capítulos	19
2 Marco Teórico	20
2.1 Teoría de la Elasticidad lineal	20
2.1.1 Definición	20
2.1.2 Matriz constitutiva	21
2.1.3 Planteamiento del problema elástico lineal cuasi-estático	22
2.1.3.1 Ecuaciones de gobierno	23
2.1.3.2 Condiciones de contorno	23
2.1.4 Resolución del problema elástico lineal cuasi-estático	24
2.2 Elasticidad plana	24
2.2.1 Tensión plana	24
2.2.2 Deformación plana	25
2.2.3 Campo de desplazamientos	26
2.2.4 Campo de deformaciones	26
2.2.5 Campo de tensiones	26
2.2.6 Relación entre tensión-deformación	27

2.3	Formulación fuerte y débil de las ecuaciones	27
2.3.1	Formulación fuerte	27
2.3.2	Formulación débil	28
2.3.2.1	Principio de la trabajos virtuales	28
2.4	Método de los Elementos Finitos	29
2.4.1	Introducción al método de los elementos finitos	29
2.4.2	Proceso de análisis del MEF	29
2.4.3	Formulación del modelo numérico	31
2.4.4	Matriz de rigidez y vectores de fuerza	34
2.5	Elementos isoparamétricos de cuatro nodos	35
2.5.1	Funciones de forma	35
2.5.2	Formulación Isoparamétrica	37
2.6	Extrapolación de los esfuerzos	37
2.7	Métodos de integración numérica	39
2.7.1	Cuadratura de Gauss	39
2.7.2	Coordenadas y pesos de la cuadratura de Gauss	40
2.7.3	Integración numérica de la matriz de rigidez	41
2.7.4	Integración numérica del vector de fuerzas equivalentes	41
3	Metodología	42
3.1	Programas utilizados	42
3.1.1	Lenguajes de programación	42
3.1.2	Preprocesamiento	43
3.1.3	Procesamiento	43
3.1.4	Posprocesamiento	44
3.2	Estado inicial del software FSIPY	44
3.2.1	Preprocesamiento	45
3.2.1.1	Entrada de datos	45
3.2.1.2	Lectura de datos	48
3.2.1.3	Inicialización de variables	49
3.2.2	Cálculo	49
3.2.3	Posprocesamiento	50
3.3	Identificación de las modificaciones y requerimientos	50

3.3.1	Ingreso de variables	50
3.3.2	Transición a un caso cuasi-estático	51
3.3.3	Implementaciones requeridas	52
3.4	Diseño de la solución	52
3.4.1	Implementación de la matriz de rigidez.....	53
3.4.2	Implementación de fuerzas superficiales.....	55
3.4.3	Extrapolación de esfuerzos	59
3.4.4	Exportación de esfuerzos a ParaView	61
3.5	Validación.....	61
3.5.1	Validación con la solución teórica	61
3.5.2	Validación con programas comerciales.....	62
4	Resultados y discusión	63
4.1	Caso 1: Solución analítica en tensión plana	63
4.1.1	Información de entrada	64
4.1.2	Generación de la malla	65
4.1.3	Condiciones de contorno.....	66
4.1.4	Validación con la solución analítica	66
4.2	Caso 2: Placa en tensión plana.....	70
4.2.1	Información de entrada	70
4.2.2	Generación de la malla	71
4.2.3	Condiciones de contorno.....	71
4.2.4	Convergencia de los resultados	72
4.2.5	Comparación	76
4.3	Caso 3: Muro de gravedad	80
4.3.1	Información de entrada	80
4.3.2	Generación de la malla	81
4.3.3	Condiciones de contorno.....	82
4.3.4	Convergencia de resultados	82
4.3.5	Comparación	85
5	Conclusiones.....	89

Referencias	91
Anexos	93
A Desplazamientos	93
A.1 Caso 1: Desplazamientos en el nodo 3.....	93
A.2 Caso 2: Desplazamientos en los nodos 2 y 3.....	94
A.3 Caso 3: Desplazamientos en los nodos 3 y 4.....	95
B Esfuerzos	96
B.1 Caso 2: Esfuerzos Software FSIPY	96
B.2 Caso 2: Esfuerzos Software ABAQUS.....	97
B.3 Caso 2: Esfuerzos Software SAP2000	98
B.4 Caso 3: Esfuerzos Software FSIPY	99
B.5 Caso 3: Esfuerzos Software ABAQUS.....	100
B.6 Caso 3: Esfuerzos Software SAP2000	101
C Manual de usuario	102
C.1 Ejecución detallada del caso 2.....	102
D Manual de instalación	108
D.1 Comandos para la instalación.....	108
D.2 Componentes extras	109
D.3 Instalación de ParaView	110
D.4 Compilación y ejecución	110

Índice de figuras

Figura 2.1	Estado tensional de un elemento diferencial (Coronel, 2006)	22
Figura 2.2	Ejemplos de tensión plana (Oñate, 2009)	25
Figura 2.3	Ejemplos de deformación plana (Oñate, 2009)	26
Figura 2.4	Proceso de análisis de elemento finito (Bathe, 1996)	30
Figura 2.5	Fuerzas actuantes sobre un triángulo de tres nodos (Oñate, 2009)	33
Figura 2.6	Funciones de forma de un elemento rectangular (Fish & Belytschko, 2007)	35
Figura 2.7	Elemento rectangular. Sistema de coordenadas naturales y cartesianas (Oñate, 2009)	36
Figura 2.8	Extrapolación local para Coordenadas de Gauss 2x2 (Oñate, 2009)	38
Figura 2.9	Cuadratura de Gauss-Legendre sobre elementos cuadriláteros: a) 1x1 b) 2x2 c) 3x3 y d) 4x4 puntos de integración (Coronel, 2006)	40
Figura 3.1	Etiquetas en un elemento cuadrilátero	47
Figura 3.2	Método de Newton-Raphson (Belytschko <i>et al.</i> , 2014)	51
Figura 3.3	Flujograma condicional para teoría de elasticidad	53
Figura 3.4	Ejemplo de elemento con fuerzas superficiales	56
Figura 3.5	Relación lineal entre dos puntos (Oñate, 2009)	60
Figura 4.1	Caso 1: Placa delgada de acero en tensión plana.	63
Figura 4.2	a) Etiquetas de nodos y bordes, b) Malla para nueve elementos. Caso 1	65
Figura 4.3	Variación del coeficiente de Poisson	67
Figura 4.4	Desplazamientos obtenidos en FSIPY para el caso 1	68
Figura 4.5	Desplazamientos en X vs Carga Aplicada	68
Figura 4.6	Desplazamientos en Y vs Carga Aplicada	69
Figura 4.7	Esfuerzos normales en la dirección X para el caso 1	69
Figura 4.8	Caso 2: Placa delgada de acero en tensión plana	70
Figura 4.9	a) Etiquetas de nodos y bordes, b) Malla para nueve elementos. Caso 2	71

Figura 4.10	Desplazamientos en la dirección X	73
Figura 4.11	Desplazamientos en la dirección Y	73
Figura 4.12	Desplazamientos obtenidos en FSIPY para el caso 2	73
Figura 4.13	Esfuerzos normales en los extremos libres	74
Figura 4.14	Esfuerzos normales en los extremos empotrados	74
Figura 4.15	Esfuerzos normales en la dirección X y Y para el caso 2	75
Figura 4.16	Esfuerzos tangenciales para el caso 2	75
Figura 4.17	Desplazamiento en X del nodo 2	76
Figura 4.18	Desplazamiento en Y del nodo 2	77
Figura 4.19	Desplazamiento en X del nodo 3	78
Figura 4.20	Desplazamiento en Y del nodo 3	78
Figura 4.21	Esfuerzos normales en X del nodo 3.....	78
Figura 4.22	Esfuerzos normales en Y del nodo 3	79
Figura 4.23	Esfuerzos tangenciales del nodo 3.....	79
Figura 4.24	Caso 3: Muro de gravedad.....	80
Figura 4.25	Etiquetas de nodos y bordes para el caso 3	81
Figura 4.26	Desplazamientos en la dirección X	83
Figura 4.27	Desplazamientos en la dirección Y	83
Figura 4.28	Desplazamientos obtenidos en FSIPY para el caso 3	84
Figura 4.29	Esfuerzos normales en la dirección X y Y para el caso 3	84
Figura 4.30	Esfuerzos tangenciales para el caso 3	85
Figura 4.31	Desplazamientos en X del nodo 3.....	86
Figura 4.32	Desplazamientos en Y del nodo 3	86
Figura 4.33	Desplazamientos en X del nodo 4	87
Figura 4.34	Desplazamientos en Y del nodo 4	87
Figura 4.35	Esfuerzos normales en la dirección X del nodo 3	88
Figura 4.36	Esfuerzos normales en la dirección Y del nodo 3.....	88
Figura 4.37	Esfuerzos tangenciales del nodo 3.....	88
Figura C.1	Caso 2.....	102
Figura C.2	Archivos de entrada de datos.....	102
Figura C.3	Archivo caso2.geo	103
Figura C.4	Opciones de generación de malla en GMSH	103

Figura C.5	Malla de nueve elementos GMSH.....	104
Figura C.6	Archivo caso2.conf.in	104
Figura C.7	Archivo caso2.boun.in	105
Figura C.8	Terminal de Ubuntu	106
Figura C.9	Variables de salida	106
Figura C.10	Interfaz ParaView.....	107
Figura C.11	Interfaz ParaView - Filtro "Warp By Vector"	107

Índice de tablas

Tabla 2.1	Elemento rectangular de cuatro nodos de Lagrange (Oñate, 2009)	36
Tabla 2.2	Coordenadas y pesos de la cuadratura de Gauss (Oñate, 2009)	40
Tabla 3.1	Parámetros físicos	45
Tabla 3.2	Parámetros numéricos	46
Tabla 3.3	Sources	46
Tabla 3.4	Outputs	46
Tabla 3.5	Condiciones de contorno	47
Tabla 3.6	Configuración para el mallado	48
Tabla 3.7	Configuración elemento línea	48
Tabla 3.8	Configuración de las fuerzas superficiales de la Figura 3.4	56
Tabla 3.9	Configuración de las condiciones de contorno de la Figura 3.4	56
Tabla 4.1	Rango de valores de las variables a analizar	64
Tabla 4.2	Parámetros físicos para el caso 1	64
Tabla 4.3	Parámetros numéricos para el caso 1	65
Tabla 4.4	Sources para el caso 1	65
Tabla 4.5	Condiciones de contorno para el caso 1	66
Tabla 4.6	Fuerzas superficiales para el caso 1	66
Tabla 4.7	Desplazamientos para el caso 1	67
Tabla 4.8	Esfuerzos en la dirección X obtenidos por FSIPY	69
Tabla 4.9	Parámetros físicos para el caso 2	70
Tabla 4.10	Parámetros numéricos para el caso 2	71
Tabla 4.11	Sources para el caso 2	71
Tabla 4.12	Condiciones de contorno para el caso 2	72
Tabla 4.13	Fuerzas superficiales para el caso 2	72
Tabla 4.14	Comparación desplazamientos Nodo 2 para el caso 2	76
Tabla 4.15	Comparación desplazamientos Nodo 3 para el caso 2	77
Tabla 4.16	Parámetros físicos para el caso 3	80
Tabla 4.17	Parámetros numéricos para el caso 3	81
Tabla 4.18	Sources para el caso 3	81
Tabla 4.19	Condiciones de contorno para el caso 3	82
Tabla 4.20	Fuerzas superficiales para el caso 3	82

Tabla 4.21	Comparación desplazamientos Nodo 3 para el caso 3	85
Tabla 4.22	Comparación desplazamientos Nodo 4 para el caso 3	86
Tabla A.1	Desplazamientos en X para el caso 1 (FSIPY)	93
Tabla A.2	Desplazamientos en Y para el caso 1 (FSIPY).....	93
Tabla A.3	Desplazamientos obtenidos al variar la carga aplicada para el caso 1 (FSIPY)	93
Tabla A.4	Desplazamientos en los nodos 2 y 3 para el caso 2 (FSIPY)	94
Tabla A.5	Desplazamientos en los nodos 2 y 3 para el caso 2 (ABAQUS) ..	94
Tabla A.6	Desplazamientos en los nodos 2 y 3 para el caso 2 (SAP2000) ..	94
Tabla A.7	Desplazamientos en los nodos 3 y 4 para el caso 3 (FSIPY)	95
Tabla A.8	Desplazamientos en los nodos 3 y 4 para el caso 3 (ABAQUS) ..	95
Tabla A.9	Desplazamientos en los nodos 3 y 4 para el caso 3 (SAP2000) ..	95
Tabla B.1	Esfuerzos normales en la dirección X para el caso 2 (FSIPY)	96
Tabla B.2	Esfuerzos normales en la dirección Y para el caso 2 (FSIPY)	96
Tabla B.3	Esfuerzos tangenciales para el caso 2 (FSIPY)	96
Tabla B.4	Esfuerzos normales en la dirección X para el caso 2 (ABAQUS) .	97
Tabla B.5	Esfuerzos normales en la dirección Y para el caso 2 (ABAQUS) .	97
Tabla B.6	Esfuerzos tangenciales para el caso 2 (ABAQUS)	97
Tabla B.7	Esfuerzos normales en la dirección X para el caso 2 (SAP2000)	98
Tabla B.8	Esfuerzos normales en la dirección Y para el caso 2 (SAP2000) .	98
Tabla B.9	Esfuerzos tangenciales para el caso 2 (SAP2000)	98
Tabla B.10	Esfuerzos normales en la dirección X para el caso 3 (FSIPY)	99
Tabla B.11	Esfuerzos normales en la dirección Y para el caso 3 (FSIPY)	99
Tabla B.12	Esfuerzos tangenciales para el caso 3 (FSIPY)	99
Tabla B.13	Esfuerzos normales en la dirección X para el caso 3 (ABAQUS) .	100
Tabla B.14	Esfuerzos normales en la dirección Y para el caso 3 (ABAQUS) .	100
Tabla B.15	Esfuerzos tangenciales para el caso 3 (ABAQUS)	100
Tabla B.16	Esfuerzos normales en la dirección X para el caso 3 (SAP2000)	101
Tabla B.17	Esfuerzos normales en la dirección Y para el caso 3 (SAP2000) .	101
Tabla B.18	Esfuerzos tangenciales para el caso 3 (SAP2000)	101

Dedicatoria

A mis padres, Guillermo y Edita, que siempre me han brindado su amor y apoyo a lo largo de todos los proyectos que he emprendido.

A mi hermano, Sebastián, por brindarme confianza y motivarme a seguir adelante.

A mis primos y primas, quienes han estado presentes en todas las etapas de mi vida.

A mis amigos y amigas, por las conversaciones motivadoras y por los años de amistad.

A todos aquellos que han formado parte de todo este proceso.

Pedro Miguel Asmal Rodas

Dedicado a mis padres, Manuel y Zoila por todo el apoyo que me han brindado durante mi vida de estudiante. En especial a mi madre quien me ha demostrado como se ve una guerrera.

Dedicado a mi hermano Fernando, por siempre haberme acompañado en todas mis ideas.

Dedicado a mi tía Nube, que siempre ha estado ahí para guiarme y apoyarme en cada etapa de mi vida. Quien ha llegado a ser mi segunda madre, mi madrina.

Dedicado a mis abuelitos Octavio, Rosa y Luz, quienes siempre han estado pendientes de mí y que gracias a sus consejos he logrado llegar ser quien soy.

Dedicado a mis amigos que a lo largo de mi carrera han sido una guía y gracias a quienes la vida fue más llevadera, en especial al team Azogues por las risas en los viajes.

Leonel Octavio Yumbla Cadme

Agradecimientos

Agradecemos a Dios por habernos brindado la sabiduría y fortaleza necesarias para alcanzar este objetivo.

A la Universidad de Cuenca por acogernos durante estos diez semestres de formación académica y personal.

A nuestros tutores Ing. Fernando Zalamea e Ing. Esteban Samaniego por el apoyo brindado en la realización de este trabajo.

A nuestros familiares, quienes han sido los pilares fundamentales de nuestra vida y nos han brindado su apoyo incondicional.

A nuestros amigos y amigas, con quienes hemos forjado fuertes lazos y disfrutado de diversas experiencias a lo largo de este trayecto.

Una mención especial al Ing. Cristóbal Samaniego, quien nos ha brindado su apoyo para llevar a cabo este proyecto.

Simbología y Nomenclatura

\mathbf{x}	Coordenadas materiales
\mathbf{X}	Coordenadas espaciales
∇	Gradiente
∇^S	Gradiente simétrico
ρ_0	Densidad
\mathbf{E}	Tensor de deformación material
\mathbf{e}	Tensor de deformación espacial
ε	Tensor de deformación
$\boldsymbol{\sigma}$	Tensor de esfuerzos
\mathbf{u}	Vector de desplazamiento
σ_0	Esfuerzos iniciales
ε_0	Deformación inicial
u_0	Desplazamiento inicial
$\sigma_x, \sigma_y, \sigma_z$	Esfuerzos normales
$\tau_{xy}, \tau_{xz}, \tau_{yz}$	Esfuerzos tangenciales
$\varepsilon_x, \varepsilon_y, \varepsilon_z$	Deformaciones normales
$\gamma_{xy}, \gamma_{xz}, \gamma_{yz}$	Deformaciones tangenciales
E	Módulo de elasticidad lineal
ν	Coefficiente de Poisson
\mathbb{D}	Matriz constitutiva
λ, μ	Constantes de Lamé
\mathbf{I}	Tensor identidad
\mathbf{b}	Fuerzas másicas
\mathbf{t}	Fuerzas superficiales
\mathbf{P}	Vector de fuerzas puntuales
W_i	Trabajo virtual interno
W_e	Trabajo virtual externo
\mathbf{F}_{int}	Fuerzas internas
\mathbf{F}_{ext}	Fuerzas externas
$\delta \mathbf{u}$	Desplazamiento virtual
$\delta \boldsymbol{\varepsilon}$	Deformación virtual

Ω	Dominio
Γ	Contorno de Ω
$A^{(e)}$	Área de un elemento finito
$N^{(e)}$	Funciones de forma de desplazamientos elementales
N	Funciones de forma de desplazamientos Globales
$B^{(e)}$	Funciones de forma de deformaciones elementales
B	Funciones de forma de deformaciones Globales
G	Derivada natural de las funciones de forma
$\mathbf{a}^{(e)}$	Desplazamiento de los nodos
$\mathbf{q}^{(e)}$	Vector de fuerzas puntuales externas
$\mathbf{K}^{(e)}, \mathbf{K}$	Matriz de rigidez elemental / Global
$\mathbf{f}^{(e)}, \mathbf{f}$	Vector de fuerzas elementales / Globales
\mathbf{r}	Vector de reacciones
$\mathbf{J}^{(e)}, \mathbf{J}$	Jacobiano elemental / Global
ξ, η	Coordenadas cartesianas
n_{pg}	Cantidad de puntos de integración de Gauss
w_i	Pesos de Gauss
N_{ext}	Funciones de forma para extrapolación

Capítulo 1

1. Introducción

1.1 Antecedentes y justificación

El método de los elementos finitos (MEF) es, sin duda, una de las herramientas más utilizadas en el análisis y diseño de sistemas mecánicos mediante la resolución numérica de ecuaciones que modelan estructuras como edificios, puentes, presas, macizos rocosos, entre otros. Permite realizar pruebas y optimizar diseños de forma rápida y versátil, lo que proporciona a los investigadores información valiosa sobre el rendimiento del sistema y la influencia de aspectos como la geometría, las propiedades del material, cargas, etc. (Abasolo, 2016).

El MEF es un procedimiento de aproximación de problemas de medios continuos, de tal manera que el medio continuo se divide en un número finito de elementos cuyo comportamiento está determinado por un número finito de parámetros, y la solución del sistema se basa en el ensamblaje de los elementos (Zienkiewicz & Taylor, 1992). Dependiendo del problema, los elementos pueden ser unidimensionales, tipo Frame, tipo Shell o volumétricos (Wilson, 2008).

En el mercado existe una gran variedad de software basados en el MEF. Sin embargo, su uso requiere adquirir licencias costosas. La Facultad de Ingeniería de la Universidad de Cuenca carece de un código o programa basado en el MEF, editable para estudios e investigación. Por esta razón, se obtuvo el permiso para editar un software de acceso libre basado en el MEF para la solución de problemas dinámicos en fluidos. Este software fue desarrollado por Barcelona SuperComputing Center (BSC) de España, para el proyecto Fluid Structure Interaction (FSI).

El desarrollo del presente trabajo de titulación se enfoca en la implementación del MEF para la generación de un software para la solución de problemas elástico-lineales isótropos bidimensionales en equilibrio, mediante el uso de bibliotecas y subrutinas de libre acceso, y del software FSIPY. Se enfatiza que el trabajo desarrollado está orientado a proporcionar un programa gratuito para uso educativo y de investigación.

1.2 Objetivos

1.2.1 *Objetivo general*

Implementar y validar un programa basado en el método de elementos finitos para la solución de problemas cuasi-estáticos en cuerpos bidimensionales.

1.2.2 *Objetivos específicos*

1. Profundizar en la teoría e implementación del método de elementos finitos.
2. Adaptar el código para resolver problemas cuasi-estáticos con estructura sólida aplicando el método de elementos finitos.
3. Validar la implementación utilizando benchmarking.
4. Elaborar un manual de instalación y de uso del software desarrollado.

1.3 Alcance

Se busca generar un código basado en el MEF con fines netamente académicos, utilizando herramientas de libre acceso para preproceso y posproceso. Para ello, se estudiará el software FSIPY y sus herramientas. El software FSIPY está desarrollado para solucionar problemas de grandes deformaciones considerando un comportamiento dinámico de los medios continuos (Oks, 2022).

Se plantea implementar una subrutina que permita obtener soluciones cuasi-estáticas (caso particular de análisis dinámico). El problema será abordado en el análisis elástico lineal, que se basa en pequeñas deformaciones, comportamiento lineal del material, efectos dinámicos despreciables y sin generación de traslapes durante la deformación. Quedan fuera del alcance de este trabajo de titulación, problemas no lineales o en grandes deformaciones.

Dentro de los contenidos, se tiene la búsqueda de herramientas y/o subrutinas, de libre acceso, prediseñadas para reducir el tiempo de programación y procesos a desarrollarse, como Fortran 90, para el ensamblaje y tratamiento de matrices. Para el mallado en elementos se utiliza GMSH. Para la graficación y presentación de resultados se emplea la herramienta ParaView. Se requiere el uso del sistema operativo Ubuntu 20.04, que es compatible con el software FSIPY en cuanto a compilación y ejecución.

1.4 Organización de capítulos

El presente trabajo de titulación se enfoca en el estudio de un software llamado FSIPY, que utiliza el MEF para la resolución de ecuaciones diferenciales del medio continuo en fluidos, con el propósito de la implementación de un nuevo módulo que resuelva problemas cuasi-estáticos en cuerpos bidimensionales. A continuación, se brinda un pequeño resumen de cada capítulo.

- 1. Introducción:** este capítulo aborda de manera general los antecedentes del trabajo de titulación, se identifica el estado del arte, se justifica la importancia de este trabajo y se establece el alcance del mismo. Además, se incluyen el objetivo general y los objetivos específicos.
- 2. Marco teórico:** se recurre a definiciones y conceptos básicos para la implementación de un programa basado en el MEF. También se explica la teoría de la elasticidad y su aplicación en la tensión y deformación plana.
- 3. Metodología del Desarrollo:** se explica y detalla el proceso para la adaptación del estado cuasi-estático a partir de software ya desarrollado. Incluye también procesos como la implementación de cargas superficiales y la extrapolación de esfuerzos.
- 4. Resultados:** se presentan los resultados obtenidos mediante el uso del nuevo módulo, los cuales se validan mediante su comparación con la solución analítica. Además, se lleva a cabo un proceso de benchmarking con los programas SAP 2000 y ABAQUS.
- 5. Conclusiones:** se presentan las conclusiones del trabajo de titulación.

Capítulo 2

2. Marco Teórico

2.1 Teoría de la Elasticidad lineal

2.1.1 Definición

La elasticidad es una propiedad característica de los materiales estructurales. Se puede describir como la capacidad de los cuerpos para recuperar su estado inicial después de retirar las fuerzas que se aplican sobre ellos (Timoshenko & Goodier, 1970).

En el campo de la ingeniería se adopta la elasticidad lineal como simplificación de las teorías más generales, con las siguientes hipótesis (Olivella & de Saracíbar Bosch, 2000):

- **Deformaciones infinitesimales:** los desplazamientos y sus gradientes son pequeños, por lo que no se diferencia la configuración material de la espacial. Las coordenadas materiales son las mismas de las espaciales (desplazamientos pequeños).

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \Rightarrow \mathbf{x} \approx \mathbf{X} \quad (2.1)$$

Con los gradientes pequeños no existe diferencia entre los tensores material y espacial de deformación, por lo cual se adopta el tensor de deformación infinitesimal.

$$\mathbf{E}(\mathbf{X}, t) \approx \mathbf{e}(\mathbf{x}, t) = \boldsymbol{\varepsilon}(\mathbf{x}, t) \quad (2.2)$$

- **Existencia de un estado neutro:** en la mayoría de los casos, las deformaciones y las tensiones iniciales son nulas en la configuración de referencia.

$$\begin{cases} \varepsilon_0(x, t_0) = 0 \\ \sigma_0(x, t_0) = 0 \end{cases} \quad (2.3)$$

- **Consideraciones de un proceso de deformación isotérmico y adiabático:** el término isotérmico hace referencia a que la temperatura permanece constante con el paso del tiempo y el término adiabático indica que no existe generación de calor en todo punto e instante de tiempo. Por lo general, los procesos de deformación lentos son considerados adiabáticos.

2.1.2 Matriz constitutiva

En el caso multidimensional, la Ley de Hooke generalizada mantiene la relación de linealidad entre el tensor de tensiones y de deformaciones. Una propiedad fundamental del comportamiento elástico es la relación entre las tensiones y las deformaciones en un cierto punto e instante, las cuales no dependen del historial de deformaciones.

$$\begin{cases} \boldsymbol{\sigma}(\mathbf{x}, t) = \mathbb{D} : \boldsymbol{\varepsilon}(\mathbf{x}, t) \\ \sigma_{ij} = \mathbb{D}_{ijkl} \varepsilon_{kl} \quad i, j \in \{1, 2, 3\} \end{cases} \quad (2.4)$$

Donde el tensor \mathbb{D} se lo conoce como el tensor de constantes elásticas, de cuarto orden. Posee $3^4 = 81$ componentes. Al considerar un material isotrópico el tensor no varía las componentes en cualquier sistema de coordenadas cartesiano. Se define como (Olivella & de Saracíbar Bosch, 2000):

$$\mathbb{D} = \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbf{I} \quad (2.5)$$

Donde λ y μ son conocidas como las constantes de Lamé, las cuales se obtienen de manera experimental. De acuerdo con la condición de isotropía, el tensor pasa a tener dos componentes (i y j). De tal forma que, la ecuación constitutiva para un material elástico lineal isótropo se puede expresar como:

$$\begin{aligned} \boldsymbol{\sigma} &= \lambda \text{Tr}(\boldsymbol{\varepsilon}) \mathbf{1} + 2\mu \boldsymbol{\varepsilon} \\ \sigma_{ij} &= \lambda \delta_{ij} \varepsilon_{ll} + 2\mu \varepsilon_{ij} \quad i, j \in \{1, 2, 3\} \end{aligned} \quad (2.6)$$

Para obtener las deformaciones en función de las tensiones se realiza una inversión en la ley de Hooke. Con ello, se definen las nuevas propiedades elásticas, módulo de Young (E) y coeficiente de Poisson (ν).

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \quad (2.7)$$

$$\nu = \frac{\lambda}{2(\lambda + \mu)} \quad (2.8)$$

Reescribiendo la ecuación constitutiva, se obtiene: la Ley de Hooke inversa.

$$\boldsymbol{\varepsilon} = -\frac{\nu}{E} \text{Tr}(\boldsymbol{\sigma}) \mathbf{1} + \frac{1 + \nu}{E} \boldsymbol{\sigma} \quad (2.9)$$

La ecuación se la puede expresar en función de sus tres componentes:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{bmatrix} = \begin{bmatrix} \frac{1}{E} & \frac{-\nu}{E} & \frac{-\nu}{E} \\ \frac{-\nu}{E} & \frac{1}{E} & \frac{-\nu}{E} \\ \frac{-\nu}{E} & \frac{-\nu}{E} & \frac{1}{E} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} \quad (2.10)$$

Al realizar el análisis de un elemento diferencial de un cuerpo en equilibrio, se observan tres tensiones normales (σ_x , σ_y y σ_z) y seis esfuerzos cortantes (τ_{xy} , τ_{yx} , τ_{xz} , τ_{zx} , τ_{yz} y τ_{zy}). Sin embargo, dado que el cuerpo está en equilibrio, únicamente se consideran tres de dichos esfuerzos para su estudio (Coronel, 2006).

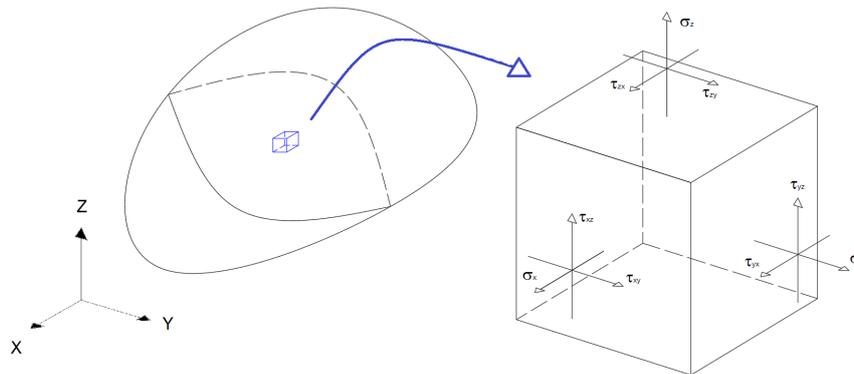


Figura 2.1: Estado tensional de un elemento diferencial (Coronel, 2006)

Considerando que el cuerpo cuenta con restricciones suficientes para impedir su movimiento como un cuerpo rígido, la aplicación de una fuerza produce desplazamientos asociados con la deformación. Por lo tanto, dichos esfuerzos generan tres deformaciones normales (ε_x , ε_y y ε_z) y tres deformaciones tangenciales (γ_{xy} , γ_{xz} y γ_{yz}) (Coronel, 2006).

