



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería
Industrial

OPTIMIZACIÓN TOPOLÓGICA DE MATERIALES PIEZOELÉCTRICOS EN EL SOFTWARE FREEFEM

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS
INDUSTRIALES

Autor: Antonio Gallego Munuera
Director: Rogelio Ortigosa Martínez
Codirector: Jesús Martínez Frutos



Universidad
Politécnica
de Cartagena

Cartagena, agosto 2021

Índice

RESUMEN	8
CAPÍTULO 1: INTRODUCCIÓN	9
1.1. Motivación	9
1.2. Objetivos	11
1.3. Estructura del proyecto.....	11
CAPÍTULO 2: MECÁNICA DE CONTINUO	14
2.1. Cinemática	14
2.2. Gradiente de deformaciones	16
2.3. Deformación	18
2.4. Campo de desplazamientos	21
CAPÍTULO 3: FUNDAMENTOS DE ELASTICIDAD LINEAL	25
3.1. Energía de deformación elástica	25
3.2. Tensor de tensiones de Cauchy	25
3.3. Tensor de elasticidad	27
3.4. Forma fuerte y forma débil en elasticidad lineal.....	28
CAPÍTULO 4: FUNDAMENTOS DE LA ELECTRO-MECÁNICA LINEAL	31
4.1. Campo eléctrico, ley de Faraday y vector de polarización.....	31
4.2. Energía elástica del problema electromecánico	33
4.3. Tensor de Cauchy y Desplazamiento eléctrico	33
4.4. Tensores de orden superior	36
4.5. Forma fuerte y forma débil en electro-mecánica lineal.	38
CAPÍTULO 5: FUNDAMENTOS DE ELEMENTOS FINITOS	41
5.1. Método de los elementos finitos	41
5.2. Discretización del problema de la piezoelectricidad	45
CAPÍTULO 6: SIMULACIÓN NUMÉRICA EN PIEZOELECTRICIDAD LINEAL EN FREEFEM	49
6.1. Código genérico	49
6.2. Sensor piezoeléctrico bidimensional	52
6.3. Sensor piezoeléctrico tridimensional	60
6.4. Actuador piezoeléctrico tridimensional.....	64
6.5. Problema dinámico.....	70
CAPÍTULO 7: OPTIMIZACIÓN TOPOLÓGICA DEL PROBLEMA MECÁNICO	76
7.1. Formulación del problema	76

7.2. Formulación continua	78
7.3. Filtrado de densidades.....	80
7.4. Formulación final y resolución del problema.....	81
7.5. Filtrado de sensibilidades.....	88
CAPÍTULO 8: IMPLEMENTACIÓN DE OPTIMIZACIÓN TOPOLOGICA DEL PROBLEMA MECÁNICO	90
8.1. Código genérico	90
8.2. Benchmark 1.....	95
8.3. Benchmark 2.....	96
8.4. Benchmark 3.....	98
8.5. Benchmark 4.....	99
CAPÍTULO 9: OPTIMIZACIÓN TOPOLOGICA DEL PROBLEMA ELECTRO-MECÁNICO.....	101
9.1. Formulación y resolución del problema	101
9.2. Cálculo de la sensibilidad con filtrado de densidades	104
CAPÍTULO 10: IMPLEMENTACIÓN DE OPTIMIZACIÓN TOPOLOGICA DEL PROBLEMA ELECTRO-MECÁNICO	105
10.1. Código genérico	105
10.2. Caso sin muelle	112
10.3. Caso muelle blando.....	113
10.4. Caso muelle intermedio	115
10.5. Caso muelle duro	116
10.6. Influencia del vector de polarización piezoeléctrica.....	119
10.7. Problema de compresión	121
10.8. Problema de tracción.....	123
CAPÍTULO 11: CONCLUSIONES	125
BIBLIOGRAFÍA	128
ANEXOS	129
Anexo 1: Sensor piezoeléctrico bidimensional	129
Anexo 2: Sensor piezoeléctrico tridimensional	132
Anexo 2.1: MeshSurface	135
Anexo 3: Actuador piezoeléctrico tridimensional	138
Anexo 3.1: Script GMSH	141
Anexo 4: Problema dinámico.....	143
Anexo 5: TO Benchmark 1	148
Anexo 6: TO Benchmark 2 (malla).....	152

Anexo 7: TO Benchmark 3 (malla).....	153
Anexo 8: TO Benchmark 4 (malla).....	154
Anexo 9: TO piezoeléctrico	155

Índice de Figuras

Figura 1.1. Aplicaciones de optimización topológica a) Moto diseñada buscando ligereza y resistencia b) Puente optimizado frente a solicitaciones de carga c) Proyecto de torre biónica optimizada en Shanghai	9
Figura 1.2. Aplicaciones de materiales piezoeléctricos a) Sensor b) Actuador	10
Figura 2.1. Configuración inicial y final del continuo	14
Figura 2.2. Enfoque Lagrangiano	15
Figura 2.3. Vectores dX y dx del ejemplo 1	18
Figura 2.4. Vector desplazamiento u	21
Figura 3.1. Dominio del problema elástico	28
Figura 4.1. Superficie definida por el vector N	32
Figura 4.2. Fronteras mecánicas y eléctricas del dominio Ω	38
Figura 5.1. Richard Courant	41
Figura 5.2. Problema de la cuerda sujeta en los extremos	42
Figura 5.3. Funciones de aproximación del problema de la cuerda sujeta a extremos ...	43
Figura 6.1. Sensor piezoeléctrico bidimensional	52
Figura 6.2. Parámetros físicos de la matriz del sensor	53
Figura 6.3. Malla del sensor piezoeléctrico bidimensional	54
Figura 6.4. Malla deformada a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	55
Figura 6.5. Desplazamiento horizontal a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	56
Figura 6.6. Desplazamiento vertical a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	57
Figura 6.7. Potencial eléctrico a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	58
Figura 6.8. Tensión equivalente a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	59

Figura 6.9. Malla del sensor piezoeléctrico tridimensional a) Entera b) Plano de corte al centro.....	60
Figura 6.10. Cuerpo deformado a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	61
Figura 6.11. Campo eléctrico a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	62
Figura 6.12. Tensión equivalente a) $\zeta = 10 - 3$ b) $\zeta = 10 - 2$ c) $\zeta = 1$ d) $\zeta = 10$	63
Figura 6.13. Actuador piezoeléctrico tridimensional.....	64
Figura 6.14. Parámetros del actuador piezoeléctrico tridimensional.....	64
Figura 6.15. Malla del actuador piezoeléctrico tridimensional	65
Figura 6.16. Deformación del actuador piezoeléctrico (x100).....	65
Figura 6.17. Deformación del actuador piezoeléctrico (x200).....	65
Figura 6.18. Potencial eléctrico del actuador piezoeléctrico	66
Figura 6.19. Desplazamiento horizontal del actuador a) en x b) en y	67
Figura 6.20. Desplazamiento vertical del actuador.....	67
Figura 6.21. Tensión equivalente del actuador.....	68
Figura 6.22. Campo eléctrico del actuador	68
Figura 6.23. Gripper piezoeléctrico a) desde arriba b) desde abajo	69
Figura 6.24. Problema dinámico.....	70
Figura 6.25. Energías en el problema mecánico.....	73
Figura 6.26. Desplazamiento vertical del problema mecánico.....	74
Figura 6.27. Potencial eléctrico del problema mecánico	74
Figura 6.28. Tensión equivalente del problema mecánico	75
Figura 7.1. Espacio de diseño de optimización topológica	76
Figura 7.2. Función de proyección de ρ	79
Figura 7.3. Patrón de ajedrez	80
Figura 7.4. Filtro de densidades.....	80

Figura 8.1. Malla del problema de optimización topológica mecánico	91
Figura 8.2. Benchmark 1	95
Figura 8.3. Evolución en la distribución de densidades del benchmark 1	96
Figura 8.4. Benchmark 2	96
Figura 8.5. Evolución en la distribución de densidades del benchmark 2	97
Figura 8.6. Benchmark 3	98
Figura 8.7. Evolución en la distribución de densidades del benchmark 3	98
Figura 8.8. Benchmark 4	99
Figura 8.9. Evolución en la distribución de densidades del benchmark 4	100
Figura 9.1. Espacio de diseño de optimización topológica en piezoeléctricos	101
Figura 10.1. Espacio de diseño de optimización topológica en piezoeléctricos	105
Figura 10.2. Evolución en la distribución de densidades del caso sin muelle	112
Figura 10.3. Evolución en la distribución de densidades del caso sin muelle	113
Figura 10.4. Evolución en la distribución de densidades del caso de muelle intermedio	114
Figura 10.5. Evolución de J en el caso de muelle blando	114
Figura 10.6. Evolución en la distribución de densidades del caso de muelle intermedio	115
Figura 10.7. Evolución de J en el caso de muelle intermedio	116
Figura 10.8. Evolución en la distribución de densidades del caso de muelle duro	117
Figura 10.9. Evolución de J en el caso de muelle duro.....	117
Figura 10.10. Malla deformada para el caso de muelle duro	118
Figura 10.11. Malla deformada para el caso de muelle duro quitando el muelle.....	118
Figura 10.12. Evolución en la distribución de densidades del caso de muelle duro y $N = (0,1)$	119

Figura 10.13. Evolución de J en el caso de muelle duro y $N = (0,1)$	120
Figura 10.14. Evolución en la distribución de densidades del problema de compresión	121
Figura 10.15. Evolución de J en el problema de compresión.....	122
Figura 10.16. Malla deformada para problema de compresión.....	122
Figura 10.17. Evolución en la distribución de densidades del problema de tracción...	123
Figura 10.18. Evolución de J en el problema de tracción	124
Figura 10.19. Malla deformada para el problema de tracción.....	124
Figura 11.1. Gripper piezoeléctrico	125
Figura 11.2. Distribución óptima de densidades para distintas solicitaciones mecánicas	126
Figura 11.3. Distribución óptima de densidades para distintas solicitaciones electro- mecánicas	126

RESUMEN

La optimización topológica es una herramienta matemática que permite el diseño de estructuras o materiales más ligeras o con unas prestaciones más adecuadas para el propósito que están diseñadas. En el caso de materiales piezoeléctricos, estos pueden utilizarse como actuadores o sensores en importantes aplicaciones industriales. La optimización topológica en este contexto permite el diseño de actuadores y sensores óptimos para, dado una restricción de material (coste), maximizar las propiedades de actuación o la generación de potencial eléctrico como consecuencia de la aplicación de deformación. Este trabajo considera una serie de benchmarks en piezoelectricidad para proponer, mediante la aplicación de optimización topológica, diseños eficientes.

CAPÍTULO 1: INTRODUCCIÓN

1.1. Motivación

La ingeniería es concebida como una disciplina que aplica el conocimiento para la resolución de problemas reales, apoyándose en el método científico y en las ciencias puras, así como en aspectos económicos y administrativos. En definitiva, se puede considerar como el enlace necesario para que se lleven a la práctica ideas científicas, más o menos abstractas, consiguiendo un beneficio tangible en la sociedad. Aunque esta denominación y su reconocimiento como campo de estudio se dio en la Revolución Industrial (1760), la ingeniería existe desde la antigüedad, ya que es una cualidad inherente a los humanos como seres inteligentes y racionales. Desde la rueda, la palanca o los puentes hasta la más sofisticada de las máquinas en la actualidad, todo es producto de la ingeniería.

Como todas las ciencias, la ingeniería ha ido evolucionando con los años, y donde antes se buscaba una herramienta o máquina que hiciera una función, hoy se buscan diseños eficientes que permitan realizar dicha función y, además, reducir al máximo los costes de producción manteniendo a la vez las mejores prestaciones posibles. Esto es, por ejemplo, disminuir el peso de un coche para mejorar su consumo, mejorar el rendimiento de un ciclo Rankine o crear un puente lo más ligero y resistente posible para poder cargar más peso. Es aquí donde entra en juego la Optimización Topológica (véase la *Figura 1.1*, donde se muestran distintas aplicaciones), que consiste en obtener la mejor distribución de un material en un espacio de diseño sujeto a una restricción de volumen (costes, peso).

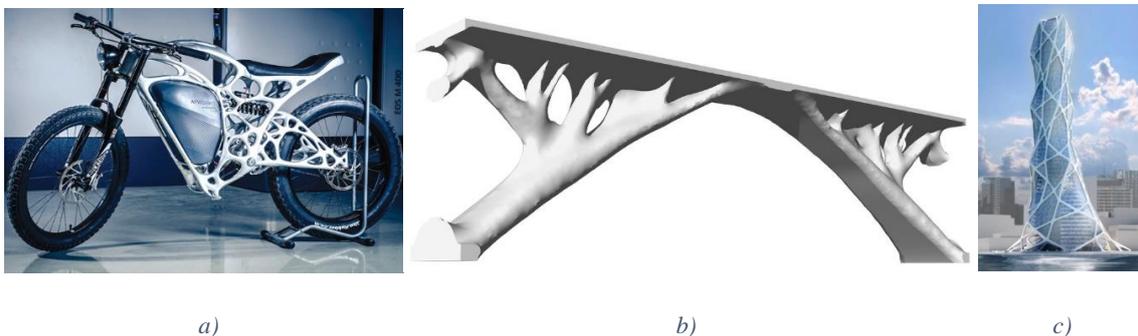


Figura 1.1. Aplicaciones de optimización topológica a) Moto diseñada buscando ligereza y resistencia b) Puente optimizado frente a solicitaciones de carga c) Proyecto de torre biónica optimizada en Shanghai

En cuanto a los materiales piezoeléctricos, fueron descubiertos en 1880 por Jaques y Pierre Curie, que observaron cómo ciertos cristales generaban un potencial eléctrico al someterse a tensión mecánica, y además el potencial producido era proporcional a la tensión aplicada. De la misma forma, en el piezoeléctrico se produce tensión mecánica, y por tanto se puede llegar a deformar ligeramente, al aplicarle una diferencia de potencial eléctrico. Desde entonces, han sido estudiados en ciencia e ingeniería y se utilizan para fabricar, entre otras cosas, sensores y actuadores piezoeléctricos (véase la *Figura 1.1*), los cuales funcionan gracias a los efectos descritos anteriormente, y bajo las hipótesis de linealidad y pequeñas deformaciones.



Figura 1.2. Aplicaciones de materiales piezoeléctricos a) Sensor b) Actuador

Mediante este trabajo se pretende modelizar matemáticamente el comportamiento de estos materiales, así como resolver distintos problemas de condiciones de contorno considerando sus leyes constitutivas en FreeFEM. De la misma forma, se explicarán los fundamentos matemáticos de la optimización topológica y se aplicará para el problema mecánico y para el problema piezoeléctrico en geometrías sencillas, para distintas funciones objetivo y condiciones de contorno.

Toda la realización del trabajo parte de los conocimientos adquiridos en la asignatura de métodos matemáticos para la resolución de modelos, la cual introduce el método de elementos finitos como una herramienta matemática extremadamente útil en todos los ámbitos de la ingeniería, ya que permite resolver problemas complejos numéricamente, obteniendo resultados satisfactorios.

1.2. Objetivos

Este trabajo persigue los siguientes objetivos:

- **Objetivo 1:** Familiarizarse con el sistema de EDPs y condiciones de contorno que gobiernan la electro-mecánica lineal (problema multifísico).
- **Objetivo 2:** Expresar las leyes constitutivas de materiales piezoeléctricos como invariantes del tensor de pequeñas deformaciones y el campo eléctrico, y deducir, a través de sus derivadas, el tensor de tensiones de Cauchy y el vector desplazamiento eléctrico.
- **Objetivo 3:** Deducir la forma débil del problema y condiciones de contorno de la electro-mecánica lineal.
- **Objetivo 4:** Implementación del problema de la electro-mecánica lineal en FreeFEM, a partir de su forma débil.
- **Objetivo 5:** Entender el problema de optimización topología mediante el método SIMP (Solid Isotropic Material with Penalization) como un problema de minimización sujeto a una serie de restricciones.
- **Objetivo 6:** Implementar el problema de optimización topológica en elasticidad lineal en FreeFEM.
- **Objetivo 7:** Implementar el problema multifísico de optimización topológica en electro-mecánica lineal en FreeFEM.

1.3. Estructura del proyecto

- **Capítulo 2: Mecánica de continuo.** Se introducirán los conceptos previos básicos necesarios para abordar el problema de elasticidad, comenzando por el *mapping de deformaciones*, que define y da nombre a los puntos de un sólido antes y después de ser deformado. Además, se explicarán los principales enfoques de la mecánica de continuo, Lagrangiano y Euleriano. Se definirá el tensor de deformaciones, así como el resto de tensores que guardan relación con el mismo (desplazamientos, *Green-Lagrange strain tensor*) hasta llegar al tensor de pequeñas deformaciones, *small strain tensor*.

- **Capítulo 3: Fundamentos de la elasticidad lineal.** En este capítulo se explicará el concepto de energía de deformación elástica, relacionado con el *small strain tensor* mediante los coeficientes de Lamé. A partir de esta energía se definirá el tensor de tensiones de Cauchy, así como el tensor de elasticidad. Finalmente, se mostrará la forma fuerte del problema de EDP para la elasticidad lineal isótropa junto con sus condiciones de contorno, y se derivará hacia la forma débil para su posterior implementación en FreeFEM.
- **Capítulo 4: Fundamentos de la electro-mecánica lineal.** Aquí se introducirán los parámetros eléctricos y leyes constitutivas que darán lugar al problema acoplado de la electro-mecánica. Se definirán el tensor de Cauchy y el vector desplazamiento eléctrico a partir de la expresión de la energía elástica para materiales piezoeléctricos, y se presentará la forma fuerte del problema acoplado que se transformará en la forma débil para su posterior implementación en FreeFEM.
- **Capítulo 5: Fundamentos de elementos finitos.** En este capítulo se introducirá la teoría de modelización matemática y el método de elementos finitos para la posterior implementación en FreeFEM de los problemas estudiados. También se realizará la discretización de un problema variacional sencillo, así como la del problema piezo-eléctrico.
- **Capítulo 6: Simulación numérica en piezoelectricidad lineal en FreeFEM.** Aplicando los conceptos estudiados en los capítulos anteriores, se resolverán en FreeFEM una serie de problemas y se comentarán los resultados. Se explicarán los aspectos más importantes del código genérico, así como las particularidades para cada uno de los ejemplos bidimensionales, tridimensionales y dinámicos.
- **Capítulo 7: Optimización topológica del problema mecánico.** Se introducirá la formulación matemática del problema de optimización topológica mecánico como la minimización de una función objetivo mediante una distribución discreta de propiedades. Se transformará el problema para poder usar distribuciones de densidades continuas mediante el método SIMP y la proyección de densidades para formular el problema compatible con la resolución mediante el método de elementos finitos, y se explicará el concepto de filtro de densidades para obtener un problema matemáticamente bien puesto. Finalmente, se resolverá el problema

Forward y Adjunto y se calculará la dirección de descenso. Además, se explicará la resolución del problema mediante filtro de sensibilidades.

- **Capítulo 8: Implementación de optimización topológica del problema mecánico.** Aplicando los conceptos estudiados en los capítulos 7, se resolverán en FreeFEM una serie benchmarks de optimización topológica mecánica con filtrado de sensibilidades, se explicarán los aspectos más importantes del código y se mostrarán los resultados.
- **Capítulo 9: Optimización topológica del problema electro-mecánico.** Se adaptará todo lo explicado en el capítulo 7 al problema piezoeléctrico, calculando las ecuaciones mecánica y eléctrica para el problema forward, así como para el adjunto.
- **Capítulo 10: Implementación de optimización topológica del problema electro-mecánico.** En este capítulo se implementará el problema de optimización topológica para materiales piezoeléctricos con filtrado de densidades y se mostrarán como resultados la evolución de la distribución de densidades y la evolución de la función objetivo. Se resolverán distintos casos de resistencia a la deformación y distintas funciones objetivo, y se estudiará la influencia del vector de polarización piezoeléctrica en los resultados.
- **Capítulo 11: Conclusiones.** En este capítulo se describirán las principales conclusiones derivadas del trabajo realizado, recopilando todo lo aprendido y desarrollado durante la realización del TFG y expresando las dificultades que han aparecido. Finalmente, se proponen posibles continuaciones en la línea de investigación.