2.1.3 Planteamiento del problema elástico lineal quasi-estático

Un sólido elástico lineal sometido a un vector de fuerzas másicas y a un vector de fuerzas superficiales en su contorno posee un conjunto de ecuaciones que permiten obtener la evolución de desplazamientos, deformaciones y tensiones a lo largo del tiempo. A estas ecuaciones se las conoce como: problema elástico lineal (Olivella & de Saracíbar Bosch, 2000).

El problema elástico lineal cuasi-estático considera que las variaciones en las fuerzas másicas, fuerzas superficiales y condiciones de contorno, así como las variaciones en las respuestas (u , ε y σ) son muy lentas en el tiempo, es decir, son problemas en los que se analiza la respuesta de un sistema a cargas o condiciones que cambian lentamente en el tiempo. Por lo que, la segunda derivada del vector de desplazamientos es igual a cero (Olivella & de Saracíbar Bosch, 2000).

$$\frac{\partial^2 \mathbf{u}(\mathbf{x}, t)}{\partial t^2} \approx 0 \quad (2.11)$$

2.1.3.1 Ecuaciones de gobierno

Son las ecuaciones fundamentales que rigen el comportamiento de un sólido. Estas ecuaciones incluyen (Olivella & de Saracíbar Bosch, 2000):

- **Ecuación de Cauchy (Ecuación de equilibrio):** ecuación encargada de describir el equilibrio estático.

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) + \rho_0 \mathbf{b}(\mathbf{x}, t) = \mathbf{0} \quad (2.12)$$

- **Ecuación constitutiva:** ecuación que relaciona las tensiones internas y las deformaciones en un material elástico lineal, que se comporta de manera isotrópica.

$$\boldsymbol{\sigma}(\mathbf{x}, t) = \lambda \text{Tr}(\boldsymbol{\varepsilon}) \mathbf{1} + 2\mu \boldsymbol{\varepsilon} \quad (2.13)$$

- **Ecuación geométrica:** ecuación utilizada para describir las relaciones geométricas entre las variables de posición y las variables de deformación de un sistema continuo.

$$\boldsymbol{\varepsilon}(\mathbf{x}, t) = \nabla^S \mathbf{u}(\mathbf{x}, t) = \frac{1}{2} (\mathbf{u} \otimes \nabla + \nabla \otimes \mathbf{u}) \quad (2.14)$$

Ninguna de estas ecuaciones involucra una derivada en el tiempo por lo que la solución del sistema de ecuaciones diferenciales se realiza solamente en términos del espacio. Las acciones que se toman en cuenta son: $\mathbf{b}(\mathbf{x})$, $\mathbf{t}(\mathbf{x})$ y $u_0(\mathbf{x})$. Obteniéndose los resultados de $\mathbf{u}(\mathbf{x})$, $\boldsymbol{\varepsilon}(\mathbf{x})$ y $\boldsymbol{\sigma}(\mathbf{x})$ (Olivella & de Saracíbar Bosch, 2000).

2.1.3.2 Condiciones de contorno

Se deben considerar las condiciones de contorno en el espacio, las cuales hacen referencia a deformaciones y tensiones (Olivella & de Saracíbar Bosch, 2000).

- **Condiciones de contorno en el espacio:**

Condiciones de contorno en desplazamientos (Γ_u):

$$\begin{cases} \mathbf{u}(\mathbf{x}, t) = u_0(\mathbf{x}, t) \\ u_i(\mathbf{x}, t) = u_{0i}(\mathbf{x}, t) \quad i \in \{1, 2, 3\} \end{cases} \Rightarrow \forall \mathbf{x} \in \Gamma_u \quad \forall t \quad (2.15)$$

Condiciones de contorno en tensiones (Γ_σ):

$$\begin{cases} \boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n} = \mathbf{t}(\mathbf{x}, t) \\ \sigma_{ij}(\mathbf{x}, t) \cdot n_j = t_j(\mathbf{x}, t) \quad i, j \in \{1, 2, 3\} \end{cases} \Rightarrow \forall \mathbf{x} \in \Gamma_\sigma \quad \forall t \quad (2.16)$$

Condiciones de contorno mixtas (desplazamiento-tensión) ($\Gamma_{u\sigma}$):

$$\begin{cases} u_i(\mathbf{x}, t) = u_{0i}(\mathbf{x}, t) \\ \sigma_{jk}(\mathbf{x}, t) \cdot \mathbf{n}_k = \mathbf{t}_j(\mathbf{x}, t) \end{cases} \Rightarrow (i, j, k \in \{1, 2, 3\} \quad i \neq j) \quad \forall \mathbf{x} \in \Gamma_{u\sigma} \quad \forall t \quad (2.17)$$

2.1.4 Resolución del problema elástico lineal cuasi-estático

Se puede abordar el problema desde un planteamiento de desplazamientos o tensiones. Actualmente, los métodos de resolución numérica están basados en el planteamiento de los desplazamientos (Olivella & de Saracíbar Bosch, 2000).

Planteamiento de desplazamientos: Ecuaciones de Navier

Partiendo de las ecuaciones de gobierno, se plantea un sistema reducido en el que la incógnita sea sólo el campo de desplazamientos. Dando lugar a las ecuaciones de Navier:

$$(\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu\nabla^2\mathbf{u} + \rho_0\mathbf{b} = \mathbf{0} \quad (2.18)$$

2.2 Elasticidad plana

La teoría de la elasticidad plana se fundamenta en el análisis de dos direcciones principales (X y Y), mientras que la tercera dimensión (Z) se considera irrelevante debido a la geometría del cuerpo y las condiciones de contorno. Los casos tratados son la tensión plana y la deformación plana (Olivella & de Saracíbar Bosch, 2000), cada uno con sus respectivas hipótesis de simplificación.

2.2.1 Tensión plana

La tensión plana se aplica a estructuras que tienen una dimensión (espesor) significativamente menor que las otras dimensiones. En este caso, las fuerzas másicas y

superficiales actúan únicamente sobre el plano donde se encuentra la estructura, y son nulas en la dirección Z . Entre las aplicaciones de la tensión plana se tiene: análisis de vigas peraltadas, placas, paredes bajo cargas axiales, presas en contrafuerte, entre otros (Oñate, 2009).

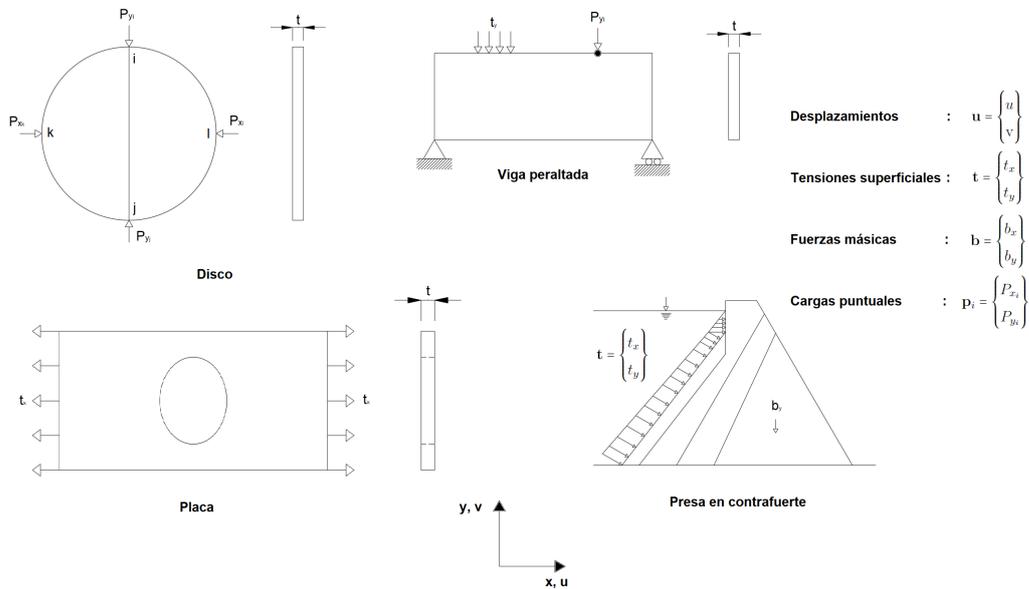


Figura 2.2: Ejemplos de tensión plana (Oñate, 2009)

El estado tensional de las estructuras queda definido por σ_x , σ_y y τ_{xy} . Estos tres componentes son independientes de Z , es decir, que no varían a través del espesor de la estructura. Mientras que las componentes σ_z , τ_{xz} y τ_{yz} , son nulas.

2.2.2 Deformación plana

La deformación plana se presenta en estructuras prismáticas que tienen una dimensión muy grande, en la dirección Z , en comparación con las otras dos dimensiones. Las estructuras que siguen la suposición de deformación plana son: presas de gravedad, tuberías presurizadas. En la geotecnia se tiene aplicaciones en túneles, cimentaciones, entre otros (Olivella & de Saracíbar Bosch, 2000).

Las cargas son perpendiculares y uniformemente distribuidas a lo largo del eje Z , es decir, no varían en esa dirección. Por lo tanto, todas las secciones están en iguales condiciones, lo que permite analizar una sección unitaria. En un modelo de deformación plana se considera las deformaciones ϵ_{xx} , ϵ_{yy} y γ_{xy} (Oñate, 2009).

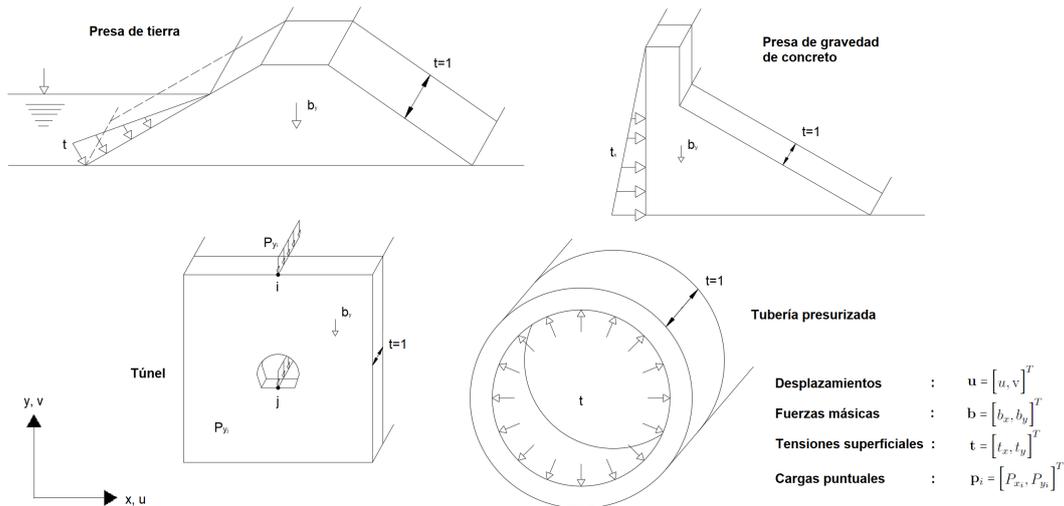


Figura 2.3: Ejemplos de deformación plana (Oñate, 2009)

2.2.3 Campo de desplazamientos

El vector de desplazamientos queda definido por $u(x, y)$ y $v(x, y)$. Para el caso de un punto el vector correspondiente es (Oñate, 2009):

$$\mathbf{u}(x, y) = \begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix} \tag{2.19}$$

2.2.4 Campo de deformaciones

A partir del campo de desplazamientos, se deducen las deformaciones mediante la teoría general de la elasticidad (Oñate, 2009):

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} \quad , \quad \varepsilon_y = \frac{\partial v}{\partial y} \\ \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad , \quad \gamma_{xz} = \gamma_{yz} = 0 \end{aligned} \tag{2.20}$$

Dentro de la deformación plana, la deformación longitudinal es cero, mientras que, para tensión plana, la tensión es cero. En los dos casos, la deformación ε_z es irrelevante ya que el producto escalar es cero ($\sigma_z \cdot \varepsilon_z = 0$). Por lo tanto, el vector de deformaciones significativas en un punto se define como (Oñate, 2009):

$$\boldsymbol{\varepsilon} = [\varepsilon_x, \varepsilon_y, \gamma_{xy}]^T \tag{2.21}$$

2.2.5 Campo de tensiones

En el campo de tensiones, se considera que las tensiones de corte τ_{xz} y τ_{yz} son cero. Debido a que el producto escalar entre esfuerzos y deformaciones es cero, no se tiene

en cuenta el valor de σ_z . Por lo que el campo de tensiones está definido por (Oñate, 2009):

$$\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \tau_{xy}]^T \quad (2.22)$$

2.2.6 Relación entre tensión-deformación

La relación entre esfuerzos y deformaciones se deriva de la teoría de la elasticidad en tres dimensiones, de acuerdo a las simplificaciones realizadas tanto para tensión plana ($\sigma_z = 0$) como para deformación plana ($\varepsilon_z = 0$). Se puede expresar como (Oñate, 2009):

$$\boldsymbol{\sigma} = \mathbb{D}\boldsymbol{\varepsilon} \quad (2.23)$$

Donde \mathbb{D} , es conocida como matriz de parámetros elásticos o matriz constitutiva. De acuerdo con (Fish & Belytschko, 2007), para los casos de tensión plana y deformación plana se tiene:

$$\mathbb{D} = \frac{\bar{E}}{1 - \bar{\nu}^2} \begin{bmatrix} 1 & \bar{\nu} & 0 \\ \bar{\nu} & 1 & 0 \\ 0 & 0 & \frac{1-\bar{\nu}}{2} \end{bmatrix} \Rightarrow \begin{cases} \bar{E} = E \\ \bar{\nu} = \nu \\ \bar{E} = \frac{E}{1-\nu^2} \\ \bar{\nu} = \frac{\nu}{1-\nu} \end{cases} \begin{matrix} \Rightarrow \text{Tensión plana} \\ \\ \Rightarrow \text{Deformación plana} \end{matrix} \quad (2.24)$$

2.3 Formulación fuerte y débil de las ecuaciones

2.3.1 Formulación fuerte

El problema bidimensional elástico lineal isótropo se rige por las siguientes ecuaciones de gobierno:

- Ecuación de equilibrio

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) + \rho_0 \mathbf{b}(\mathbf{x}, t) = \mathbf{0}$$

$$\begin{cases} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \rho b_x = 0 \\ \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \rho b_y = 0 \end{cases} \quad (2.25)$$

- Ecuación constitutiva

$$\boldsymbol{\sigma} = \mathbb{D}\boldsymbol{\varepsilon}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (2.26)$$

- Ecuación geométrica

$$\boldsymbol{\varepsilon} = \nabla^s \mathbf{u}$$

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (2.27)$$

2.3.2 Formulación débil

2.3.2.1 Principio de la trabajos virtuales

Este principio establece que si un punto material se encuentra sometido a un conjunto de fuerzas y se mantiene en equilibrio, entonces el trabajo total realizado por esas fuerzas será igual a cero para cualquier desplazamiento infinitesimal de una partícula. Un cuerpo se encuentra en equilibrio si el trabajo virtual interno es igual al trabajo virtual externo (Coronel, 2006).

$$W_i = W_e \quad (2.28)$$

Donde W_i indica el trabajo virtual interno y W_e el trabajo virtual externo.

$$W_e = \sum_{i=1}^n F_{e_i} \delta_i \quad (2.29)$$

$$W_i = \int_{\Omega} (\sum_{ij} \sigma_{ij} \varepsilon_{ij}) d\Omega \quad (2.30)$$

Para el caso de elasticidad en dos dimensiones, el principio de los trabajos virtuales se puede expresar como:

$$\int_{\Omega} (\delta \varepsilon_x \sigma_x + \delta \varepsilon_y \sigma_y + \delta \gamma_{xy} \tau_{xy}) d\Omega = \int_{\Omega} (\delta u b_x + \delta v b_y) d\Omega + \oint_{\Gamma} (\delta u t_x + \delta v t_y) d\Gamma \quad (2.31)$$

$$+ \sum_i (\delta u_i P x_i + \delta v_i P y_i)$$

La integral en el lado izquierdo de la ecuación representa el trabajo realizado por las tensiones σ_x , σ_y y τ_{xy} sobre las deformaciones virtuales $\delta \varepsilon_x$, $\delta \varepsilon_y$ y $\delta \gamma_{xy}$. La ecuación (2.31) se puede representar como:

$$\int_{\Omega} (\nabla^s \delta \mathbf{u})^T \mathbb{D} \nabla^s \mathbf{u} d\Omega = \oint_{\Gamma} \delta \mathbf{u}^T \mathbf{t} d\Gamma + \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega \quad (2.32)$$

2.4 Método de los Elementos Finitos

2.4.1 Introducción al método de los elementos finitos

El método de elementos finitos (Finite Element Method, por sus siglas en inglés) es un proceso de resolución numérica aproximada de las ecuaciones que gobiernan los problemas en la naturaleza. A menudo, el comportamiento de la naturaleza se puede describir mediante ecuaciones diferenciales o integrales. Por ello, el MEF se encuentra orientado a la resolución de dichas ecuaciones, obteniendo la evolución en el espacio y/o tiempo de una o más variables que representan el comportamiento de un sistema físico (Oñate, 2009).

En el caso de análisis de estructuras, el MEF resuelve las ecuaciones diferenciales parciales, determinando desplazamientos, tensiones y deformaciones en una estructura sometida a diferentes cargas. Actualmente, la resolución de los sistemas de ecuaciones se ha complementado con la incorporación del cálculo computacional de ordenadores (Ruffrán, 2020).

Un elemento finito se puede entender como la porción más pequeña de un medio continuo, como un sólido o una estructura. La geometría de un medio continuo está conformada por el ensamblaje de una colección de elementos finitos de geometría simple. Para su representación, se suele usar triángulos y cuadriláteros (2D) o tetraedros y hexaedros (3D) (Olivella & de Saracíbar Bosch, 2000).

2.4.2 Proceso de análisis del MEF

El MEF se ejecuta en tres fases principales, el preprocesamiento de datos, el procesamiento de datos y el posprocesamiento de resultados. En este sentido, se proporciona un flujograma (Figura 2.4) que ilustra el procedimiento descrito.

- **Preprocesamiento de datos:**

En esta etapa se define la red de elementos discretos conectados entre sí, cada elemento posee propiedades geométricas y mecánicas conocidas, todo esto constituye los datos del problema y debe definirse de manera sencilla (Oñate, 2009).

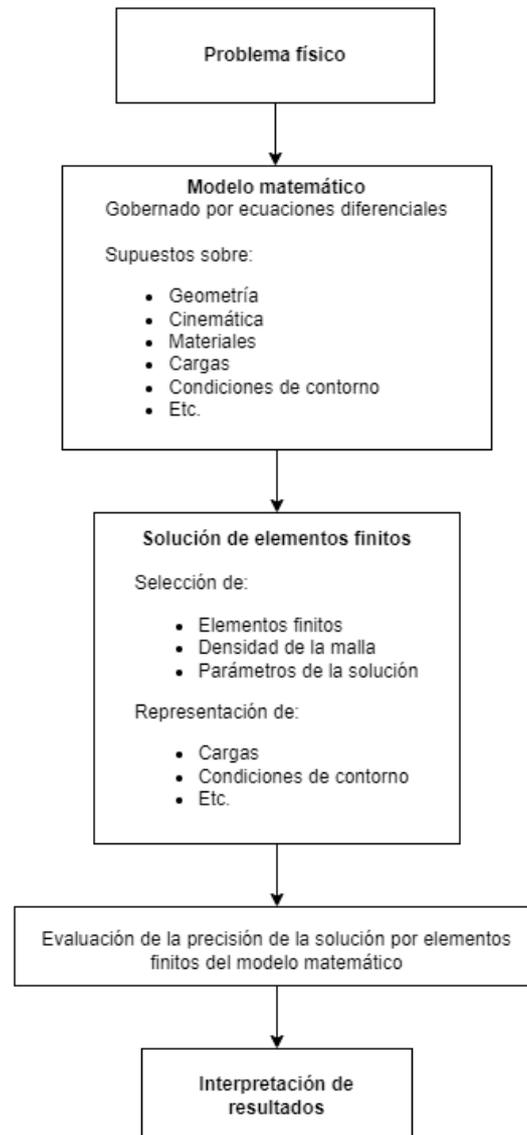


Figura 2.4: Proceso de análisis de elemento finito (Bathe, 1996)

• **Procesamiento de datos:**

La solución del sistema de ecuaciones involucra el cálculo de la matriz de rigidez $\mathbf{K}^{(e)}$ y del vector de fuerzas $\mathbf{f}^{(e)}$ para cada elemento del sistema. Se ensambla y resuelve la matriz global ($\mathbf{Ku} = \mathbf{f}$) para obtener los desplazamientos en cada nodo (Oñate, 2009).

• **Posprocesamiento de resultados:**

Como etapa final, la validación del MEF debe presentar los resultados del análisis en forma gráfica para que su interpretación sea sencilla (Oñate, 2009).

2.4.3 Formulación del modelo numérico

• **Discretización del campo de desplazamientos:**

$$u(x, y) = \sum_{i=1}^n N_i u_i \quad ; \quad \nu(x, y) = \sum_{i=1}^n N_i \nu_i \tag{2.33}$$

Donde u_i, ν_i son los desplazamientos horizontales y verticales, y N_i es la función de forma o de interpolación de los desplazamientos del nodo i . En la práctica es común utilizar la misma interpolación para representar tanto los desplazamientos horizontales como verticales.

Estas ecuaciones se pueden escribir de forma matricial:

$$\mathbf{u} = \begin{Bmatrix} u \\ \nu \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & \dots & N_n & 0 \\ 0 & N_1 & \dots & 0 & N_n \end{bmatrix} \begin{Bmatrix} u_1 \\ \nu_1 \\ \dots \\ u_n \\ \nu_n \end{Bmatrix} \tag{2.34}$$

$$\mathbf{u} = \mathbf{N} \mathbf{a}^{(e)}$$

Donde,

N: es la matriz de funciones de forma

a^(e): es el vector de desplazamientos en los nodos

• **Discretización del campo de deformaciones:**

Reemplazando la ecuación del campo de los desplazamientos en la ecuación

(2.27), se obtiene que:

$$\begin{aligned} \epsilon_x &= \frac{\partial u}{\partial x} = \sum_{i=1}^n \frac{\partial N_i}{\partial x} \cdot u_i \\ \epsilon_y &= \frac{\partial \nu}{\partial y} = \sum_{i=1}^n \frac{\partial N_i}{\partial y} \cdot \nu_i \\ \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial \nu}{\partial x} = \sum \left[\frac{\partial N_i}{\partial y} \cdot u_i + \frac{\partial N_i}{\partial x} \cdot \nu_i \right] \end{aligned} \tag{2.35}$$

Se puede expresar de manera matricial:

$$\varepsilon = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \dots & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & \dots & 0 & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \dots & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \dots \\ \dots \\ u_n \\ v_n \end{Bmatrix} \quad (2.36)$$

$$\varepsilon = \mathbf{B}\mathbf{a}^{(e)}$$

Donde,

B: es la matriz de deformación del elemento.

$\mathbf{a}^{(e)}$: es el vector de desplazamientos en los nodos.

- **Discretización del campo de tensiones:**

La expresión discretizada del vector de tensiones se obtiene de la ecuación constitutiva:

$$\boldsymbol{\sigma} = \mathbb{D}\boldsymbol{\varepsilon} = \mathbb{D}\mathbf{B}\mathbf{a}^{(e)} \quad (2.37)$$

Si se cuentan con tensiones o deformaciones iniciales, la ecuación se deduce como:

$$\boldsymbol{\sigma} = \mathbb{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 = \mathbb{D}\mathbf{B}\mathbf{a}^{(e)} - \mathbb{D}\boldsymbol{\varepsilon}_0 + \boldsymbol{\sigma}_0 \quad (2.38)$$

- **Discretización de la Ecuación de equilibrio:**

De manera habitual, el equilibrio de las fuerzas que actúan sobre el elemento se establece únicamente en los nodos, se crean fuerzas nodales específicas que compensan las fuerzas internas debidas a la deformación del elemento y a cualquier otra fuerza externa presente, como se observa en la figura 2.5.

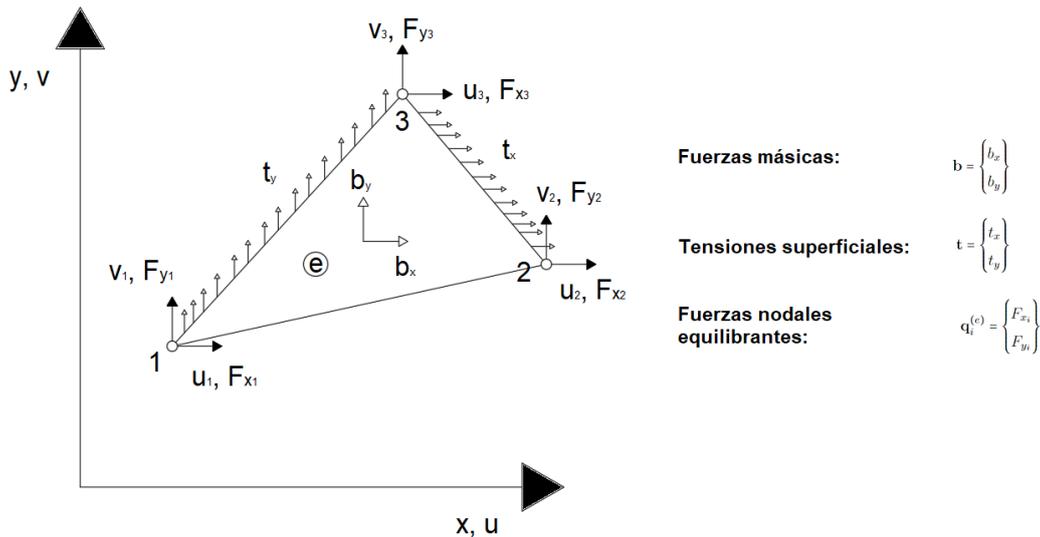


Figura 2.5: Fuerzas actuantes sobre un triángulo de tres nodos (Oñate, 2009)

Estas fuerzas nodales equilibrantes se obtienen a partir de la forma débil de la ecuación de la elasticidad lineal para problemas bidimensionales, la cual es:

$$\int_{\Omega} \delta \boldsymbol{\varepsilon} \cdot \boldsymbol{\sigma} d\Omega = \int_{\Omega} \delta \mathbf{u} \cdot \mathbf{b} d\Omega + \oint_{\Gamma} \delta \mathbf{u} \cdot \mathbf{t} d\Gamma + \sum_i (\delta u_i P_{x_i} + \delta v_i P_{y_i}) \quad (2.39)$$

Donde,

δu_i y δv_i : son los desplazamientos nodales virtuales.

P_{x_i} y P_{y_i} : son las fuerzas equilibrantes en los nodos. La ecuación (2.39), se puede expresar como:

$$\int_{\Omega^{(e)}} \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega^{(e)}} \delta \mathbf{u}^T \mathbf{b} d\Omega - \oint_{\Gamma^{(e)}} \delta \mathbf{u}^T \mathbf{t} d\Gamma = [\delta \mathbf{a}^{(e)}]^T \mathbf{q}^{(e)} \quad (2.40)$$

Los desplazamientos virtuales se interpolan en función de los valores nodales, por lo que las ecuaciones (2.34) y (2.36), se expresan como:

$$\begin{aligned} \delta \mathbf{u} &= \mathbf{N} \delta \mathbf{a}^{(e)} \quad ; \quad \delta \boldsymbol{\varepsilon} = \mathbf{B} \delta \mathbf{a}^{(e)} \\ \delta \mathbf{u}^T &= [\delta \mathbf{a}^{(e)}]^T \mathbf{N}^T \quad ; \quad \delta \boldsymbol{\varepsilon}^T = [\delta \mathbf{a}^{(e)}]^T \mathbf{B}^T \end{aligned} \quad (2.41)$$

Sustituyendo (2.41) en (2.40) y desarrollando, se obtiene la expresión (2.42):

$$\begin{aligned} \left[\int_{\Omega^{(e)}} \mathbf{B}^{(e)T} \mathbb{D} \mathbf{B}^{(e)} d\Omega \right] \mathbf{a}^{(e)} - \int_{\Omega^{(e)}} \mathbf{B}^{(e)T} \mathbb{D} \boldsymbol{\varepsilon}_0 d\Omega + \int_{\Omega^{(e)}} \mathbf{B}^{(e)T} \boldsymbol{\sigma}_0 d\Omega \\ - \int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \mathbf{b} d\Omega - \oint_{\Gamma^{(e)}} \mathbf{N}^{(e)T} \mathbf{t} d\Gamma = \mathbf{q}^{(e)} \end{aligned} \quad (2.42)$$

2.4.4 Matriz de rigidez y vectores de fuerza

La ecuación (2.42), se puede escribir como:

$$\mathbf{K}^{(e)} \mathbf{a}^{(e)} - \mathbf{f}^{(e)} = \mathbf{q}^{(e)} \quad (2.43)$$

Donde,

$$\mathbf{K}^{(e)} = \int_{\Omega} \mathbf{B}^{(e)T} \mathbb{D} \mathbf{B}^{(e)} d\Omega \quad (2.44)$$

Se la conoce como matriz de rigidez del elemento y,

$$\mathbf{f}^{(e)} = \mathbf{f}_{\varepsilon}^{(e)} + \mathbf{f}_{\sigma}^{(e)} + \mathbf{f}_{\mathbf{b}}^{(e)} + \mathbf{f}_{\mathbf{t}}^{(e)} \quad (2.45)$$

Es el vector de fuerzas nodales equivalentes, el cual se debe a deformaciones iniciales, esfuerzos iniciales, fuerzas másicas y fuerzas superficiales.

$$\begin{aligned} \mathbf{f}_{\varepsilon}^{(e)} &= \int_{\Omega^{(e)}} \mathbf{B}^{(e)T} \mathbb{D} \varepsilon_0 d\Omega \\ \mathbf{f}_{\sigma}^{(e)} &= \int_{\Omega^{(e)}} \mathbf{B}^{(e)T} \sigma_0 d\Omega \\ \mathbf{f}_{\mathbf{b}}^{(e)} &= \int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \mathbf{b} d\Omega \\ \mathbf{f}_{\mathbf{t}}^{(e)} &= \oint_{\Gamma^{(e)}} \mathbf{N}^{(e)T} \mathbf{t} d\Gamma \end{aligned} \quad (2.46)$$

En los nodos se establece el equilibrio, de tal manera que la suma de todas las fuerzas nodales equilibrantes se compensa con las cargas puntuales “ P_i ” que se aplican en el nodo:

$$\sum_e q_j^{(e)} = P_i \quad , \quad i = 1, N \quad (2.47)$$

El vector “ P ” incluye las reacciones en los nodos prescritos, estas se calculan a posteriori de acuerdo a:

$$\mathbf{r} = \mathbf{F}_{int} - \mathbf{F}_{ext} \quad (2.48)$$

Donde,

\mathbf{F}_{ext} : son las fuerzas externas debido a las fuerzas másicas, fuerzas superficiales y fuerzas puntuales.