CAPÍTULO 2: MECÁNICA DE CONTINUO

2.1. Cinemática

En la Figura 2.1 se encuentra representado un continuo genérico en su configuración inicial no deformada en el dominio Ω_0 . Su contorno es $\partial\Omega_0$, y el vector normal a la superficie del mismo es N . El punto P viene definido por su posición $\mathbf{X}_P(X_1, X_2, X_3)$, y el punto Q por $\mathbf{X}_Q(X_1 + dX_1, X_2 + dX_2, X_3 + dX_3)$. Su contorno es $\partial\Omega_0$ y el vector normal exterior a dicho contorno es N . Este continuo es sometido a un movimiento tras el cual ocupa una configuración final deformada definida por el dominio Ω , cuyo contorno es $\partial\Omega$ y donde el vector normal a la superficie es \mathbf{n} . Ahora la posición de P es $x_P(x_1, x_2, x_3)$, y la de Q es $x_Q(x_1 + dx_1, x_2 + dx_2, x_3 + dx_3)$.

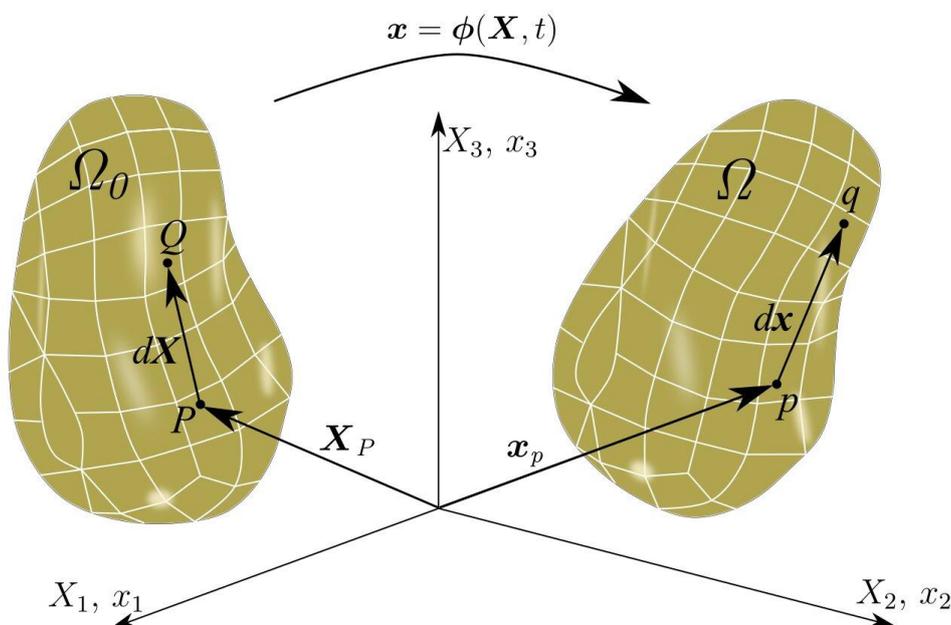


Figura 2.1. Configuración inicial y final del continuo

Este movimiento y deformación del continuo lo definimos mediante el **mapping de deformaciones** ϕ , una función que relaciona la posición de los puntos del dominio en su configuración inicial, $\mathbf{X} \in \Omega_0$, con la de la configuración final, $\mathbf{x} \in \Omega$. Por tanto, se podría expresar la posición de cualquier punto en función del tiempo como:

$$\mathbf{x} = \phi(\mathbf{X}, t) \quad (2.1)$$

Para este trabajo se aplicará la hipótesis de elasticidad lineal, que implica pequeñas deformaciones y pequeños desplazamientos, por lo que a efectos de cálculo se podrán considerar iguales \mathbf{X} y \mathbf{x} , así como \mathbf{N} y \mathbf{n} . Esta hipótesis es válida para los materiales piezoeléctricos que vamos a estudiar, así como para la mayoría de materiales en construcción e ingeniería (hierro, cemento, etc). Sin embargo, no lo es para otros materiales como los polímeros, donde el estudio de la elasticidad es no lineal.

En cuanto al estudio del comportamiento de un continuo, se pueden adoptar dos enfoques o interpretaciones: el enfoque *Lagrangiano* y el enfoque *Euleriano*.

El enfoque *Lagrangiano* se fija en los puntos materiales, cuya posición varía con respecto al tiempo. Por lo tanto, un punto material que en un instante inicial $t = t_0$ está situado en \mathbf{r}_0 variará su posición con respecto al tiempo. La posición del punto material define una función $\mathbf{r}(t)$ dependiente del tiempo y de condición inicial \mathbf{r}_0 . De esta forma se define una trayectoria (*Figura 2.2*) y se pueden conocer las magnitudes del punto material en cada momento. Este es el enfoque utilizado típicamente en mecánica de sólidos, y el que se adoptará en este trabajo.

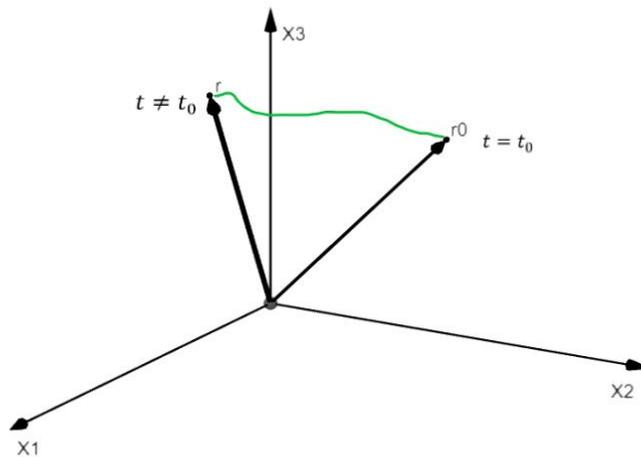


Figura 2.2. Enfoque Lagrangiano

El enfoque *Euleriano* no se fija en puntos materiales, sino que define los puntos mediante su posición en el espacio, \mathbf{r} . Esto implica que son fijos, y que la posición de los mismos \mathbf{r} es independiente del tiempo. La información del campo no está ligada al punto material, sino a la posición. Este enfoque es muy útil en mecánica de fluidos, donde se aplica normalmente.

2.2. Gradiente de deformaciones

Como se aprecia en la Figura 2.1, la posición relativa entre P y Q en la configuración inicial no deformada está definida por el vector $d\mathbf{X}$. Asimismo, dichos puntos materiales en la configuración final son p y q , y el vector $d\mathbf{x}$ define la posición relativa de los mismos. Esto es:

$$d\mathbf{X} = \mathbf{X}_Q - \mathbf{X}_P \quad d\mathbf{x} = \mathbf{x}_q - \mathbf{x}_p \quad (2.2)$$

A estas expresiones, ecuación (2.2) les podemos aplicar la función *mapping de deformaciones*, ecuación (2.1), quedando la expresión (2.3). (Nótese la que se ha simplificado la escritura de la función, $\boldsymbol{\phi}(\mathbf{X}, t) = \boldsymbol{\phi}(\mathbf{X})$, pero la función depende del tiempo en todo momento).

$$d\mathbf{x} = \boldsymbol{\phi}(\mathbf{X}_Q) - \boldsymbol{\phi}(\mathbf{X}_P) = \boldsymbol{\phi}(\mathbf{X}_P + d\mathbf{X}) - \boldsymbol{\phi}(\mathbf{X}_P) \quad (2.3)$$

El primer término de esta diferencia se puede reescribir como su desarrollo de Taylor de primer orden para una función continua, definido como:

$$f(x) \approx f(x_0) + \frac{\partial f(x_0)}{\partial x}(x - x_0) + \frac{1}{2} \frac{\partial^2 f(x_0)}{\partial x^2}(x - x_0)^2 \quad (2.4)$$

Aplicando la ecuación (2.4) podemos obtener la posición de la partícula \mathbf{x}_Q , tomando $f(x) = \boldsymbol{\phi}(\mathbf{X}_P + d\mathbf{X})$, $x_0 = \mathbf{X}_P$ y $(x - x_0) = d\mathbf{X}$:

$$\boldsymbol{\phi}(\mathbf{X}_Q) = \boldsymbol{\phi}(\mathbf{X}_P + d\mathbf{X}) \approx \boldsymbol{\phi}(\mathbf{X}_P) + \frac{\partial \boldsymbol{\phi}(\mathbf{X}_P)}{\partial \mathbf{X}} d\mathbf{X} + o(\|d\mathbf{X}\|^2) \quad (2.5)$$

Sustituyendo la expresión (2.5) en (2.3) se obtiene:

$$d\mathbf{x} = \boldsymbol{\phi}(\mathbf{X}_P) + \frac{\partial \boldsymbol{\phi}(\mathbf{X}_P)}{\partial \mathbf{X}} d\mathbf{X} + o(d\mathbf{X}^2) - \boldsymbol{\phi}(\mathbf{X}_P) \quad (2.6)$$

Anulando términos queda la expresión:

$$d\mathbf{x} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}} d\mathbf{X} + o(\|d\mathbf{X}\|^2) \quad (2.7)$$

Por último, al trabajar en elasticidad lineal se pueden despreciar los términos de orden superior, quedando la expresión final:

$$dx = \frac{\partial \phi}{\partial X} dX \quad (2.8)$$

En la expresión (2.8) podemos identificar el **gradiente de deformaciones F** , el cual queda definido como:

$$F = \frac{\partial \phi}{\partial X} = \nabla_0 \phi \quad (2.9)$$

El subíndice 0 de ∇ nos indica que la derivada es con respecto a X . Sin embargo, como consideramos las hipótesis de pequeñas deformaciones y pequeños desplazamientos de la elasticidad lineal, se puede tomar:

$$F = \frac{\partial \phi}{\partial X} = \nabla_0 \phi \approx \frac{\partial \phi}{\partial x} = \nabla \phi \quad (2.10)$$

Sustituyendo (2.10) en (2.8) se obtiene la expresión:

$$dx = F dX \quad (2.11)$$

La función obtenida, el **gradiente de deformaciones F** , por tanto, nos permite transformar vectores en su configuración inicial no deformada a su configuración final deformada bajo las hipótesis tomadas para la elasticidad lineal.

Ejemplo 1:

Dada la siguiente función ϕ , se calcula el gradiente de deformaciones F .

$$\phi = \begin{bmatrix} \frac{1}{3}(12 + 9X_1 + 3X_2) \\ \frac{1}{3}(16 + 6X_2) \end{bmatrix}$$

Aplicando la expresión (2.9) se obtiene:

$$F = \frac{\partial \phi}{\partial X} = \begin{bmatrix} \frac{\partial \phi_1}{\partial X_1} & \frac{\partial \phi_1}{\partial X_2} \\ \frac{\partial \phi_2}{\partial X_1} & \frac{\partial \phi_2}{\partial X_2} \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$

Conociendo el vector $d\mathbf{X}$ en la configuración inicial, se puede obtener el vector $d\mathbf{x}$ en su configuración final. Suponiendo como vector $d\mathbf{X}$ el siguiente:

$$d\mathbf{X} = \begin{bmatrix} 5/3 \\ 3 \end{bmatrix}$$

Aplicando la ecuación (2.11) se obtiene:

$$d\mathbf{x} = \mathbf{F}d\mathbf{X} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 5/3 \\ 3 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

Se puede observar en este resultado que el gradiente de deformaciones \mathbf{F} produce un cambio en el módulo y rota el vector $d\mathbf{X}$. Su configuración final, $d\mathbf{x}$, está representada en la Figura 2.3 junto a la configuración inicial.

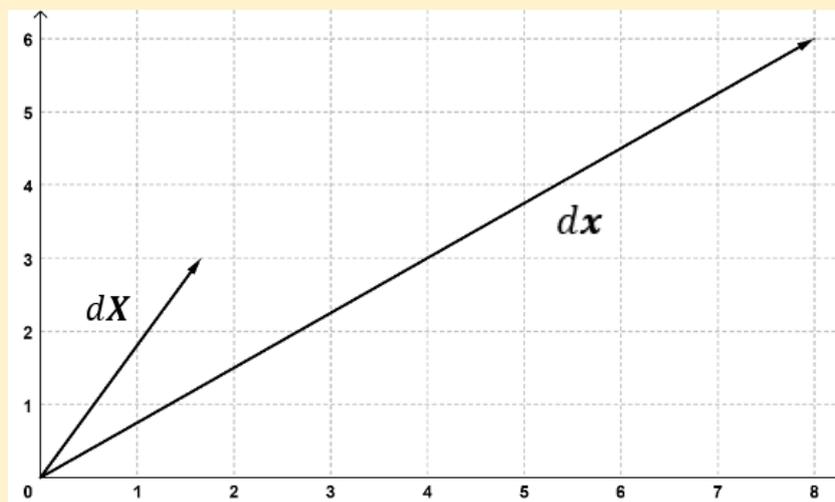


Figura 2.3. Vectores $d\mathbf{X}$ y $d\mathbf{x}$ del ejemplo 1

2.3. Deformación

A partir del gradiente de deformaciones \mathbf{F} , se obtiene un tensor simétrico y definido positivo llamado **Right-Cauchy deformation tensor** \mathbf{C} , definido como:

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} \quad (2.12)$$

Asimismo, el tensor \mathbf{C} está relacionado con el **Green-Lagrange strain tensor** \mathbf{E} , que se obtiene mediante la expresión (2.13), donde \mathbf{I} es la matriz identidad:

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) \quad (2.13)$$

Este resultado es de vital importancia, ya que aporta información sobre el campo de deformaciones del dominio estudiado. Si se toma un vector $d\mathbf{X}$ de la configuración inicial no deformada, y su vector análogo $d\mathbf{x}$ de la configuración final deformada, podemos definir sus módulos, dL y dl respectivamente, como:

$$d\mathbf{X} \cdot d\mathbf{X} = dL^2 \qquad d\mathbf{x} \cdot d\mathbf{x} = dl^2 \qquad (2.14)$$

La diferencia de estos módulos es, por tanto,

$$dl^2 - dL^2 = d\mathbf{x} \cdot d\mathbf{x} - d\mathbf{X} \cdot d\mathbf{X} \qquad (2.15)$$

Aplicando las expresiones (2.11) y (2.12) queda:

$$\begin{aligned} dl^2 - dL^2 &= \mathbf{F}d\mathbf{X} \cdot \mathbf{F}d\mathbf{X} - d\mathbf{X} \cdot d\mathbf{X} \\ &= d\mathbf{X} \cdot \mathbf{F}^T \mathbf{F}d\mathbf{X} - d\mathbf{X} \cdot \mathbf{I}d\mathbf{X} \\ &= d\mathbf{X} \cdot \mathbf{C}d\mathbf{X} - d\mathbf{X} \cdot \mathbf{I}d\mathbf{X} \end{aligned} \qquad (2.16)$$

Agrupando términos y aplicando la ecuación (2.13) se llega a la expresión:

$$dl^2 - dL^2 = d\mathbf{X} \cdot (\mathbf{C} - \mathbf{I})d\mathbf{X} = d\mathbf{X} \cdot 2\mathbf{E}d\mathbf{X} \qquad (2.17)$$

Por último, dividiendo ambos términos de la igualdad por dL^2 obtenemos la expresión:

$$\frac{1}{2} \frac{dl^2 - dL^2}{dL^2} = \frac{d\mathbf{X} \cdot \mathbf{E}d\mathbf{X}}{dL^2} \qquad (2.18)$$

Como se aprecia en la ecuación (2.18), el llamado *Green-Lagrange strain tensor* \mathbf{E} nos informa sobre cambios relativos en la longitud de los vectores de la configuración final deformada con respecto a la configuración inicial.

Considerando ahora dos vectores $d\mathbf{X}_1$ y $d\mathbf{X}_2$ junto con sus configuraciones finales deformadas $d\mathbf{x}_1$ y $d\mathbf{x}_2$ y realizando un desarrollo análogo, se obtiene como resultado la expresión (2.19):

$$\begin{aligned} d\mathbf{x}_1 \cdot d\mathbf{x}_2 - d\mathbf{X}_1 \cdot d\mathbf{X}_2 &= \mathbf{F}d\mathbf{X}_1 \cdot \mathbf{F}d\mathbf{X}_2 - d\mathbf{X}_1 \cdot d\mathbf{X}_2 \\ &= d\mathbf{X}_1 \cdot \mathbf{F}^T \mathbf{F}d\mathbf{X}_2 - d\mathbf{X}_1 \cdot \mathbf{I}d\mathbf{X}_2 \\ &= d\mathbf{X}_1 \cdot (\mathbf{C} - \mathbf{I})d\mathbf{X}_2 \\ &= d\mathbf{X}_1 \cdot 2\mathbf{E}d\mathbf{X}_2 \end{aligned} \qquad (2.19)$$

La expresión ahora obtenida relaciona el tensor \mathbf{E} con la diferencia del producto escalar de los dos vectores considerados en su configuración final deformada con respecto a su configuración inicial. Como $d\mathbf{x}_1 \cdot d\mathbf{x}_2 = |d\mathbf{x}_1||d\mathbf{x}_2| \cos \alpha$, el tensor también nos informa acerca del cambio entre las proyecciones de los vectores, es decir, de la distorsión angular.