\mathbf{F}_{int} : son las fuerzas internas debida a la deformación del elemento y se obtienen mediante:

$$\mathbf{F}_{int}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}^{(e)T} \sigma d\Omega \quad (2.49)$$

2.5 Elementos isoparamétricos de cuatro nodos

La discretización de un medio continuo en elementos finitos se lo puede realizar empleando una serie de elementos como triángulos, cuadriláteros y elementos de orden superior. Estos elementos no siempre van a tener una forma regular, por lo que se emplea la formulación isoparamétrica (Fish & Belytschko, 2007).

2.5.1 Funciones de forma

Las funciones de forma para dos dimensiones se obtienen mediante el producto de dos funciones de forma unidimensionales (Fish & Belytschko, 2007). Este producto se lo ilustra en la Figura 2.6.

$$N_{[I,J]}^{(e)}(x, y) = N_I^{(e)}(x)N_J^{(e)}(y) \quad \text{para } I = 1, 2 \quad \text{y } J = 1, 2$$

$$N_1^{(e)}(x, y) = \frac{x - x_2^{(e)}}{x_1^{(e)} - x_2^{(e)}} \frac{y - y_4^{(e)}}{y_1^{(e)} - y_4^{(e)}} = \frac{1}{A^{(e)}}(x - x_2^{(e)})(y - y_4^{(e)}),$$

$$N_2^{(e)}(x, y) = \frac{x - x_1^{(e)}}{x_2^{(e)} - x_1^{(e)}} \frac{y - y_4^{(e)}}{y_1^{(e)} - y_4^{(e)}} = -\frac{1}{A^{(e)}}(x - x_1^{(e)})(y - y_4^{(e)}),$$

$$N_3^{(e)}(x, y) = \frac{x - x_1^{(e)}}{x_2^{(e)} - x_1^{(e)}} \frac{y - y_1^{(e)}}{y_4^{(e)} - y_1^{(e)}} = \frac{1}{A^{(e)}}(x - x_1^{(e)})(y - y_1^{(e)}),$$

$$N_4^{(e)}(x, y) = \frac{x - x_2^{(e)}}{x_1^{(e)} - x_2^{(e)}} \frac{y - y_1^{(e)}}{y_4^{(e)} - y_1^{(e)}} = -\frac{1}{A^{(e)}}(x - x_2^{(e)})(y - y_1^{(e)})$$
(2.50)

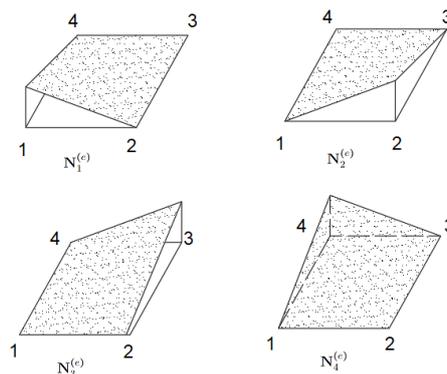


Figura 2.6: Funciones de forma de un elemento rectangular (Fish & Belytschko, 2007)

Se define un sistema de coordenadas locales (ξ, η) con el objetivo de facilitar la derivación de las funciones de forma. Este sistema de coordenadas natural se encuentra normalizado, por lo que el elemento esta limitado por: $\xi = \pm 1$ y $\eta = \pm 1$, tal como se observa en la Figura 2.7.

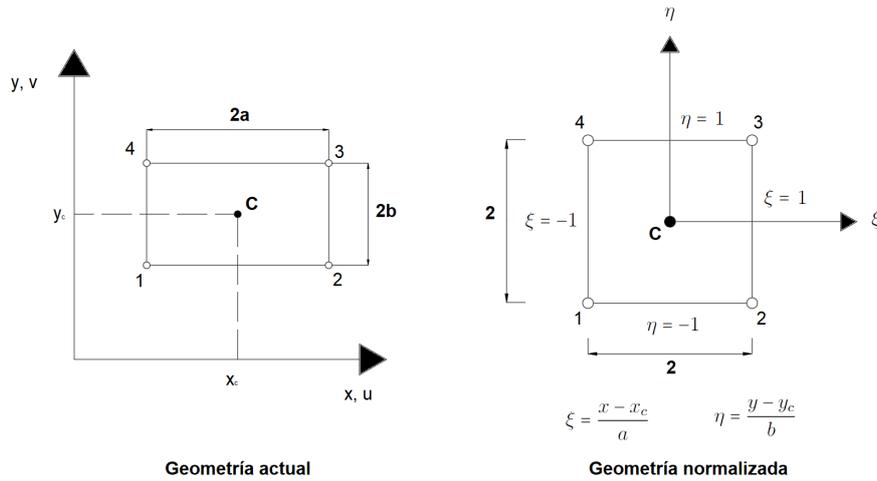


Figura 2.7: Elemento rectangular. Sistema de coordenadas naturales y cartesianas (Oñate, 2009)

Las funciones de forma en coordenadas naturales para elementos continuos deben cumplir:

- Condición de compatibilidad nodal
- Condición de cuerpo rígido

Se pueden expresar como:

$$N_i = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) \tag{2.51}$$

Tabla 2.1: Elemento rectangular de cuatro nodos de Lagrange (Oñate, 2009)

Número de nodos	ξ_i	η_i
1	-1	-1
2	1	-1
3	1	1
4	-1	1

Las funciones bilineales en términos de X y Y son adecuadas para rectángulos, pero no para elementos cuadriláteros arbitrarios, lo que hace que dos nodos comunes no sean suficientes para garantizar la continuidad. Para solucionar este problema se considera el elemento isoparamétrico (Fish & Belytschko, 2007).

2.5.2 Formulación Isoparamétrica

La formulación de elementos finitos isoparamétricos relaciona los desplazamientos del elemento en cualquier punto y los desplazamientos nodales mediante la utilización directa de las funciones de forma (Bathe, 1996). Esta relación se puede expresar en forma matricial como (Oñate, 2009):

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = \mathbf{J}^{(e)} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} \quad (2.52)$$

Donde, $\mathbf{J}^{(e)}$ es la matriz jacobiana, que permite la transformación de las derivadas de N_i en los ejes naturales y globales. Esta matriz es calculada para cada elemento. Despejando las derivadas cartesianas de las funciones de forma se tiene que:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = (\mathbf{J}^{(e)})^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix}, \text{ donde } \mathbf{J}^{(e)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (2.53)$$

Por lo tanto, $\mathbf{J}^{(e)}$:

$$\mathbf{J}^{(e)} = \begin{bmatrix} \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \quad (2.54)$$

La matriz Jacobiana y las derivadas cartesianas de las funciones de forma dependen de las coordenadas de los elementos. Mientras que las funciones de forma y las derivadas naturales de las funciones de forma no dependen de las coordenadas de los elementos, por lo que son iguales para cada elemento cuadrilátero. Las derivadas cartesianas para un elemento cuadrilátero de cuatro nodos son (Fish & Belytschko, 2007):

$$\begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} \eta - 1 & 1 - \eta & 1 + \eta & -\eta - 1 \\ \xi - 1 & -\xi - 1 & 1 + \xi & 1 - \xi \end{bmatrix} \quad (2.55)$$

2.6 Extrapolación de los esfuerzos

Las deformaciones se calculan a partir de las derivadas de los desplazamientos. Por lo que su orden de aproximación es menor. Si las funciones de forma son polinomios de grado Q , la aproximación para las deformaciones será un polinomio de grado $Q - 1$ o $Q - 2$, dependiendo si son obtenidos a través de las primera o segunda derivada del

campo de los desplazamientos (Oñate, 2009).

Los esfuerzos son obtenidos a partir de las deformaciones en los nodos mediante la ecuación constitutiva. Se considera el caso unidimensional de la Figura 2.8, donde los esfuerzos en los puntos de Gauss σ_1 y σ_2 , en las coordenadas $\xi = \pm p$ ($p = \frac{\sqrt{3}}{3}$) son linealmente interpolados en el rango $-p < \xi < p$, resultando:

$$\sigma = \begin{bmatrix} \frac{1-s}{2} & \frac{1+s}{2} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix} \tag{2.56}$$

Donde $s = \xi/p$

Los esfuerzos en los nodos A y B se calculan utilizando $s = \pm 1/p$.

$$\begin{Bmatrix} \sigma_A \\ \sigma_B \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix} \tag{2.57}$$

Donde, $a = 1 + \frac{1}{p}$ y $b = 1 - \frac{1}{p}$

La interpolación bilineal usa valores en los 4 puntos de Gauss de un elemento cuadrilátero, de tal forma que:

$$\sigma(s, t) = \sum_{i=1}^4 N_i(s, t) \sigma_i \tag{2.58}$$

Se representa de manera gráfica en la Figura 2.8.

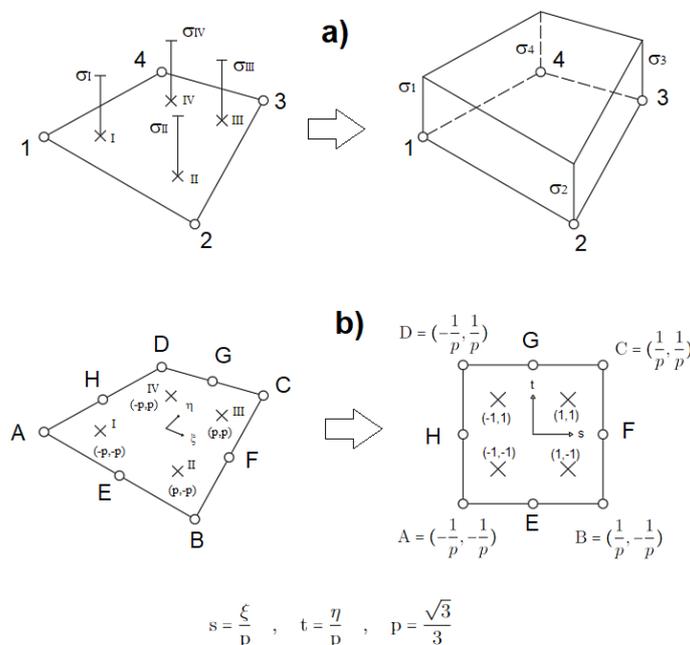


Figura 2.8: Extrapolación local para Coordenadas de Gauss 2x2 (Oñate, 2009)

Es posible apreciar que en el i-ésimo punto de Gauss el valor que se toma es de uno y el resto, cero. Para obtener el valor de la tensión en cualquier punto dentro del

elemento se sustituye los valores adecuados de las coordenadas s y t . Para el caso del nodo A de la Figura 2.8 se tendría:

$$\sigma_A = \begin{bmatrix} a^2 & ab & b^2 & ab \end{bmatrix} \begin{bmatrix} \sigma^1 & \sigma^2 & \sigma^3 & \sigma^4 \end{bmatrix}^T \quad (2.59)$$

Donde, $a = 1 + \frac{1}{p}$ y $b = 1 - \frac{1}{p}$

2.7 Métodos de integración numérica

2.7.1 Cuadratura de Gauss

La cuadratura de Gauss utiliza una menor cantidad de puntos de integración para alcanzar una precisión específica. Esta propiedad es fundamental, ya que se realiza una alta cantidad de operaciones matemáticas en cada punto de integración (Davis, 1984).

- **Cuadratura de Gauss en una dimensión:**

La cuadratura de Gauss en una dimensión se realiza cuando se integra con respecto a una sola variable, en este caso (ξ):

$$\int_{-1}^1 F(\xi) d\xi = \sum_{i=1}^{n_{pg}} w_i F(\xi_i) \quad (2.60)$$

Donde,

n_{pg} : cantidad de puntos de integración de Gauss utilizados en la cuadratura

w_i : pesos para cada punto de integración.

ξ_i : coordenadas de cada punto de integración

Debido a que la integral está definida en el intervalo [-1,1], se omite la transformación lineal del dominio.

- **Cuadratura de Gauss en dos dimensiones:**

Para resolver integrales en dos dimensiones, se aplica cuadraturas unidimensionales a cada variable independiente (Cuadratura del producto):

$$\begin{aligned} \int_{-1}^1 \int_{-1}^1 F(\xi, \eta) d\xi d\eta &= \int_{-1}^1 \left(\int_{-1}^1 F(\xi, \eta) d\xi \right) d\eta = \int_{-1}^1 \left(\sum_{i=1}^{n_{pg1}} w_i F(\xi_i, \eta) \right) d\eta \quad (2.61) \\ &= \sum_{i=1}^{n_{pg1}} \sum_{j=1}^{n_{pg2}} w_i w_j F(\xi_i, \eta_j) \end{aligned}$$

Si las funciones tienen el mismo grado en ξ y η , se adopta $n_{pg1} = n_{pg2}$. Por lo que los puntos de Gauss son: 1 punto, 2 x 2, 3 x 3, 4 x 4, etc.

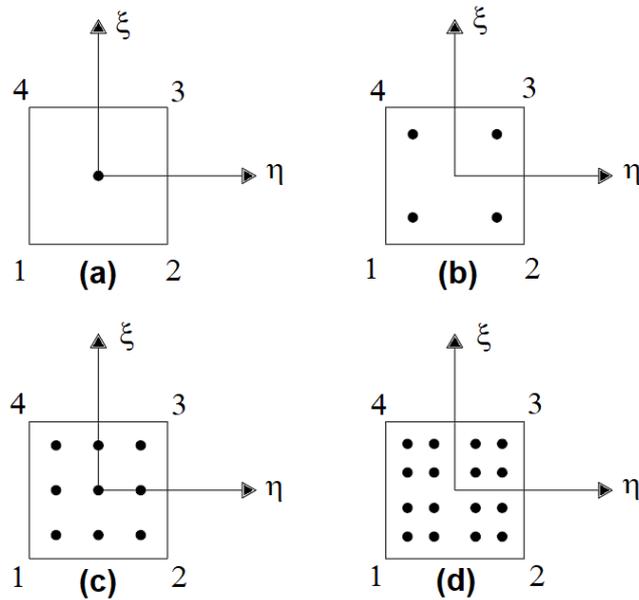


Figura 2.9: Cuadratura de Gauss-Legendre sobre elementos cuadriláteros: a) 1x1 b) 2x2 c) 3x3 y d) 4x4 puntos de integración (Coronel, 2006)

2.7.2 Coordenadas y pesos de la cuadratura de Gauss

En el caso bidimensional los puntos de Gauss pasan a formar coordenadas de ξ y η . Los valores de las coordenadas y los pesos de Gauss dependen del número de puntos de integración utilizados en la cuadratura. En la Tabla 2.2 se puede observar los valores de x_i y los pesos asociados w_i , para distintos puntos de integración.

Tabla 2.2: Coordenadas y pesos de la cuadratura de Gauss (Oñate, 2009)

Número de puntos, n	Puntos x_i	Pesos w_i
1	0	2
2	$\pm\sqrt{\frac{1}{3}}$	1
3	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	$\pm\sqrt{\frac{3 - 2\sqrt{\frac{6}{5}}}{7}}$	$\frac{18 + \sqrt{30}}{36}$
	$\pm\sqrt{\frac{3 + 2\sqrt{\frac{6}{5}}}{7}}$	$\frac{18 - \sqrt{30}}{36}$
5	0	$\frac{128}{225}$
	$\pm\frac{1}{3}\sqrt{5 - 2\sqrt{\frac{10}{7}}}$	$\frac{322 + 13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5 + 2\sqrt{\frac{10}{7}}}$	$\frac{322 - 13\sqrt{70}}{900}$

2.7.3 Integración numérica de la matriz de rigidez

Para poder realizar una integración numérica de la ecuación (2.44) utilizando la cuadratura de Gauss, la matriz de rigidez se expresa como:

$$\mathbf{K}^{(e)} = \int_{\eta=-1}^1 \int_{\xi=-1}^1 \mathbf{B}^{(e)T}(\xi, \eta) \mathbb{D} \mathbf{B}^{(e)}(\xi, \eta) |\mathbf{J}^{(e)}(\xi, \eta)| d\xi d\eta \quad (2.62)$$

El elemento diferencial de área $d\Omega$ se relaciona con un elemento en el espacio isoparamétrico mediante el determinante de la matriz jacobiana:

$$d\Omega = dx dy = \mathbf{J} d\xi d\eta \quad (2.63)$$

2.7.4 Integración numérica del vector de fuerzas equivalentes

En el cálculo del vector de fuerzas nodales, se pueden identificar dos componentes distintas:

- **Cargas volumétricas (másicas):** la ecuación (2.46) se puede expresar en función de las coordenadas isoparamétricas.

$$\mathbf{f}_{vol}^{(e)} = \int_{\eta=-1}^1 \int_{\xi=-1}^1 G_i(\xi, \eta) d\xi d\eta \quad (2.64)$$

$$\mathbf{G} = \mathbf{N}^T \mathbf{b} \mathbf{J}$$

- **Cargas distribuidas en el contorno:** este vector correspondiente a las cargas distribuidas se lo expresa como:

$$\mathbf{f}_{dist}^{(e)} = \int_{\Gamma^{(e)}} \mathbf{N}^{(e)T} \mathbf{t} d\Gamma \quad (2.65)$$

Es necesario transformar tanto la expresión del vector \mathbf{t} como el elemento diferencial $d\Gamma$ para que sea posible expresar la integral en el espacio isoparamétrico usando las coordenadas naturales.

$$\mathbf{f}_{dist}^{(e)} = \int_{\eta=-1}^1 \mathbf{N}^{(e)}(\xi)^T \mathbf{J} d\xi \mathbf{t} \quad (2.66)$$

Capítulo 3

3. Metodología

En este capítulo se presenta la metodología utilizada en la implementación y validación del nuevo módulo para la solución de problemas en estado cuasi-estático de sólidos bidimensionales del software FSIPY (Fluid Structure Interaction Python).

Se inicia con una etapa de comprensión de los módulos del software FSIPY, donde se identificó los cambios necesarios para adaptar las condiciones cuasi-estáticas. Luego, se procedió con la implementación de casos de tensión o deformación plana. Posteriormente, se integró funciones que incorporen fuerzas superficiales y extrapolen los esfuerzos a los nodos.

3.1 Programas utilizados

3.1.1 Lenguajes de programación

Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, que se utiliza en una amplia gama de aplicaciones en informática, ciencia de datos, inteligencia artificial, desarrollo web y otros campos. Fue creado en 1991 por Guido van Rossum. Entre las ventajas que ofrece Python se incluyen una sintaxis simple y legible, y la capacidad de ejecutarse en diferentes plataformas.

Python ofrece una gran cantidad de bibliotecas y módulos que simplifican tareas comunes y complejas. Entre estas se incluyen GMSH, NumPy, Pandas, Matplotlib y Scikit-Learn, que son utilizadas por el software FSIPY (Foundation, 2023).

FORTRAN

FORTRAN (FORmula TRANslation) es un lenguaje de programación de alto nivel y propósito general, diseñado específicamente para la computación científica y numérica. Fue creado en la década de 1950 por International Business Machines Corporation (IBM).

FORTRAN es conocido por su eficiencia en el procesamiento de grandes cantidades de datos numéricos y su capacidad para manejar números en coma flotante con una alta precisión (Backus & Heising, 1964). Por lo que, se emplea este lenguaje para realizar los cálculos del software FSIPY.

MATLAB

MATLAB (MATrix LABoratory) es un entorno de computación numérica y programación utilizado principalmente en ingeniería, matemáticas y ciencias aplicadas. Cuenta con interfaz gráfica, análisis numérico, simulación y modelado. Además, presenta un lenguaje de programación de alto nivel basado en vectores, arreglos y matrices (Pérez, 2002).

MATLAB presenta un entorno de desarrollo integrado (IDE) y un lenguaje de programación propio (.M). Es una herramienta muy completa para la computación numérica y la programación en general (MathWorks, 2023).

3.1.2 Preprocesamiento

GMSH

GMSH es un programa de código abierto que se utiliza tanto para el preprocesamiento como para el posprocesamiento en el ámbito de los elementos finitos. Cuenta con un motor CAD integrado que permite la creación de mallas de elementos finitos. Una de sus principales ventajas de GMSH es su capacidad para generar mallas de alta calidad (Geuzaine & Remacle, 2009). Cuenta con cuatro módulos que son: geometría, malla, solucionador y posprocesamiento.

3.1.3 Procesamiento

FSIPY

FSIPY es un software desarrollado por Barcelona SuperComputing Center, para la simulación de sistemas físicos como fluidos, sólidos y la interacción entre ellos. Es un software gratuito y de código abierto, utilizado en proyectos de investigación.

FSIPY consta de varios módulos que permiten la simulación de diversos sistemas. Su programación se basa en diferentes lenguajes como Python y Fortran. Para el preprocesamiento de datos utiliza GMSH y para el posprocesamiento, ParaView.

Se eligió este software por su disponibilidad gratuita y su eficiencia en el procesamiento de problemas. Al tener la capacidad de modificar el código, es posible adaptar la resolución de problema a necesidades específicas utilizando fórmulas o bibliografías particulares.

3.1.4 Posprocesamiento

ParaView

ParaView es un software desarrollado por Kitware, el cual permite la visualización y análisis de datos. Con su interfaz de usuario abierta, flexible e intuitiva, se puede analizar conjuntos de datos extremadamente grandes de forma interactiva (Inc., 2023). ParaView utiliza el Visualization Toolkit (VTK) como motor de procesamiento y representación de datos.

3.2 Estado inicial del software FSIPY

El presente trabajo de titulación se basa en el aprovechamiento de funciones desarrolladas en el software FSIPY. Es importante resaltar que gran parte del tiempo conlleva el entendimiento de un programa realizado por terceros para modificar las funciones integradas en los distintos módulos.

El software FSIPY está diseñado para resolver problemas relacionados a fluidos y la interacción fluido-sólido, por lo que sus rutinas están enfocadas a resolver este tipo de problemas. Referente a sólidos este software se basa en los criterios del capítulo 6 del Libro de Ted Belytschko “Nonlinear Finite Elements for Continua and Structures” (Belytschko *et al.*, 2014). Este software permite obtener los vectores de desplazamiento, velocidad y aceleración de una barra sometida a una fuerza másica equivalente al peso propio.

Como se mencionó el software FSIPY utiliza las herramientas de libre acceso GMSH y ParaView para el preprocesamiento y posprocesamiento de información. Los módulos programados en Python y Fortran se encargan de la importación de información proveniente de GMSH. Después, se calculan los parámetros de desplazamiento, velocidad y aceleración en los nodos. Al final, se exportan los datos en el formato VTK para su visualización en ParaView.

Desde este punto se realizaron cambios en los módulos con el fin de adaptar la resolución de sólidos bidimensionales isótropos en condiciones cuasi-estáticas. Se usó el subsistema de Windows para Linux (WSL) para la instalación y compilación del software FSIPY. El proceso se detalla en el manual técnico, Anexo D.

3.2.1 Preprocesamiento

En una simulación computacional de un problema de elementos finitos, todos los pasos involucrados en la definición del modelo, antes de la solución de la ecuación diferencial algebraica, forman parte del preprocesamiento. Para el caso del software FSIPY en esta etapa se incluye la entrada y lectura de datos, así como la generación de la malla. Además, se realiza la inicialización de variables.

3.2.1.1 Entrada de datos

La modelación de ensayos requiere de cierta información de entrada, la cual incluye la configuración de los materiales, las condiciones de contorno y el mallado del cuerpo a estudiarse. Cada uno de estos aspectos debe ser especificado con la siguiente información:

- **Configuración de los materiales:** consta de parámetros físicos, parámetros numéricos, sources y parámetros de salida.

Los parámetros físicos presentan la información básica respecto al tipo de material, como la densidad, el módulo de Young y el coeficiente de Poisson. Además, se especifica el número dimensiones espaciales del problema, el cual debe ser igual a dos para sólidos bidimensionales. Como se muestra en la Tabla 3.1.

Tabla 3.1: Parámetros físicos

Parámetros físicos		Observaciones
Physics	solid	Problem physics
Material	isolinear	Material model
Rho	1000.0	Density
Young	1.4e8	Young's modulus
Poisson	0.4	Poisson ratio
ndime	2	Number of spatial dimensions

En los parámetros numéricos se ingresan los valores necesarios para la solución del problema, como se muestra en la Tabla 3.2. Los parámetros más importantes para el caso de estudio cuasi-estático son volumen quadrature y surface quadrature, estos valores indican el polinomio de aproximación para la integración numérica de la matriz de rigidez elemental y el vector de fuerzas elementales.

Tabla 3.2: Parámetros numéricos

Parámetros Numéricos		Observaciones
Time treatment	implicit	Time treatment scheme
Tolerance	1e-9	Tolerance for iterations
Nitermax	1000	Max number of iterations
Volumen quadrature	Gauss2	Quadrature for volume numerical integration
Surface quadrature	Gauss2	Quadrature for surface numerical integration

En la Tabla 3.3 se muestra el apartado sources, donde se indica la dirección de las fuerzas másicas. Estos corresponden a las componentes “X” y “Y” de un vector. Si se desea considerar el peso propio se ingresa 0 en la componente “X” y -9.8 en “Y”, de tal forma que se colocan las fuerzas gravitatorias en la dirección vertical.

Tabla 3.3: Sources

Sources		Observaciones
Type	uniform	Type of source
Nparameters	2	Number of parameters for source
Parameters	0.0 -1.0	Source parameters

En los outputs se especifica el nombre de la carpeta donde se exportan los resultados. Además, se debe indicar el formato de estos archivos, para este caso, VTK. En la Figura 3.4 se puede observar la configuración del mismo.

Tabla 3.4: Outputs

Parámetros Numéricos		Observaciones
Path outs	resultados	Path to outputs directory
Output datatype	VTK	Output datatype
Ndump	1	Number of timesteps between outputs
In volumen	displacement velocity acceleration	Saved variables

- **Condiciones de contorno:** en este documento se definen las restricciones en los grados de libertad. De acuerdo al caso implementado, se etiquetan primero los nodos y después los lados correspondientes de la barra. Tal como se muestra en la Figura 3.1.

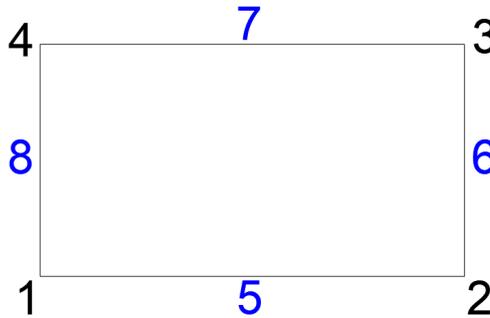


Figura 3.1: Etiquetas en un elemento cuadrilátero

Las condiciones de contorno se ingresan de acuerdo a la Tabla 3.5, en la cual se considera un empotramiento en el extremo izquierdo. Donde el primer valor indica la etiqueta del elemento, ya sea un punto o línea. Seguido a este se indica la restricción en los grados de libertad “X” y “Y” respectivamente, si existe una restricción se ingresa un valor de “1” y en caso de no existir se coloca “0”.

Tabla 3.5: Condiciones de contorno

Nodes		Lines	
Tag	DOF (XY)	Tag	DOF (XY)
1	11	5	00
2	00	6	00
3	00	7	00
4	11	8	11

- **Mallado:** en este archivo se configura el cuerpo a discretizarse. Para realizar el mallado en GMSH se debe definir la geometría del cuerpo bidimensional en un archivo de extensión (.geo). En este archivo se ingresa: la cantidad de nodos en cada dirección, coordenadas del sólido, líneas, superficies y etiquetas.

El ingreso de los nodos de la geometría mostrada en la Figura 3.1, se realiza de acuerdo a la sintaxis: $Point(Tag) = \{x, y, z\}$; con los valores de la Tabla 3.6.

Tabla 3.6: Configuración para el mallado

Tag	X	Y	Z
1	0.0	0.0	0.0
2	0.5	0.0	0.0
3	0.5	0.3	0.0
4	0.0	0.3	0.0

Se considera una malla rectangular de 355 nodos (71 en la dirección “X” y 5 en la dirección “Y”). Para el caso de las líneas se debe indicar los puntos que corresponden a los extremos de las mismas y la cantidad de nodos en cada línea, de acuerdo a la sintaxis: $Line(Tag) = \{P_{inicial}, P_{final}, Type, Number\}$; con los valores de la Tabla 3.7.