Ejemplo 2:

Recogiendo los resultados obtenidos en el Ejemplo 1 se calcula el tensor \mathbf{E} . Aplicando la ecuación (2.12), el *Right-Cauchy deformation tensor* \mathbf{C} es:

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}^T \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 9 & 3 \\ 3 & 5 \end{bmatrix}$$

El tensor *Green-Lagrange* \mathbf{E} se obtiene mediante la expresión (2.11):

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) = \frac{1}{2} \left(\begin{bmatrix} 9 & 3 \\ 3 & 5 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \begin{bmatrix} 4 & 3/2 \\ 3/2 & 2 \end{bmatrix}$$

La longitud al cuadrado del vector $d\mathbf{X}$ es:

$$dL^2 = d\mathbf{X} \cdot d\mathbf{X} = \begin{bmatrix} 5/3 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 5/3 \\ 3 \end{bmatrix} = 11,78$$

La longitud del vector en la configuración deformada será:

$$dl^2 - dL^2 = d\mathbf{X} \cdot 2\mathbf{E}d\mathbf{X} = \begin{bmatrix} 5/3 \\ 3 \end{bmatrix} \cdot 2 \begin{bmatrix} 4 & 3/2 \\ 3/2 & 2 \end{bmatrix} \begin{bmatrix} 5/3 \\ 3 \end{bmatrix} = 88,22$$

$$dl = \sqrt{88,22 + dL^2} = \sqrt{88,22 + 11,78} = 10$$

Calculando el módulo del vector $d\mathbf{x}$ mediante su definición se obtiene el mismo resultado:

$$dl = \sqrt{\begin{bmatrix} 8 \\ 6 \end{bmatrix} \cdot \begin{bmatrix} 8 \\ 6 \end{bmatrix}} = 10$$

2.4. Campo de desplazamientos

Considerando el sistema de coordenadas (X_1, X_2, X_3) de la *Figura 2.4* en el que están representadas la configuración inicial $\mathbf{X} \in \Omega_0$ y la configuración final $\mathbf{x} \in \Omega$, se define el **vector de desplazamiento** \mathbf{u} como:

$$\mathbf{u} = \mathbf{x} - \mathbf{X} \quad (2.20)$$

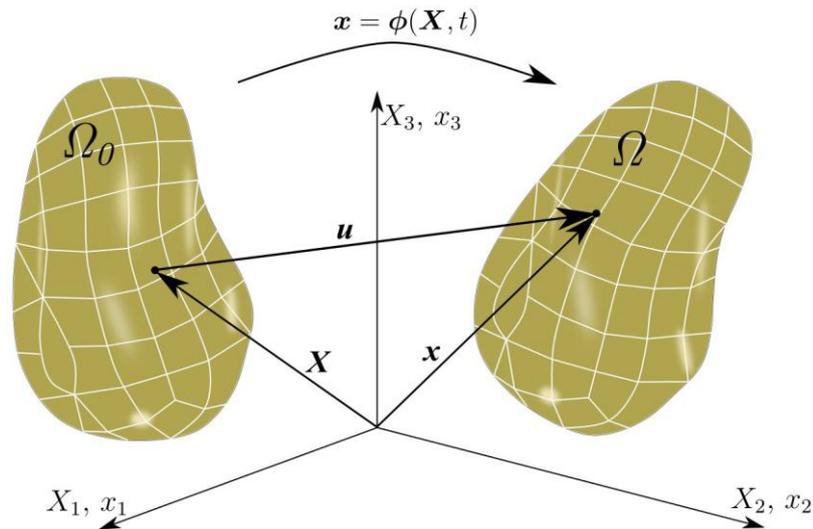


Figura 2.4. Vector desplazamiento \mathbf{u}

El gradiente de dicho vector es el siguiente:

$$\nabla_0 \mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial}{\partial \mathbf{X}} (\mathbf{x} - \mathbf{X}) = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} - \frac{\partial \mathbf{X}}{\partial \mathbf{X}} \quad (2.21)$$

Aplicado en (2.21) la definición del *mapping de deformaciones* $\boldsymbol{\phi}(\mathbf{X})$ (2.1), *gradiente de deformaciones* \mathbf{F} (2.10) y matriz identidad \mathbf{I} , se obtiene:

$$\nabla_0 \mathbf{u} = \frac{\partial \boldsymbol{\phi}(\mathbf{X})}{\partial \mathbf{X}} - \frac{\partial \mathbf{X}}{\partial \mathbf{X}} = \mathbf{F} - \mathbf{I} \quad (2.22)$$

Se observa en esta expresión que el vector de desplazamiento \mathbf{u} está relacionado con el gradiente de deformaciones \mathbf{F} , lo que implica que también lo está con el tensor \mathbf{E} . Desarrollando su expresión, y siendo $\mathbf{F} = \nabla_0 \mathbf{u} + \mathbf{I}$:

$$\begin{aligned}
\mathbf{E} &= \frac{1}{2}(\mathbf{C} - \mathbf{I}) = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) \\
&= \frac{1}{2}((\nabla_0 \mathbf{u} + \mathbf{I})^T (\nabla_0 \mathbf{u} + \mathbf{I}) - \mathbf{I}) \\
&= \frac{1}{2}(\mathbf{I} + \nabla_0 \mathbf{u}^T + \nabla_0 \mathbf{u} + \nabla_0 \mathbf{u}^T \nabla_0 \mathbf{u} - \mathbf{I})
\end{aligned} \tag{2.23}$$

Finalmente, agrupamos términos:

$$\mathbf{E} = \frac{1}{2}(\nabla_0 \mathbf{u}^T + \nabla_0 \mathbf{u}) + \frac{1}{2} \nabla_0 \mathbf{u}^T \nabla_0 \mathbf{u} \tag{2.24}$$

Nuevamente, se aplican las simplificaciones propias de la elasticidad lineal, despreciándose el término no lineal $\nabla_0 \mathbf{u}^T \nabla_0 \mathbf{u}$ y tomando $\nabla_0 \approx \nabla$ por la hipótesis de pequeños desplazamientos. Tras simplificar la expresión se obtiene el **tensor de pequeñas deformaciones** o **small strain tensor** $\boldsymbol{\varepsilon}$.

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u}) \tag{2.25}$$

Se puede reescribir este tensor, aplicando la expresión (2.22), como:

$$\begin{aligned}
\boldsymbol{\varepsilon} &= \frac{1}{2}((\mathbf{F} - \mathbf{I})^T + (\mathbf{F} - \mathbf{I})) \\
&= \frac{1}{2}(\mathbf{F} + \mathbf{F}^T) - \mathbf{I}
\end{aligned} \tag{2.26}$$

Ejemplo 3:

A partir del tensor \mathbf{F} calculado de los anteriores ejemplos, se calcula tensor de pequeñas deformaciones $\boldsymbol{\varepsilon}$.

$$\mathbf{F} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} 4 & 3/2 \\ 3/2 & 2 \end{bmatrix}$$

Aplicando la expresión (2.26) se puede obtener $\boldsymbol{\varepsilon}$.

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\mathbf{F} + \mathbf{F}^T) - \mathbf{I} = \frac{1}{2} \left(\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}^T \right) - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

Se observa que, pese a que el tensor $\boldsymbol{\varepsilon}$ es una aproximación del tensor \boldsymbol{E} , distan bastante de ser iguales. Se puede calcular el error relativo β en su término a_{22} , donde el error es mayor, obteniéndose:

$$\beta = \frac{|1/2 - 3/2|}{3/2} \cdot 100 = 66,67 \%$$

La razón de un error tan alto es que, en el ejemplo propuesto, no se cumple la hipótesis de pequeñas deformaciones. Por tanto, el tensor obtenido bajo estas hipótesis, $\boldsymbol{\varepsilon}$, no es válido. Para realizar una simplificación válida, debemos tener una función *mapping de deformaciones* $\boldsymbol{\phi}$ que se ajuste a las condiciones de elasticidad lineal.

$$\boldsymbol{\phi} = \begin{bmatrix} \frac{1}{5}(15 + 5X_1 + 10^{-3}X_2) \\ \frac{1}{5}(10 + 5.05X_2) \end{bmatrix}$$

Aplicando la expresión (2.9) se obtiene el gradiente de deformaciones:

$$\boldsymbol{F} = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial \phi_1}{\partial X_1} & \frac{\partial \phi_1}{\partial X_2} \\ \frac{\partial \phi_2}{\partial X_1} & \frac{\partial \phi_2}{\partial X_2} \end{bmatrix} = \begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 0 & 1.01 \end{bmatrix}$$

Con la ecuación (2.12) se calcula *Right-Cauchy deformation tensor*:

$$\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F} = \begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 0 & 1.01 \end{bmatrix}^T \begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 0 & 1.01 \end{bmatrix} = \begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 2 \cdot 10^{-4} & 1.0201 \end{bmatrix}$$

Con las ecuaciones (2.13) y (2.26) se obtienen el tensor *Green-Lagrange* \boldsymbol{E} y el tensor de pequeñas deformaciones $\boldsymbol{\varepsilon}$ respectivamente.

$$\begin{aligned} \boldsymbol{E} &= \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}) = \frac{1}{2} \left(\begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 2 \cdot 10^{-4} & 1.0201 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \begin{bmatrix} 0 & 1 \cdot 10^{-4} \\ 1 \cdot 10^{-4} & 0.01005 \end{bmatrix} \\ \boldsymbol{\varepsilon} &= \frac{1}{2}(\boldsymbol{F} + \boldsymbol{F}^T) - \boldsymbol{I} = \frac{1}{2} \left(\begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 0 & 1.01 \end{bmatrix} + \begin{bmatrix} 1 & 2 \cdot 10^{-4} \\ 0 & 1.01 \end{bmatrix}^T \right) - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix} \end{aligned}$$

Si ahora se evalúa el error relativo del término a_{22} nuevamente:

$$\beta = \frac{|10^{-2} - 1.005 \cdot 10^{-2}|}{1.005 \cdot 10^{-2}} \cdot 100 = 0,498\%$$

Se observa que el error ha disminuido significativamente. Por tanto, se puede considerar la aproximación del tensor de pequeñas deformaciones ε para elasticidad lineal.

CAPÍTULO 3: FUNDAMENTOS DE ELASTICIDAD LINEAL

3.1. Energía de deformación elástica

La ley constitutiva desarrollada en este apartado es aplicable a sólidos que cumplan las hipótesis de la elasticidad lineal, y que además tengan un comportamiento isótropo, es decir, que se comportan igual en todas las direcciones. Para estos casos, se define la **energía de deformación elástica por unidad de volumen** $\Psi(\boldsymbol{\varepsilon})$ como:

$$\Psi(\boldsymbol{\varepsilon}) = \mu \operatorname{tr}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}) + \frac{\lambda}{2} (\operatorname{tr}(\boldsymbol{\varepsilon}))^2 \quad (3.1)$$

En esta expresión, λ es el primer parámetro de Lamé, relacionado con los cambios volumétricos del sólido, y μ es el segundo parámetro de Lamé, también llamado módulo de cizalladura, relacionado con la resistencia a esfuerzos cortantes. Los parámetros de Lamé se pueden expresar en función del módulo de Young E y el coeficiente de Poisson ν , como se indica en las expresiones (3.2). Todos estos parámetros son constantes para cada material, y dependen de la naturaleza del mismo.

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)} \quad (3.2)$$

El término tr es el operador traza. La traza de una matriz se calcula sumando las componentes de su diagonal principal:

$$\operatorname{tr}(A) = \sum_{i=1} A_{ii} \quad (3.3)$$

3.2. Tensor de tensiones de Cauchy

El **tensor de tensiones de Cauchy** $\boldsymbol{\sigma}$ y su relación con el tensor de pequeñas deformaciones $\boldsymbol{\varepsilon}$ es lo que define la ley constitutiva de la elasticidad lineal isótropa.

$$\boldsymbol{\sigma} = f(\boldsymbol{\varepsilon}(\mathbf{u})) \quad (3.4)$$

Para calcular el tensor de tensiones σ se deriva la energía elástica definida en (3.1) con respecto al tensor de pequeñas deformaciones:

$$\sigma = \frac{\partial \Psi(\epsilon)}{\partial \epsilon} \quad (3.5)$$

Por lo tanto, derivando y desarrollando la expresión:

$$\begin{aligned} \sigma &= \frac{\partial}{\partial \epsilon} \left(\mu \operatorname{tr}(\epsilon \epsilon) + \frac{\lambda}{2} (\operatorname{tr}(\epsilon))^2 \right) \\ &= \mu \frac{\partial}{\partial \epsilon} \operatorname{tr}(\epsilon \epsilon) + \frac{\lambda}{2} \frac{\partial}{\partial \epsilon} (\operatorname{tr}(\epsilon))^2 \\ &= \mu \frac{\partial (\operatorname{tr}(\epsilon \epsilon))}{\partial \epsilon} + \lambda \operatorname{tr}(\epsilon) \frac{\partial (\operatorname{tr}(\epsilon))}{\partial \epsilon} \end{aligned} \quad (3.6)$$

Se puede demostrar que, derivando las trazas en la expresión (3.6), se obtiene la ley constitutiva que relaciona el tensor de tensiones de Cauchy σ y el tensor de pequeñas deformaciones ϵ :

$$\sigma = 2\mu \epsilon + \lambda \operatorname{tr}(\epsilon) I \quad (3.7)$$

Ejemplo 4:

Considerando un tensor ϵ , cualquiera, como el del ejemplo 3:

$$\epsilon = \begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix}$$

Se puede calcular σ mediante la expresión (3.7).

$$\sigma = 2\mu \begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix} + (0 + 10^{-2}) \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Considerando el módulo de elasticidad y coeficiente de Poisson del acero, y aplicando las expresiones (3.2):

$$E \approx 210 \text{ GPa} = 210 \cdot 10^9 \text{ Pa} \quad \nu = 0.3$$

$$\lambda = \frac{210 \cdot 10^9 \cdot 0.3}{(1 + 0.3)(1 - 2 \cdot 0.3)} = 121,15 \text{ GPa} \quad \mu = \frac{210 \cdot 10^9}{2(1 + 0.3)} = 80,77 \text{ GPa}$$

Sustituyendo estos valores en el tensor de tensiones de Cauchy queda:

$$\boldsymbol{\sigma} = \begin{bmatrix} 1.2115 & 0.01615 \\ 0.01615 & 2.8269 \end{bmatrix} \text{GPa:}$$

3.3. Tensor de elasticidad

El **tensor de elasticidad** \mathcal{C} , de cuarto orden, agrupa las propiedades materiales de un sólido dado, y se define como la derivada del tensor de tensiones de Cauchy $\boldsymbol{\sigma}$ con respecto al tensor de pequeñas deformaciones $\boldsymbol{\varepsilon}$:

$$\mathcal{C} = \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial \boldsymbol{\varepsilon}} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \quad (3.8)$$

Sustituyendo la expresión del tensor de tensiones de Cauchy (3.7) y derivando la expresión se obtiene la expresión del tensor de elasticidad \mathcal{C} :

$$\mathcal{C} = 2\mu \boldsymbol{\mathfrak{I}}^{sym} + \lambda \mathbf{I} \otimes \mathbf{I} \quad \boldsymbol{\mathfrak{I}}^{sym} = \frac{1}{2}(\boldsymbol{\mathfrak{I}} + \boldsymbol{\mathfrak{I}}^T) \quad (3.9)$$

Cuya notación de índices es:

$$\mathcal{C}_{ijkl} = 2\mu \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{ju}) + \lambda \delta_{ij}\delta_{ul} \quad (3.10)$$

Además, teniendo en cuenta que el modelo constitutivo es lineal, se puede escribir $\boldsymbol{\sigma}$ como:

$$\boldsymbol{\sigma} = \mathcal{C} : \boldsymbol{\varepsilon}(\mathbf{u}) \quad (3.11)$$

3.4. Forma fuerte y forma débil en elasticidad lineal

La Ecuación en Derivadas Parciales (EDP) que gobierna el comportamiento de un sólido elástico lineal definido por el dominio Ω (Figura 3.1), junto con las condiciones de contorno en tipo Neumann y Dirichlet constituyen la forma fuerte de problema de condiciones de contorno en elasticidad lineal:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) + \mathbf{f} = \mathbf{0} & \text{en } \Omega \\ \mathbf{u} = \mathbf{u}^* & \text{en } \partial\Omega_D \\ \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) \cdot \mathbf{n} = \mathbf{t} & \text{en } \partial\Omega_N \end{cases} \quad (3.12)$$

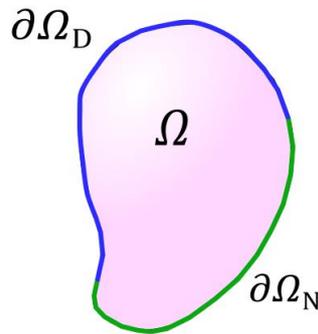


Figura 3.1. Dominio del problema elástico

En este sistema se encuentran los tensores definidos en los apartados anteriores, así como el vector desplazamiento. El operador " $\nabla \cdot$ " es la divergencia de un tensor, en este caso el tensor de Cauchy, por lo que la ecuación matricial constituye un sistema de tres ecuaciones, donde las tres incógnitas serían cada una de las componentes del vector desplazamiento \mathbf{u} . El vector \mathbf{f} constituye la fuerza por unidad de volumen ejercida en el sólido, el vector \mathbf{u}^* es el desplazamiento impuesto en la frontera Dirichlet $\partial\Omega_D$ y el vector \mathbf{t} es la fuerza por unidad de superficie impuesta en la frontera Neumann $\partial\Omega_N$.

Una vez definida la forma fuerte del problema (3.12) se obtendrá su forma débil, también llamada formulación variacional, para posteriormente poder implementar el problema de elasticidad lineal en FreeFEM.

Para este desarrollo se ha de definir una función test o función virtual \mathbf{v} para la incógnita del problema, que es en este caso el desplazamiento \mathbf{u} . Multiplicando la ecuación del problema fuerte por la función test e integrando:

$$\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) + \mathbf{f}) \cdot \mathbf{v} dV = 0 \quad (3.13)$$

Siendo $\mathbf{u} \in H^1(\Omega)$ y $\mathbf{v} \in H_0^1(\Omega)$. Esto es, que tanto el vector desplazamiento como su derivada primera pertenecen al dominio estudiado, y de la misma forma para función test. Además, el valor de la función test es nulo en la frontera Dirichlet.

Continuando con la expresión (3.13):

$$\int_{\Omega} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) \cdot \mathbf{v} dV + \int_{\Omega} \mathbf{f} \cdot \mathbf{v} dV = 0 \quad (3.14)$$

Desarrollando el primer término de la expresión (3.14):

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) \cdot \mathbf{v} dV = \nabla \cdot (\boldsymbol{\sigma}^T \cdot \mathbf{v}) - \boldsymbol{\sigma} : \nabla \mathbf{v} \quad (3.15)$$

El último término de la expresión (3.15) se puede reescribir aplicando la definición del tensor de pequeñas deformaciones para la función test:

$$\boldsymbol{\sigma} : \nabla \mathbf{v} = \boldsymbol{\sigma} : \frac{1}{2} (\nabla \mathbf{v} + \nabla \mathbf{v}^T) = \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{v}) \quad (3.16)$$

Reescribiendo la ecuación (3.15) aplicando (3.16) queda:

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) \cdot \mathbf{v} dV = \nabla \cdot (\boldsymbol{\sigma}^T \cdot \mathbf{v}) - \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{v}) \quad (3.17)$$

Aplicando los cambios a la expresión (3.14), el primer término quedaría

$$\int_{\Omega} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) \cdot \mathbf{v} dV = \int_{\Omega} \nabla \cdot (\boldsymbol{\sigma}^T \cdot \mathbf{v}) dV - \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{v}) dV \quad (3.18)$$

Ahora se puede aplicar el teorema de la divergencia al primero de los términos de la expresión (3.18):

$$\int_{\Omega} \nabla \cdot (\boldsymbol{\sigma}^T \cdot \mathbf{v}) dV = \int_{\partial\Omega_D} \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \mathbf{v} da + \int_{\partial\Omega_N} \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \mathbf{v} da \quad (3.19)$$

Como se ha dicho anteriormente, la función test en todo el contorno Dirichlet es nula, por lo que se anula el primer término de (3.19). En cuanto al término de la frontera Neumann, está constituido por la condición de contorno del problema impuesta para esa frontera, quedando:

$$\int_{\Omega} \nabla \cdot (\boldsymbol{\sigma}^T \cdot \boldsymbol{v}) dV = \int_{\partial\Omega_N} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot \boldsymbol{v} da = \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v} da \quad (3.20)$$

Reagrupando todos los términos desarrollados, se obtiene la expresión:

$$\int_{\Omega} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\boldsymbol{u})) \cdot \boldsymbol{v} dV = \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v} da - \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\boldsymbol{v}) dV \quad (3.21)$$

Añadiendo el término de la fuerza volumétrica y reordenando, obtenemos la **forma débil o formulación variacional del problema de elasticidad lineal**:

$$\int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\boldsymbol{v}) dV - \int_{\partial\Omega_N} \boldsymbol{t} \cdot \boldsymbol{v} da - \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v} dV = 0 \quad (3.22)$$

En esta ecuación, el primer término representa el **trabajo de deformación elástica** que ejerce el material, el cual se opone a deformarse. Los dos términos que restan representan el **trabajo ejercido por las fuerzas externas** sobre el material. De esta forma, el trabajo de deformación elástica del material está en equilibrio con el trabajo ejercido por las fuerzas externas, por eso la ecuación del problema elástico también se conoce como ecuación de equilibrio.

CAPÍTULO 4: FUNDAMENTOS DE LA ELECTRO-MECÁNICA LINEAL

4.1. Campo eléctrico, ley de Faraday y vector de polarización

El *campo eléctrico* E es una magnitud vectorial, que para el problema estudiado se puede definir como gradiente de un *potencial eléctrico* φ .

$$\mathbf{E}(\varphi) = -\nabla\varphi = -\frac{\partial\varphi}{\partial\mathbf{x}} \quad (4.1)$$

Este potencial eléctrico es una de las variables estudiadas en el problema de la electro-mecánica, así como una de las condiciones de contorno que se pueden imponer para el dominio estudiado.

Ejemplo 5:

Dado el potencial eléctrico φ , calcular el campo eléctrico E

$$\varphi = 2 \cdot 10^6 + 10^6 X_1 + 5 \cdot 10^5 X_2$$

Aplicando la expresión (4.1) se obtiene:

$$\mathbf{E} = -\nabla\varphi = -\left[\begin{array}{c} 10^6 \\ 5 \cdot 10^5 \end{array} \right] \text{V/m}$$

Una de las leyes constitutivas del problema de piezoelectricidad es la *Ley de Faraday*, la cual relaciona el rotacional del vector campo eléctrico E con la derivada temporal del campo magnético B .