Tabla 3.7: Configuración elemento línea

Tag	$P_{inicial}$	P_{final}	Type	Nnodes
1	1	2	Transfinite {1}	71
2	2	3	Transfinite {2}	5
3	3	4	Transfinite {3}	71
4	4	1	Transfinite {4}	5

En las superficies se debe indicar las líneas que lo conforman, estas deben formar una figura cerrada y se las enumera en sentido antihorario, de acuerdo a:

$$Curve Loop (1) = \{1, 2, 3, 4\}$$

$$Plane Surface (1) = \{1\}$$

Al ejecutar el archivo tipo (.geo) en GMSH da como resultado un archivo de extensión (.msh) en el cual se especifican las coordenadas de los puntos, el tipo de elemento y la conectividad entre estos puntos. La importancia de este archivo (.msh), se debe a que mediante este se importa la información de la malla a Python a través de la API de GMSH.

3.2.1.2 Lectura de datos

La información de los archivos “conf.in, boun.in y .msh” se incorpora al software FSIPY mediante el módulo parser, que realiza una lectura de cada uno de los archivos de extensión (.in). Previo a la lectura de datos, se ejecutan los siguientes módulos:

- **Main:** primer módulo en ejecutarse, en este se indica el caso que se desea resolver. Se especifica las etapas: lectura de datos, inicialización de variables y la ejecución de la simulación.
- **Parser:** permite el ingreso de la información referente a la configuración de los materiales, las condiciones de contorno y la configuración de la malla. Para cada una de estas existe una función mediante la cual se almacena la información en los diccionarios: d params, d bconds y d mesh respectivamente.

3.2.1.3 Inicialización de variables

En esta etapa, de acuerdo a la información de entrada, se declaran las variables a ser usadas, las cuales tienen un valor inicial de cero. Los módulos en los cuales se realiza la inicialización son:

- **Pyproblem:** se almacenan las funciones que se encargan de inicializar y ejecutar la simulación.
- **PySolidImplicit:** se inicializan las propiedades de los materiales tanto en Python como en Fortran.
- **FortranSolidImplicit:** se asignan las variables y los valores iniciales al módulo de Fortran, el cual se usa para realizar las operaciones con vectores.

3.2.2 Cálculo

Se realizan los cálculos iterativos de acuerdo al caso que se esté analizando. Los módulos que integran el software FSIPY realizan funciones tanto de inicialización como ejecución, en esta sección se presentan los módulos que realizan la simulación:

- **FortranSolidImplicit:** cálculo de la matriz de rigidez elemental y del vector de fuerzas elementales.
- **FortanKernel:** se ensambla la matriz de rigidez y el vector de fuerzas, que es llamado desde FortranSolidImplicit.
- **PyEquation:** invoca las funciones disponibles en Fortran y procesa los resultados que retornan a este. La función más importante es la solución del sistema $Ku = f$, mediante el cual se obtiene los desplazamientos en los nodos.

- **PyProblem:** engloba las funciones requeridas para la ejecución de la simulación de todos los casos que analiza FSIPY.
- **PySolidImplicit:** se especifica las funciones del módulo PyProblem que se utilizarán para la simulación.

En los módulos de Python y Fortran para referirse a los miembros del sistema $Ku = f$ se utiliza las denominaciones LHS (left-hand side) como la matriz de rigidez y RHS (right-hand side) para el vector de fuerzas.

3.2.3 Posprocesamiento

El posprocesamiento de resultados utiliza el formato de archivos VTK, que es compatible con ParaView. En el software FSIPY emplea la función “End timestep”, donde se guardan los valores de las variables de desplazamiento, velocidad y aceleración, como vectores.

3.3 Identificación de las modificaciones y requerimientos

3.3.1 Ingreso de variables

Los archivos de entrada de datos del software FSIPY incluyen información para la solución de ecuaciones en el tiempo, la cual se realiza mediante iteraciones hasta llegar a un punto de convergencia o un número máximo de iteraciones. La solución de un caso cuasi-estático se realiza en una sola iteración por lo que, en los parámetros numéricos se utiliza una tolerancia elevada y un cantidad máxima de iteraciones igual a “1”.

El software FSIPY cuenta con el análisis de las fuerzas másicas, es decir, peso propio del sólido. Sin embargo, para considerar fuerzas externas es necesario ingresar la información relacionada a la dirección y la magnitud de las mismas. En el caso de cargas distribuidas, se debe indicar los elementos de borde donde se aplican, mientras que, en el caso de fuerzas puntuales, se deben especificar los nodos correspondientes.

3.3.2 Transición a un caso cuasi-estático

El software FSIPY se basa en los criterios implementados en el libro de (Belytschko *et al.*, 2014), partiendo de la ecuación:

$$\mathbf{0} = \mathbf{r}(\mathbf{a}^{n+1}, t^{n+1}) = S_D \mathbf{M} \ddot{\mathbf{u}}^{n+1} + \mathbf{F}_{int}(\mathbf{a}^{n+1}, t^{n+1}) - \mathbf{F}_{ext}(\mathbf{a}^{n+1}, t^{n+1}) \quad (3.1)$$

En la ecuación (3.1), S_D actúa como switch entre el estado dinámico o en equilibrio.

$$S_D = \begin{cases} 0 & \text{Caso estático} \\ 1 & \text{Caso dinámico} \end{cases} \quad (3.2)$$

De acuerdo a la ecuación (3.2), para el caso cuasi-estático, la ecuación (3.1) se puede expresar como:

$$\mathbf{0} = \mathbf{r}(\mathbf{a}^{n+1}, t^{n+1}) = \mathbf{F}_{int}(\mathbf{a}^{n+1}, t^{n+1}) - \mathbf{F}_{ext}(\mathbf{a}^{n+1}, t^{n+1}) \quad (3.3)$$

Se puede observar que la ecuación (3.3) no depende de la aceleración ni de la masa, por lo que se encuentra en equilibrio. Donde $\mathbf{r}(\mathbf{a}, t)$ es el residuo, que converge si tiende a cero. El método utilizado para resolver esta ecuación es Newton-Raphson, que mediante aproximaciones tangenciales determina el punto de intersección con el eje de las abscisas, tal como se ilustra en la Figura 3.2.

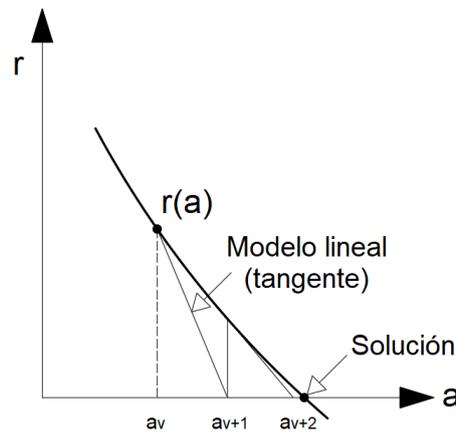


Figura 3.2: Método de Newton-Raphson (Belytschko *et al.*, 2014)

Las aproximaciones se realizan mediante la serie de Taylor, de tal forma que los desplazamientos en el paso $n + 1$ vienen dados por:

$$\mathbf{a}^{n+1} = \mathbf{a}^n - \left(\frac{\partial \mathbf{r}(\mathbf{a}^n)}{\partial \mathbf{a}} \right)^{-1} \mathbf{r}(\mathbf{a}^n) \quad (3.4)$$

La derivada del residuo con respecto al desplazamiento es la matriz de rigidez y el residuo es la diferencia entre las fuerzas externas e internas (Belytschko *et al.*, 2014). Por lo que la ecuación (3.4) puede ser reescrita de la forma:

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \mathbf{K}^{-1} \mathbf{f} \quad (3.5)$$

El software FSIPY es capaz de resolver la ecuación (3.5), por lo que, a partir de este punto se pueden llevar a cabo modificaciones e implementaciones. En este trabajo de titulación, se pretende analizar sólidos en dos dimensiones, ya sea en tensión plana o deformación plana. Para ello, se modificó la matriz de rigidez “ K ” y se disminuyó el número de iteraciones a uno para obtener los desplazamientos de acuerdo a:

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{f} \quad (3.6)$$

3.3.3 Implementaciones requeridas

En el software FSIPY se realiza análisis únicamente con fuerzas másicas, sin embargo, para poder simular la respuesta de un cuerpo sometido a fuerzas superficiales se requiere implementar el análisis de fuerzas distribuidas o puntuales sobre el cuerpo sólido.

El cálculo de esfuerzos no se encuentra programado, el software FSIPY se limita a la obtención de desplazamientos en los nodos. La importancia del cálculo de los esfuerzos radica en analizar las zonas con mayor fatiga, en las cuales se pueden dar las fracturas.

El posprocesamiento disponible permite visualizar los desplazamientos, la velocidad y la aceleración en los nodos que conforman el sólido. Al ser un estado cuasi-estático los valores de velocidad y aceleración son iguales a cero, por lo que no es necesario exportarlos. Sin embargo, se debe visualizar los valores de esfuerzos en los nodos.

3.4 Diseño de la solución

La adaptación del software FSIPY a las condiciones cuasi-estáticas requirió de modificaciones en Fortran para obtener una matriz de rigidez concorde con la teoría de la elasticidad en dos dimensiones. Además, se implementó funciones en el lenguaje de programación Python para calcular las fuerzas superficiales y esfuerzos. Para ello, se tomó como modelo base el caso denominado “SolidImplicit”, del cual se aprovechó

tanto el preprocesamiento como el posprocesamiento de la información.

Las modificaciones se realizaron en los módulos desarrollados en Python y Fortran. Para obtener la matriz de rigidez, se modificaron los módulos: “FortranSolidImplicit”, “PySolidImplicit” y “FortranKernel”. Las implementaciones de las fuerzas superficiales y esfuerzos se realizó en los módulos: “PySolidImplicit”, “PyProblem” y “PyEquation”.

3.4.1 Implementación de la matriz de rigidez

Modificación de la matriz constitutiva

El software FSIPY realiza el cálculo de la matriz de rigidez en el módulo “FortranSolid Implicit”, bajo la denominación de LHS. La función a tomar en cuenta se denomina “compute gauss lhs rhs”. Esta función realiza el producto de las derivadas cartesianas por la matriz constitutiva.

Se modificó la matriz constitutiva para adaptarla a condiciones cuasi-estáticas en estado bidimensional. En la ecuación (2.24) se muestra la relación entre el caso de tensión plana y deformación plana. La matriz \mathbb{D} se puede expresar en términos de “ \bar{E} y $\bar{\nu}$ ”. A partir de esta relación se implementó un condicional en el módulo “FortranSolidImplicit”, representado en la Figura 3.3.

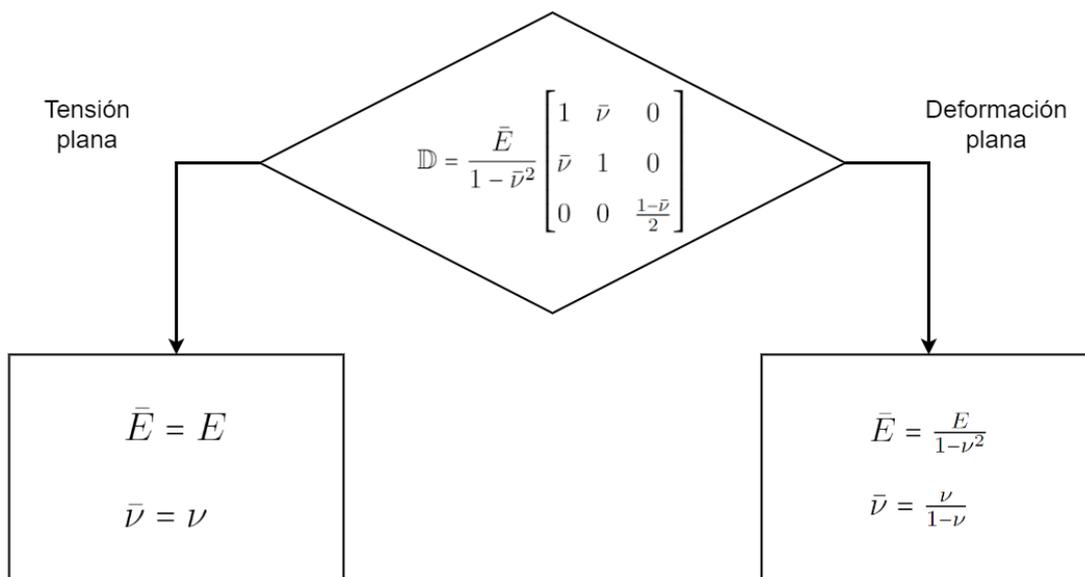


Figura 3.3: Flujograma condicional para teoría de elasticidad

El software FSIPY incorpora una contribución de la matriz de masa a la matriz de rigidez LHS, sin embargo, la ecuación (3.3) no la considera para el caso cuasi-estático. Por lo tanto, se debe suprimir esta contribución de tal forma que LHS elemental sea

igual a la matriz de rigidez evaluada en los puntos de integración de Gauss, la cual está dada por:

$$\mathbf{LHS}^{(e)} = \mathbf{B}^{(e)T} \mathbb{D} \mathbf{B}^{(e)} \quad (3.7)$$

Como se mencionó anteriormente el software FSIPY realiza el preproceso mediante el programa GMSH, el cual permite obtener las derivadas naturales y los jacobianos para cada elemento. Esta capacidad es aprovechada por Fortran, ya que en este módulo se obtienen las derivadas cartesianas para cada elemento. Aunque es necesario llevar a cabo la multiplicación de $\mathbf{J}^{-1} \mathbf{G}$ para obtener las derivadas cartesianas, no es necesario implementar alguna función relacionada con la generación de la malla.

El programa GMSH permite la evaluación de las funciones de forma en distintos puntos de Gauss. Se realizaron modificaciones en los puntos de integración, ya que según (Oñate, 2009), una aproximación de cuatro puntos es suficiente para elementos cuadriláteros.

Reducción de fuerzas externas

El software FSIPY denomina como RHS al vector de fuerzas resultante de la suma de fuerzas internas, externas, inerciales y de amortiguamiento Rayleigh. En condiciones cuasi-estáticas, solo se consideran las fuerzas externas e internas. El vector de fuerzas se calcula de acuerdo a:

$$\mathbf{RHS} = \mathbf{F}_{ext} - \mathbf{F}_{int} \quad (3.8)$$

Según la ecuación (2.49), las fuerzas internas en un sólido están relacionadas con una deformación inicial presente en el cuerpo. La ecuación constitutiva permite expresar los esfuerzos en términos de las deformaciones, pero es necesario ingresar los esfuerzos o deformaciones internas de todos los nodos del sólido, lo cual no es viable. Por lo tanto, para un elemento en particular, el vector RHS se limita a las fuerzas externas. En el estado inicial del software FSIPY, estas fuerzas externas solo incluyen el peso propio del sólido, tal como se muestra en la ecuación (3.9).

$$\mathbf{RHS}^{(e)} = \rho \mathbf{N}^{(e)} \mathbf{b} \quad (3.9)$$

De tal forma que la matriz de rigidez y el vector de fuerzas másicas se calculan de acuerdo al Algoritmo 1.

Algorithm 1 Implementación de la matriz de rigidez y vector de fuerzas másicas

```

1: function MATRIZRIGIDEZ_VECTORFUERZAS(b, x, y, w, gp, N, B, J)
2:   ke, fe ← 0                                ▷ Matriz de rigidez y vector de fuerzas elemental
3:   b ← Fuerzas_masicas
4:   Coordenadas ← [x; y]
5:   [w, gp] ← pesos_y_puntos_de_Gauss
6:   for i = 1 to numero_de_puntos_de_Gauss do
7:     for j = 1 to numero_de_puntos_de_Gauss do
8:        $\eta \leftarrow gp(i)$ 
9:        $\xi \leftarrow gp(j)$ 
10:       $\mathbf{N} \leftarrow \mathbf{N}(\xi, \eta)$ 
11:       $\mathbf{B} \leftarrow \mathbf{B}(\xi, \eta)$ 
12:       $detJ \leftarrow |\mathbf{J}(\xi, \eta)|$ 
13:       $\mathbf{ke} \leftarrow \mathbf{ke} + w(i) \cdot w(j) \cdot \mathbf{B}^T \cdot \mathbb{D} \cdot \mathbf{B} \cdot detJ$ 
14:       $\mathbf{fe} \leftarrow \mathbf{fe} + w(i) \cdot w(j) \cdot \mathbf{N} \cdot \mathbf{b} \cdot detJ$ 
15:    end for
16:  end for
17: end function
18: return ke, fe

```

3.4.2 Implementación de fuerzas superficiales

La implementación de fuerzas superficiales se llevó a cabo siguiendo los principios descritos en el libro de (Fish & Belytschko, 2007), y en el software creado en Matlab por Haim Waisman y Rensselaer (Algoritmo 3). A partir de la metodología empleada en Matlab, se adaptaron las funciones del software FSIPY para programar un código que incorpore fuerzas superficiales.

Modificación en el archivo Boun.in

Previo al análisis de fuerzas superficiales, distribuidas y puntuales, se debe ingresar los valores de magnitud y dirección en el archivo Boun.in, de acuerdo a:

Etiqueta *Magnitud en X* *Magnitud en Y*

Se debe especificar si se ingresan fuerzas distribuidas “Loads” o puntuales “Punctual”. En el caso de las fuerzas distribuidas, se ingresa la etiqueta del elemento sobre el cual se aplican, seguido de la magnitud de la carga en la dirección “X” y “Y”. Si se trata de tensión plana las unidades corresponden a [Pa], mientras que para deformación plana, se usa un espesor unitario por lo que esta carga se ingresa en fuerza por unidad de distancia, en el sistema internacional corresponde a Newtons por metro [N/m]. Por

otro lado, en las fuerzas puntuales se indica la etiqueta del nodo y la magnitud en Newtons [N].

En la Figura 3.4 se presenta una placa sometida a cargas distribuidas y fuerzas puntuales. La configuración del archivo boun.in que contiene las fuerzas superficiales se muestra en la Tabla 3.8, mientras que en la Tabla 3.9 se puede observar la configuración de las condiciones de contorno.

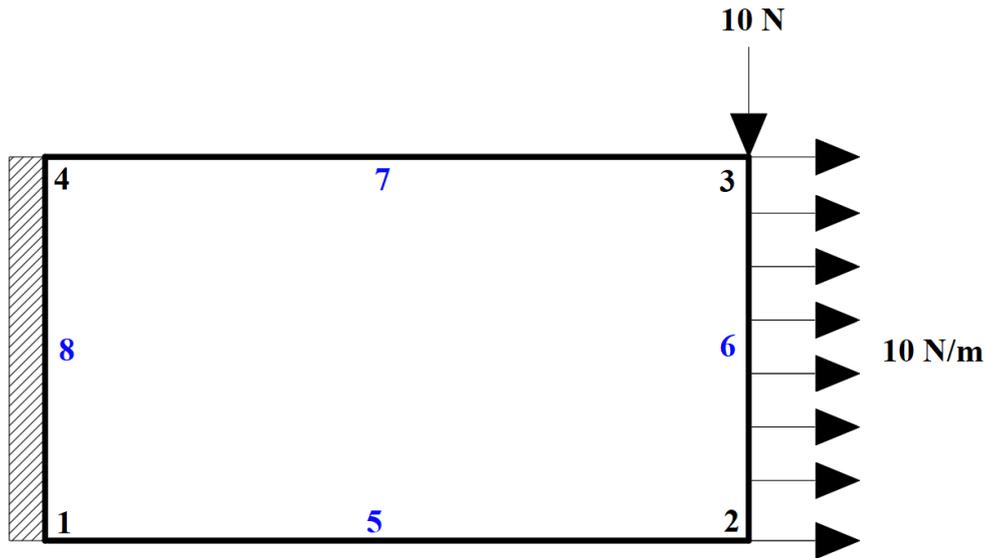


Figura 3.4: Ejemplo de elemento con fuerzas superficiales

Tabla 3.8: Configuración de las fuerzas superficiales de la Figura 3.4

Forces	Tag	X	Y
Loads	6	10.0	0.0
Punctual	3	0.0	-10.0

Tabla 3.9: Configuración de las condiciones de contorno de la Figura 3.4

Nodes		Lines	
Tag	DOF (XY)	Tag	DOF (XY)
1	11	5	00
2	00	6	00
3	00	7	00
4	11	8	11

Modificación en Parser.py

Se modificó el módulo Parser del software FSIPY para permitir la lectura de la información relacionada con el tipo de carga, su dirección y magnitud. Para lograr la lectura de las cargas superficiales, se aprovechó las funciones existentes y se agregó nuevas secciones al código para el almacenamiento en diccionarios. Esta implementación se realizó de acuerdo al algoritmo 2.

Algorithm 2 Lectura de fuerzas distribuidas y puntuales

```

1: procedure LEERFUERZAS(.boun.in)
2:   Loads  $\leftarrow [0, 0, 0]$ 
3:   Puntual  $\leftarrow [0, 0, 0]$ 
4:   Etiqueta  $\leftarrow$  LeerLínea ▷ Lectura Línea por Línea
5:   while Etiqueta  $\neq$  'EndForces' do
6:     if Etiqueta = 'Loads' then ▷ Fuerzas distribuidas
7:       Loads  $\leftarrow$  Elemento , Fuerza_X , Fuerza_Y
8:     end if
9:     if Etiqueta = 'Puntual' then ▷ Fuerzas puntuales
10:      Puntual  $\leftarrow$  Nodo , Fuerza_X , Fuerza_Y
11:    end if
12:  end while
13:  d_bforces  $\leftarrow$  {'Loads', 'Puntual'} ▷ Se guarda en el Diccionario
14: end procedure
15: return d_bforces

```

Traducción del algoritmo de Matlab a Python

Se aprovechó el software creado en Matlab por Haim Waisman y Rensselaer para el cálculo de las cargas distribuidas. Este software cuenta con subrutinas de preprocesamiento, cálculo y posprocesamiento para solucionar problemas en sólidos bidimensionales mediante tensión plana.

Luego de analizar las funciones que conforman el mencionado programa, se seleccionó aquellas que permiten el cálculo de las fuerzas superficiales. Las cuales se tradujeron del lenguaje de programación Matlab a Python, de acuerdo con el algoritmo 3.

Por consiguiente, se puede obtener el vector de fuerzas superficiales para cada elemento unidimensional analizado. Si se considera varios elementos de contorno, se generarán múltiples vectores. Los cuales se ensamblan en un vector de fuerzas global mediante una matriz de conectividad. Esta matriz se genera a partir de la geometría de la malla.

Algorithm 3 Cálculo del vector de fuerzas superficiales según MATLAB

```

1: function FUERZAS SUPERFICIALES(n_bce, x, y, w, gp, N)
2:   ft ← 0
3:   for i = 1 to numero_de_elementos_con_carga_actuante do
4:     n_bce ← Vector_de_fuerza
5:     x1 ← x(i)           y1 ← y(i)
6:     x2 ← x(i + 1)       y2 ← y(i + 1)
7:     leng ← √(x2 - x1)2 + (y2 - y1)2
8:     J ← leng/2
9:     [w, gp] ← pesos_y_puntos_de_Gauss
10:    for i = 1 to ngp do                                     ▷ Integración 1D
11:      ξ ← gp(i)
12:      N ← N(ξ)
13:      ft ← ft + w(i) · N · n_bce · J
14:    end for
15:  end for
16: end function
17: return ft

```

Adaptación a FSIPY

Se aprovechó las funciones disponibles en la API de GMSH, para importar las funciones de forma, Jacobiano, puntos de integración y pesos de Gauss de elementos unidimensionales. Estos valores se importan a Python como un vector de una sola fila, por lo que es necesario realizar un preproceso para convertirlos en matrices.

El vector elemental de fuerzas superficiales se debe ensamblar de acuerdo a la matriz de conectividad proveniente de GMSH. De tal forma que se obtiene el vector global de fuerzas superficiales, el cual se debe sumar al vector de fuerzas másicas.

Fuerzas puntuales

Las fuerzas puntuales al ser aplicadas directamente en los nodos se suman al vector de fuerzas superficiales obtenido anteriormente. Por lo que, el vector RHS queda conformado por fuerzas másicas, fuerzas distribuidas y fuerzas puntuales:

$$RHS = f_{\text{másicas}} + f_{\text{distribuidas}} + f_{\text{puntuales}} \quad (3.10)$$

Algoritmo desarrollado en FSIPY para fuerzas superficiales

La implementación de las fuerzas distribuidas y puntuales se basa en la lógica del algoritmo 3, a partir del cual se desarrolló el algoritmo 4 que realiza un preprocesamiento de las fuerzas superficiales y el algoritmo 5 realiza los cálculos del vector **RHS**.

Algorithm 4 Preprocesamiento de Fuerzas superficiales

```

1: procedure FUERZAS SUPERFICIALES(dimension, nodos)
2:    $Fuerzas\_d, Fuerzas\_p \leftarrow 0$ 
3:   for i to dimension do
4:      $L \leftarrow Loads[X, Y]$ 
5:      $Fuerzas\_d[i] \leftarrow [i, L(X), L(Y)]$ 
6:   end for
7:   for i to nodos do
8:      $L \leftarrow Punctual[X, Y]$ 
9:      $Fuerzas\_P[i] \leftarrow [i, L(X), L(Y)]$ 
10:  end for
11: end procedure
12: return  $Fuerzas\_d$  ,  $Fuerzas\_p$ 

```

Algorithm 5 Cálculo de Fuerzas superficiales

```

1: procedure CONTRIBUCIÓN(RHS,  $Fuerzas\_d$ ,  $Fuerzas\_p$ , N, J, w)
2:   for i to dimension do
3:      $fe \leftarrow 0$  ▷ Vector de fuerzas elementales
4:     for j to nodos do
5:       for k to numero_dimensiones do
6:          $fe[j] = fe[j] + N[j, k] \cdot Fuerzas\_d[i][j + 1]$ 
7:       end for
8:        $fe[j + 2] = fe[j]$ 
9:     end for
10:     $Fuerzas\_d \leftarrow Fuerzas\_d + fe \cdot detJ \cdot w$  ▷ Ensamblaje del vector
11:  end for
12:   $RHS = RHS + Fuerzas\_d + Fuerzas\_P$ 
13: end procedure
14: return RHS

```

3.4.3 Extrapolación de esfuerzos

La evaluación de los esfuerzos se realizó en los puntos de Gauss, los cuales tienen una mayor precisión que cualquier otro punto del elemento. Los esfuerzos son discontinuos entre los elementos adyacentes. Esto se debe a que en la formulación de elementos finitos solo se requiere la continuidad en los desplazamientos. Por lo que, los valores del esfuerzo nodal se obtienen mediante una extrapolación de los valores en los puntos de Gauss (Oñate, 2009).

Según (Oñate, 2009), existen varios métodos de extrapolación. Para este trabajo de titulación se optó por el método de extrapolación directa, el cual asume una relación

lineal entre dos puntos, tal como se muestra en la Figura 3.5.

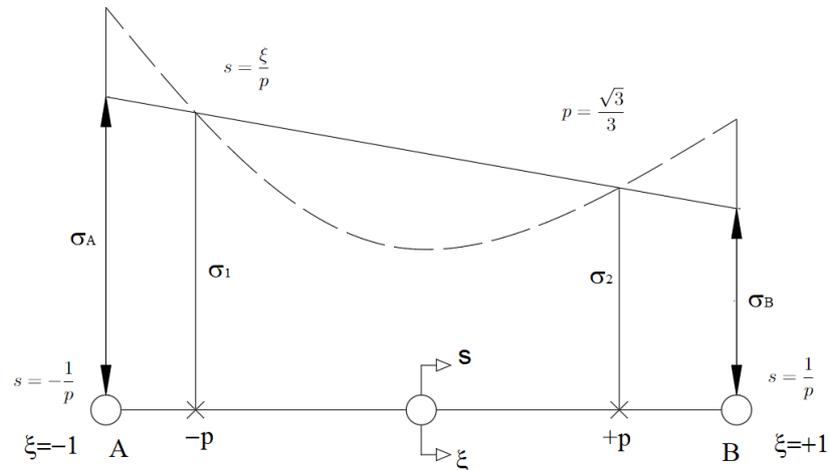


Figura 3.5: Relación lineal entre dos puntos (Oñate, 2009)

La extrapolación directa utiliza las funciones de forma evaluadas en los puntos $(1/s)$ y $(1/p)$. Debido a que no existe compatibilidad en los valores de esfuerzos en los nodos, se realizó un promedio aritmético de los resultados obtenidos desde distintos elementos para un mismo nodo. De tal forma que se obtiene un solo valor de esfuerzos para cada nodo. La implementación se basa en el algoritmo 6.

Algorithm 6 Get Stress

```

1: procedure GET STRESS( $a, \mathbb{D}, \mathbf{J}^{-1}, \mathbf{N}, \mathbf{B}$ )
2:    $a \leftarrow$  Desplazamiento en los nodos
3:    $\mathbb{D} \leftarrow$  Matriz constitutiva
4:    $\mathbf{J}^{-1}, \mathbf{N}, \mathbf{B} \leftarrow \mathbf{J}^{-1}(\xi, \eta), \mathbf{N}(\xi, \eta), \mathbf{B}(\xi, \eta)$ 
5:    $p \leftarrow \sqrt{3}$ 
6:    $\mathbf{N}_{\text{ext}} \leftarrow [[-p, -p, -0], [p, -p, 0], [p, p, 0], [-p, p, 0]]$  ▷ Función extrapolación
7:   for  $i$  to Numero_elementos_finitos do
8:      $\varepsilon_x^{(e)} \leftarrow a_x \cdot B_1$ 
9:      $\varepsilon_y^{(e)} \leftarrow a_y \cdot B_2$ 
10:     $\tau_{xy}^{(e)} \leftarrow a_x \cdot B_2 + a_y \cdot B_1$ 
11:     $\sigma^{(e)} \leftarrow \mathbb{D} \cdot \varepsilon^{(e)}$ 
12:     $\sigma_{\text{total}} \leftarrow \sigma_{\text{total}} + \sigma^{(e)} \cdot \mathbf{N}_{\text{ext}}$  ▷ Extrapolación
13:  end for
14:  for  $i$  to Numero_nodos do
15:     $\sigma_{\text{final}} = \bar{\sigma}_{\text{total}}$  ▷ Promedio de esfuerzos en el nodo
16:  end for
17: end procedure
18: return  $\bar{\sigma}_{\text{total}}$ 

```

3.4.4 Exportación de esfuerzos a ParaView

La representación gráfica de los desplazamientos y esfuerzos se llevó a cabo mediante la exportación de resultados en formato .vtu. Se utilizó la librería VTK en Python que permite la exportación de valores como vectores o escalares.

El software FSIPY exporta los resultados correspondientes a los desplazamientos en los nodos, considerados de tipo vectorial. Para los esfuerzos se aprovechó una función que permite exportar los resultados de tipo escalar a un archivo de extensión .vtu. Debido a que los resultados de σ_x , σ_y y τ_{xy} se deben representar en gráficos independientes.

3.5 Validación

La validación del nuevo módulo incorporado en el software FSIPY se basa en el contraste de resultados, tanto para el caso unidimensional que cuenta con una solución analítica, como para el caso bidimensional los cuales se comparan con los resultados obtenidos por SAP2000 y ABAQUS. Esta comparación se llevó a cabo exportando los resultados a una hoja de cálculo.

3.5.1 Validación con la solución teórica

En la validación de la solución analítica se evaluó el caso unidimensional desde dos perspectivas. La primera, no considera el efecto Poisson, lo que resultó en un alargamiento en la dirección axial. Para el cálculo de este se utilizó la ecuación (3.11).

$$\Delta L = \frac{F \cdot L}{E \cdot A} \quad (3.11)$$

La expresión (3.11) relaciona la deformación total ΔL con la fuerza aplicada F , la longitud de la barra L , el área de la sección A y el módulo de elasticidad E . La validez de esta ecuación esta condicionada por (Singer, 2006):

1. Carga axial.
2. Barra homogénea y de sección constante.
3. El esfuerzo debe ser inferior al límite de proporcionalidad.

Para el segundo caso se recurrió a la relación de Poisson, que considera los esfuerzos biaxiales. La relación de deformaciones unitarias viene dada de acuerdo a la ecuación

(3.12). Por lo tanto, los desplazamientos transversales se expresan en la ecuación

(3.13).

$$\nu = - \frac{\varepsilon_y}{\varepsilon_x} \quad (3.12)$$

$$\Delta H = - \nu \cdot H \cdot \frac{\Delta L}{L} \quad (3.13)$$

Donde,

ΔH : Desplazamiento vertical

H : Altura de la barra

Se llevó a cabo pruebas para analizar el comportamiento del sistema con valores de ν cercanos a los límites establecidos, dado que según (Singer, 2006), el coeficiente de Poisson puede tomar valores en el rango de 0 a 0.5.

Además, se realizó pruebas variando la carga aplicada sobre el sólido y se validó los resultados obtenidos con la solución analítica, calculada con las ecuaciones (3.11) y (3.13) para desplazamientos horizontales y verticales. En todos los casos se refinó la malla con el propósito de mejorar la precisión de los resultados, ya que al aumentar la cantidad de elementos se espera una mejor precisión en los resultados.

3.5.2 Validación con programas comerciales

En el proceso de benchmarking, se impuso las mismas condiciones y cantidad de elementos finitos para todos los casos en estudio, con el fin de analizar el rendimiento del software a medida que se aumenta la cantidad de elementos. Es importante destacar que el uso de un elevado número de elementos finitos implica el manejo de conjuntos de datos de gran tamaño, lo que hace inviable su comparación mediante tablas convencionales.

Los resultados obtenidos por FSIPY, ABAQUS y SAP2000 se compararon en los nodos ubicados en bordes de la figura, los cuales no se modifican a pesar del refinamiento de la malla. La validación se realizó mediante ejemplos que muestren las capacidades del software FSIPY, las cuales incluyen el análisis en tensión plana y deformación plana. Así como la incorporación de las fuerzas superficiales y másicas.

Capítulo 4

4. Resultados y discusión

El Método de Elementos Finitos (MEF) no proporciona la solución exacta del problema en cuestión, sino una solución basada en la idealización realizada. A medida que se refine la malla se conduce a una representación más cercana al problema real que se busca resolver.

Este capítulo presenta la ejecución del nuevo módulo del software FSIPY para la obtención de resultados y validación de los mismos. Se presenta tres ejemplos en los que se analizó los casos de tensión y deformación plana. En el primer ejercicio, se comparó la solución obtenida por FSIPY con la solución analítica para evaluar su comportamiento. En los ejercicios dos y tres, al carecer de soluciones analíticas, se comparó con los resultados obtenidos por SAP2000 y ABAQUS.

Se presentan los datos de entrada usados y los resultados de la simulación del software FSIPY para los distintos casos, el proceso detallado de compilación y ejecución se presenta en el Anexo C.

4.1 Caso 1: Solución analítica en tensión plana

Se formuló un caso para el análisis de una placa delgada de acero sujeta del lado izquierdo, mientras que en el lado derecho es sometida a una tensión constante de 200 MPa . El material del sólido presenta un módulo de elasticidad de $E = 210 \text{ GPa}$ y un coeficiente de Poisson de $\nu = 0.3$. La geometría del objeto corresponde a un rectángulo con dimensiones de 0.4 m en el eje horizontal y 0.3 m en el eje vertical, como se ilustra en la Figura 4.1.

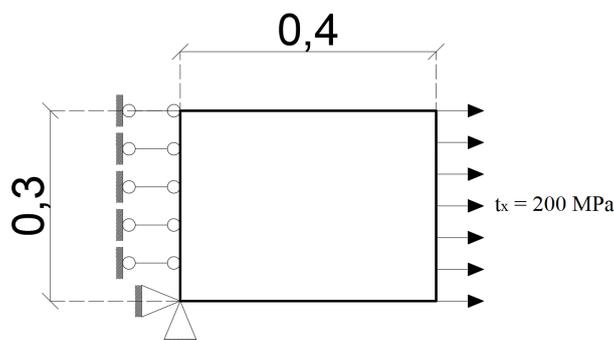


Figura 4.1: Caso 1: Placa delgada de acero en tensión plana.

Como se puede ver en la Figura 4.1, se trata de un caso unidimensional de geometría sencilla. El cual se encuentra articulado en su esquina inferior y con restricciones en el eje X a lo largo de su borde izquierdo.

Estas condiciones de contorno permiten que existan tensiones paralelas a la dirección de la carga aplicada, lo que hace posible comparar los resultados con la solución teórica. Con el fin de estudiar el comportamiento del software FSIPY, se llevó a cabo un análisis variando la cantidad de elementos, el coeficiente de Poisson y la carga aplicada, como se indica en la Tabla 4.1.

Tabla 4.1: Rango de valores de las variables a analizar.

Variable	Rango
Coeficiente de Poisson	0 - 0.49
Número de elementos	1 - 400
Carga aplicada	20 kPa - 200 MPa

4.1.1 Información de entrada

De acuerdo a la sección 3.2.1.1, para correr el ejemplo en el software FSIPY, se identifican las características del material y las consideraciones para su solución.

- **Parámetros físicos**

Tabla 4.2: Parámetros físicos para el caso 1.

Parámetros físicos		Observaciones
Physics	solid	El valor de densidad no es relevante debido que no se considera peso propio.
Material	isolinear	
Rho	0.0 kg/m^3	
Young	210.0e9 Pa	El valor de Poisson varía de acuerdo al rango 0 – 0.49
Poisson	0.3	
ndime	2	

- **Parámetros numéricos**

Tabla 4.3: Parámetros numéricos para el caso 1.

Parámetros numéricos		Observaciones
Volume quadrature	Gauss3	Se especifica Gauss3 debido a que se utiliza una integración de 4 puntos.
Surface quadrature	Gauss3	
Plane Elasticity	Stress	Stress indica que se resolverá por tensión plana.

- **Sources**

Tabla 4.4: Sources para el caso 1.

Sources			Observaciones
Parameters	0.0	0.0	Valores de ponderación para el peso propio en la dirección X y Y

4.1.2 Generación de la malla

La Figura 4.2a muestra las etiquetas de nodos y bordes correspondientes a las condiciones de contorno, mientras que en la Figura 4.2b contiene la malla y las etiquetas para las fuerzas superficiales.

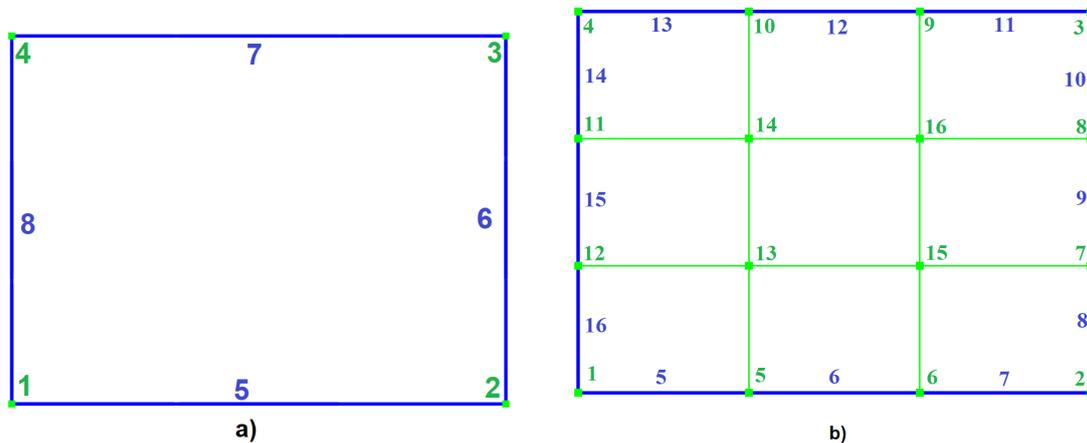


Figura 4.2: a) Etiquetas de nodos y bordes, b) Malla para nueve elementos. Caso 1

En el diseño de la malla, se deben colocar los elementos de tal forma que cada uno sea lo más cuadrado posible para garantizar su rendimiento. Según (Coronel, 2006), los elementos presentan errores a medida que su forma se distorsiona respecto a un cuadrado.

4.1.3 Condiciones de contorno

En el caso particular de una malla de nueve elementos, se tienen las consideraciones:

- **Condiciones de contorno:** se aplican en los nodos 1 y 4, así como en el borde lateral izquierdo, el cual corresponde al número 8 de la Figura 4.2a. Tal como se indica en la Tabla 4.5.

Tabla 4.5: Condiciones de contorno para el caso 1

Nodes		Lines	
Tag	DOF (XY)	Tag	DOF (XY)
1	11	5	00
2	00	6	00
3	00	7	00
4	10	8	10

- **Fuerzas superficiales:** se aplican sobre los bordes 8, 9 y 10 de la Figura 4.2b; en la dirección X , generando una tensión axial distribuida sobre el sólido. De acuerdo con la Tabla 4.6.

Tabla 4.6: Fuerzas superficiales para el caso 1

Forces	Tag	X	Y
Loads	8	200000000.0	0.0
	9	200000000.0	0.0
	10	200000000.0	0.0

4.1.4 Validación con la solución analítica

- **Desplazamientos:** La solución analítica se obtiene de acuerdo con las ecuaciones (3.11) y (3.13). Con las cuales se tiene los valores de desplazamiento indicados en la Tabla 4.7.

Los desplazamientos en X son iguales para cualquier coeficiente de Poisson, ya que la ecuación (3.11) no depende de este. Por otro lado, los desplazamientos en la dirección Y muestran una dependencia lineal con el coeficiente de Poisson, a medida que este valor aumenta, también lo hacen los desplazamientos en Y como se ilustra en la Figura 4.3.

Tabla 4.7: Desplazamientos para el caso 1

Coefficiente de Poisson	Desplazamiento en X [m]	Desplazamiento en Y [m]
0.00	3.81E-04	0.00E+00
0.10	3.81E-04	-2.86E-05
0.20	3.81E-04	-5.71E-05
0.30	3.81E-04	-8.57E-05
0.40	3.81E-04	-1.14E-04
0.49	3.81E-04	-1.40E-04

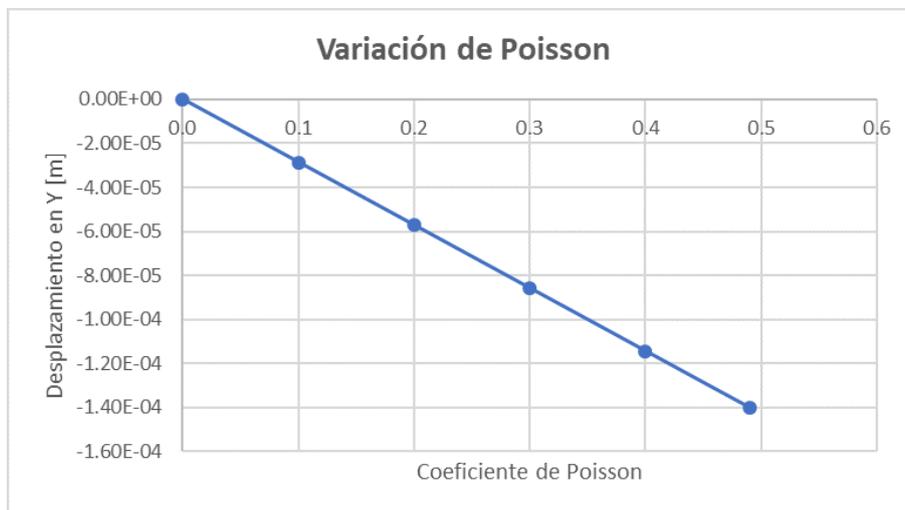


Figura 4.3: Variación del coeficiente de Poisson

En el software FSIPY se obtuvieron los desplazamientos en la dirección X y Y del nodo 3, tal como se muestra en las Tablas A.1 y A.2 respectivamente. Como se puede observar estos valores son exactos respecto a la solución teórica obtenida para distintos valores de Poisson.

Asimismo, es posible notar que la solución obtenida es precisa para cualquier número de elementos utilizados, por lo que no es necesario realizar un análisis de convergencia. Debido a las condiciones de contorno, el desplazamiento en el nodo 2 se da solo en la dirección X y coincide con el desplazamiento del nodo 3 en dicha dirección. En la Figura 4.4, se muestran los desplazamientos visualizados en ParaView para una malla de 400 elementos con un coeficiente de Poisson de 0.3.

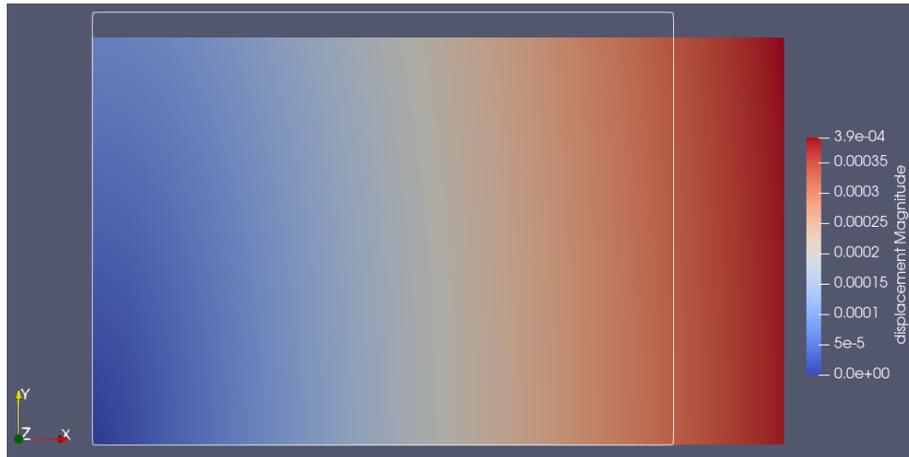


Figura 4.4: Desplazamientos obtenidos en FSIPY para el caso 1

En el análisis de la variación de la carga aplicada, se utilizó una malla conformada por un elemento y con un coeficiente de Poisson $\nu = 0.3$. Al aumentar la carga, se constató que se mantiene una relación de proporcionalidad directa entre esfuerzos y desplazamientos.

Los valores de desplazamiento en el nodo 3 se encuentran en la Tabla A.3, en la cual se puede observar que el error en entre los resultados del software FSIPY y la solución analítica es cero. En las Figuras 4.5 y 4.6 puede observar su comportamiento lineal.

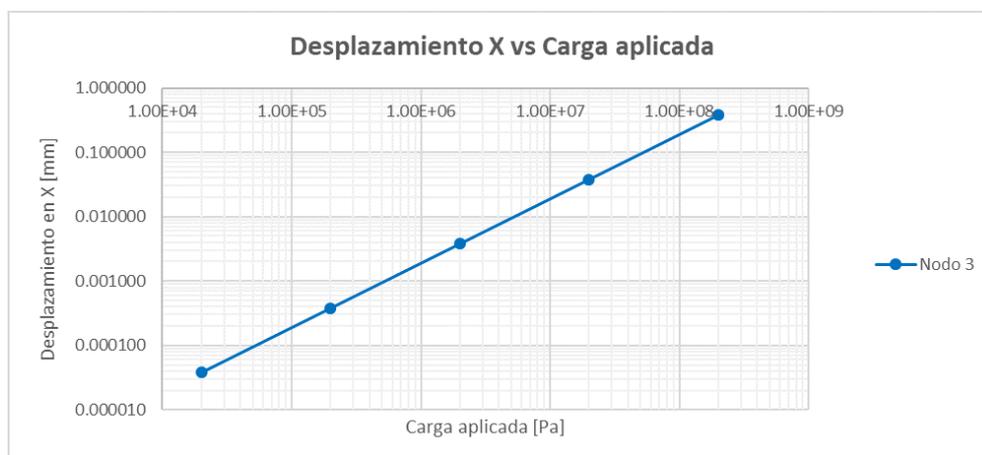


Figura 4.5: Desplazamientos en X vs Carga Aplicada

Se utilizó una escala logarítmica en ambos ejes, dado que la carga y los desplazamientos varían de manera exponencial. Para el caso de los desplazamientos en Y se cambió el signo debido a la escala logarítmica.

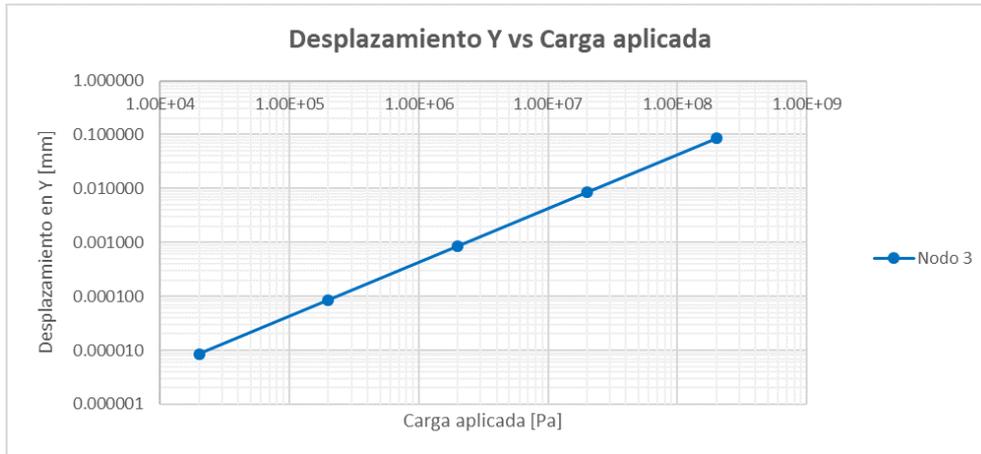


Figura 4.6: Desplazamientos en Y vs Carga Aplicada

- **Esfuerzos:** dado que se está analizando un caso unidimensional existen tensiones en la dirección paralela (dirección X), iguales a la tensión aplicada (200 MPa). Mientras que los esfuerzos normales en la dirección perpendicular y los esfuerzos cortantes son nulos, lo cual se pudo comprobar con los resultados obtenidos por FSIPY.

Se obtuvieron los resultados de esfuerzos para una malla de 1 y 400 elementos, los cuales se presentan en la Tabla 4.8. En la Figura 4.7 se muestran los esfuerzos en ParaView. Se observa que la distribución de los esfuerzos es constante a lo largo de toda la placa y es igual a la carga constante aplicada de 200 MPa .

Tabla 4.8: Esfuerzos en la dirección X obtenidos por FSIPY

Solución analítica	Esfuerzos en X	
	1 elemento	400 elementos
2.00E+08	2.00E+08	2.00E+08



Figura 4.7: Esfuerzos normales en la dirección X para el caso 1

4.2 Caso 2: Placa en tensión plana

Este caso se basa en el ejemplo presentado en la sección 9.2 del libro de (Fish & Belytschko, 2007). En dicho ejemplo se presenta una placa delgada de acero empotrada y sometida a una carga constante de 20 Pa. La placa cuenta con un módulo de elasticidad $E = 30 MPa$ y un coeficiente de Poisson $\nu = 0.3$. La geometría del sólido está compuesta por un cuadrilátero cuyas dimensiones se muestran en la Figura 4.8.

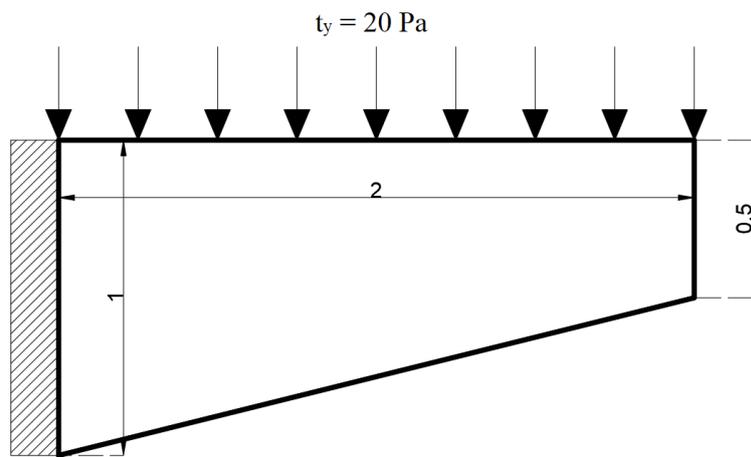


Figura 4.8: Caso 2: Placa delgada de acero en tensión plana

4.2.1 Información de entrada

De acuerdo a la sección 3.2.1.1 se identifican las características del material y las consideraciones para su solución.

• **Parámetros físicos**

Tabla 4.9: Parámetros físicos para el caso 2

Parámetros físicos		Observaciones
Physics	solid	El valor de densidad no es relevante debido a que no se considera peso propio.
Material	isolinear	
Rho	0.0 kg/m ³	
Young	30.0e6 Pa	
Poisson	0.3	
ndime	2	

• **Parámetros numéricos**

Tabla 4.10: Parámetros numéricos para el caso 2

Parámetros numéricos		Observaciones
Volume quadrature	Gauss3	Se utiliza Gauss3 debido a que se recurre a la integración de 4 puntos.
Surface quadrature	Gauss3	
Plane Elasticity	Stress	Stress indica que se resolverá por tensión plana.

• **Sources**

Tabla 4.11: Sources para el caso 2

Sources		Observaciones
Parameters	0.0 0.0	Valores de ponderación para el peso propio en la dirección X y Y

4.2.2 Generación de la malla

La Figura 4.9a muestra las etiquetas de nodos y bordes correspondientes a las condiciones de contorno, mientras que en la Figura 4.9b contiene la malla y las etiquetas para las fuerzas superficiales.

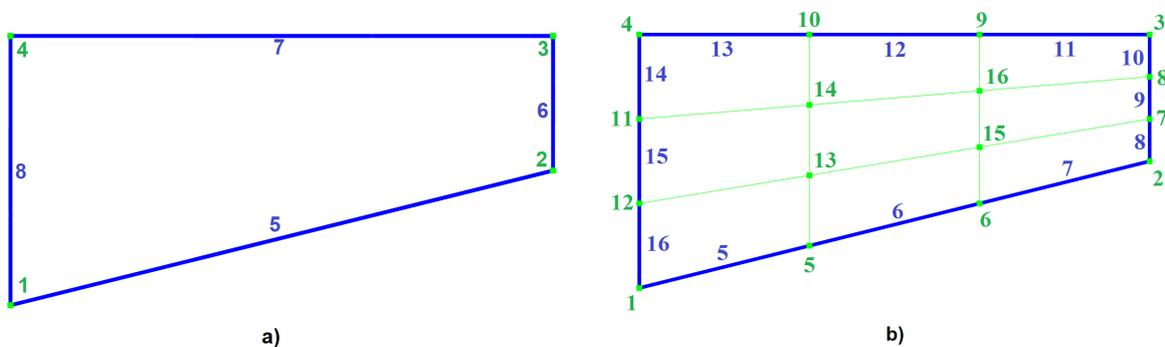


Figura 4.9: a) Etiquetas de nodos y bordes, b) Malla para nueve elementos. Caso 2

4.2.3 Condiciones de contorno

- **Condiciones de contorno:** de acuerdo a la Figura 4.9a, el caso 2 requiere de un empotramiento en el lado izquierdo. Esto implica restringir los movimientos de los nodos 1 y 4, así como del elemento 8, en las direcciones X y Y . Como se muestra en la Tabla 4.12.
- **Fuerzas superficiales:** la Tabla 4.13 presenta los valores de las cargas aplicadas sobre la malla de la Figura 4.9b.

Tabla 4.12: Condiciones de contorno para el caso 2

Nodes		Lines	
Tag	DOF (XY)	Tag	DOF (XY)
1	11	5	00
2	00	6	00
3	00	7	00
4	11	8	11

Tabla 4.13: Fuerzas superficiales para el caso 2

Forces	Tag	X	Y
Loads	11	0.0	-20.0
	12	0.0	-20.0
	13	0.0	-20.0

4.2.4 Convergencia de los resultados

- **Desplazamientos:** se escogió los nodos 2 y 3, ubicados en las esquinas de la figura, ya que presentan los mayores desplazamientos. En la Tabla A.4 se observan los desplazamientos obtenidos en estos nodos, para su análisis de convergencia.

Se aprecia que a medida que se refina la malla, se da una tendencia hacia una solución en los resultados. Dado que la cantidad de elementos se aumentó de manera exponencial, se optó por presentar las gráficas en una escala semi-logarítmica.

En cuanto a los desplazamientos en el eje X , los resultados del nodo 2 y 3 tienen direcciones opuestas. Dado que la placa está sometida a una carga vertical y se encuentra empotrado su lado izquierdo, se da un alargamiento en el nodo 3 (borde superior) mientras que en el nodo 2 se presenta un acortamiento. Con respecto a los desplazamientos verticales, estos son similares y en la misma dirección que se aplica la carga. Los desplazamientos horizontales y verticales se muestran en las Figuras 4.10 y 4.11, respectivamente.

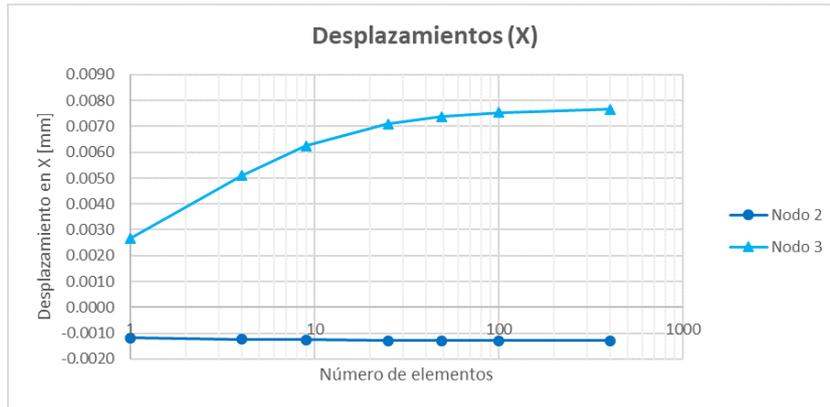


Figura 4.10: Desplazamientos en la dirección X

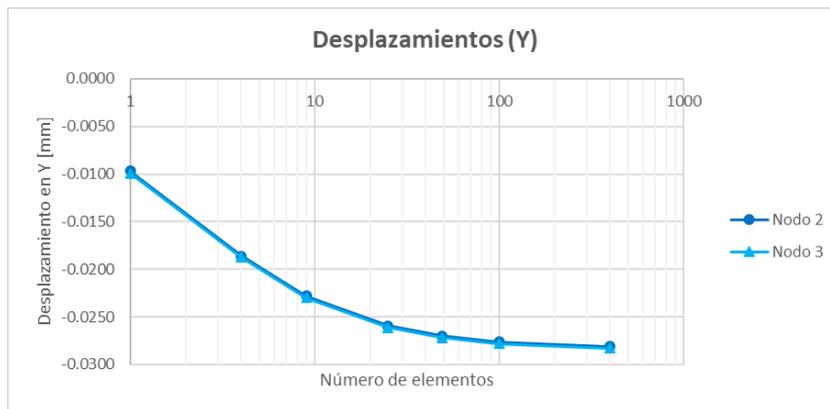


Figura 4.11: Desplazamientos en la dirección Y

Los desplazamientos verticales y horizontales muestran una tendencia de resultados conforme aumenta el número de elementos utilizados. La Figura 4.12 muestra los resultados de los desplazamientos de cada uno de los nodos en una malla de 400 elementos, exhibidos en ParaView. Se utilizó una escala de 10.000 para facilitar la observación de los desplazamientos. Se puede apreciar que en el extremo libre de la placa se produce una deformación y una rotación con respecto al eje vertical.

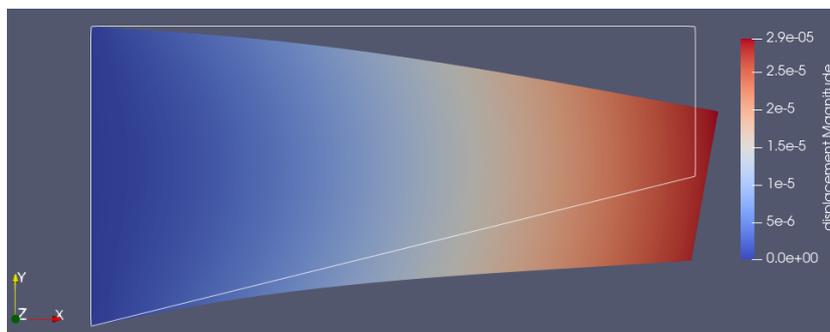


Figura 4.12: Desplazamientos obtenidos en FSIPY para el caso 2

- **Esfuerzos:** los resultados de esfuerzos normales en la dirección X y Y , así como los esfuerzos tangenciales en la dirección XY se detallan en el Anexo B.1.