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (4.2)$$

Como se ha definido el campo eléctrico E como el gradiente de un potencial φ y, por definición, el rotacional del gradiente de un potencial es nulo, se obtiene:

$$\nabla \times \mathbf{E} = \nabla \times (-\nabla\varphi) = -\nabla \times (\nabla\varphi) = \mathbf{0} \quad (4.3)$$

Como se ha definido el campo eléctrico \mathbf{E} como el gradiente de un potencial φ y, por definición, el rotacional del gradiente de un potencial es nulo, se obtiene:

$$\nabla \times \mathbf{E} = \nabla \times (-\nabla\varphi) = -\nabla \times (\nabla\varphi) = \mathbf{0} \quad (4.4)$$

Si se sustituye la expresión (4.4) en (4.2):

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} = \mathbf{0} \quad (4.5)$$

Este resultado significa que, al haber definido el campo eléctrico como el gradiente de un potencial, para que se satisfaga la *Ley de Faraday* debe ser nula la derivada temporal del campo magnético. Para el problema estudiado se considera que no existe campo magnético en el dominio, y por tanto se cumple la ley constitutiva.

El problema de la electro-mecánica para materiales piezoeléctricos estudia un modelo en el que las deformaciones inducen un campo eléctrico en el dominio (sensores piezoeléctricos). De la misma forma, al generarse un campo eléctrico en el dominio se inducen deformaciones (actuadores piezoeléctricos). Además, el modelo requiere materiales transversalmente isótropos, es decir, que se comportan como un isótropo en planos o superficies definidas por su vector normal \mathbf{N} (*Figura 4.1*). Este vector se denomina *vector de polarización* \mathbf{N} o dirección preferencial.

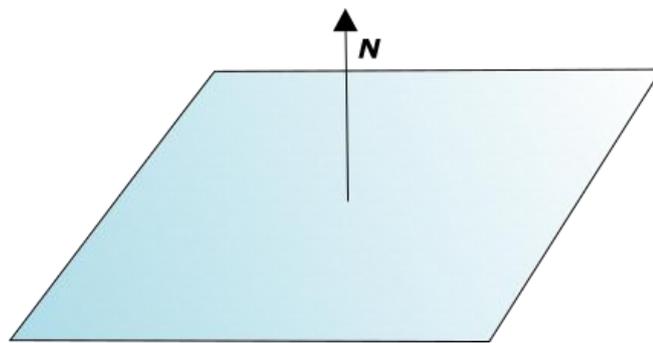


Figura 4.1. Superficie definida por el vector \mathbf{N}

4.2. Energía elástica del problema electromecánico

De forma análoga a la del apartado 3 (ecuación 3.1), se define una **energía elástica** $\Psi(\boldsymbol{\varepsilon}, \mathbf{E}, \mathbf{N})$ para el problema electromecánico:

$$\Psi(\boldsymbol{\varepsilon}, \mathbf{E}, \mathbf{N}) = \mu \operatorname{tr}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}) + \frac{\lambda}{2} (\operatorname{tr}(\boldsymbol{\varepsilon}))^2 - \frac{\varepsilon}{2} \mathbf{E} \cdot \mathbf{E} - \alpha_1 (\mathbf{E} \cdot \mathbf{N}) \operatorname{tr}(\boldsymbol{\varepsilon}) - \alpha_2 (\mathbf{E} \cdot \boldsymbol{\varepsilon} \cdot \mathbf{N}) \quad (4.6)$$

En la ecuación (4.6) se define la energía elástica mediante cuatro términos. Los dos primeros constituyen la parte mecánica, igual que la definida para el problema de elasticidad lineal. El tercer término corresponde a la energía eléctrica, siendo ε la permitividad eléctrica del material estudiado. En dicho término interviene el campo eléctrico \mathbf{E} definido en la expresión (4.1). Los dos últimos términos corresponden a la energía debida al comportamiento piezoelectrico del material, y en ellos intervienen tanto los parámetros mecánicos como los eléctricos, $\boldsymbol{\varepsilon}$ y \mathbf{E} respectivamente, así como el vector de polarización \mathbf{N} . Por último, α_1 y α_2 son los llamados *coeficientes piezoelectricos*.

4.3. Tensor de Cauchy y Desplazamiento eléctrico

Al igual que en el problema de la elasticidad lineal, se obtiene la expresión del tensor de tensiones de Cauchy $\boldsymbol{\sigma}$ mediante la derivada de la expresión de la energía elástica Ψ con respecto al parámetro mecánico $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi), \mathbf{N}) = \frac{\partial \Psi(\boldsymbol{\varepsilon}, \mathbf{E}, \mathbf{N})}{\partial \boldsymbol{\varepsilon}} \quad (4.7)$$

Se sustituyendo la expresión de la energía elástica (4.6) y derivando, se puede demostrar la obtención de la expresión:

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi), \mathbf{N}) = 2\mu\boldsymbol{\varepsilon} + \lambda \operatorname{tr}(\boldsymbol{\varepsilon})\mathbf{I} - \alpha_1 (\mathbf{E} \cdot \mathbf{N})\mathbf{I} - \frac{\alpha_2}{2} (\mathbf{E} \otimes \mathbf{N} + \mathbf{N} \otimes \mathbf{E}) \quad (4.8)$$

Donde el operador $\mathbf{A} \otimes \mathbf{B}$ es un *dyadic product*, un operador matemático aplicable entre vectores definido como:

$$\mathbf{A} \otimes \mathbf{B} = [A_1 B_1, A_1 B_2, A_2 B_1, A_2 B_2] \quad (4.9)$$

Siendo $\mathbf{A} = [A_1, A_2]$ y $\mathbf{B} = [B_1, B_2]$.

En la expresión (4.8) se puede observar cómo el tensor de Cauchy σ , el cual dependía únicamente de parámetros mecánicos en el problema de la elasticidad lineal, ahora depende de parámetros mecánicos y eléctricos, así como de los coeficientes piezoeléctricos y el vector de polarización.

Ejemplo 6:

Se va a calcular el tensor de Cauchy σ para el problema piezoeléctrico considerando los tensores y vectores obtenidos en los ejemplos anteriores:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix} \quad \mathbf{E} = - \begin{bmatrix} 10^6 \\ 5 \cdot 10^5 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Aplicando las expresiones (4.8) y (4.9).

$$\begin{aligned} \boldsymbol{\sigma} = & 2\mu \begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix} + (0 + 10^{-2}) \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \alpha_1 \left(\begin{bmatrix} -10^6 \\ -5 \cdot 10^5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \mathbf{I} \\ & - \alpha_2 \left(\begin{bmatrix} -10^6 \\ -5 \cdot 10^5 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} -10^6 \\ -5 \cdot 10^5 \end{bmatrix} \right) \end{aligned}$$

Considerando $\alpha_1 = 0,1$ y $\alpha_2 = 0,2$, así como el módulo de elasticidad y coeficiente de Poisson del acero, y aplicando las expresiones (3.2):

$$E \approx 210 \text{ GPa} = 210 \cdot 10^9 \text{ Pa} \qquad \nu = 0.3$$

$$\lambda = 121,15 \cdot 10^9 \text{ Pa} \qquad \mu = 80,77 \cdot 10^9 \text{ Pa}$$

Se obtiene el tensor de tensiones de Cauchy:

$$\boldsymbol{\sigma} = \begin{bmatrix} 1.5115 & 0.066154 \\ 0.066154 & 2.9269 \end{bmatrix} \text{ GPa}$$

Un nuevo vector que se define para el problema electromecánico es el **vector desplazamiento eléctrico D** , cuya ley constitutiva lo define como la derivada de la energía elástica Ψ con respecto al parámetro eléctrico, que es el campo eléctrico E .

$$D(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi), \mathbf{N}) = - \frac{\partial \Psi(\boldsymbol{\varepsilon}, \mathbf{E}, \mathbf{N})}{\partial \mathbf{E}} \quad (4.10)$$

Sustituyendo la expresión de la energía elástica (4.6) y derivando, se puede demostrar el resultado:

$$D(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi), \mathbf{N}) = \epsilon \mathbf{E} + \alpha_1 \text{tr}(\boldsymbol{\varepsilon}) \mathbf{N} + \alpha_2 (\boldsymbol{\varepsilon} \mathbf{N}) \quad (4.11)$$

Nuevamente se aprecia que el vector desplazamiento eléctrico D no sólo depende de los parámetros eléctricos, sino que también influyen en él parámetros mecánicos debido a los términos piezoeléctricos de la expresión de la energía elástica (4.6).

Ejemplo 7:

Calcular el vector desplazamiento eléctrico D considerando los tensores y vectores obtenidos en los ejemplos anteriores:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix} \quad \mathbf{E} = - \begin{bmatrix} 10^6 \\ 5 \cdot 10^5 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Aplicando la expresión (4.11).

$$\mathbf{D} = \epsilon \begin{bmatrix} -10^6 \\ -5 \cdot 10^5 \end{bmatrix} + \alpha_1 (0 + 10^{-2}) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_2 \left(\begin{bmatrix} 0 & 10^{-4} \\ 10^{-4} & 10^{-2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$$

Considerando $\alpha_1 = \alpha_2 = 0,1$, y $\epsilon = 3,52 \cdot 10^{-11}$, se obtiene un vector desplazamiento eléctrico:

$$\mathbf{D} = \begin{bmatrix} 9,684 \cdot 10^{-4} \\ -7,6 \cdot 10^{-6} \end{bmatrix} \text{C/m}^2$$

4.4. Tensores de orden superior

Al igual que para el problema de elasticidad, a partir de las expresiones de la energía elástica Ψ así como del tensor de Cauchy y del vector desplazamiento eléctrico se pueden obtener una serie de tensores de orden superior, los cuales agrupan las propiedades del sólido piezoeléctrico.

El primero de ellos es el ya conocido **tensor de elasticidad** \mathcal{C} , de cuarto orden, informa sobre las propiedades mecánicas del sólido y se obtiene al derivar tensor de tensiones de Cauchy σ con respecto al tensor de pequeñas deformaciones $\boldsymbol{\varepsilon}$:

$$\mathcal{C} = \frac{\partial^2 \Psi}{\partial \boldsymbol{\varepsilon} \partial \boldsymbol{\varepsilon}} = \frac{\partial \sigma}{\partial \boldsymbol{\varepsilon}} \quad (4.12)$$

Sustituyendo la expresión del tensor de tensiones de Cauchy (4.7) y derivando la expresión se obtiene la expresión del tensor de elasticidad \mathcal{C} , cuya notación de índices es:

$$\mathcal{C} = 2\mu \boldsymbol{\mathfrak{I}}^{sym} + \lambda \mathbf{I} \otimes \mathbf{I} \quad \boldsymbol{\mathfrak{I}}^{sym} = \frac{1}{2}(\boldsymbol{\mathfrak{I}} + \boldsymbol{\mathfrak{I}}^T) \quad (4.13)$$

cuya notación de índices es:

$$\mathcal{C}_{ijkl} = 2\mu \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{ju}) + \lambda \delta_{ij}\delta_{ul} \quad (4.14)$$

Derivando la expresión de la energía elástica Ψ con respecto al tensor de pequeñas deformaciones $\boldsymbol{\varepsilon}$ y el campo eléctrico \boldsymbol{E} se obtiene el llamado **tensor piezoeléctrico** \mathcal{P} , tensor de tres dimensiones que informa sobre el comportamiento piezoeléctrico del material.

$$\mathcal{P} = -\frac{\partial^2 \Psi}{\partial \boldsymbol{E} \partial \boldsymbol{\varepsilon}} \quad (4.15)$$

Sustituyendo la expresión de la energía elástica (4.6) y derivando, se obtiene la expresión del tensor piezoeléctrico:

$$\mathcal{P}_{ijk} = \alpha_1 \mathbf{N} \otimes \mathbf{I} + \frac{\alpha_2}{2} (\delta_{ij} N_k + \delta_{ik} N_j) = \alpha_1 N_i \delta_{jk} + \frac{\alpha_2}{2} (\delta_{ij} N_k + \delta_{ik} N_j) \quad (4.16)$$

Por último, derivando el vector desplazamiento eléctrico \mathbf{D} de nuevo con respecto al vector campo eléctrico \mathbf{E} se obtiene el llamado **tensor dieléctrico** \mathcal{G} .

$$\mathcal{G} = -\frac{\partial^2 \Psi}{\partial \mathbf{E} \partial \mathbf{E}} = \frac{\partial \mathbf{D}}{\partial \mathbf{E}} \quad (4.17)$$

Sustituyendo la expresión del desplazamiento eléctrico la energía elástica (4.10) y derivando, se obtiene la expresión del tensor dieléctrico:

$$\mathcal{G} = \epsilon \mathbf{I} \quad (4.18)$$

Además, teniendo en cuenta el modelo constitutivo lineal, se pueden expresar $\boldsymbol{\sigma}$ y \mathbf{D} de la siguiente forma:

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) - \mathcal{P}^T \cdot \mathbf{E}(\varphi) \quad \mathbf{D} = -\mathcal{P} : \boldsymbol{\varepsilon}(\mathbf{u}) - \mathcal{G} \cdot \mathbf{E}(\varphi) \quad (4.19)$$

4.5. Forma fuerte y forma débil en electro-mecánica lineal.

El sistema de ecuaciones en derivadas parciales que gobierna el comportamiento de un sólido piezoeléctrico definido por el dominio Ω , junto con las condiciones de contorno en tipo Neumann y Dirichlet constituyen la forma fuerte del problema de condiciones de contorno en electro-mecánica lineal:

$$\left\{ \begin{array}{ll} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi)) + \mathbf{f} = \mathbf{0} & \text{en } \Omega \\ \nabla \cdot \mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi)) - \rho = 0 & \text{en } \Omega \\ \mathbf{u} = \mathbf{u}^* & \text{en } \partial\Omega_{Du} \\ \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi)) \cdot \mathbf{n} = \mathbf{t} & \text{en } \partial\Omega_{Nu} \\ \varphi = \varphi^* & \text{en } \partial\Omega_{D\varphi} \\ \mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi)) \cdot \mathbf{n} = -w & \text{en } \partial\Omega_{N\varphi} \end{array} \right. \quad (4.20)$$

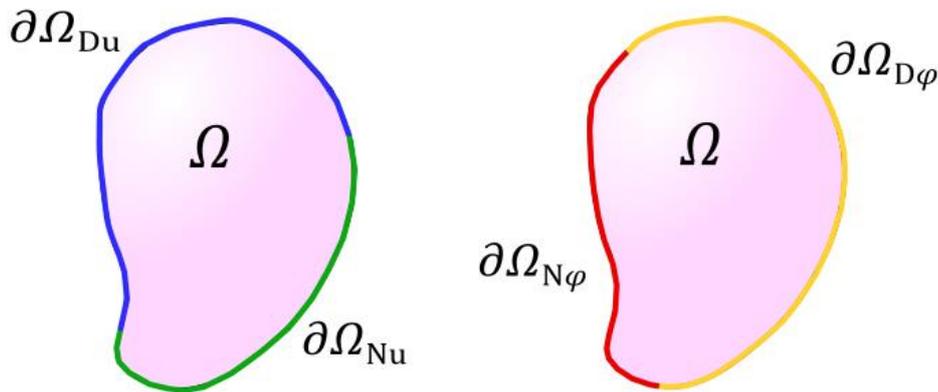


Figura 4.2. Fronteras mecánicas y eléctricas del dominio Ω

Este sistema está constituido por la EDP de la elasticidad lineal expuesta en el capítulo 3.4, junto con la nueva ecuación correspondiente a la divergencia del desplazamiento eléctrico \mathbf{D} . El término ρ es una carga eléctrica por unidad de volumen, mientras que φ^* es un potencial eléctrico impuesto en la frontera Dirichlet $\partial\Omega_{D\varphi}$, y w una carga eléctrica por unidad de superficie impuesta en la frontera Neumann $\partial\Omega_{N\varphi}$.

Otro aspecto importante del sistema (4.12) es que las fronteras del dominio son independientes para cada ecuación (Figura 4.2). Esto quiere decir que existe una frontera Dirichlet y Neumann para la ecuación mecánica, $\partial\Omega_{Du}$ y $\partial\Omega_{Nu}$ respectivamente, y de la

misma forma ocurre con la ecuación eléctrica, $\partial\Omega_{D\varphi}$ y $\partial\Omega_{N\varphi}$, donde se aplicarán las respectivas condiciones de contorno.

Por último, cabe destacar que el operador " $\nabla \cdot$ " para la segunda ecuación representa la divergencia de un vector, es decir, un escalar. Como ya se expuso, para la primera ecuación representa la divergencia de un tensor, que es un vector de tres componentes para el caso tridimensional, esto es, una ecuación vectorial equivalente a tres ecuaciones escalares. Por tanto, el sistema constaría de cuatro ecuaciones, donde las cuatro incógnitas son cada una de las componentes del vector desplazamiento \mathbf{u} junto con el potencial eléctrico φ . La denominación de problema acoplado quiere decir que, tal y como se ha descrito en el capítulo 4.3, los cambios en los parámetros eléctricos afectan a la ecuación mecánica (tensor de Cauchy) y viceversa, ya que $\boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi))$ y $\mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi))$.

Para obtener la forma débil del problema, se ha de asignar una función test a cada una de las incógnitas, que van a ser $\delta\mathbf{u}$ para el desplazamiento \mathbf{u} , y $\delta\varphi$ para el potencial eléctrico φ . Para la primera ecuación ya se obtuvo su formulación variacional en el capítulo 3.4 (ecuación 3.22). Para la ecuación del desplazamiento eléctrico se sigue un procedimiento análogo, ya que la ecuación tiene la misma forma, y se obtiene:

$$-\int_{\Omega} \mathbf{D} \cdot \mathbf{E}(\delta\varphi) dV + \int_{\partial\Omega_N} w \delta\varphi da + \int_{\Omega} \rho \delta\varphi dV = 0 \quad (4.21)$$

Finalmente, sumando las formas débiles de ambas ecuaciones (3.22) y (4.21) se obtiene la *forma débil del problema multifísico de la electro-mecánica lineal para materiales piezoeléctricos*:

$$\begin{aligned} & \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta\mathbf{u}) dV - \int_{\partial\Omega_{Nu}} \mathbf{t} \cdot \delta\mathbf{u} da - \int_{\Omega} \mathbf{f} \cdot \delta\mathbf{u} dV \\ & - \int_{\Omega} \mathbf{D} \cdot \mathbf{E}(\delta\varphi) dV + \int_{\partial\Omega_{N\varphi}} w \delta\varphi da + \int_{\Omega} \rho \delta\varphi dV = 0 \end{aligned} \quad (4.22)$$

De nuevo, los términos del tensor de Cauchy $\boldsymbol{\sigma}$ y vector desplazamiento eléctrico \mathbf{D} constituyen el **trabajo interno del material**, mientras que el resto de términos representan el **trabajo ejercido por las fuerzas y cargas externas**.

Para modelizar un problema dinámico (sin disipación), habría que añadir un término de aceleración:

$$\int_{\Omega} \rho \mathbf{a} \cdot \delta \mathbf{u} \, dV + \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, dV - \int_{\partial \Omega_{Nu}} \mathbf{t} \cdot \delta \mathbf{u} \, da - \int_{\Omega} \mathbf{f} \cdot \delta \mathbf{u} \, dV - \int_{\Omega} \mathbf{D} \cdot \mathbf{E}(\delta \varphi) \, dV + \int_{\partial \Omega_{N\varphi}} w \delta \varphi \, da + \int_{\Omega} \rho \delta \varphi \, dV = 0 \quad (4.23)$$

donde ρ es la densidad másica del material y \mathbf{a} es la aceleración, es decir, la derivada temporal segunda del desplazamiento \mathbf{u}

$$\mathbf{a} = \frac{\partial^2 \mathbf{u}}{\partial t^2} \quad (4.24)$$

CAPÍTULO 5: FUNDAMENTOS DE ELEMENTOS FINITOS

Este capítulo pretende introducir la teoría básica de elementos finitos, explicar la razón para escribir la forma débil del problema y expresar la discretización del problema de la piezoelectricidad.

5.1. Método de los elementos finitos

La modelización matemática es una herramienta muy potente a la hora de entender fenómenos físicos en ciencia. Estos modelos se describen mediante sistemas de ecuaciones en derivadas parciales (EDPs), a partir de los cuales se plantean problemas de gran dificultad. Estos problemas, en muchas ocasiones, no tienen solución exacta clásica, o bien son extremadamente complejos. El *método de los elementos finitos* (MEF), desarrollado por el matemático alemán Richard Courant en 1943 (*Figura 5.1*), es uno de los métodos numéricos más recurridos en ingeniería resolver numéricamente dichos modelos.

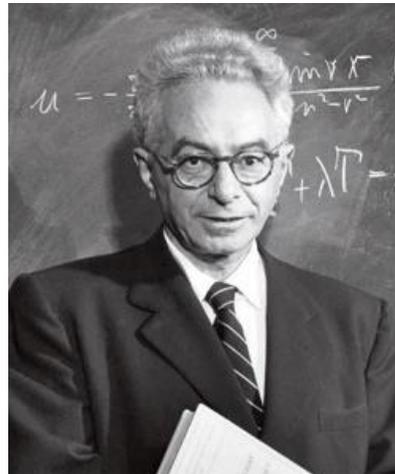


Figura 5.1. Richard Courant

El método de los elementos finitos consiste en desarrollar la forma fuerte del problema, consistente de un sistema de ecuaciones en derivadas parciales, hasta obtener una expresión del tipo:

$$KU = F \tag{5.1}$$

En la ecuación matricial (5.1), U representa el **vector incógnita**, que contiene todas las incógnitas del problema estudiado, mientras que K es la llamada **matriz de rigidez**, que multiplicada por el vector incógnita da como resultado el **vector de términos independientes** F . La forma matricial de la ecuación se debe a que esta operación se lleva a cabo en todos los puntos del dominio estudiado, cuyo espacio se discretiza mediante mallas.

Para explicar el método, se va a utilizar como ejemplo el problema unidimensional de una cuerda elástica sujeta en los extremos, sobre la cual se aplica una carga puntual f en el centro, tal y como se aprecia en la (Figura 5.2), cuya forma fuerte es:

$$\begin{cases} -(ku')' = f & \text{en }]0, L[\\ u(0) = u(L) = 0 \end{cases} \quad (5.2)$$

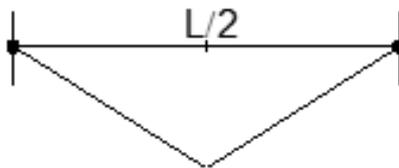


Figura 5.2. Problema de la cuerda sujeta en los extremos

El primer paso es escribir el **Problema Variacional**, que consiste en aplicar el principio de los trabajos virtuales e integrar las EDPs de la forma fuerte del problema. Para este problema, queda la expresión:

$$\int_0^L k(x)u'(x)v'(x)dx = \int_0^L f(x)v(x)dx \quad (5.3)$$

Al igual que sucedía en los capítulos anteriores, para este problema el término a la izquierda de la igualdad representa el trabajo interno de deformación de la cuerda, mientras que el término a la derecha es el trabajo ejercido por las fuerzas exteriores. Si a f le damos como valor la delta de Dirac, $f = \delta$, entonces se simplifica la expresión:

$$\int_0^L k(x)u'(x)v'(x)dx = v(L/2) \quad (5.4)$$

Para todo el espacio definido, se cumple que $u \in H^1(\Omega)$ y $v \in H_0^1(\Omega)$. Esto es, que tanto la incógnita como su derivada primera pertenecen al dominio estudiado, y de la misma forma para función test. Además, la función test es nula en el contorno Dirichlet, $v(0) = v(L) = 0$.

Una vez obtenida la expresión (5.4), se procede a la **discretización del problema variacional**. Para ello, se definen todos los puntos que se van a evaluar de forma discreta para la resolución del problema. A estos puntos se les llama nodos, y todos ellos cubren el dominio del problema y constituyen lo que se conoce como malla. Hay muchos tipos distintos de mallado, pero el más sencillo y el que se utiliza normalmente es el compuesto por triángulos y tetraedros. Para el problema del ejemplo, el mallado consiste en dividir la línea en segmentos, cuyos extremos son los nodos. A los segmentos, triángulos o tetraedros que unen los nodos se les conoce como espacios de aproximación, y en ellos se aproxima el valor que toma la incógnita en ese punto según el de los nodos adyacentes. Los más sencillos y usados son los que aproximan mediante polinomios de grado 1, y se les conoce como elementos finitos de Lagrange de grado 1.

Para el caso unidimensional del ejemplo, estas funciones de aproximación serían:

$$\phi_j(x) = \begin{cases} 1 - \frac{|x - c_j|}{h} & \text{si } c_{j-1} \leq x \leq c_{j+1} \\ 0 & \text{en caso contrario} \end{cases} \quad (5.5)$$

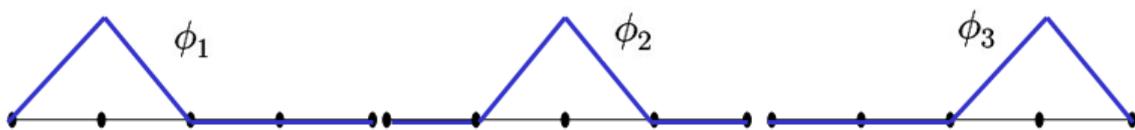


Figura 5.3. Funciones de aproximación del problema de la cuerda sujeta a extremos

En la expresión (5.5), ϕ_j es la función de aproximación para el nodo j , c_j es la posición del nodo que se analiza, x la posición del punto que aproxima la función y h es la distancia entre nodos. Como se aprecia en la *Figura 5.3*, la función define una forma triangular donde el vértice tiene valor 1 y está en el nodo estudiado y es nula en los nodos adyacentes. De esta forma, superponiendo el valor de la función en cada nodo y sus aproximaciones en estos espacios, se obtiene una distribución suave en los resultados.

Además de las funciones de aproximación, se utilizan las llamadas funciones de forma para asignar el valor de la incógnita en cada punto, siendo estas funciones:

$$\phi_i(c_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad (5.6)$$

Por tanto, se pueden discretizar las funciones u y v , para un espacio finito H_h de dimensión n (n nodos), de la forma:

$$u_h(c_j) = \sum_{i=1}^n u_h^i \phi_i \quad v(x) = \phi_j(x) \quad (5.7)$$

Donde u_h^i es el valor de la incógnita en cada nodo del dominio finito. Si ahora se sustituyen las expresiones (5.7) en la formulación variacional del problema (5.4) y se ordenan los términos, se obtiene:

$$\sum_{i=1}^n u_h^i \int_0^L \phi_i'(x) \phi_j'(x) dx = \phi_j(L/2) \quad (5.8)$$

La expresión obtenida es la discretización del problema. Si se observa de nuevo la expresión (5.1), se aprecia la similitud con la (5.8), ya que el término del sumatorio es vector incógnita U , el término de la integral de las funciones de forma es la matriz de rigidez K y el término a la derecha de la igualdad es el vector de términos independientes F .

Esta discretización a partir de la forma débil del problema se puede hacer para problemas sencillos, pero a medida que aumenta la complejidad, así como el número de nodos, es necesaria la capacidad computacional de un ordenador, y programas especializados en elementos finitos. Con el programa de software libre FreeFEM, basta con escribir la forma débil del problema para su resolución.

5.2. Discretización del problema de la piezoelectricidad

Como se ha explicado anteriormente, y a partir de la forma débil del problema de la electro-mecánica lineal, se busca encontrar $\mathbf{u}_h = \sum_{e=1}^n \mathbf{u}_h^i \phi_i$ y $\varphi_h = \sum_{e=1}^n \varphi_h^i \phi_i$ para todos los nodos del espacio finito H_h tal que:

$$\begin{aligned}
 D\Pi[\delta\mathbf{u}] &= \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta\mathbf{u}) dV - \int_{\partial\Omega_{Nu}} \mathbf{t} \cdot \delta\mathbf{u} da - \int_{\Omega} \mathbf{f} \cdot \delta\mathbf{u} dV = \mathbf{0} \\
 D\Pi[\delta\varphi] &= - \int_{\Omega} \mathbf{D} \cdot \mathbf{E}(\delta\varphi) dV + \int_{\partial\Omega_{N\varphi}} w \delta\varphi da + \int_{\Omega} \rho \delta\varphi dV = 0
 \end{aligned} \tag{5.9}$$

La misma discretización que se utiliza para los desplazamientos \mathbf{u} y el potencial eléctrico φ se usa para sus respectivas funciones test, $\delta\mathbf{u}$ y $\delta\varphi$. En cuanto al tensor de Cauchy $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi))$ y el vector desplazamiento eléctrico $\mathbf{D} = \mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}), \mathbf{E}(\varphi))$, se pueden reescribir en función de los tensores de orden superior como:

$$\begin{aligned}
 \boldsymbol{\sigma} &= \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) - \mathcal{P}^T \cdot \mathbf{E}(\varphi) \\
 \mathbf{D} &= -\mathcal{P} : \boldsymbol{\varepsilon}(\mathbf{u}) - \mathcal{G} \cdot \mathbf{E}(\varphi)
 \end{aligned} \tag{5.10}$$

Si se sustituyen las ecuaciones (5.10) en (5.9), operando queda:

$$\begin{aligned}
 D\Pi[\delta\mathbf{u}] &= \int_{\Omega} \boldsymbol{\varepsilon}(\delta\mathbf{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) dV - \int_{\Omega} \boldsymbol{\varepsilon}(\delta\mathbf{u}) : \mathcal{P}^T \cdot \mathbf{E}(\varphi) dV \\
 &\quad - \int_{\partial\Omega_{Nu}} \mathbf{t} \cdot \delta\mathbf{u} da - \int_{\Omega} \mathbf{f} \cdot \delta\mathbf{u} dV = \mathbf{0} \\
 D\Pi[\delta\varphi] &= - \int_{\Omega} \mathbf{E}(\delta\varphi) \cdot \mathcal{P} : \boldsymbol{\varepsilon}(\mathbf{u}) dV - \int_{\Omega} \mathbf{E}(\delta\varphi) \cdot \mathcal{G} \cdot \mathbf{E}(\varphi) dV \\
 &\quad + \int_{\partial\Omega_{N\varphi}} w \delta\varphi da + \int_{\Omega} \rho \delta\varphi dV = 0
 \end{aligned} \tag{5.11}$$

Si se escriben las variables y funciones test utilizando la discretización de elementos finitos estándar:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}^b \phi_u^b & \delta \mathbf{u} &= \delta \mathbf{u}^a \phi_u^a \\ \varphi &= \varphi^b \phi_\varphi^b & \delta \varphi &= \delta \varphi^a \phi_\varphi^a \end{aligned} \quad (5.12)$$

Se pueden discretizar los primeros términos de las ecuaciones (5.11), quedando:

$$\begin{aligned} D\Pi[\delta \mathbf{u}] &= \sum_{e=1}^n \delta \mathbf{u}^a \cdot \left(\int_{\Omega_e} \nabla \phi_u^a : \mathbf{c}^{sym} : \nabla \phi_u^b dV \right) \mathbf{u}^b \\ &\quad + \delta \mathbf{u}^a \left(\int_{\Omega_e} \nabla \phi_u^a (-\mathcal{P}^T)^{sym} \nabla \phi_\varphi^b dV \right) \varphi^b \\ D\Pi[\delta \varphi] &= \sum_{e=1}^n \delta \varphi^a \cdot \left(\int_{\Omega_e} \nabla \phi_\varphi^a (-\mathcal{P})^{sym} \nabla \phi_u^b dV \right) \mathbf{u}^b \\ &\quad + \delta \varphi^a \left(\int_{\Omega_e} \nabla \phi_\varphi^a (-\mathcal{G}) \nabla \phi_\varphi^b dV \right) \varphi^b \end{aligned} \quad (5.13)$$

Donde:

$$\mathbf{c}_{ijkl}^{sym} = \frac{1}{4} (\mathbf{c}_{ijkl} + \mathbf{c}_{jikl} + \mathbf{c}_{ijlk} + \mathbf{c}_{jilk}) \quad \mathcal{P}_{ijk}^{sym} = \frac{1}{2} (\mathcal{P}_{ijk} + \mathcal{P}_{ikj}) \quad (5.14)$$

Las ecuaciones (5.13) pueden escribirse agrupando las funciones de forma como términos de la matriz de rigidez \mathbf{K} :

$$\begin{aligned} D\Pi[\delta \mathbf{u}] &= \sum_{e=1}^n \delta \mathbf{u}^a \cdot \mathbf{K}_{uu}^{e,ab} \mathbf{u}^b + \delta \mathbf{u}^a \mathbf{K}_{u\varphi}^{e,ab} \varphi^b \\ D\Pi[\delta \varphi] &= \sum_{e=1}^n \delta \varphi^a \cdot \mathbf{K}_{\varphi u}^{e,ab} \mathbf{u}^b + \delta \varphi^a \mathbf{K}_{\varphi\varphi}^{e,ab} \varphi^b \end{aligned} \quad (5.15)$$

Donde:

$$\begin{aligned}
\mathbf{K}_{uu}^{e,ab} &= \int_{\Omega_e} \nabla \phi_u^a : \mathbf{C}^{sym} : \nabla \phi_u^b dV & \mathbf{K}_{u\varphi}^{e,ab} &= \int_{\Omega_e} \nabla \phi_u^a (-\mathcal{P}^T)^{sym} \nabla \phi_\varphi^b dV \\
\mathbf{K}_{\varphi u}^{e,ab} &= \int_{\Omega_e} \nabla \phi_\varphi^a (-\mathcal{P})^{sym} \nabla \phi_u^b dV & \mathbf{K}_{\varphi\varphi}^{e,ab} &= \int_{\Omega_e} \nabla \phi_\varphi^a (-\mathcal{G}) \nabla \phi_\varphi^b dV
\end{aligned} \tag{5.16}$$

Finalmente, el ensamblaje a lo largo de todos los elementos e de la malla lleva a $\mathbf{K}\mathbf{U} = \mathbf{F}$, donde la matriz de rigidez tiene una componente puramente mecánica, y otra eléctrica.

$$\begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{u\varphi} \\ \mathbf{K}_{\varphi u} & \mathbf{K}_{\varphi\varphi} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \varphi \end{bmatrix} = \mathbf{F} \tag{5.17}$$

Para la discretización del problema dinámico, la aceleración ya se definió en capítulos anteriores como la derivada temporal segunda del desplazamiento:

$$\mathbf{a} = \frac{\partial^2 \mathbf{u}}{\partial t^2} \tag{5.18}$$

Al discretizarla queda:

$$\mathbf{a} = \mathbf{a}^b \phi_u^b = \frac{\partial^2 \mathbf{u}^b}{\partial t^2} \phi_u^b \tag{5.19}$$

Introduciendo el término discretizado en su término de la forma débil y discretizando, queda:

$$\int_{\Omega} \rho \mathbf{a} \cdot \delta \mathbf{u} dV = \sum_{e=1}^n \int_{\Omega} \rho \frac{\partial^2 \mathbf{u}^b}{\partial t^2} \phi_u^b \cdot \delta \mathbf{u}^a \phi_u^a dV \tag{5.20}$$

Reordenando términos queda:

$$\sum_{e=1}^n \delta \mathbf{u}^a \cdot \left(\int_{\Omega} \rho \phi_u^a \phi_u^b \mathbf{I} dV \right) \frac{\partial^2 \mathbf{u}^b}{\partial t^2} \tag{5.21}$$

Siendo

$$\int_{\Omega} \rho \phi_u^a \phi_u^b \mathbf{I} dV = \mathbf{M}_u^{e,ab} \quad (5.22)$$

La matriz \mathbf{M} se conoce como matriz de masas, y es la que multiplica al término de la aceleración en el problema dinámico, quedando finalmente:

$$\mathbf{M} \frac{\partial^2 \mathbf{U}}{\partial t^2} + \mathbf{K} \mathbf{U} = \mathbf{F} \quad (5.23)$$

Por último, para determinar cómo se relaciona la derivada temporal segunda del desplazamiento con respecto al mismo, se recurre a la llamada discretización temporal. Entre los distintos integradores temporales, en este caso se aplicará el *integrador temporal Newmar-beta*. Este integrador consiste en expresar la aceleración y velocidad en un instante n+1 mediante las variables en el instante n. Las expresiones que relacionan estas variables son:

$$\begin{aligned} \mathbf{a}_{n+1} &= \frac{\mathbf{u}_{n+1} - \mathbf{u}_p}{\beta \Delta t^2} & \mathbf{u}_p &= \mathbf{u}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2} (1 - 2\beta) \mathbf{a}_n \\ \mathbf{v}_{n+1} &= \mathbf{v}_p + \frac{\gamma}{\beta \Delta t} (\mathbf{u}_{n+1} - \mathbf{u}_p) & \mathbf{v}_p &= \mathbf{v}_n + (1 - \gamma) \Delta t \mathbf{a}_n \end{aligned} \quad (5.24)$$

En estas expresiones, β y γ pueden tomar distintos valores, normalmente 0,25 y 0,5 respectivamente, y el parámetro Δt es el salto temporal entre un los instantes discretizados.

CAPÍTULO 6: SIMULACIÓN NUMÉRICA EN PIEZOELECTRICIDAD LINEAL EN FREEFEM

En este capítulo se expondrán los resultados de aplicar los conceptos estudiados en los capítulos 4 y 5 a una serie de ejemplos en el software FreeFEM. Se explicarán los aspectos más importantes del código para un problema genérico y, si procede, las particularidades de cada ejemplo junto a sus resultados visualizados en el software gráfico *Paraview*. Todos los programas están en el apartado ANEXOS al final del documento.

6.1. Código genérico

El primer paso es cargar los paquetes de FreeFEM necesarios para el problema. En este caso, necesitaremos el paquete “iovtk” para exportar los datos a Paraview, así como “gmsht” y “msh3” para los problemas cuya malla esté hecha en el software GMSH. Una vez cargados los paquetes, el primer paso es crear una malla en código, o bien importarla desde un programa externo.