En particular, la Tabla B.1 muestra los resultados de los esfuerzos normales en la dirección X . Debido a que la carga se aplica en la dirección vertical, se escogió los extremos libres, ya que estos no cuentan con restricciones en sus grados de libertad. Los esfuerzos normales en X tienden a cero, como se muestra en la Figura 4.13.

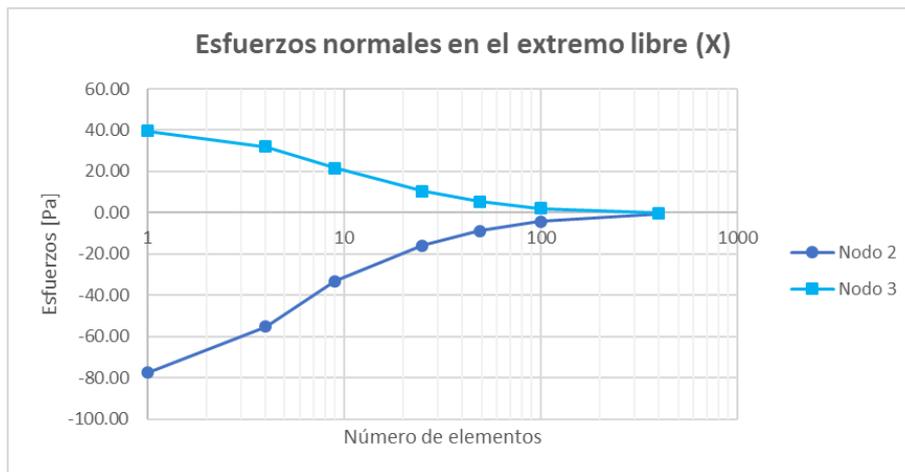


Figura 4.13: Esfuerzos normales en los extremos libres

Respecto a los extremos que se encuentran empotrados, se puede apreciar un aumento en la magnitud de la tensión como consecuencia de su condición de empotramiento. El nodo 1 se encuentra bajo compresión y el nodo 4 en tensión, como se muestra en la Figura 4.14.

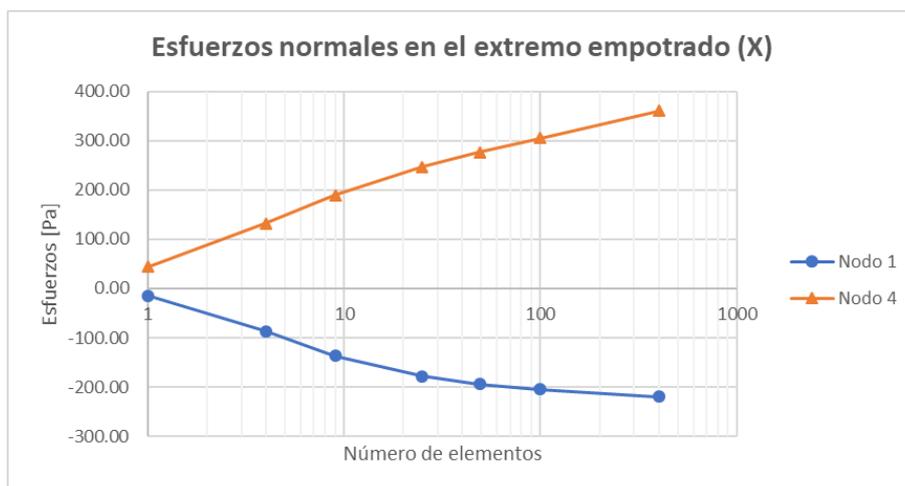


Figura 4.14: Esfuerzos normales en los extremos empotrados

Las Figuras 4.15 y 4.16 muestran los resultados visualizados en ParaView, correspondientes al estado tensional, obtenidos a partir de una malla conformada por 400 elementos.

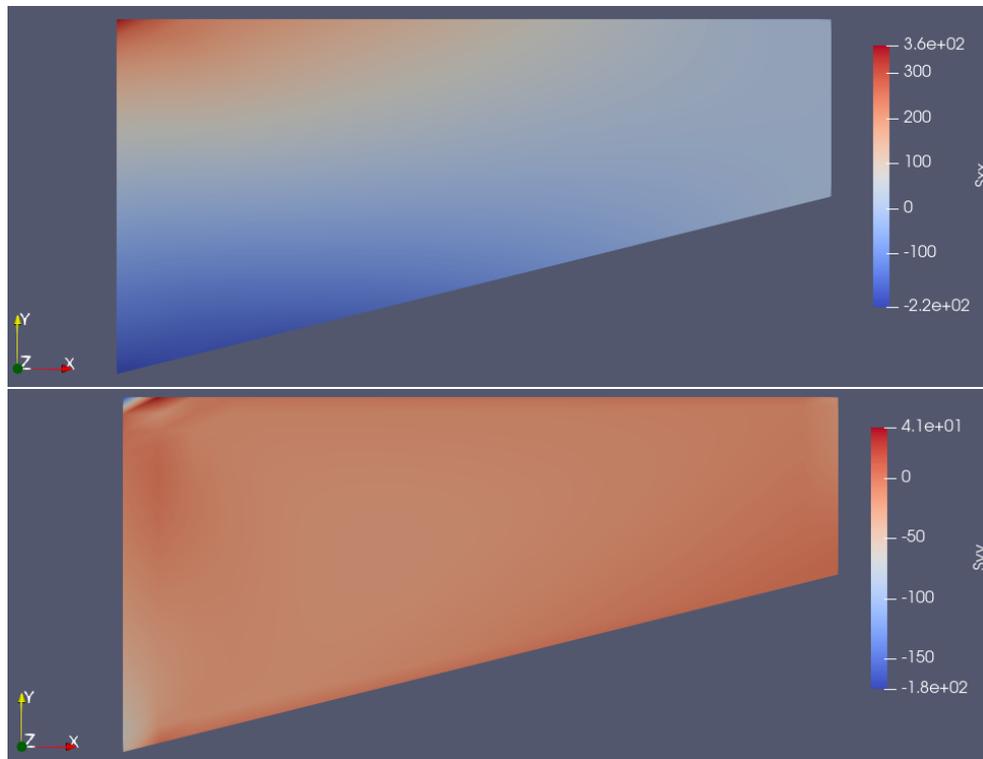


Figura 4.15: Esfuerzos normales en la dirección X y Y para el caso 2

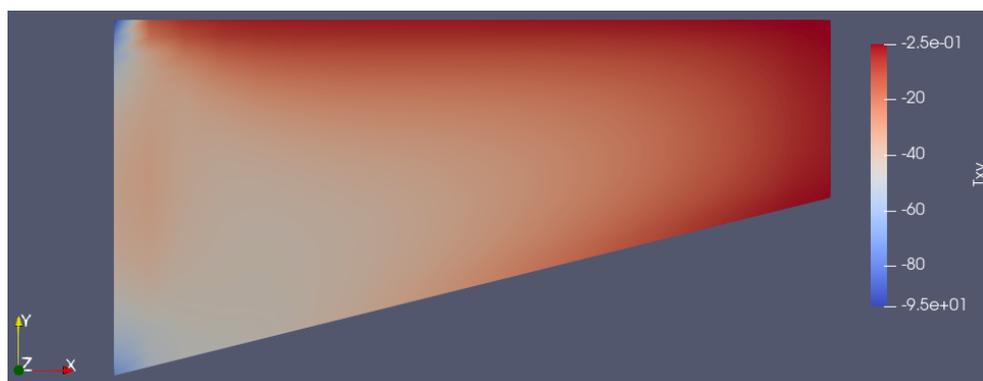


Figura 4.16: Esfuerzos tangenciales para el caso 2

4.2.5 Comparación

- **Desplazamientos:** en la tabla 4.14 se muestran las diferencias entre los desplazamientos en el nodo 2 obtenidos por los programas FSIPY, ABAQUS y SAP2000 con una malla de 400 elementos. Los desplazamientos horizontales de FSIPY y SAP2000 convergen a 0.0077 mm, mientras que los resultados obtenidos por ABAQUS presentan una diferencia porcentual de 23% con respecto a FSIPY. Por otro lado, los resultados de los desplazamientos horizontales presentan diferencias porcentuales del orden de 2%.

Tabla 4.14: Comparación desplazamientos Nodo 2 para el caso 2

Comparación Nodo 2					
Dirección	FSIPY [mm]	ABAQUS [mm]	SAP2000 [mm]	FSIPY/ABAQUS [%]	FSIPY/SAP2000 [%]
X	0.00766	0.01000	0.00771	23.39	0.64
Y	-0.02833	-0.02900	-0.02800	2.32	1.16

Los desplazamientos horizontales y verticales en el nodo 2 se presentan en las Figuras 4.17 y 4.18 respectivamente. Los desplazamientos verticales evidencian una convergencia uniforme, mientras que los horizontales presentan diferencias respecto a los resultados obtenidos por ABAQUS.

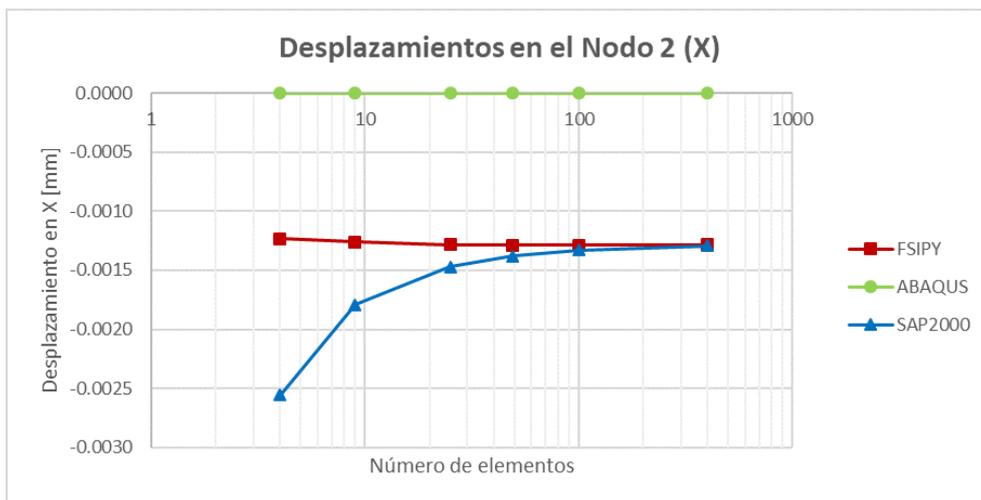


Figura 4.17: Desplazamiento en X del nodo 2

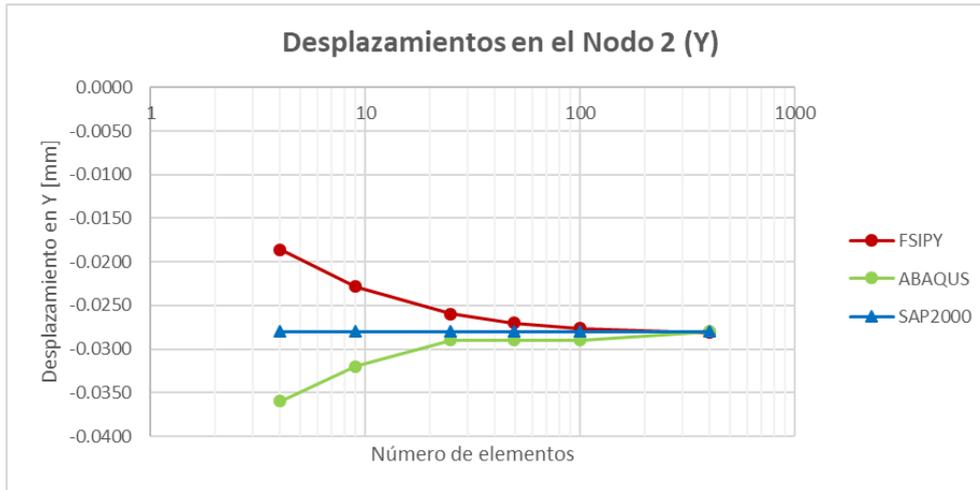


Figura 4.18: Desplazamiento en Y del nodo 2

Los desplazamientos correspondientes al nodo 3 se muestran en la Tabla 4.15, al igual que en el nodo 2 existe una convergencia uniforme en los desplazamientos verticales. Dado que el desplazamiento en X obtenido por ABAQUS es cero, no se pudo obtener la diferencia relativa entre FSIPY y ABAQUS. Sin embargo, los resultados obtenidos por SAP2000 tienen una diferencia mínima respecto a FSIPY.

Tabla 4.15: Comparación desplazamientos Nodo 3 para el caso 2

Comparación Nodo 3					
Dirección	FSIPY [mm]	ABAQUS [mm]	SAP2000 [mm]	FSIPY/ABAQUS [%]	FSIPY/SAP2000 [%]
X	-0.00128	0.00000	-0.00129	-	0.79
Y	-0.02814	-0.02800	-0.02800	0.48	0.48

Las Figuras 4.19 y 4.20 representan los desplazamientos del nodo 3.

Es importante mencionar que, tanto el software FSIPY como SAP2000 muestran resultados similares para el caso de tensión plana. Por lo que, los resultados de desplazamientos se consideran válidos. Los valores de los desplazamientos se detallan en el Anexo A.2.

- **Esfuerzos:** en la Figura 4.21, los esfuerzos en la dirección X muestran una convergencia a cero. Esto se debe a que la carga se aplica de manera vertical y el nodo 3 al no contar con restricciones al movimiento, no genera reacciones.

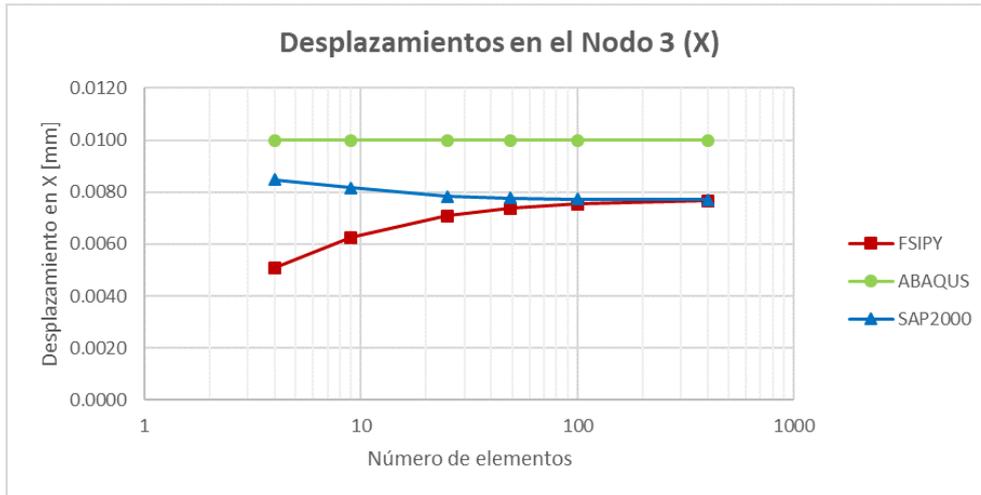


Figura 4.19: Desplazamiento en X del nodo 3

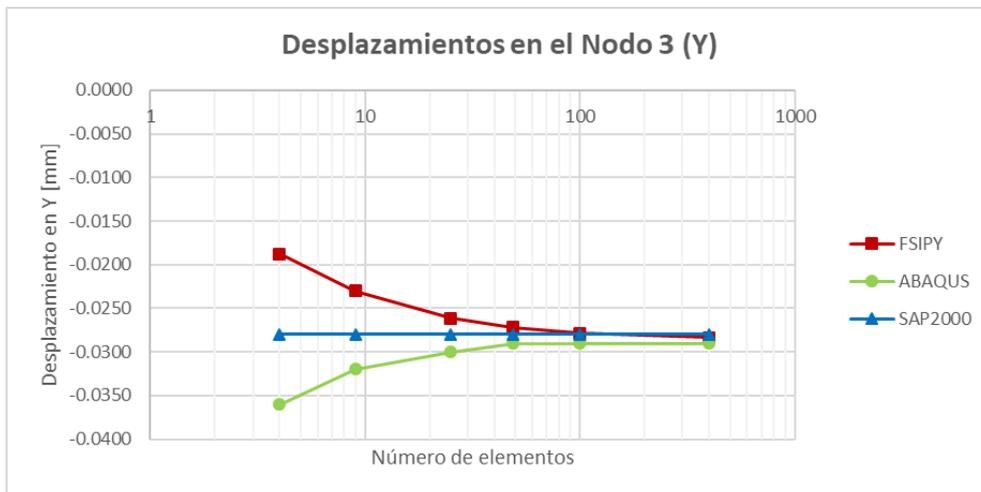


Figura 4.20: Desplazamiento en Y del nodo 3

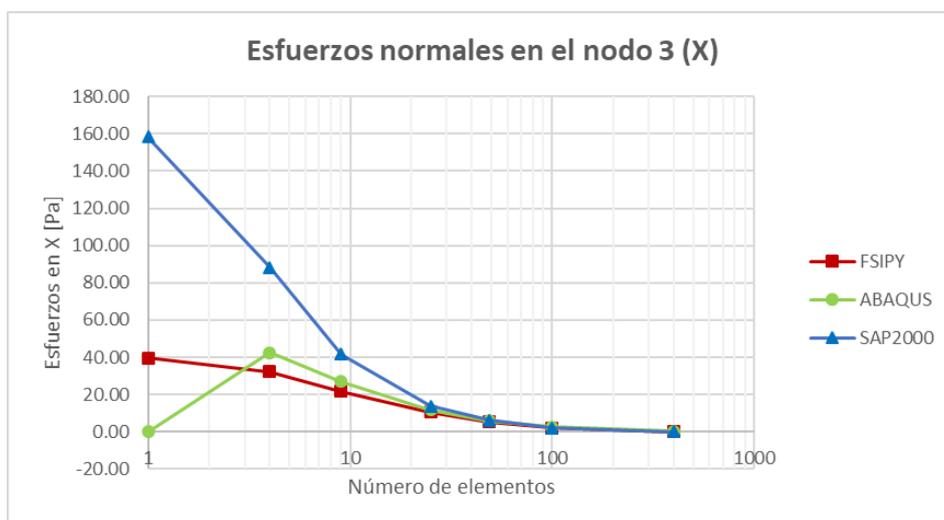


Figura 4.21: Esfuerzos normales en X del nodo 3

Con respecto a la Figura 4.22, los esfuerzos en la dirección Y convergen a un valor de $-20 Pa$, el cual coincide con el valor de la carga aplicada sobre la placa. Los resultados del software FSIPY para pocos elementos, difieren respecto a los obtenidos por ABAQUS y SAP2000. Sin embargo, al aumentar la cantidad de elementos se nota que existe similitud en las soluciones.

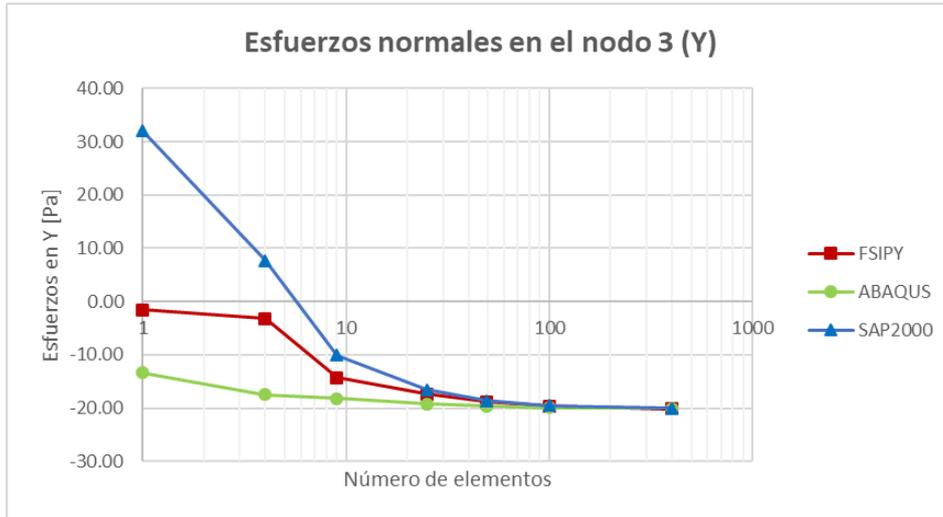


Figura 4.22: Esfuerzos normales en Y del nodo 3

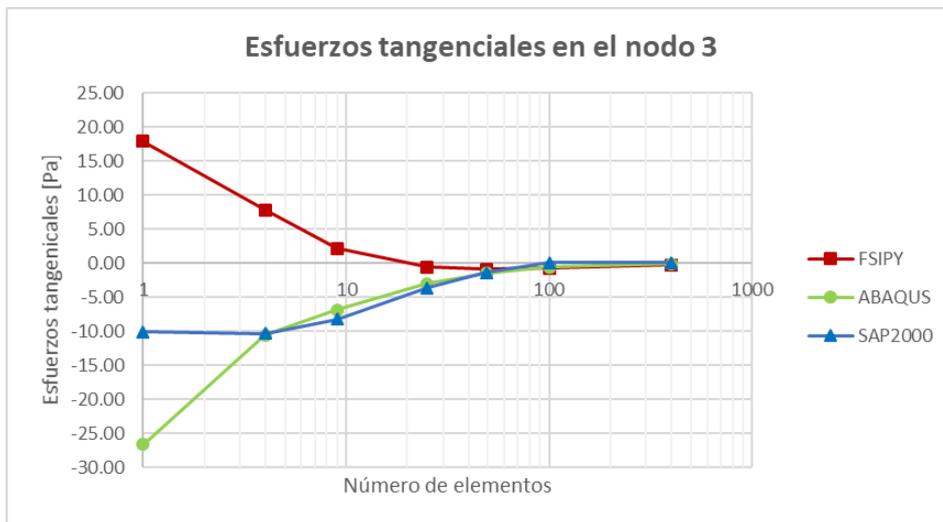


Figura 4.23: Esfuerzos tangenciales del nodo 3

En cuanto a los esfuerzos tangenciales, se puede observar en la Figura 4.23 una tendencia hacia un valor de cero. En términos generales, aunque el software FSIPY muestra diferencias con SAP2000 y ABAQUS para un número reducido de elementos, a medida que estos aumentan, los resultados convergen hacia valores similares, por lo que se consideran aceptables. Los resultados de los esfuerzos en los nodos 1, 2, 3 y 4 se encuentran en el Anexo B.1.

4.3 Caso 3: Muro de gravedad

Se considera un muro de gravedad, cuyas dimensiones se ven en la Figura 4.24, que se encuentra sometido a una carga constante de 98 kPa a lo largo de su extremo izquierdo. Este caso se resuelve en dos dimensiones en términos de deformación plana. El muro cuenta con un módulo de elasticidad de 20 GPa , un coeficiente de Poisson de 0.3 y un peso específico de 24 kN/m^3 , correspondiente al hormigón.

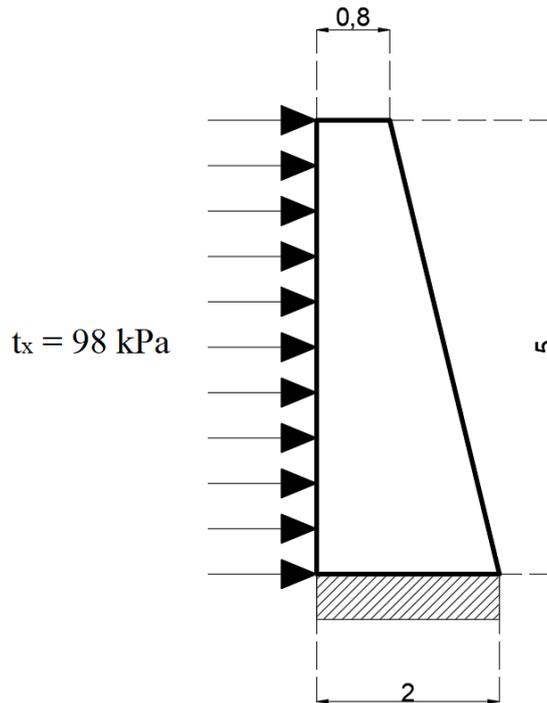


Figura 4.24: Caso 3: Muro de gravedad

4.3.1 Información de entrada

• **Parámetros físicos**

Tabla 4.16: Parámetros físicos para el caso 3

Parámetros físicos		Observaciones
Physics	solid	En este caso se considera peso propio, con una densidad de 2400 kg/m^3 .
Material	isolinear	
Rho	2400.0 kg/m^3	
Young	$20.0e9 \text{ Pa}$	
Poisson	0.3	
ndime	2	

• **Parámetros numéricos**

Tabla 4.17: Parámetros numéricos para el caso 3

Parámetros numéricos		Observaciones
Volume quadrature	Gauss3	Se utiliza Gauss3 debido a que se recurre a la integración de 4 puntos.
Surface quadrature	Gauss3	
Plane Elasticity	Strain	Strain indica que se resolverá por deformación plana.

• **Sources**

Tabla 4.18: Sources para el caso 3

Sources		Observaciones
Parameters	0.0 -9.8	Valores de ponderación para el peso propio en la dirección X y Y . Al considerar peso propio se debe indicar la magnitud y dirección, para el caso -9.8 en Y .

4.3.2 Generación de la malla

La Figura 4.25a muestra las etiquetas de nodos y bordes correspondientes a las condiciones de contorno. Dado que la dimensión de la base es menor a la altura, se utilizó una relación 1H:3V para el mallado, con el fin de que los elementos se asemejen a cuadrados. En el caso de una malla de 27 elementos se tiene la Figura 4.25b.

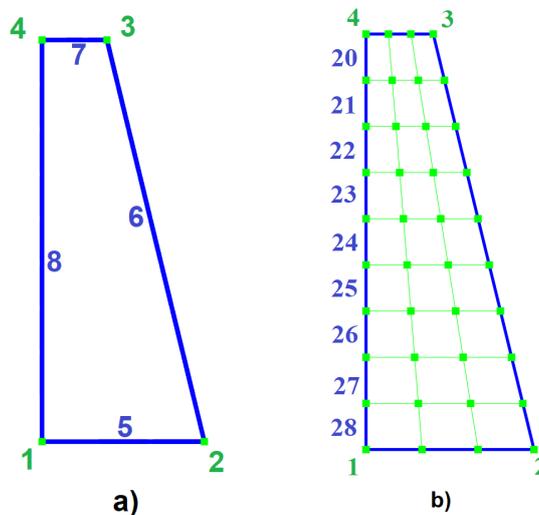


Figura 4.25: Etiquetas de nodos y bordes para el caso 3

4.3.3 Condiciones de contorno

- **Condiciones de contorno:** la Tabla 4.19 indica que existen restricciones en los grados de libertad X y Y en la base de la Figura 4.25a, la cual representa un muro empotrado en su base.

Tabla 4.19: Condiciones de contorno para el caso 3

Nodes		Lines	
Tag	DOF (XY)	Tag	DOF (XY)
1	11	5	11
2	11	6	00
3	00	7	00
4	00	8	00

- **Fuerzas superficiales:** la Tabla 4.20 muestra la aplicación de las cargas sobre la pared izquierda del muro analizado. Se considera una fuerza lateral constante de 98 kPa aplicada desde el borde 20 al 28 de la Figura 4.25b.

Tabla 4.20: Fuerzas superficiales para el caso 3

Forces	Tag	X	Y
Loads	20	98000.0	0.0
	21	98000.0	0.0
	22	98000.0	0.0
	23	98000.0	0.0
	24	98000.0	0.0
	25	98000.0	0.0
	26	98000.0	0.0
	27	98000.0	0.0
	28	98000.0	0.0

4.3.4 Convergencia de resultados

- **Desplazamientos:** los resultados de desplazamiento se muestran en la Tabla A.7, estos corresponden a los desplazamientos en los nodos 3 y 4, los cuales son los bordes superiores del muro. En los nodos 1 y 2 no se presentan los resultados ya que se consideran empotrados.

Los resultados corresponden al análisis en deformación plana, ya que se considera que el espesor es mayor que las otras dimensiones. Este tipo de análisis produce valores de deformación menores que los que se obtendrían en un análisis de tensión plana, ya que la estructura es menos propensa a deformarse. En la Figura 4.26 se presenta los desplazamientos en la dirección X para varios elementos. El comportamiento es similar en los nodos 3 y 4 ya que estos no se encuentran restringidos y la carga aplicada es paralela a esta dirección.

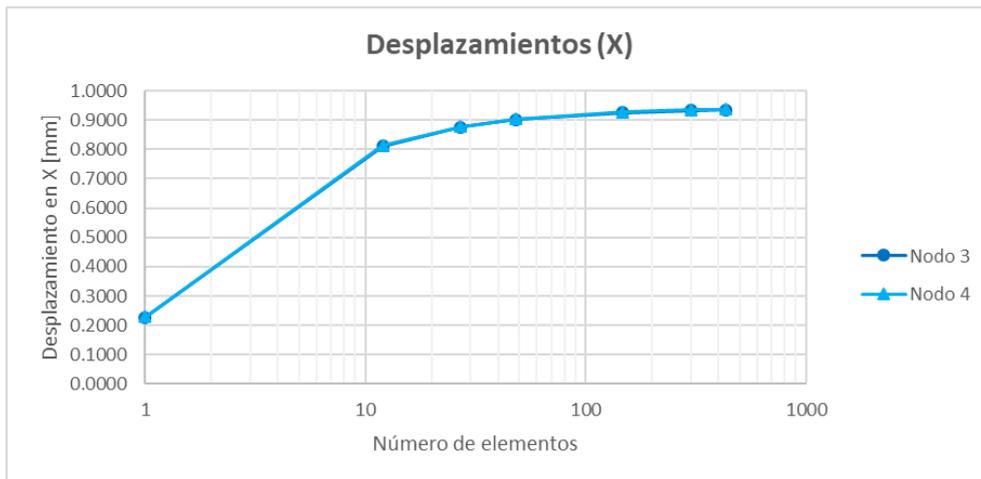


Figura 4.26: Desplazamientos en la dirección X

Los desplazamientos en la dirección Y se muestran en Figura 4.27. Estos se ven influenciados por el peso propio del muro, lo que provoca que los resultados sean distintos en los nodos 3 y 4. Los cuales muestran una tendencia a medida que se aumenta el número de elementos.