Lo siguiente es introducir todos los parámetros físicos del problema de la electromecánica lineal, los cuales fueron introducidos en el capítulo 4. Para los casos estudiados se consideran nulas la carga eléctrica por unidad de superficie w , la fuerza por unidad de volumen \mathbf{f} y la carga eléctrica por unidad de volumen ρ . El resto de términos serían:

```
real lambda=1e9; //coeficiente de Lamé
real mu=1e9; //coeficiente de Lamé
real f=0; //fuerza por unidad de volumen
real ro=0; //carga eléctrica por unidad de volumen
real epsilon= 4*8.8e-12; //Permitividad eléctrica
real alpha1= 0.1; // coeficiente piezoeléctrico
real alpha2= 0.1; //coeficiente piezoeléctrico
```

A continuación, se define el espacio de elementos finitos de elementos de lagrange P1 para la malla Th, con todas las variables incógnitas y funciones test que intervienen en el problema:

```
fespace Vh(Th,P1); //elementos de Lagrange P1  
Vh u1,u2,u3,phi,v1,v2,v3,alpha,sigmaVm;
```

Para escribir el problema de la forma más limpia posible, todos los términos que intervienen en el tensor de tensores de Cauchy σ y en el vector desplazamiento eléctrico D , así como ellos mismos, se definen previamente mediante macros:

```
macro IdentityMatrix [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0] //EOM  
macro Trace(a) (a[0] + a[4] + a[8]) //EOM  
macro epsilonmatrix(u1,u2,u3) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dz(u1) + dx(u3)), 0.5*(dy(u1) + dx(u2)), dy(u2), 0.5*(dz(u2) + dy(u3)), 0.5*(dz(u1) + dx(u3)),0.5*(dz(u2) + dy(u3)), dz(u3)] //EOM  
macro Campoelectrico(phi) [-dx(phi),-dy(phi),-dz(phi)]//EOM  
macro N [0,0,1]//EOM  
macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[0]*B[2], A[1]*B[0],A[1]*B[1],A[1]*B[2],A[2]*B[0],A[2]*B[1],A[2]*B[2]] //EOM  
macro mult (A,b)[A[0]*b[0]+A[1]*b[1]+A[2]*b[2], A[3]*b[0]+A[4]*b[1]+A[5]*b[2], A[6]*b[0]+A[7]*b[1]+A[8]*b[2]]//EOM  
macro stress  
(u1,u2,u3,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2,u3))  
+2.*mu*epsilonmatrix(u1,u2,u3) -  
alpha1*(Campoelectrico(phi))*N)*IdentityMatrix -  
alpha2*(DyadicProduct(Campoelectrico(phi),N) +  
DyadicProduct(N,Campoelectrico(phi))))//EOM
```

```
macro desplazamiento(u1,u2,u3, phi)(epsilon*Camptoelectrico(phi) +
alpha1*Trace(epsilonmatrix(u1,u2,u3))*N +
alpha2*mult(epsilonmatrix(u1,u2,u3),N) //EOM
```

Una vez definidos todos los macros, es fácil escribir la formulación variacional del problema (expresión 4.14) que FreeFEM es capaz de resolver. Para este problema, las etiquetas 2, 3 y 4 son las que se asignan a las distintas fronteras del dominio cuando se crea la malla. Se recurre a ellas para introducir las condiciones de contorno.

```
solve Problem([u1,u2,u3,phi],[v1,v2,v3,alpha])=
    int3d(Th)(stress(u1,u2,u3,phi)'*epsilonmatrix(v1,v2,v3))
    -int3d(Th)(desplazamiento(u1,u2,u3,phi)'*Camptoelectrico(alpha))
    +on(2,u1=0.0,u2=0.0,u3=0.0)
    +on(3,phi=10000000)
    +on(4,phi=0);
```

Para el análisis de los resultados, se puede calcular la tensión equivalente de Von Mises σ_{VM} como:

```
sigmaVm = sqrt(stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[0]
    +stress(u1,u2,u3,phi)[4]*stress(u1,u2,u3,phi)[4]
    +stress(u1,u2,u3,phi)[8]*stress(u1,u2,u3,phi)[8]
    -stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[4]
    -stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[8]
    -stress(u1,u2,u3,phi)[4]*stress(u1,u2,u3,phi)[8]
    +3*(stress(u1,u2,u3,phi)[1]*stress(u1,u2,u3,phi)[1]
    +stress(u1,u2,u3,phi)[2]*stress(u1,u2,u3,phi)[2]
    +stress(u1,u2,u3,phi)[5]*stress(u1,u2,u3,phi)[5]));
```

Se puede generar la malla deformada mediante el comando “movemesh”, aplicando un factor de escala para hacer visibles los desplazamientos, ya que son muy pequeños (pequeñas deformaciones). Por último, se guarda el archivo vtk para exportarlo a Paraview y mostrar los resultados.

6.2. Sensor piezoeléctrico bidimensional

El primero de los problemas propuestos consiste en un sensor piezoeléctrico bidimensional de geometría cuadrada, a la que llamaremos matriz, con una inclusión circular, cuyas propiedades variarán para distintos casos estudiados. El cuerpo está sometido a desplazamiento vertical en su cara superior, y está empotrado en la inferior. Además, la cara inferior está conectada a tierra. Las propiedades y parámetros de la matriz se encuentran reflejados en la *Figura 6.2*.

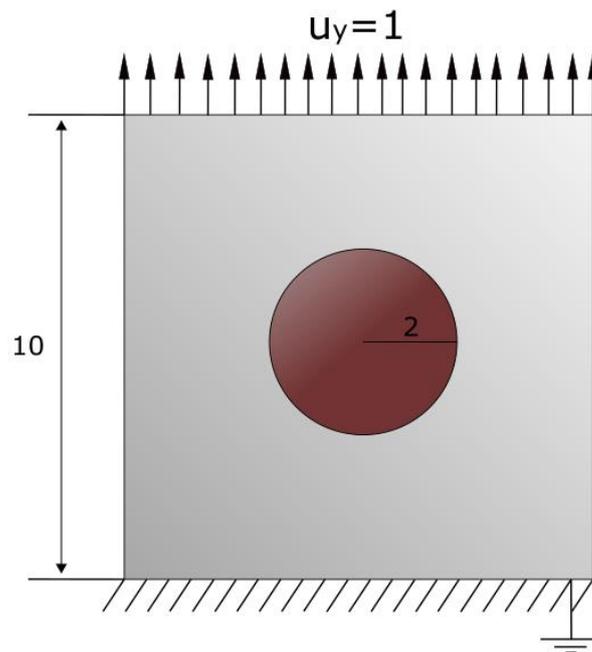


Figura 6.1. Sensor piezoeléctrico bidimensional

Dato	Valor
Arista	10 m
Radio	2 m
Desplazamiento (u_y)	1 m
Parámetro λ	10^9 Pa
Parámetro μ	10^9 Pa
Parámetro α_1	10^{-3} C/m
Parámetro α_2	10^{-3} C/m
Parámetro ϵ	$3,52 \cdot 10^{-11}$ C ² /Nm ²
Vector N	[1,0]

Figura 6.2. Parámetros físicos de la matriz del sensor

En cuanto a los parámetros físicos de la inclusión, se tomará para todos los casos que los parámetros piezoeléctricos son nulos $\alpha_1 = \alpha_2 = 0$. Para los coeficientes de Lamé, se estudiarán cuatro casos según el parámetro ζ , el cual se define como:

$$\zeta = \frac{\text{Coef. Lamé}_{\text{inclusión}}}{\text{Coef. Lamé}_{\text{matriz}}} \quad (6.1)$$

- **Caso 1: Inclusión blanda:** $\zeta = 10^{-3}$
- **Caso 2:** $\zeta = 10^{-2}$
- **Caso 3:** $\zeta = 1$
- **Caso 4: Inclusión dura:** $\zeta = 10$

Para diferenciar la matriz y la inclusión y asignar los parámetros correspondientes en cada caso, se genera un espacio de elementos finitos P0 y se recurre al comando “region” de esta forma:

```

fespace Ph(Th,P0); //Espacio de elementos de Lagrange P0 para definir la región
Ph reg=region;
int ninterior=reg(5,5);//región interior del círculo
int nexterior=reg(9,9); //región exterior del círculo
cout<<"ninterior = "<<ninterior<<"nexterior= "<<nexterior<<endl;
Ph lambda=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de lambda
Ph mu=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de mu
Ph alpha1=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha1
Ph alpha2=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha2

```

El resto del código es similar al expuesto en el capítulo 6.1. Ejecutando el programa para cada uno de los casos propuestos, se obtienen los siguientes resultados:

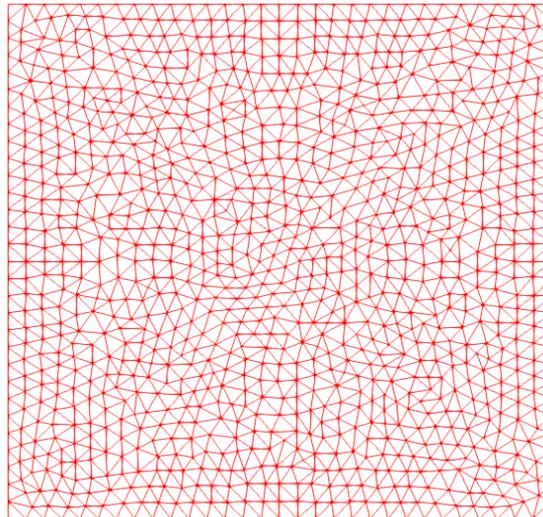


Figura 6.3. Malla del sensor piezoeléctrico bidimensional

La malla es trivial, creada directamente desde FreeFEM. Es una malla grosera, ya que no es necesario afinarla más al ser un ejemplo meramente ilustrativo. En la siguiente figura se muestran, para los cuatro casos, la malla original junto con la malla deformada.

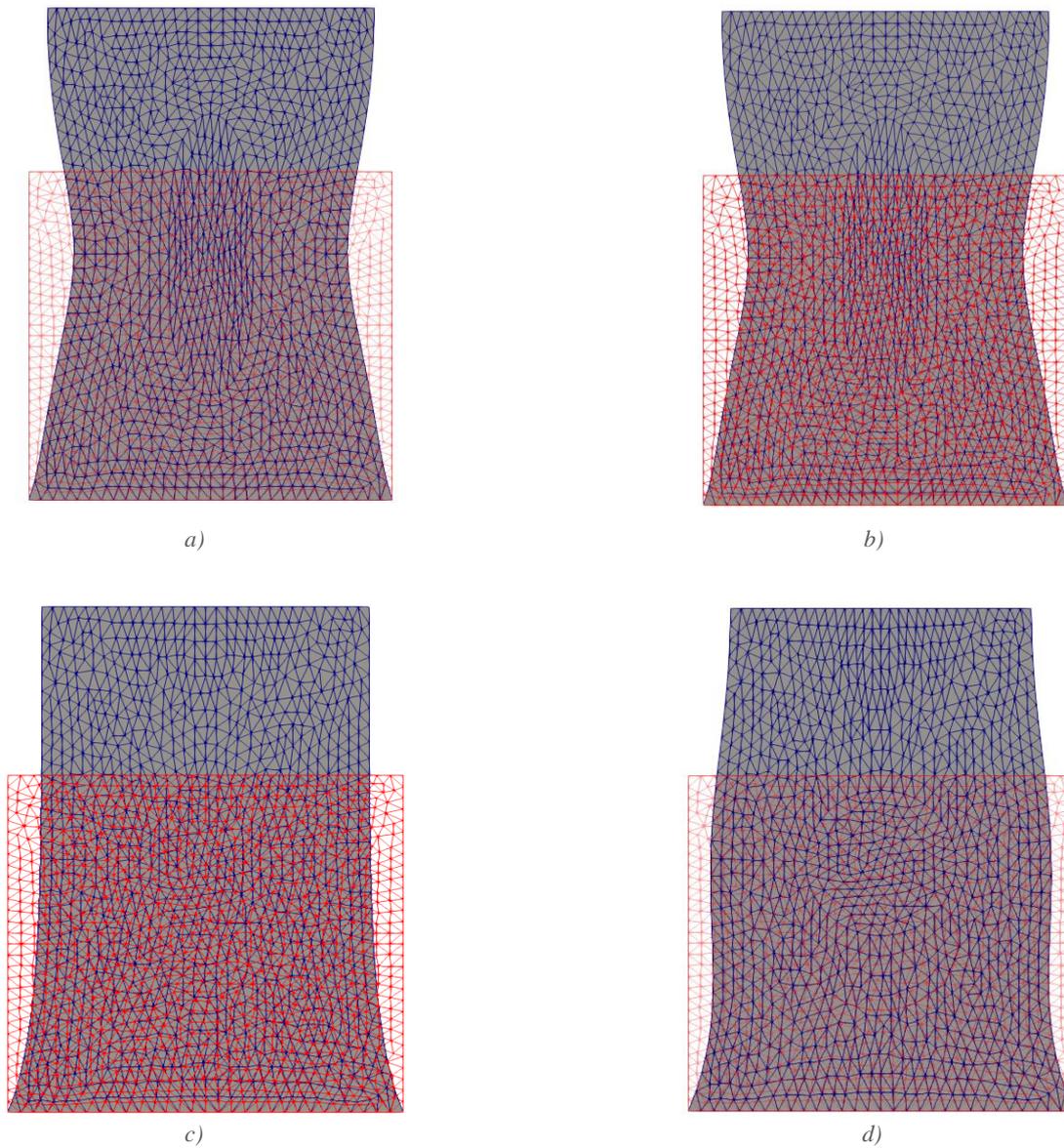


Figura 6.4. Malla deformada a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

En esta figura se representa la malla deformada (azul) junto a la no deformada (roja) para cada uno de los cuatro casos propuestos. Lo primero a comentar es que la deformación está escalada por un factor de 5 con el fin de apreciar mejor los resultados.

Para las inclusiones blandas (casos 1 y 2) se observa una mayor deformación en la parte central del sensor, lo que produce una forma cóncava en la malla deformada. Lo contrario ocurre para el caso 4, en el que, debido a la dureza de la inclusión, ésta apenas se deforma con respecto a la matriz, lo cual conlleva una forma convexa en el mallado

deformado. En cuanto al caso 3, al ser la matriz y la inclusión iguales mecánicamente, la deformación del cuerpo es uniforme.

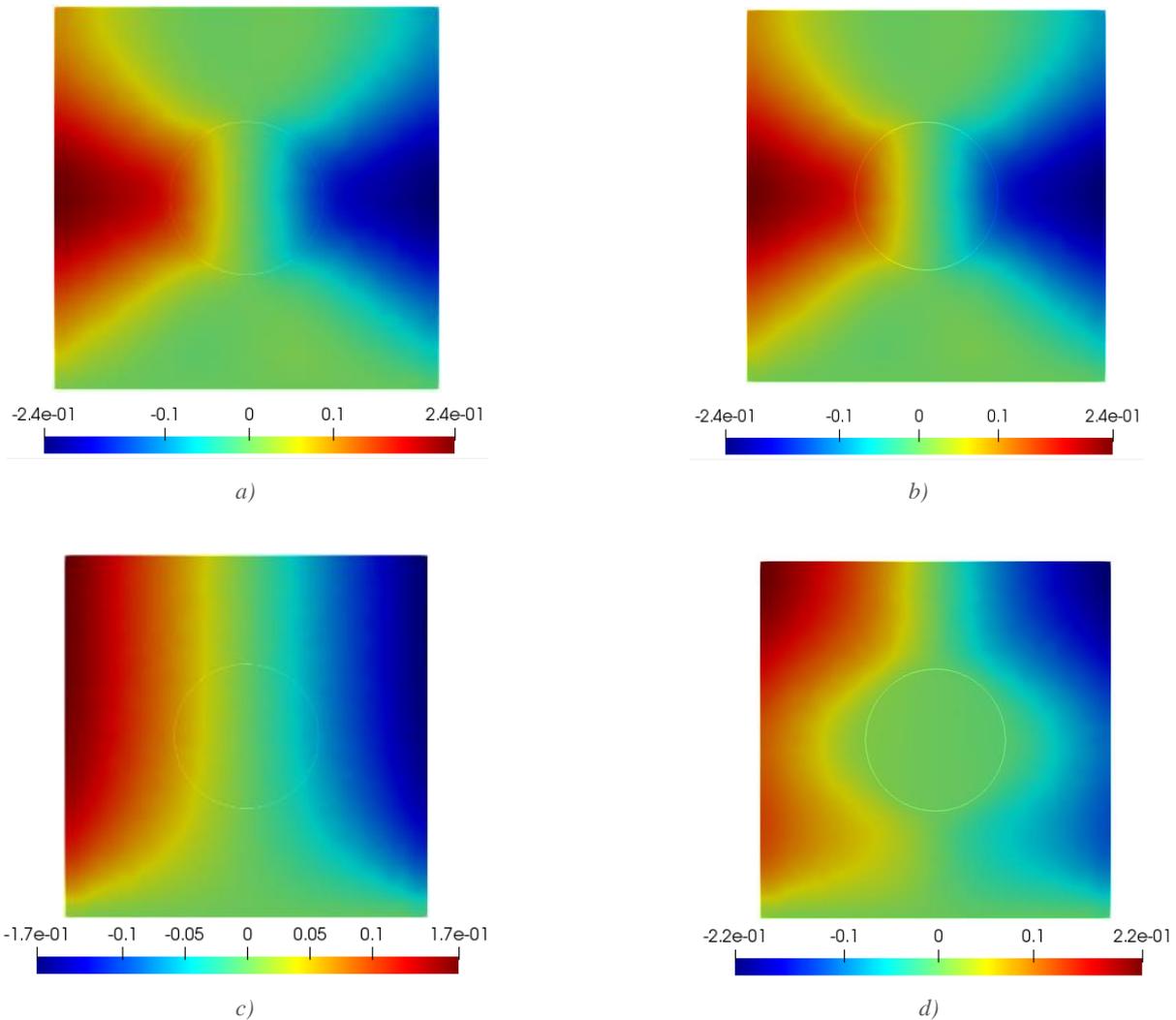


Figura 6.5. Desplazamiento horizontal a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

En cuanto al desplazamiento horizontal, poco se puede añadir que no se haya visto en la *Figura 6.4*. Lo que sí es apreciable es que el desplazamiento es nulo en la cara inferior, ya que está empotrada. Además, el caso 3 es en el que se producen los desplazamientos más pequeños al deformarse el cuerpo uniformemente. Las mayores deformaciones horizontales se producen en los casos de inclusión blanda, ya que la matriz se hunde a la altura de la inclusión de forma acusada.

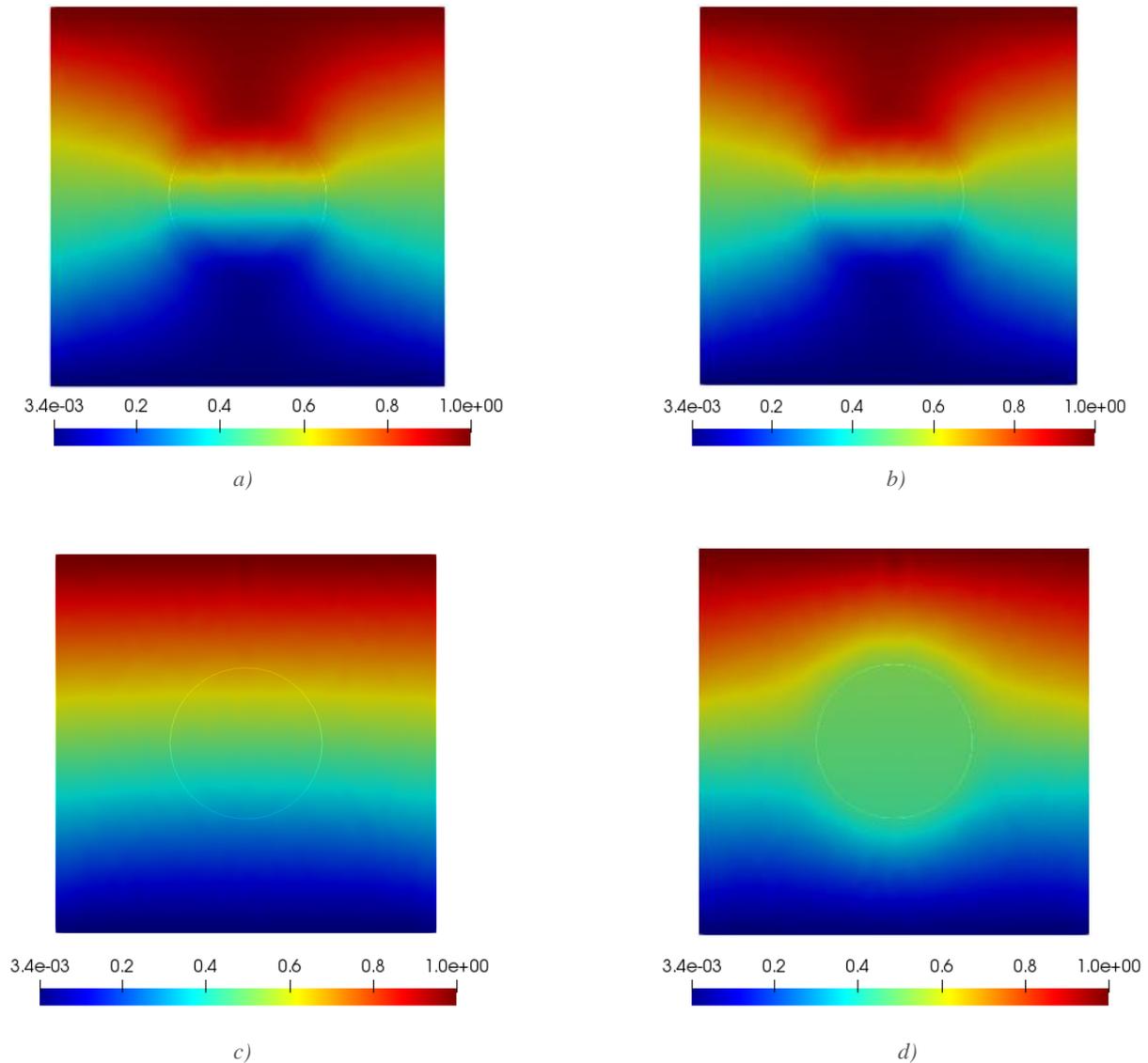


Figura 6.6. Desplazamiento vertical a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

En esta figura se aprecia de forma numérica el desplazamiento vertical establecido como condición de contorno en la cara superior, mientras que en la inferior el desplazamiento es nulo (Paraview interpreta un gran salto de valores en esa zona y por eso omite el 0 en la escala de valores). De nuevo, las mayores deformaciones se producen en la inclusión para los casos de inclusión blanda y en la matriz para el caso de inclusión dura, mientras que para el caso 3 la deformación es uniforme.

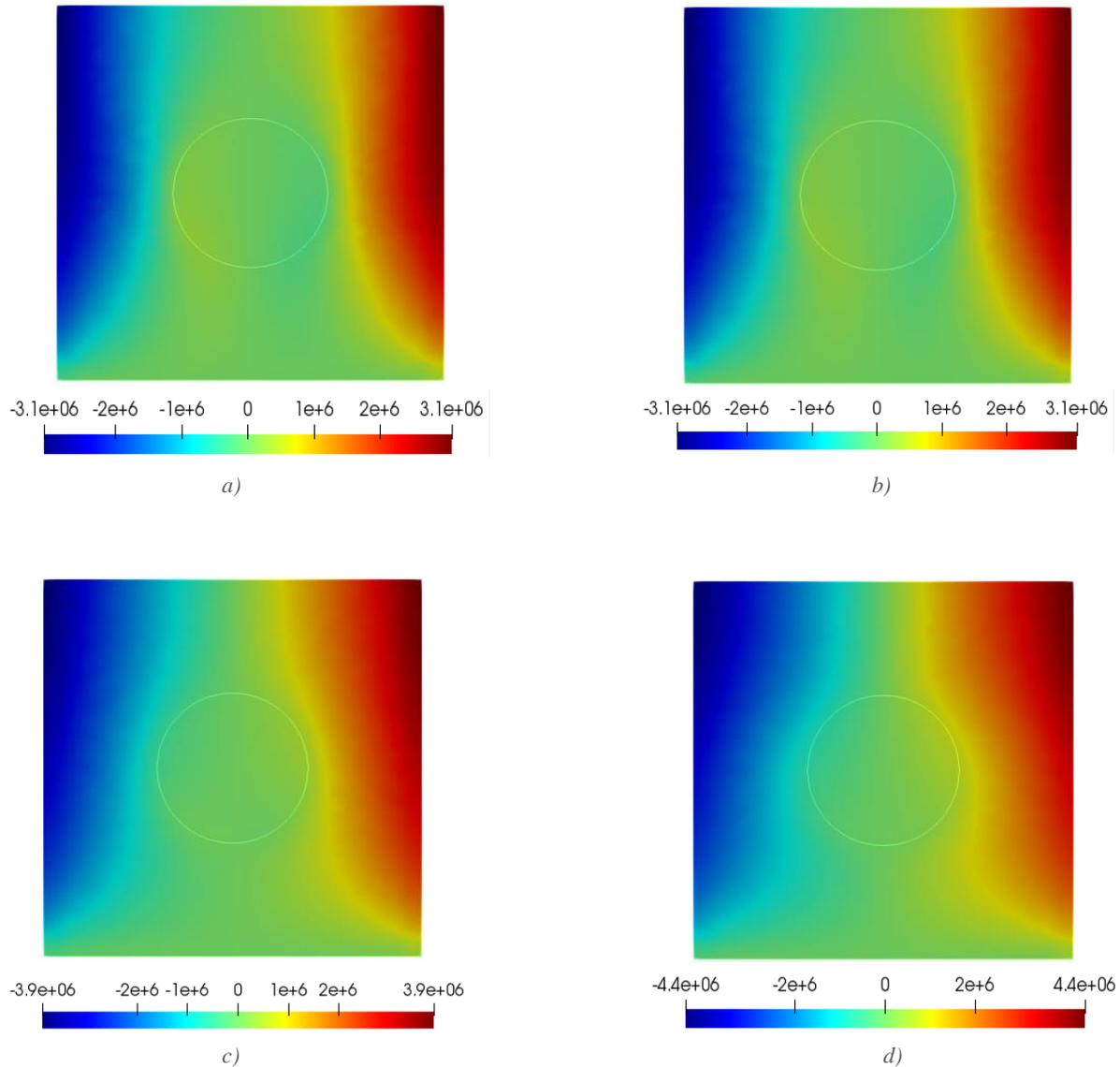


Figura 6.7. Potencial eléctrico a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

Para todos los casos, el gradiente del potencial φ , y por tanto el campo eléctrico E está en la misma dirección. Esto es debido al vector de polarización magnética N , cuyo valor es $[1,0]$ en este ejemplo. Además, en la inclusión el potencial es nulo, debido a que sus constantes piezoeléctricas son nulas para todos los casos también y no hay ningún potencial impuesto como condición de contorno.

Otro aspecto a destacar es que el potencial eléctrico generado es mayor en los casos de inclusión dura, siendo el mayor en el caso 4, donde más dura es. Esto es debido a que la matriz se deforma más en estos casos. En contraste, para los casos 1 y 2 la mayor deformación se produce en la inclusión, donde los coeficientes piezoeléctricos son nulos.

El potencial en la cara inferior es nulo para todos los casos, ya que se fijó como condición de contorno.

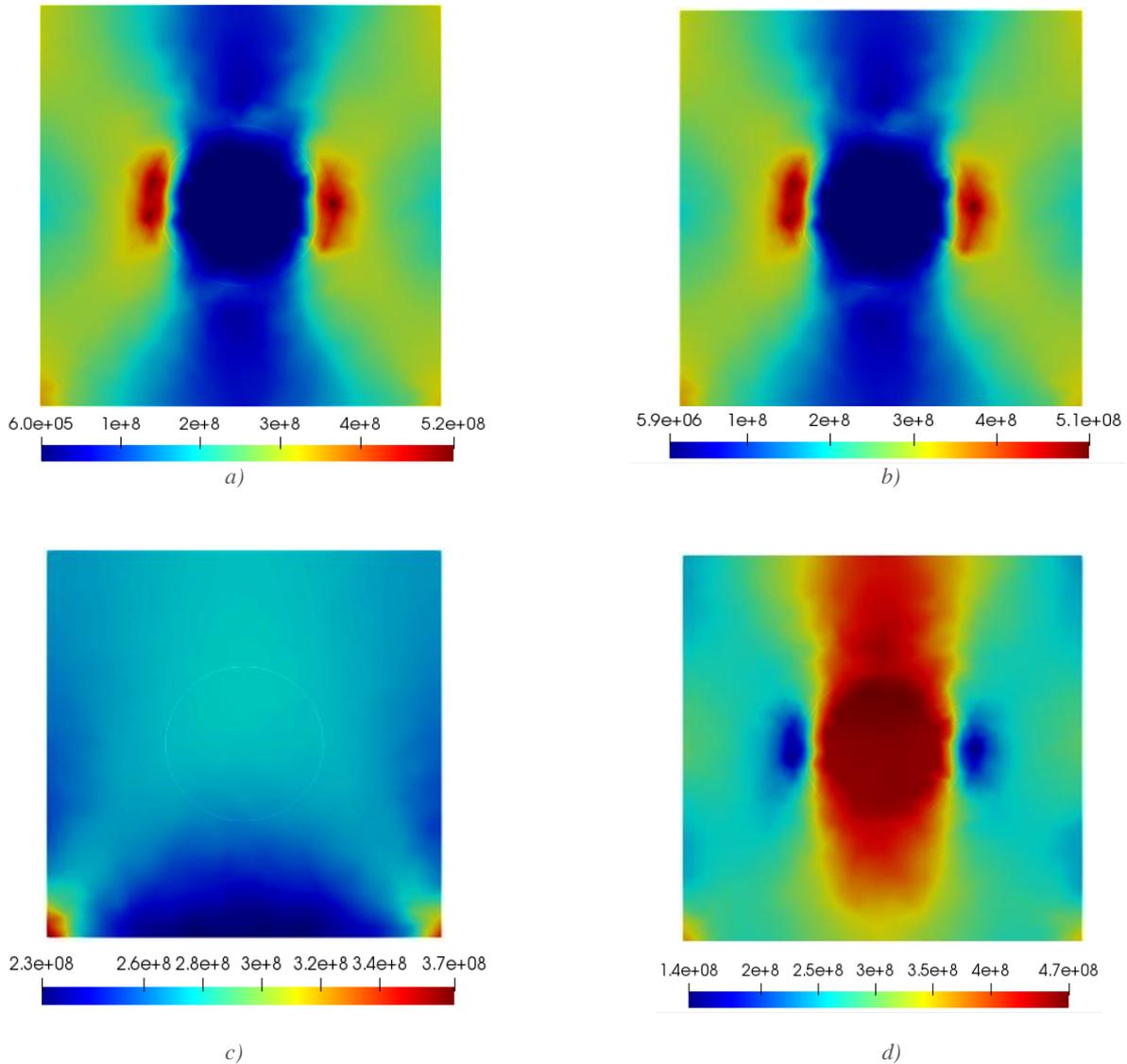


Figura 6.8. Tensión equivalente a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

El parámetro de tensión equivalente, en este caso Von Mises, es muy interesante a la hora de diseñar elementos que van a estar en tensión. La idea es no sobrepasar la tensión de fluencia en materiales dúctiles o tensión última en frágiles para evitar su fallo por sollicitaciones estáticas. En el cálculo de la tensión equivalente influyen, como se ha descrito en el capítulo 3, tanto la deformación del cuerpo como sus parámetros mecánicos, por lo que las diferencias son notables para cada caso.

Los puntos de mayor tensión equivalente se encuentran en los casos de inclusión blanda, ya que al deformarse ésta especialmente, los puntos de la matriz que se encuentran a ambos lados se deforman hacia dentro para ocupar su espacio, además de estirarse por continuidad. Es decir, hay mayor tensión en la matriz por ser más dura, y la zona de mayor tensión es la que más se deforma debido a la proximidad a la inclusión.

En el caso 4 sucede lo contrario. Al ser la inclusión más dura, las mayores tensiones se producen en ella. Además, en la matriz se producen las mayores deformaciones en el centro del cuerpo, especialmente por encima de la inclusión, por lo que la tensión equivalente también es alta en esa zona. En cuanto al caso 3, no hay ningún punto en el que la tensión equivalente sea muy alta, ya que la deformación es uniforme y las propiedades mecánicas son iguales para la matriz y la inclusión.

6.3. Sensor piezoeléctrico tridimensional

Se ha realizado el primer ejemplo con una geometría análoga en tres dimensiones. En el código se deben adaptar todos los vectores y tensores al problema tridimensional. Los resultados se mostrarán mediante un plano de corte al centro de la figura, de forma que se vea su interior. Como los resultados son análogos al caso bidimensional, se omitirán los resultados de desplazamiento horizontal y vertical. El resto de los resultados cambian debido a la nueva dimensión y al vector de polarización N , que para este caso se ha tomado como $[0,0,1]$.

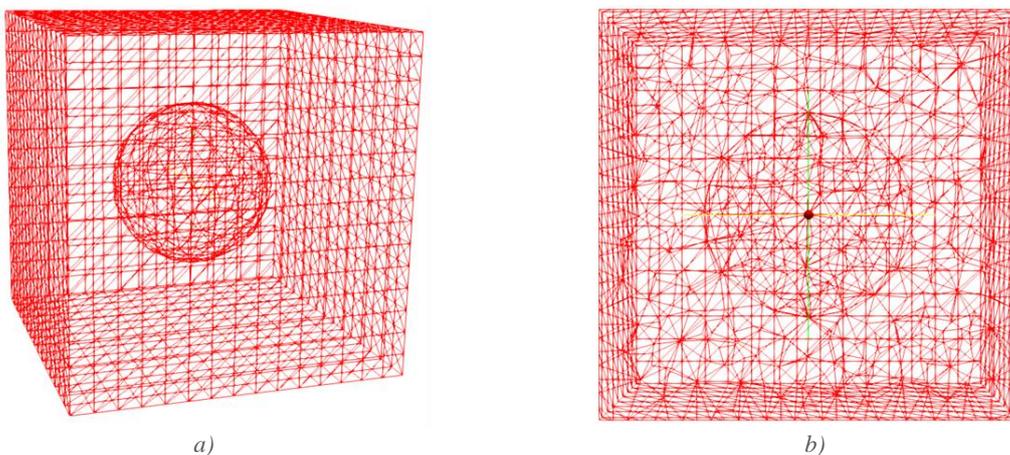


Figura 6.9. Malla del sensor piezoeléctrico tridimensional a) Entera b) Plano de corte al centro

Para este caso, la creación de la malla es más compleja, ya que para usar el comando “region” la malla debe estar incluida desde FreeFEM, por lo que no se puede importar desde GMSH. Por tanto, hay que cargar los paquetes “TetGen” y “msh3”, así como incluir el programa “MeshSurface” (anexo). Esto permite generar una malla esférica y una cúbica por separado y sumarlas para crear la malla objetivo, así como asignar etiquetas a las superficies para implementar las condiciones de contorno.

La malla es todo lo afinada que permite la capacidad computacional de la máquina con la que se han calculado los resultados. Debido a que FreeFEM utiliza factorización LU por defecto, debe almacenar en memoria todos los nodos a la vez, y éstos aumentan mucho al ser un problema tridimensional.

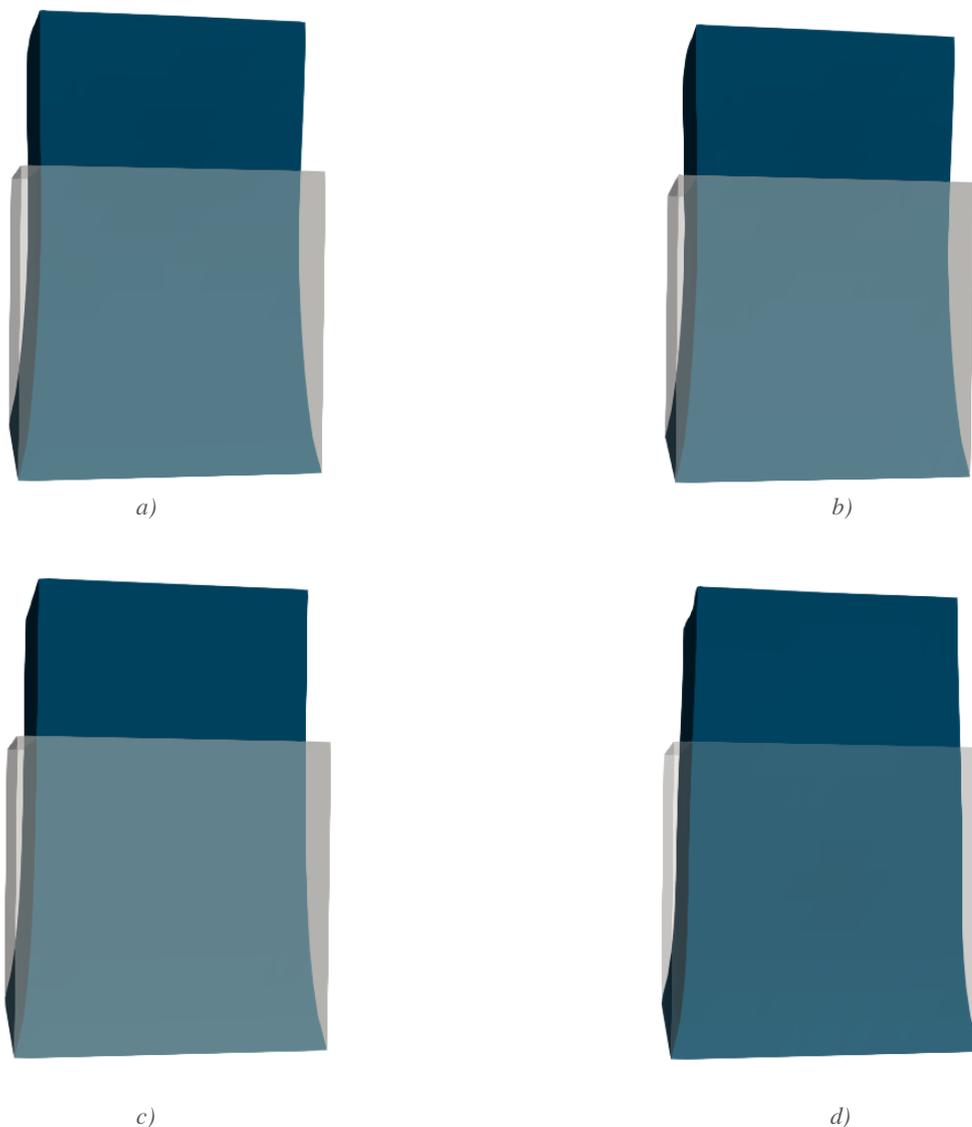


Figura 6.10. Cuerpo deformado a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

En esta figura está representado el cuerpo deformado (azul) junto al no deformado (translúcido). Nuevamente, se ha aplicado un coeficiente de escalado para la deformación de 5. Al añadir la tercera dimensión, la diferencia de volumen entre la matriz y la inclusión es mucho mayor que la diferencia de área de los elementos del ejemplo bidimensional. Es por ello que se aprecia mucho menos la forma cóncava en los casos de inclusión blanda, y la forma convexa en el caso de inclusión dura, ya que la deformación de la inclusión afecta menos a la de la matriz, pero sigue siendo notoria.

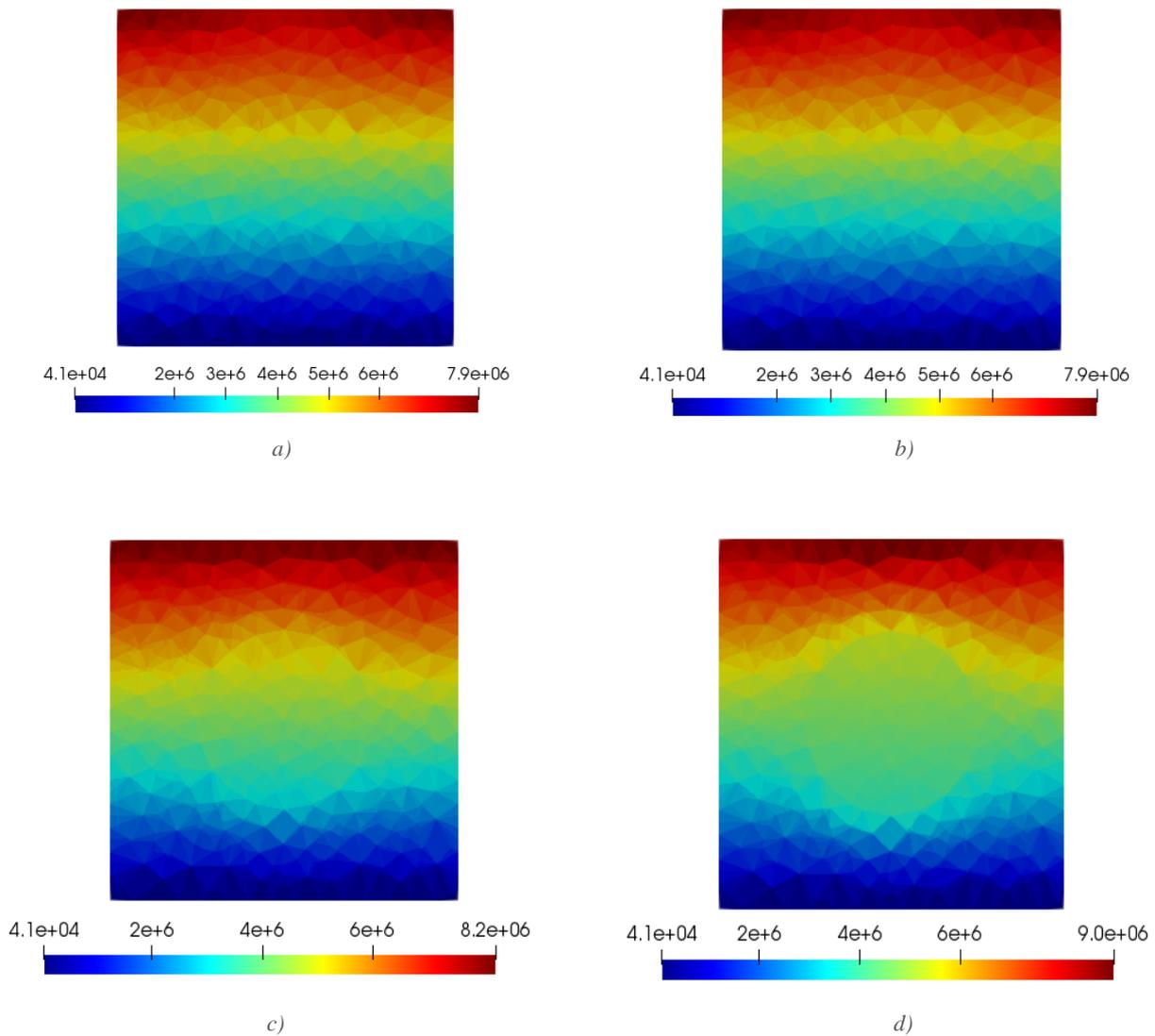


Figura 6.11. Campo eléctrico a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

Se puede comprobar en esta figura la influencia del vector de polarización N . Para este caso, su dirección es la del eje vertical, por lo que el potencial varía desde ser nulo en la cara inferior hasta su valor máximo en la cara superior. Es importante recordar que

Paraview no muestra el valor 0 en su escala de valores, sino el siguiente más próximo debido a que el salto de valores entre la cara inferior y los puntos infinitesimalmente próximos es muy grande con respecto a la horquilla de valores mostrados.