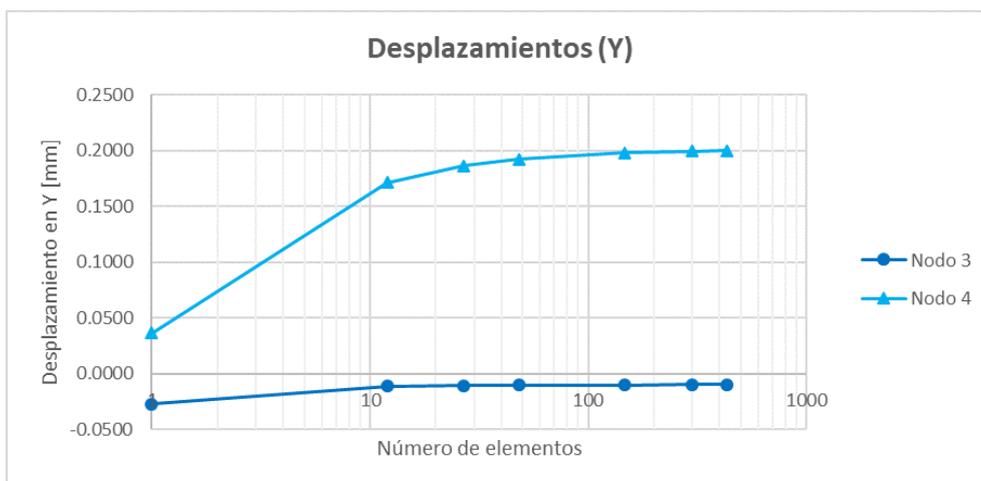


Figura 4.27: Desplazamientos en la dirección Y

En el nodo 4 se presenta un desplazamiento horizontal y una elevación debido a la carga lateral, por otro lado, el nodo 3 se desplaza hacia abajo por el peso propio de la misma. Estos resultados se graficaron en ParaView para un total de 432 elementos, como se muestra en la Figura 4.28 . Se ajustó la escala de visualización en 1000 veces, ya que los desplazamientos están en el orden de 0.2 milímetros.

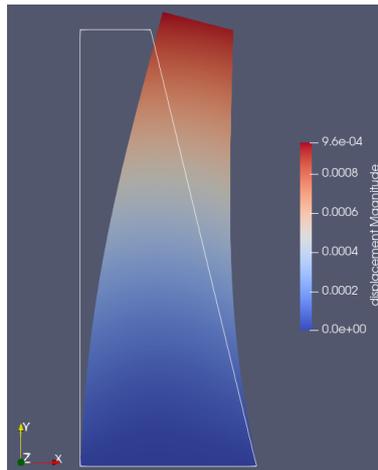


Figura 4.28: Desplazamientos obtenidos en FSIPY para el caso 3

- **Esfuerzos:** los esfuerzos en la dirección X en las esquinas de la Figura 4.24 se muestran en la Tabla B.10. Existe una convergencia de estos esfuerzos en cada nodo, concentrándose en el nodo 1 para tensión y en el nodo 2 para compresión. En el Anexo B.4 se presentan los resultados correspondientes a los esfuerzos normales y los esfuerzos tangenciales, mientras que las Figuras 4.29 y 4.30 presentan los resultados en ParaView para una malla de 432 elementos.

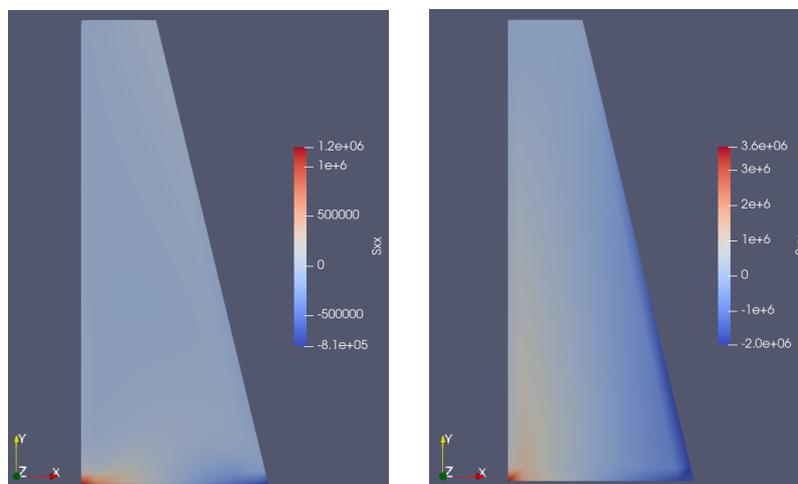


Figura 4.29: Esfuerzos normales en la dirección X y Y para el caso 3

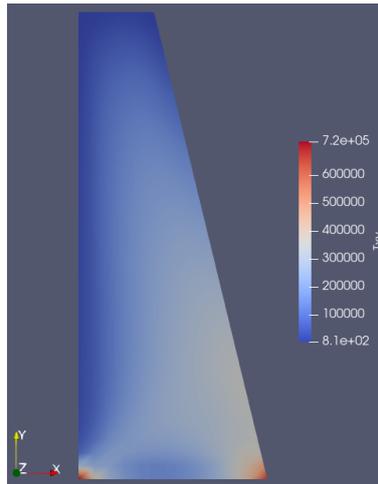


Figura 4.30: Esfuerzos tangenciales para el caso 3

4.3.5 Comparación

- **Desplazamientos:** en la Tabla 4.21 se presenta los desplazamientos horizontales y verticales del nodo 3. Se puede observar que las diferencias relativas de FSIPY respecto a SAP2000 son menores a las obtenidas con ABAQUS.

Tabla 4.21: Comparación desplazamientos Nodo 3 para el caso 3

Comparación Nodo 3					
Dirección	FSIPY [mm]	ABAQUS [mm]	SAP2000 [mm]	FSIPY/ABAQUS [%]	FSIPY/SAP2000 [%]
X	0.935015	0.945	0.939	1.06	0.42
Y	-0.00975	-0.010	-0.010	2.53	0.58

Los desplazamientos horizontales convergen a un valor de 0.94 mm y los verticales a un valor de -0.01 mm . Este comportamiento se puede visualizar en las Figuras 4.31 y 4.32. Los valores para un solo elemento no se consideran debido a que, al tratarse de un cuadrilátero irregular, los resultados varían significativamente.

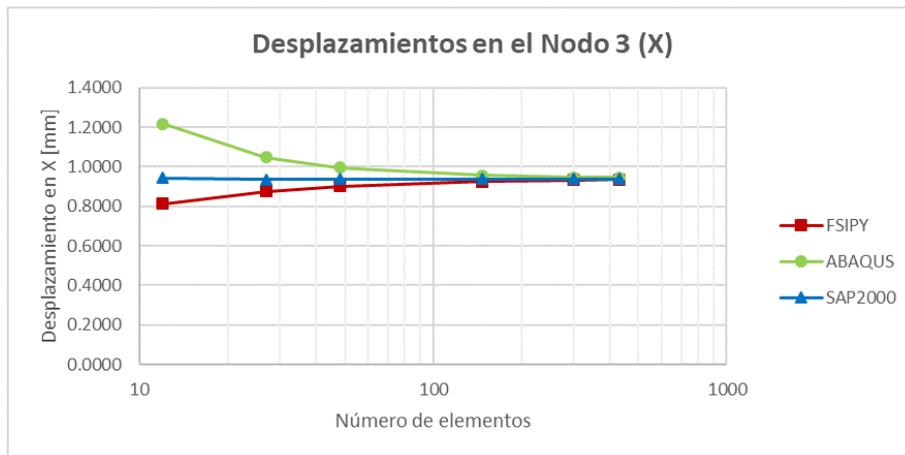


Figura 4.31: Desplazamientos en X del nodo 3

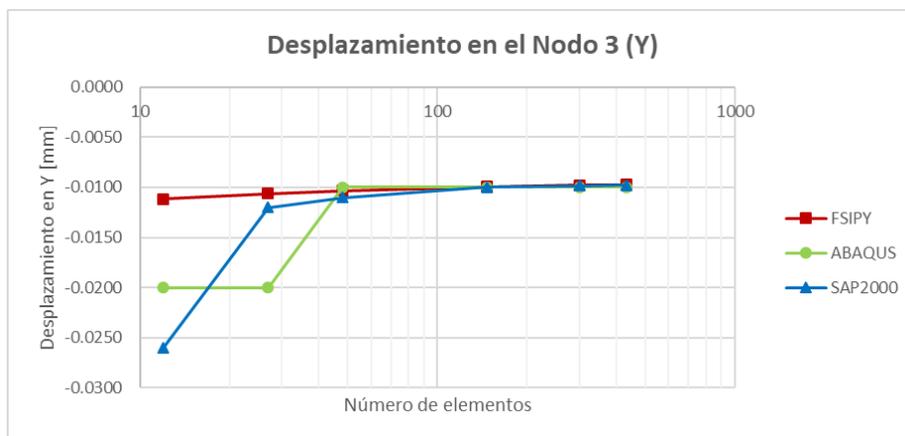


Figura 4.32: Desplazamientos en Y del nodo 3

La Tabla 4.22 y las Figuras 4.33 y 4.34 muestran la convergencia de los desplazamientos en el nodo 4. Los desplazamientos horizontales convergen a 0.94 mm y los verticales a 0.2 mm. Se puede observar que los desplazamientos obtenidos en FSIPY y los programas comerciales tienen diferencias relativas en el orden del 1%.

Tabla 4.22: Comparación desplazamientos Nodo 4 para el caso 3

Comparación Nodo 4					
Dirección	FSIPY [mm]	ABAQUS [mm]	SAP2000 [mm]	FSIPY/ABAQUS [%]	FSIPY/SAP2000 [%]
X	0.93705	0.94708	0.941	1.06	0.42
Y	0.19986	0.20000	0.201	0.07	0.57

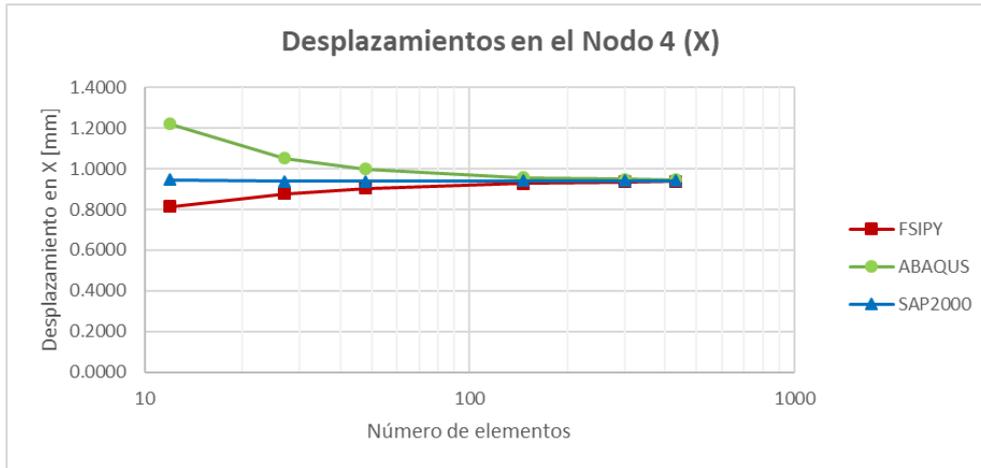


Figura 4.33: Desplazamientos en X del nodo 4

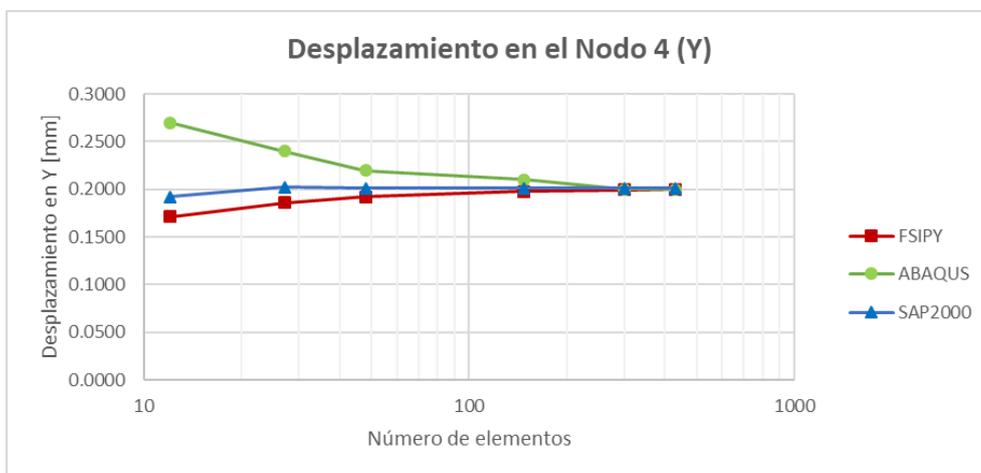


Figura 4.34: Desplazamientos en Y del nodo 4

- **Esfuerzos:** las Figuras 4.35, 4.36 y 4.37 presentan el estado de tensiones en el nodo 3. Donde se puede evidenciar que FSIPY, SAP2000 y ABAQUS muestran la misma tendencia al incrementar el número de elementos.

Se puede observar que para el caso de deformación plana los resultados obtenidos por FSIPY, SAP2000 y ABAQUS presentan un comportamiento similar en términos de desplazamientos y esfuerzos. Al igual que en el caso de tensión plana, los resultados de FSIPY mantienen una menor diferencia relativa con SAP2000.

Es importante resaltar que no se tomó en cuenta los valores correspondientes a 1 elemento ya que, al tener un cuadrilátero irregular, estos varían considerablemente.

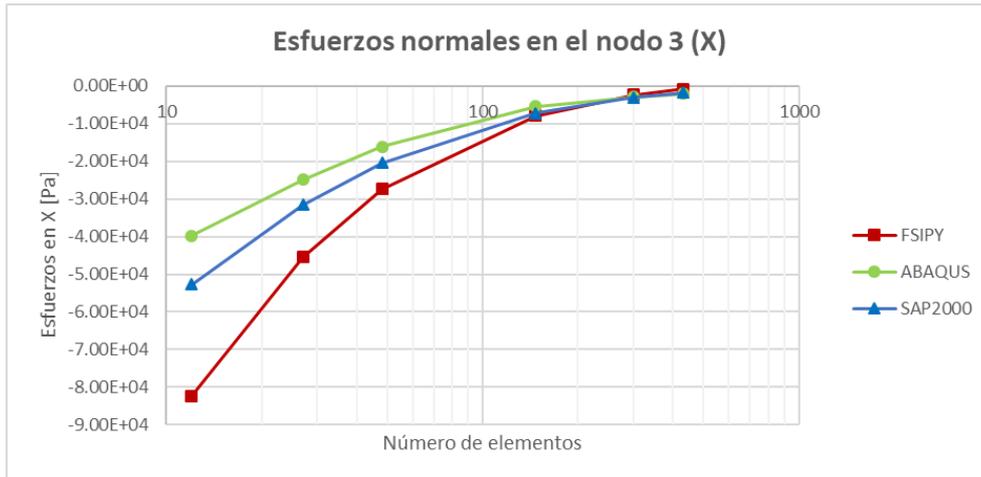


Figura 4.35: Esfuerzos normales en la dirección X del nodo 3

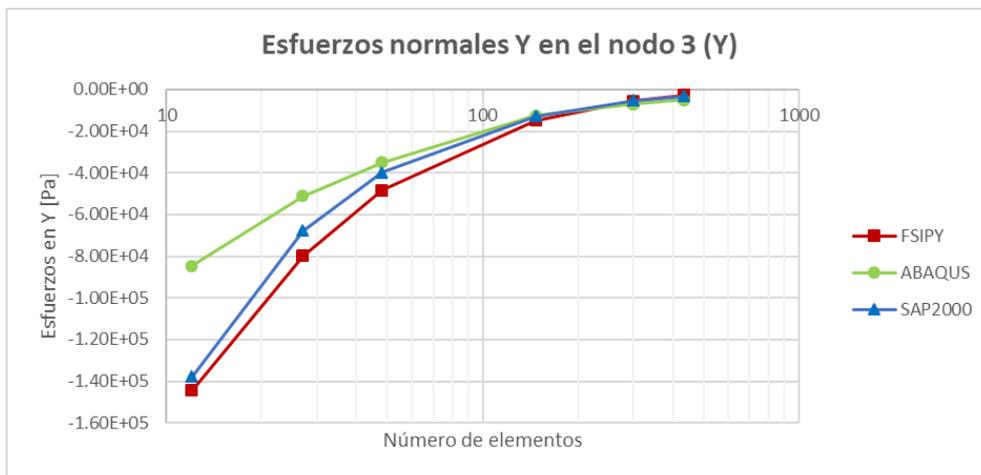


Figura 4.36: Esfuerzos normales en la dirección Y del nodo 3

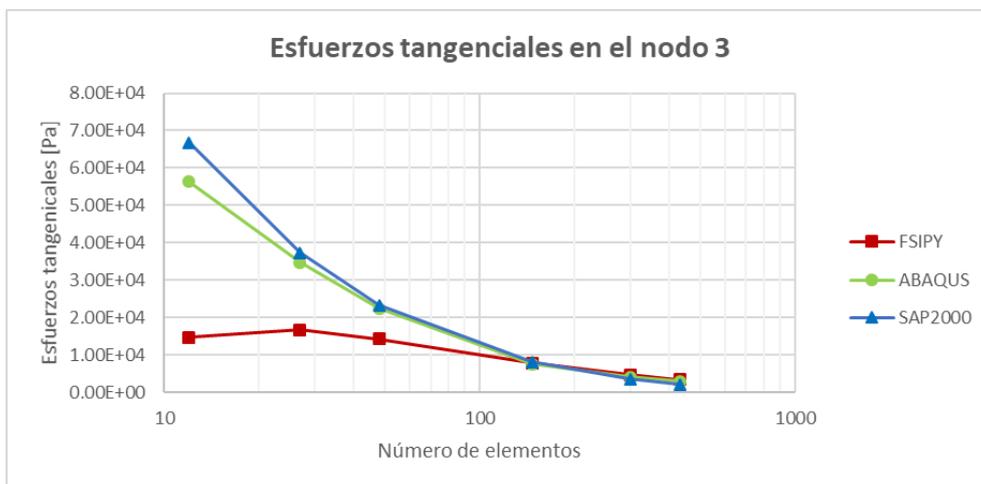


Figura 4.37: Esfuerzos tangenciales del nodo 3

Capítulo 5

5. Conclusiones

Se validó los resultados obtenidos de esfuerzos y desplazamientos por FSIPY mediante benchmarking. Basado en la convergencia de los resultados y similitud con SAP 2000 y ABAQUS, se concluye que el nuevo módulo desarrollado para el software FSIPY soluciona problemas en cuerpos bidimensionales isótropos en condiciones de tensión plana y deformación plana.

En la adaptación del software FSIPY a condiciones cuasi-estáticas se depuró los módulos y funciones preexistentes. El nuevo módulo calcula la matriz constitutiva, el vector de fuerzas superficiales y extrapola esfuerzos para los casos de elasticidad plana. Además, permite visualizar los desplazamientos y el estado tensional en ParaView.

En el presente trabajo de titulación se planteó tres casos de estudio para la validación. El primer caso se centró en el análisis unidimensional, el cual cuenta con la solución analítica para los desplazamientos. Al variar el número de elementos, se constató que los resultados obtenidos con FSIPY son exactos respecto a la respuesta analítica, independientemente de la cantidad de elementos utilizados. De igual manera, se corroboró que la relación entre la carga aplicada y los desplazamientos es lineal.

En relación al coeficiente de Poisson, se observó un aumento proporcional en los desplazamientos en la dirección transversal a medida que dicho coeficiente aumenta. Se pudo apreciar una relación lineal en el rango de 0 a 0.49, como se esperaba. Sin embargo, no se encontró una solución para un valor de 0.5, dado que este no es aplicable a sólidos.

En el segundo caso de estudio se observó que, al utilizar un cuadrilátero irregular, los resultados de desplazamientos obtenidos por el software FSIPY mantienen similitud con los resultados obtenidos por programas comerciales. En particular, la comparación de desplazamientos con el software SAP2000, demostró diferencias relativas en el orden del 1% en términos de tensión plana.

En cuanto a los resultados de tensiones, estos convergen a partir de 49 elementos. En el caso del nodo 3 (extremo libre), los esfuerzos normales en X convergen a cero

debido a que solo existe carga en el sentido vertical. Por el contrario, los esfuerzos normales en Y convergen a $-20 Pa$ correspondiente a la carga aplicada.

El último caso de estudio se enfocó en un ejemplo práctico basado en la deformación plana. Al igual que en el segundo caso, los resultados de desplazamientos obtenidos por FSIPY se acercan más a los obtenidos por SAP2000. En este sentido, se evidencia que las diferencias relativas de los desplazamientos son inferiores al 1%.

Es importante tener en cuenta que, al utilizar cualquier programa de análisis de esfuerzos, los resultados presentan diferencias notables debido a sus métodos de extrapolación y a la falta de definición del campo de esfuerzos. A pesar de esto, el software FSIPY presenta una convergencia estable de los resultados, con una tendencia similar a la obtenida por los programas comerciales.

En conclusión, aunque el software FSIPY presenta limitaciones en cuanto a su instalación y uso debido a su desarrollo en un entorno de Linux; sus ventajas de ser gratuito, tener acceso al código fuente y presentar diferencias relativas mínimas con respecto a los programas comerciales, lo convierten en una herramienta valiosa para quienes buscan desarrollar y validar problemas basados en el MEF. Es importante destacar que FSIPY carece de una interfaz de usuario y depende de otros programas para la visualización de resultados, lo que puede dificultar su uso para aquellos usuarios sin experiencia en programación.

Referencias

- Abasolo, M. (2016). *Guía práctica de elementos finitos en estática*. UNED.
- Backus, J. & Heising, W. (1964). Fortran. *IEEE Transactions on Electronic Computers*, EC-13(4):382–385.
- Bathe, K. J. (1996). *Finite Element Procedures in Engineering Analysis*. Prentice Hall.
- Belytschko, T., Liu, W., & Moran, B. (2014). *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, 2 edición.
- Coronel, G. (2006). Cálculo de la matriz de rigidez en forma analítica de un elemento finito isoparamétrico de cuatro nodos en elasticidad plana.
- Davis, P. J. (1984). *Methods of numerical integration*. Academic Press, 2nd edición.
- Fish, J. & Belytschko, T. (2007). *A First Course in Finite Elements*. Wiley, Hoboken, NJ, 2nd edición.
- Foundation, P. S. (2023). Python. <https://www.python.org/about/>. [Online; accessed 19-April-2023].
- Geuzaine, C. & Remacle, J.-F. (2009). Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. <https://gmsh.info/>. [Online; accessed 19-April-2023].
- Inc., K. (2023). Acerca de paraview. <https://www.paraview.org/about/>. [Online; accessed 19-April-2023].
- MathWorks (2023). MathWorks. <https://www.mathworks.com/>. [Online; accessed 19-April-2023].
- Oks, M. (2022). Fluid-structure interaction analysis of eccentricity and leaflet rigidity on thrombosis biomarkers in bioprosthetic aortic valve replacements. *Journal of Biomechanical Engineering*, 144(3):031008.
- Olivella, X. O. & de Saracíbar Bosch, C. A. (2000). *Mecánica de medios continuos para ingenieros*. Ediciones UPC.
- Oñate, E. (2009). *Structural Analysis with the Finite Element Method Linear Statics*. CRC Press.
- Pérez, C. (2002). *MATLAB y sus aplicaciones en las ciencias y la ingeniería*. PRENTICE HALL, Madrid, edición edición.
- Ruffrán, J. A. (2020). *Análisis aplicativo del método de los elementos finitos en un campo estático lineal e introducción a la no linealidad*. Universidad Nacional de Cuyo.

Singer, F. P. (2006). *Resistencia de materiales*. McGraw-Hill.

Timoshenko, S. & Goodier, J. (1970). *Teoría de la elasticidad*. Ediciones URMO.

Wilson, E. (2008). *Análisis estático y dinámico de estructuras*. Limusa.

Zienkiewicz, O. C. & Taylor, R. L. (1992). *El método de los elementos finitos*. Ediciones Díaz de Santos, Madrid, España, 4th edición.

Anexos A

A. Desplazamientos

A.1 Caso 1: Desplazamientos en el nodo 3

Tabla A.1: Desplazamientos en X para el caso 1 (FSIPY)

Desplazamientos en X [mm]					
Poisson	Analítica	1 elemento	400 elementos	Error % 1 elemento	Error % 400 elementos
0.00	0.380952	0.380952	0.380952	0.00	0.00
0.10	0.380952	0.380952	0.380952	0.00	0.00
0.20	0.380952	0.380952	0.380952	0.00	0.00
0.30	0.380952	0.380952	0.380952	0.00	0.00
0.40	0.380952	0.380952	0.380952	0.00	0.00
0.49	0.380952	0.380952	0.380952	0.00	0.00

Tabla A.2: Desplazamientos en Y para el caso 1 (FSIPY)

Desplazamientos en Y [mm]					
Poisson	Analítica	1 elemento	400 elementos	Error % 1 elemento	Error % 400 elementos
0.00	0.000000	0.000000	0.000000	0.00	0.00
0.10	-0.028571	-0.028571	-0.028571	0.00	0.00
0.20	-0.057143	-0.057143	-0.057143	0.00	0.00
0.30	-0.085714	-0.085714	-0.085714	0.00	0.00
0.40	-0.114286	-0.114286	-0.114286	0.00	0.00
0.49	-0.140000	-0.140000	-0.140000	0.00	0.00

Tabla A.3: Desplazamientos obtenidos al variar la carga aplicada para el caso 1 (FSIPY)

Desplazamientos [mm]						
Carga [Pa]	Analítica		FSIPY		Error %	
	X	Y	X	Y	X	Y
2.00E+04	0.000038	-0.000009	0.000038	-0.000009	0.00	0.00
2.00E+05	0.000381	-0.000086	0.000381	-0.000086	0.00	0.00
2.00E+06	0.003810	-0.000857	0.003810	-0.000857	0.00	0.00
2.00E+07	0.038095	-0.008571	0.038095	-0.008571	0.00	0.00
2.00E+08	0.380952	-0.085714	0.380952	-0.085714	0.00	0.00

A.2 Caso 2: Desplazamientos en los nodos 2 y 3

Tabla A.4: Desplazamientos en los nodos 2 y 3 para el caso 2 (FSIPY)

Desplazamientos en los nodos [mm]				
Número elementos	Nodo 2		Nodo 3	
	X	Y	X	Y
1	-0.0012	-0.0097	0.0027	-0.0099
4	-0.0012	-0.0186	0.0051	-0.0188
9	-0.0013	-0.0228	0.0063	-0.0230
25	-0.0013	-0.0260	0.0071	-0.0261
49	-0.0013	-0.0270	0.0074	-0.0272
100	-0.0013	-0.0276	0.0075	-0.0278
400	-0.0013	-0.0281	0.0077	-0.0283

Tabla A.5: Desplazamientos en los nodos 2 y 3 para el caso 2 (ABAQUS)

Desplazamientos en los nodos [mm]				
Número elementos	Nodo 2		Nodo 3	
	X	Y	X	Y
1	-0.4500	-1.8030	0.9000	-1.8040
4	0.0000	-0.0360	0.0100	-0.0360
9	0.0000	-0.0320	0.0100	-0.0320
25	0.0000	-0.0290	0.0100	-0.0300
49	0.0000	-0.0290	0.0100	-0.0290
100	0.0000	-0.0290	0.0100	-0.0290
400	0.0000	-0.0280	0.0100	-0.0290

Tabla A.6: Desplazamientos en los nodos 2 y 3 para el caso 2 (SAP2000)

Desplazamientos en los nodos [mm]				
Número elementos	Nodo 2		Nodo 3	
	X	Y	X	Y
1	-0.0044	-0.0230	0.0099	-0.0240
4	-0.0026	-0.0280	0.0085	-0.0280
9	-0.0018	-0.0280	0.0082	-0.0280
25	-0.0015	-0.0280	0.0078	-0.0280
49	-0.0014	-0.0280	0.0078	-0.0280
100	-0.0013	-0.0280	0.0077	-0.0280
400	-0.0013	-0.0280	0.0077	-0.0280

A.3 Caso 3: Desplazamientos en los nodos 3 y 4

Tabla A.7: Desplazamientos en los nodos 3 y 4 para el caso 3 (FSIPY)

Desplazamientos en los nodos [mm]				
Número elementos	Nodo 3		Nodo 4	
	X	Y	X	Y
1	0.2276	-0.0273	0.2283	0.0366
12	0.8119	-0.0112	0.8137	0.1713
27	0.8754	-0.0106	0.8774	0.1862
48	0.9013	-0.0103	0.9033	0.1922
147	0.9262	-0.0099	0.9282	0.1979
300	0.9329	-0.0098	0.9350	0.1994
432	0.9350	-0.0097	0.9371	0.1999

Tabla A.8: Desplazamientos en los nodos 3 y 4 para el caso 3 (ABAQUS)

Desplazamientos en los nodos [mm]				
Número elementos	Nodo 3		Nodo 4	
	X	Y	X	Y
1	39.4770	-6.3000	39.4826	15.7400
12	1.2160	-0.0200	1.2204	0.2700
27	1.0470	-0.0200	1.0508	0.2400
48	0.9960	-0.0100	0.9975	0.2200
147	0.9550	-0.0100	0.9575	0.2100
300	0.9470	-0.0100	0.9494	0.2000
432	0.9450	-0.0100	0.9471	0.2000

Tabla A.9: Desplazamientos en los nodos 3 y 4 para el caso 3 (SAP2000)

Desplazamientos en los nodos [mm]				
Número elementos	Nodo 3		Nodo 4	
	X	Y	X	Y
1	0.8090	-0.1170	0.8140	0.2730
12	0.9430	-0.0260	0.9450	0.1920
27	0.9360	-0.0120	0.9380	0.2020
48	0.9370	-0.0110	0.9390	0.2010
147	0.9380	-0.0100	0.9410	0.2010
300	0.9390	-0.0098	0.9410	0.2010
432	0.9390	-0.0098	0.9410	0.2010

Anexos B

B. Esfuerzos

B.1 Caso 2: Esfuerzos Software FSIPY

Tabla B.1: Esfuerzos normales en la dirección X para el caso 2 (FSIPY)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-14.13	-77.59	39.64	44.49
4	-87.29	-55.13	32.06	132.20
9	-136.39	-33.23	21.69	189.91
25	-178.50	-15.90	10.43	247.04
49	-194.41	-8.90	5.34	276.72
100	-205.11	-4.24	2.12	305.03
400	-219.60	-0.40	-0.08	360.48

Tabla B.2: Esfuerzos normales en la dirección Y para el caso 2 (FSIPY)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-3.01	-36.76	-1.59	14.57
4	-26.41	-23.36	-3.20	41.11
9	-41.52	-10.68	-14.28	58.44
25	-54.20	-5.50	-17.39	75.33
49	-58.87	-3.02	-18.80	84.01
100	-61.94	-1.36	-19.62	92.28
400	-66.10	-0.05	-20.09	108.59

Tabla B.3: Esfuerzos tangenciales para el caso 2 (FSIPY)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-62.74	20.73	17.90	-64.16
4	-82.04	7.21	7.81	-86.88
9	-80.62	-0.74	2.16	-88.60
25	-73.99	-2.95	-0.61	-85.47
49	-70.96	-2.66	-0.90	-84.42
100	-70.04	-1.97	-0.73	-85.58
400	-73.96	-0.78	-0.25	-94.63

B.2 Caso 2: Esfuerzos Software ABAQUS

Tabla B.4: Esfuerzos normales en la dirección X para el caso 2 (ABAQUS)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	0.00	0.00	0.00	0.00
4	-121.52	-42.42	42.42	121.52
9	-146.46	-24.22	26.99	154.29
25	-170.24	-13.36	11.68	193.93
49	-181.68	-7.90	5.75	219.03
100	-191.21	-4.18	2.61	245.16
400	-206.35	-1.04	0.35	296.90

Tabla B.5: Esfuerzos normales en la dirección Y para el caso 2 (ABAQUS)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-13.32	-13.32	-13.32	-13.32
4	-17.53	-10.66	-17.42	-19.25
9	-9.04	-6.14	-18.22	-14.56
25	-16.06	-3.39	-19.19	-6.44
49	-19.43	-2.01	-19.58	-1.68
100	-23.52	-1.04	-19.88	3.10
400	-26.99	-0.26	-20.00	10.03

Tabla B.6: Esfuerzos tangenciales para el caso 2 (ABAQUS)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-26.67	-26.67	-26.67	-26.67
4	-49.31	-21.38	-10.62	-19.26
9	-45.08	-12.19	-6.86	-21.35
25	-48.90	-6.73	-3.00	-24.92
49	-51.77	-3.98	-1.50	-27.74
100	-55.20	-2.09	-0.61	-31.39
400	-61.63	-0.52	-0.07	-39.48

B.3 Caso 2: Esfuerzos Software SAP2000

Tabla B.7: Esfuerzos normales en la dirección X para el caso 2 (SAP2000)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-151.50	-182.68	158.42	170.91
4	-177.74	-101.30	88.14	214.52
9	-196.13	-42.97	41.44	241.27
25	-206.85	-18.88	13.80	273.66
49	-211.44	-10.20	6.05	295.27
100	-215.64	-5.02	2.13	319.38
400	-226.04	-0.74	-0.14	372.78

Tabla B.8: Esfuerzos normales en la dirección Y para el caso 2 (SAP2000)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-62.86	-70.30	32.03	33.86
4	-62.69	-37.40	7.69	54.43
9	-62.38	-11.79	-10.09	67.11
25	-63.47	-6.04	-16.58	80.35
49	-64.03	-3.18	-18.58	87.92
100	-64.65	-1.47	-19.53	95.47
400	-67.90	-0.13	-20.05	111.85

Tabla B.9: Esfuerzos tangenciales para el caso 2 (SAP2000)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-45.38	-19.41	-10.07	-27.04
4	-48.04	-12.00	-10.34	-29.88
9	-48.44	-7.66	-8.27	-30.61
25	-53.57	-3.63	-3.65	-36.34
49	-56.61	-2.00	-1.42	-40.38
100	-60.11	-0.72	0.03	-45.42
400	-67.28	-0.01	0.05	-56.65

B.4 Caso 3: Esfuerzos Software FSIPY

Tabla B.10: Esfuerzos normales en la dirección X para el caso 3 (FSIPY)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	8.63E+04	-4.19E+04	-2.53E+05	6.72E+04
12	7.57E+05	-6.44E+05	-8.24E+04	-4.69E+04
27	8.67E+05	-7.07E+05	-4.55E+04	-7.53E+04
48	9.33E+05	-7.30E+05	-2.74E+04	-8.72E+04
147	1.05E+06	-7.65E+05	-7.88E+03	-9.76E+04
300	1.14E+06	-7.93E+05	-2.43E+03	-9.93E+04
432	1.19E+06	-8.10E+05	-8.88E+02	-9.95E+04

Tabla B.11: Esfuerzos normales en la dirección Y para el caso 3 (FSIPY)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	1.98E+05	-1.01E+05	-5.58E+05	1.90E+05
12	1.75E+06	-1.49E+06	-1.44E+05	8.21E+04
27	2.01E+06	-1.64E+06	-7.97E+04	3.98E+04
48	2.16E+06	-1.69E+06	-4.84E+04	1.97E+04
147	2.45E+06	-1.78E+06	-1.48E+04	7.63E+02
300	2.65E+06	-1.85E+06	-5.42E+03	-2.70E+03
432	2.77E+06	-1.89E+06	-2.72E+03	-3.18E+03

Tabla B.12: Esfuerzos tangenciales para el caso 3 (FSIPY)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	4.05E+05	4.04E+05	-1.32E+05	-1.29E+05
12	5.22E+05	4.96E+05	1.47E+04	6.15E+03
27	5.26E+05	5.03E+05	1.67E+04	7.97E+03
48	5.43E+05	5.21E+05	1.42E+04	6.48E+03
147	6.12E+05	5.85E+05	7.83E+03	2.81E+03
300	6.78E+05	6.39E+05	4.65E+03	1.30E+03
432	7.18E+05	6.70E+05	3.43E+03	8.06E+02

B.5 Caso 3: Esfuerzos Software ABAQUS

Tabla B.13: Esfuerzos normales en la dirección X para el caso 3 (ABAQUS)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-7.00E+04	-7.00E+04	-7.00E+04	-7.00E+04
12	-6.91E+04	-9.12E+04	-3.98E+04	-8.14E+04
27	4.55E+04	-1.65E+05	-2.49E+04	-8.82E+04
48	1.51E+05	-2.50E+05	-1.62E+04	-9.59E+04
147	2.19E+05	-3.02E+05	-5.45E+03	-9.71E+04
300	2.48E+05	-3.28E+05	-2.96E+03	-9.79E+04
432	2.79E+05	-3.43E+05	-2.08E+03	-9.79E+04

Tabla B.14: Esfuerzos normales en la dirección Y para el caso 3 (ABAQUS)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	-5.14E+04	-5.14E+04	-5.14E+04	-5.14E+04
12	1.02E+06	-1.18E+06	-8.46E+04	6.46E+04
27	1.25E+06	-1.29E+06	-5.12E+04	3.24E+04
48	1.40E+06	-1.37E+06	-3.50E+04	3.78E+03
147	1.71E+06	-1.48E+06	-1.24E+04	5.90E+02
300	1.88E+06	-1.56E+06	-7.09E+03	-1.86E+03
432	1.99E+06	-1.60E+06	-4.99E+03	-1.18E+03

Tabla B.15: Esfuerzos tangenciales para el caso 3 (ABAQUS)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	1.75E+05	1.75E+05	1.75E+05	1.75E+05
12	1.05E+05	3.68E+05	5.63E+04	3.59E+04
27	1.43E+05	3.26E+05	3.47E+04	2.02E+04
48	2.09E+05	3.92E+05	2.24E+04	4.26E+03
147	2.72E+05	4.50E+05	7.53E+03	1.85E+03
300	3.23E+05	5.02E+05	4.11E+03	1.42E+02
432	3.52E+05	5.28E+05	2.88E+03	2.38E+02

B.6 Caso 3: Esfuerzos Software SAP2000

Tabla B.16: Esfuerzos normales en la dirección X para el caso 3 (SAP2000)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	4.17E+02	-2.04E+04	-1.56E+05	-1.04E+05
12	4.23E+05	-5.14E+05	-5.27E+04	-6.83E+04
27	4.83E+05	-5.20E+05	-3.15E+04	-8.23E+04
48	5.69E+05	-5.68E+05	-2.04E+04	-8.93E+04
147	7.11E+05	-6.33E+05	-7.23E+03	-9.64E+04
300	7.95E+05	-6.67E+05	-3.07E+03	-9.80E+04
432	8.38E+05	-6.85E+05	-1.78E+03	-9.84E+04

Tabla B.17: Esfuerzos normales en la dirección Y para el caso 3 (SAP2000)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	6.72E+05	-7.75E+05	-1.86E+06	1.76E+06
12	1.35E+06	-1.83E+06	-1.38E+05	1.08E+05
27	1.82E+06	-1.59E+06	-6.76E+04	4.26E+04
48	1.94E+06	-1.62E+06	-3.98E+04	1.92E+04
147	2.20E+06	-1.69E+06	-1.27E+04	7.17E+02
300	2.39E+06	-1.76E+06	-5.36E+03	-2.25E+03
432	2.49E+06	-1.80E+06	-3.19E+03	-2.63E+03

Tabla B.18: Esfuerzos tangenciales para el caso 3 (SAP2000)

Esfuerzos en los nodos [Pa]				
Elementos	Nodo 1	Nodo 2	Nodo 3	Nodo 4
1	8.82E+04	2.62E+05	3.92E+05	-4.21E+04
12	5.65E+04	5.45E+05	6.68E+04	3.36E+04
27	1.81E+05	4.10E+05	3.73E+04	1.83E+04
48	2.16E+05	4.36E+05	2.33E+04	1.04E+04
147	2.99E+05	5.06E+05	8.10E+03	2.45E+03
300	3.61E+05	5.61E+05	3.52E+03	5.45E+02
432	3.94E+05	5.90E+05	2.12E+03	8.46E+01

Anexos C

C. Manual de usuario

C.1 Ejecución detallada del caso 2

A continuación, se ejemplificará la ejecución del caso 2. El caso presenta una placa delgada de acero empotrada y sometida a una carga constante de 20 Pa . La placa cuenta con un módulo de elasticidad $E = 30 \text{ MPa}$ y un coeficiente de Poisson $\nu = 0.3$. La geometría del sólido está compuesta por un cuadrilátero cuyas dimensiones se muestran en la Figura C.1.

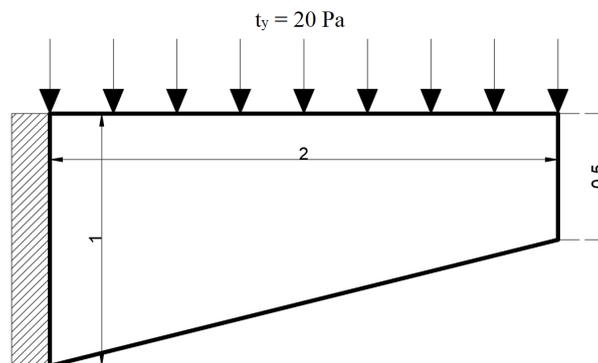


Figura C.1: Caso 2

Para cada ejemplo desarrollado se deben crear tres archivos de entrada: configuración, condiciones de contorno y geometría, como se observa en la Figura C.2.

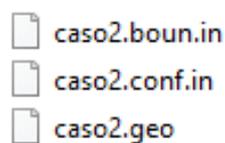


Figura C.2: Archivos de entrada de datos

El primer archivo en editarse es el archivo “caso2.geo” este contiene la geometría del sólido a discretizarse mediante GMSH.

```

*caso2.geo: Bloc de notes
Archivo Edición Formato Ver Ayuda
// Number of points
Nx = 4;
Ny = 4;

// Points
Point(1) = {0.0, 0.0, 0, 1.0};
Point(2) = {2.0, 0.5, 0, 1.0};
Point(3) = {2.0, 1.0, 0, 1.0};
Point(4) = {0.0, 1.0, 0, 1.0};

// Lines
Line(1) = {1, 2}; Transfinite Line {1} = Nx;
Line(2) = {2, 3}; Transfinite Line {2} = Ny;
Line(3) = {3, 4}; Transfinite Line {3} = Nx;
Line(4) = {4, 1}; Transfinite Line {4} = Ny;

```

```
*caso2.geo: Bloc de notas
Archivo Edición Formato Ver Ayuda
// Surfaces
Curve Loop(1) = {1, 2, 3, 4};
Plane Surface(1) = {1};
Transfinite Surface "*";
Recombine Surface "*";

// Tags
Physical Point("sw") = {1};
Physical Point("se") = {2};
Physical Point("ne") = {3};
Physical Point("nw") = {4};
Physical Curve("south") = {1};
Physical Curve("east") = {2};
Physical Curve("north") = {3};
Physical Curve("west") = {4};
Physical Surface("domain") = {1};
```

Figura C.3: Archivo caso2.geo

Dentro de GMSH se debe utilizar la función “Mesh-2D” para que se discretice el elemento y “Save” para guardar el archivo. Se crea un archivo de extensión “msh” que contiene los datos del elemento como los nodos, coordenadas y matriz de conectividad.

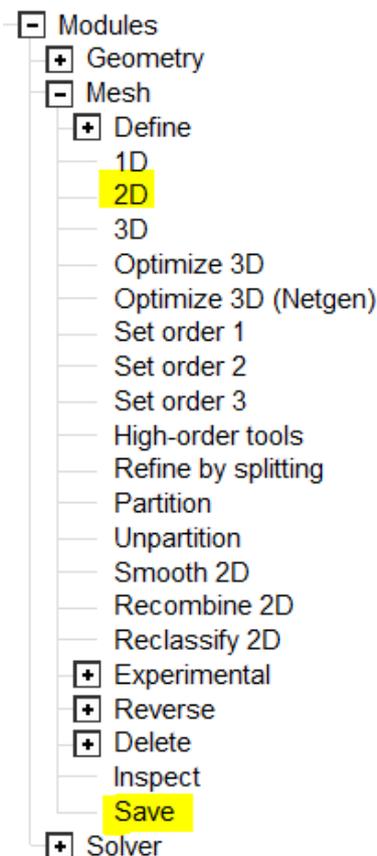


Figura C.4: Opciones de generación de malla en GMSH

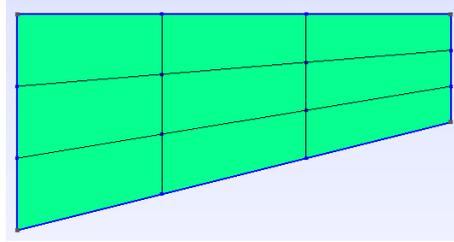


Figura C.5: Malla de nueve elementos GMSH

Después, se debe editar el archivo “caso2.conf.in” donde se detallan los parámetros físicos, parámetros numéricos, sources y outputs. Como se puede observar en la Figura C.6.

```

*caso2.conf.in: Bloc de notas
Archivo Edición Formato Ver Ayuda
$PhysicalParameters
  physics solid # problem physics
  material isolinear # material model
  rho 0.0 # density
  young 30.0e6 # young's modulus
  poisson 0.3 # poisson ratio
  ndime 2 # number of spatial dimensions
$EndPhysicalParameters

$NumericalParameters
  time_coupling fixed # time step definition [fixed'|from_critical]
  dt 0.01 # time step size
  beta 500 # beta parameter for beta-Newmark scheme
  gamma 0.5 # gamma parameter for beta-Newmark scheme
  time_treatment implicit # time treatment scheme [explicit|implicit]
  tolerance 1 # tolerance for iterations
  nitermax 1 # max number of iterations
  cfl 0.3 # CFL number (dt=CFL*dt_critical)
  tinitial 0.0 # initial time
  tfinal 20.0 # max real time
  ntmx 1 # max number of timesteps
  alpha 0.0 # Rayleigh damping mass matrix coefficient
  flag_lump_mass 0 # flag for mass-matrix lumping
  volume_quadrature Gauss3 # quadrature for volume numerical integration
  surface_quadrature Gauss3 # quadrature for surface numerical integration
  plane_elasticity Stress # Stress o Strain
$EndNumericalParameters

$Sources 1
  $Variable displacement
  type uniform # type of source
  nparameters 2 # number of parameters for source
  parameters 0.0 0.0 # source parameters
  $EndVariable
$EndSources

$Outputs
  path_outs resultados # path to outputs directory
  output_datatype vtk # output datatype
  ndump 1 # number of timesteps between outputs
  $InVolume
  displacement
  Sxx
  Syy
  Txy
  $EndInVolume
  $AtSets
  $EndAtSets
  $AtPhysicalGroups
  $EndAtPhysicalGroups
$EndOutputs

```

Figura C.6: Archivo caso2.conf.in

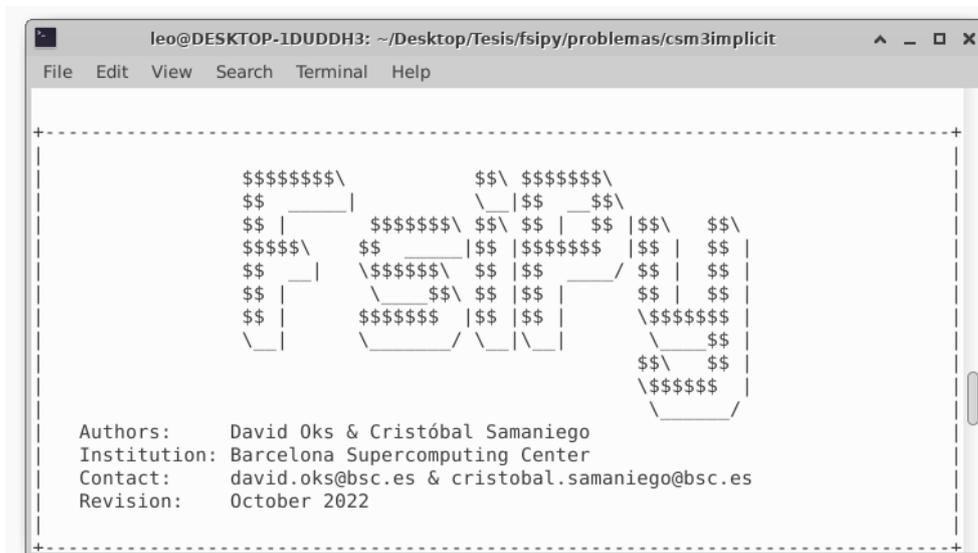
Luego, se edita el archivo “caso2.boun.in” en el cual se colocan las condiciones de contorno y las fuerzas superficiales que actúan en el elemento.

```
*caso2.boun.in: Bloc de notas
Archivo Edición Formato Ver Ayuda
$BoundaryConditions
  $Implicit_solid
    $Nodes
      1 11 0.0 0.0 0
      2 00 0.0 0.0 0
      3 00 0.0 0.0 0
      4 11 0.0 0.0 0
      5 00 0.0 0.0 0
      6 00 0.0 0.0 0
      7 00 0.0 0.0 0
      8 11 0.0 0.0 0
    $EndNodes
  $EndImplicit_solid
$EndBoundaryConditions

$Forces
  $Loads
    11 0.0 -20.0
    12 0.0 -20.0
    13 0.0 -20.0
  $EndLoads
  $Punctual
    $EndPunctual
$EndForces
```

Figura C.7: Archivo caso2.boun.in

Se ejecuta el ejercicio de acuerdo al Anexo D y se espera a que finalice el proceso como se muestra en la Figura C.8.





```
leo@DESKTOP-1DUDDH3: ~/Desktop/Tesis/fsipy/problemas/csm3implicit
File Edit View Search Terminal Help
-----
Problem Parameters:
-----
Physics:          solid
Time-treatment:  implicit
Material model:   isoline
Mass density:     1000.0
Young modulus:   30000000.0
Poisson ratio:   0.3
beta-Newmark scheme:
  Beta:           500.0
  Gamma:          0.5
  Timestep:       0.01
CFL Number:      4.263373956834367
Rayleigh alpha coef: 0.0
Number of elements: 9
Number of nodes: 16
Volume Quadrature: Gauss3
Surface Quadrature: Gauss3
Plane elasticity: Stress
```

Figura C.8: Terminal de Ubuntu

A continuación, se procede a abrir el software ParaView y se selecciona la opción “Open” para cargar los archivos generados previamente por FSIPY. Una vez cargados, se realiza la visualización de los resultados seleccionando la opción “Apply”. Se elige una de las variables de salida disponibles, las cuales se observan en la Figura C.9.

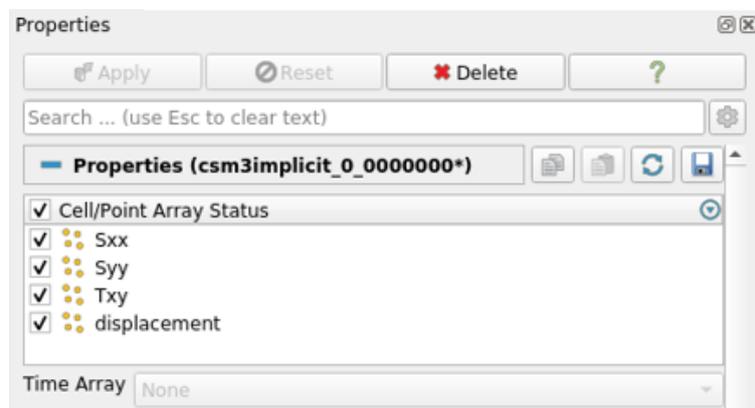


Figura C.9: Variables de salida

Se debe actualizar los valores con “Rescale to Visible Data Range”. El resultado se muestra en la Figura C.10.

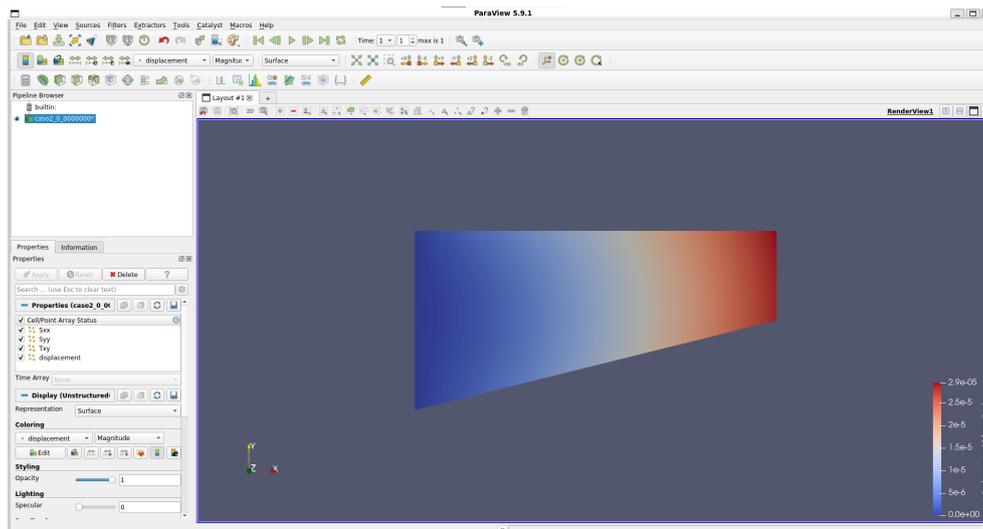


Figura C.10: Interfaz ParaView

Si se desea ver la deformación se puede utilizar el filtro “Warp By Vector”, se escoge un valor de escalas y se selecciona “Apply”. El resultado se visualiza en la Figura C.11.

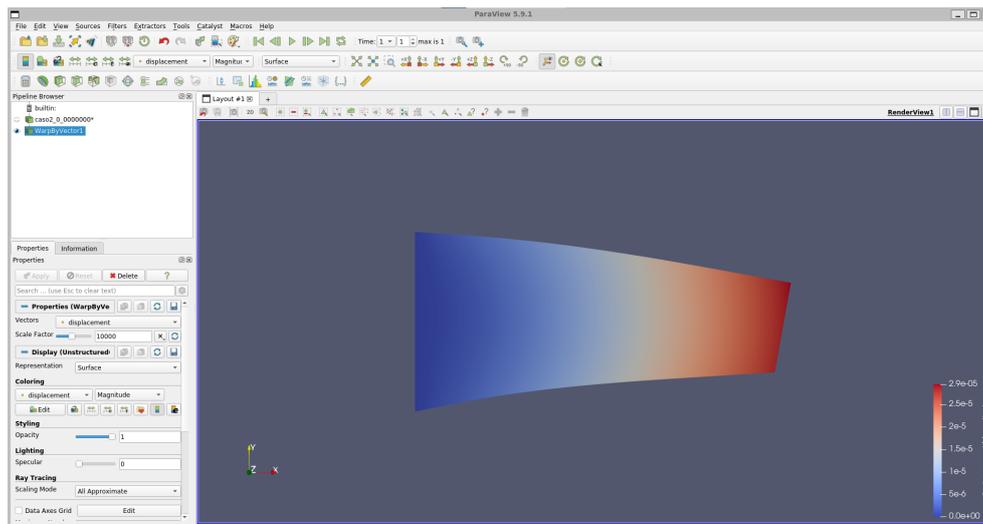


Figura C.11: Interfaz ParaView - Filtro “Warp By Vector”

Anexos D

D. Manual de instalación

Esta sección presenta el proceso de instalación del programa FSIPY. Es importante destacar que, aunque el software es de código abierto, su uso está restringido exclusivamente para fines académicos y no con fines de lucro. Para más información leer el archivo “License”, disponible en la carpeta de archivos.

El proceso de instalación puede resultar bastante laborioso, ya que involucra la ejecución de varios comandos. Con el fin de facilitar esta tarea, se ha preparado un video tutorial que muestra paso a paso el proceso de instalación. Para acceder al video, se proporciona el siguiente enlace.

Enlace Video de Instalación

Los archivos necesarios para llevar a cabo la instalación pueden obtenerse desde el repositorio de GitLab. Sin embargo, también se ha dispuesto una carpeta en Google Drive donde se encuentran dichos archivos. Para acceder a esta carpeta, se proporciona el siguiente enlace.

Enlace Carpeta en Google Drive

En el vídeo tutorial mencionado previamente, se utilizan una serie de comandos específicos. A continuación, se presentan dichos comandos en el mismo orden en el que aparecen en el vídeo.

D.1 Comandos para la instalación

```
$ wsl -l -v
$ wsl --set-version Ubuntu-20.04 2
$ wsl --set-default-version 2
$ sudo apt update && sudo apt -y upgrade
$ sudo apt-get purge xrdp
$ sudo apt install -y xrdp
$ sudo apt install -y xfce4
$ sudo apt install -y xfce4-goodies
$ sudo cp /etc/xrdp/xrdp.ini /etc/xrdp/xrdp.ini.bak
$ sudo sed -i 's/3389/3390/g' /etc/xrdp/xrdp.ini
$ sudo sed -i 's/max_bpp=32/#max_bpp=32\nmax_bpp=128/g'
/etc/xrdp/xrdp.ini
$ sudo sed -i 's/xserverbpp=24/#xserverbpp=24\nxserverbpp=128
/g'/etc/xrdp/xrdp.ini
$ echo xfce4-session > ~/.xsession
$ sudo nano /etc/xrdp/startwm.sh
$ startxfce4
```

```
$ sudo /etc/init.d/xrdp start
$ localhost:3390
$ sudo apt install firefox
```

D.2 Componentes extras

- Crear carpeta en el escritorio

```
$ cd Desktop
$ mkdir programa
```

- Entrar a la carpeta y descargar los archivos desde Gitlab

```
$ cd Desktop/programa
$ git clone https://gitlab.com/fsipy/fsipy.git
```

- Remover Python 2:

```
$ sudo apt remove python2
$ sudo apt autoremove --purge
```

- Instalación del pip:

```
$ sudo apt update
$ sudo apt install python3-pip
```

- Agregar al PATH:

```
$ /home/usuario/.bashrc
$ export PATH="$PATH:/home/user/.local/bin"
```

- Instalación GFortran:

```
$ sudo apt install gfortran
```

- Instalación GMSH (4.11):

```
$ pip install gmsh==4.11
```

- Instalación numpy (1.23):

```
$ pip install numpy==1.23
```

- Instalación pytest (7.2):

```
$ pip install pytest==7.2
```

- Cambio en el archivo “makefile”:

```
$ python3 -m numpy.f2py
```

- Librerías restantes:

```
$ sudo apt-get install python3-mpi4py  
$ sudo apt-get install python3-vtk7  
$ sudo apt-get install python3-scipy  
$ sudo apt-get install python3-pandas
```

D.3 Instalación de ParaView

```
$ sudo tar -xvzf ~/Downloads/ParaView-5.9.1  
-MPI-Linux-Python3.8-64bit.tar.gz -C /opt/
```

- Abrir carpeta opt desde la terminal:

```
$ cd ../../  
$ cd opt
```

- Renombrar la carpeta:

```
$ sudo mv ParaView-5.9.1-MPI-Linux-Python3.8-64bit  
ParaView-5.9
```

- Añadir al PATH:

```
$ /home/usuario/.bashrc  
$ export PATH=$PATH:/opt/ParaView-5.9/bin/
```

D.4 Compilación y ejecución

- Compilación

```
$ cd sources  
$ make
```

- Ejecución

```
$ python3 Main.py <PATH_TO_INPUTS>/<CASENAME>
```