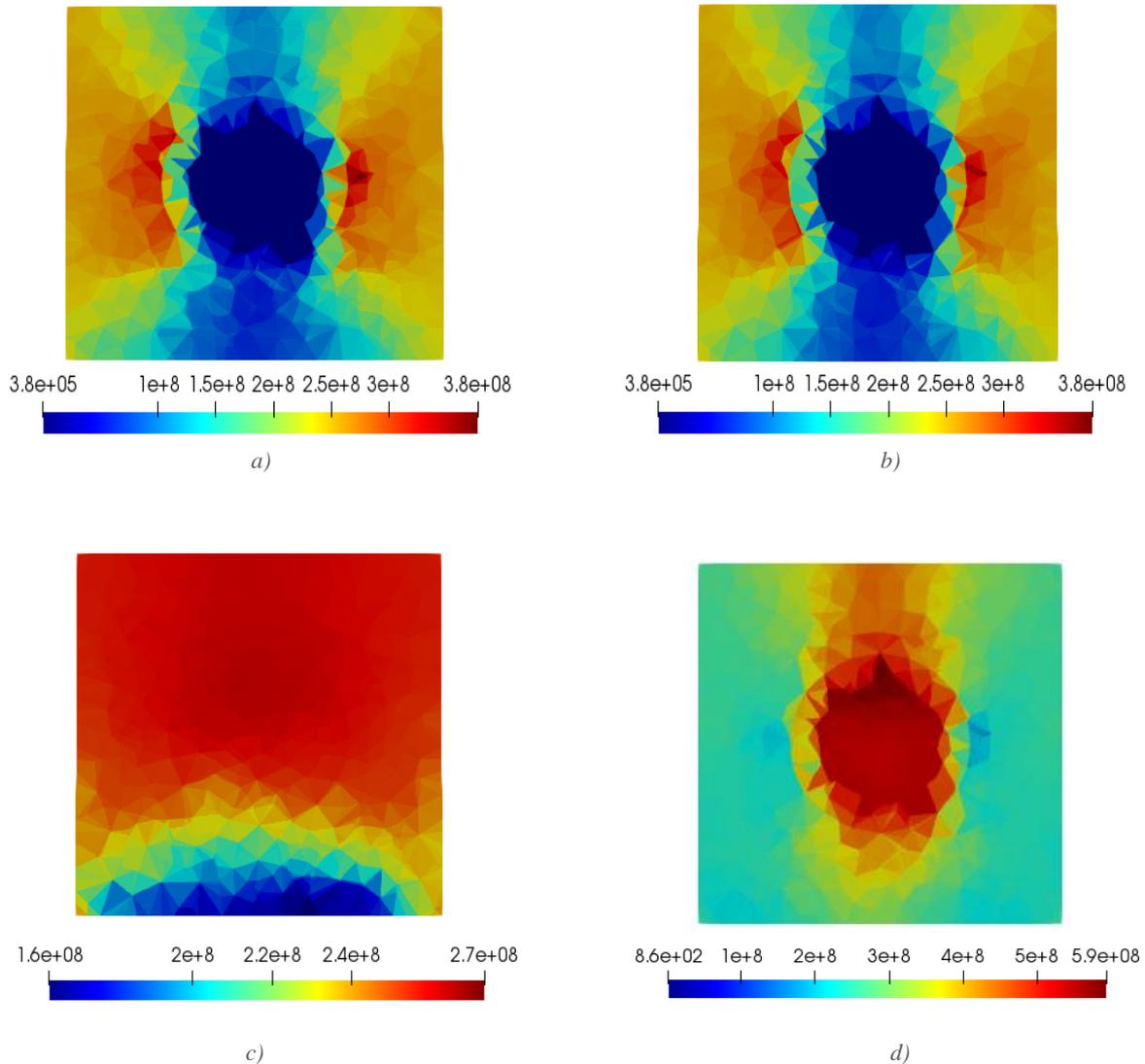


Figura 6.12. Tensión equivalente a) $\zeta = 10^{-3}$ b) $\zeta = 10^{-2}$ c) $\zeta = 1$ d) $\zeta = 10$

En cuanto a la tensión equivalente de Von Mises, el cuerpo se comporta de manera similar al bidimensional, pero hay algunos cambios debido a la nueva dimensión. Como se ha comentado anteriormente, en este caso la diferencia de volumen entre la inclusión y la matriz es bastante alta, por lo que las deformaciones en la inclusión no inducen deformaciones tan acusadas en la matriz como en el caso bidimensional. Por eso, en este ejemplo se encuentran las mayores tensiones en el caso de inclusión dura. Para los casos de inclusión blanda sigue habiendo una concentración de esfuerzos alrededor de la misma,

pero no llegan a ser tan altos como en el caso 4. El caso 3 sigue siendo donde menor es el valor de la máxima tensión equivalente debido a su uniformidad al deformarse.

6.4. Actuador piezoeléctrico tridimensional

Para este problema se ha supuesto una geometría de actuador (solapa) empotrada por un lado y sometida a unas condiciones eléctricas que inducirán una deformación. Se ha fijado un potencial de 0 voltios en su cara inferior, y de 10^7 voltios en una superficie definida que pasa por el centro del actuador. El código es el expuesto en el capítulo 6.1. Los valores de los parámetros se encuentran en la *Figura 6.14*

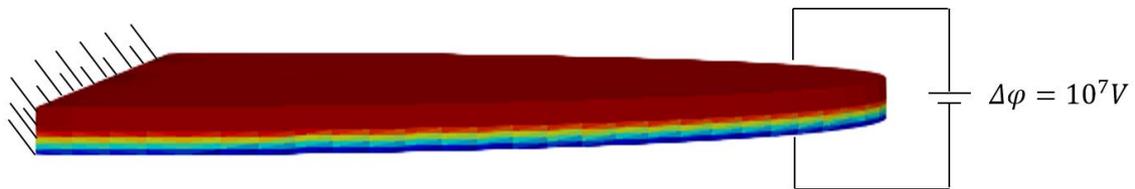


Figura 6.13. Actuador piezoeléctrico tridimensional

Dato	Valor
Largo	6 m
Profundidad	4 m
Ancho	0,18 m
Parámetro λ	10^9 Pa
Parámetro μ	10^9 Pa
Parámetro α_1	0,1 C/m
Parámetro α_2	0,1 C/m
Parámetro ϵ	$3,52 \cdot 10^{-11}$ C ² /Nm ²
Vector N	[0,0,1]

Figura 6.14. Parámetros del actuador piezoeléctrico tridimensional

Una vez definidos todos los parámetros, se puede pasar a analizar los resultados del cálculo:

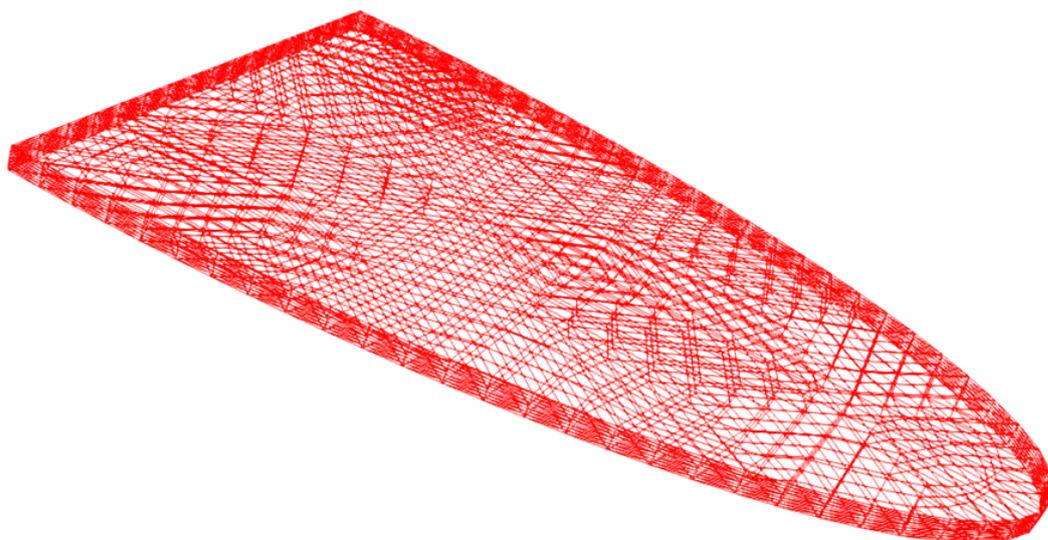


Figura 6.15. Malla del actuador piezoeléctrico tridimensional

Para la creación de esta malla se recurrió al programa GMSH (anexo). De nuevo, se afinó todo lo posible teniendo en cuenta la capacidad computacional del ordenador utilizado.



Figura 6.16. Deformación del actuador piezoeléctrico (x100)

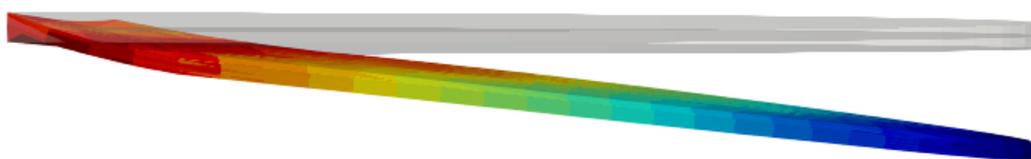


Figura 6.17. Deformación del actuador piezoeléctrico (x200)

Se puede observar una deformación en el sentido negativo del eje vertical debida a una diferencia de potencial positiva en el mismo eje, y a que el vector de polarización es $N = [0,0,1]$. Esta deformación es muy pequeña, por lo que para obtener las figuras se han aplicado factores de escalado de 100 y 200 respectivamente. Sin embargo, estas deformaciones no se pueden obtener en la realidad para materiales piezoeléctricos. Sí se conseguiría en polímeros y otros materiales, pero ya no sería régimen lineal y no se podrían considerar las hipótesis de pequeñas deformaciones, por lo que el cálculo sería distinto y más complejo.

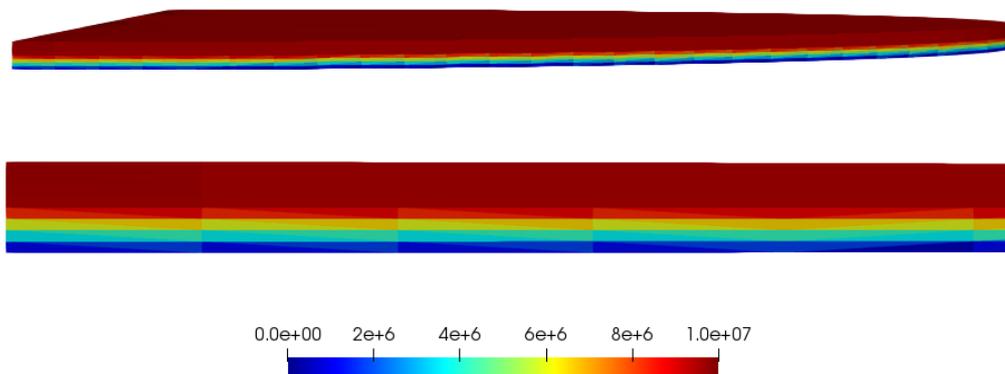


Figura 6.18. Potencial eléctrico del actuador piezoeléctrico

Se aprecia cómo se cumplen las condiciones de contorno establecidas, de potencial nulo en la cara inferior y 10^7 voltios en el centro del actuador.

En la siguiente figura se encuentran representados los desplazamientos horizontales, tanto en el eje x como en el eje y.

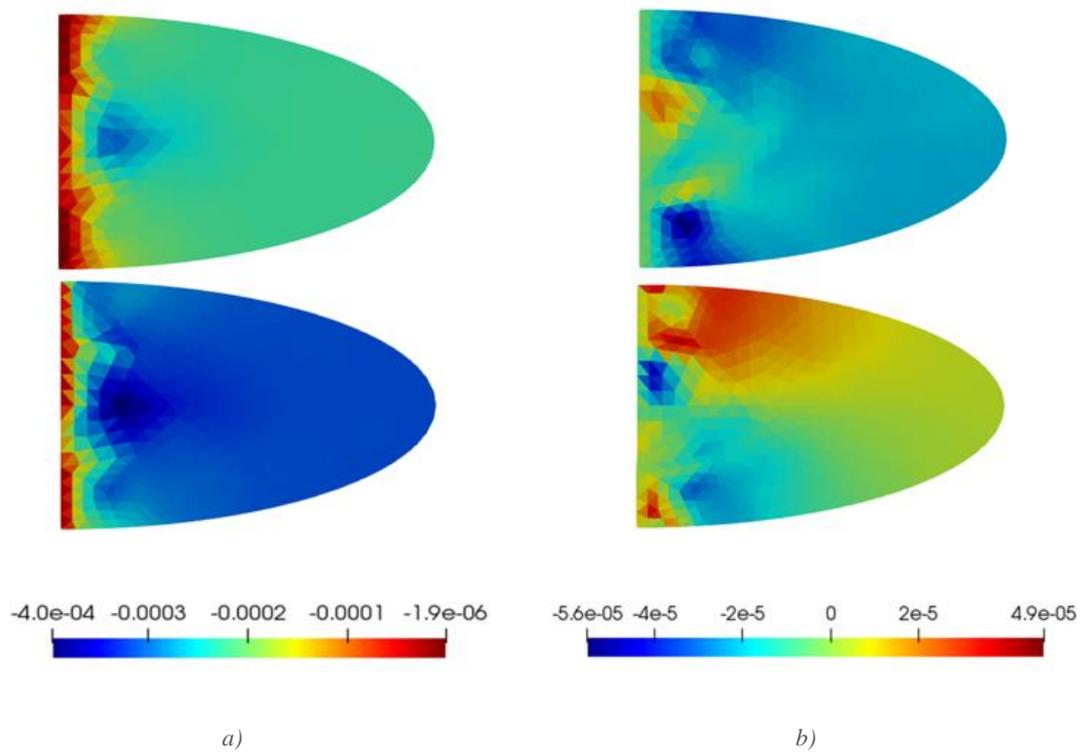


Figura 6.19. Desplazamiento horizontal del actuador a) en x b) en y

Estos desplazamientos son despreciables y no muy importantes. Los desplazamientos en el eje x se producen como consecuencia de la deformación vertical, ya que al inclinarse el actuador verticalmente, todos sus puntos quedan más cerca del empotramiento en el eje x. Los desplazamientos en el eje y se pueden considerar nulos en todo el cuerpo. Las imágenes muestran la parte superior e inferior del actuador, respectivamente. Se aprecia cómo el desplazamiento es nulo en el empotramiento.

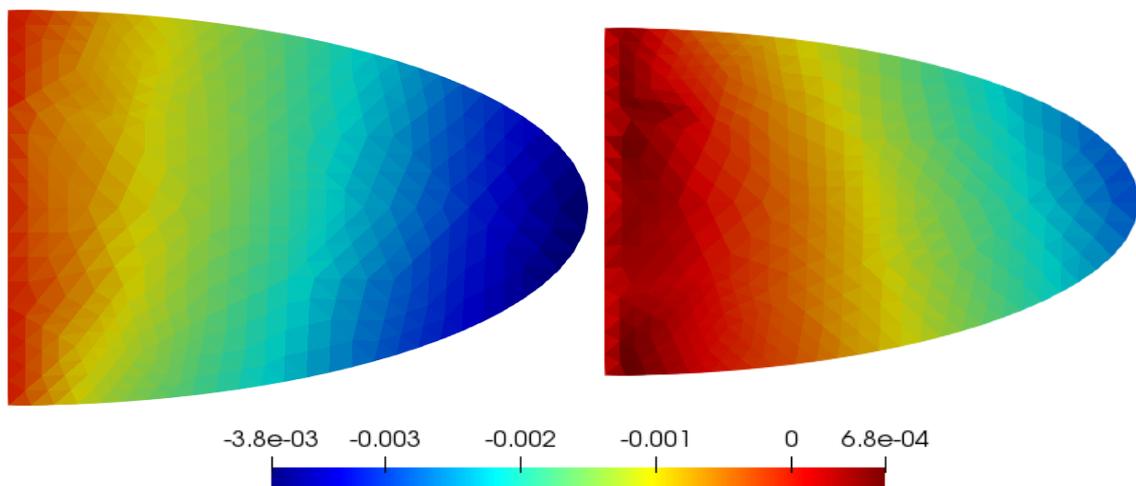


Figura 6.20. Desplazamiento vertical del actuador

Los desplazamientos verticales son los importantes en este ejemplo, ya que es lo que se busca al imponer las condiciones de contorno. Se puede apreciar, como en la *Figura 6.16*, que el desplazamiento es negativo y nulo en el empotramiento. En cuanto a la tensión equivalente, los resultados se muestran en la siguiente figura:

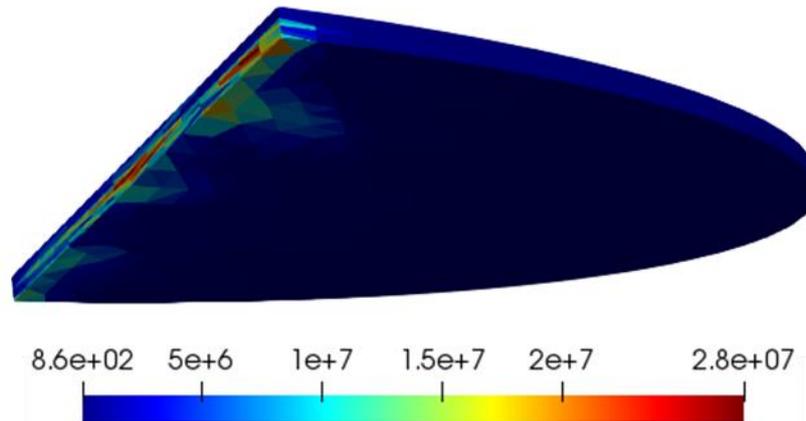


Figura 6.21. Tensión equivalente del actuador

La tensión equivalente de Von Mises es muy baja para este actuador, ya que los desplazamientos son muy pequeños. La mayor concentración de esfuerzos se encuentra en el empotramiento.

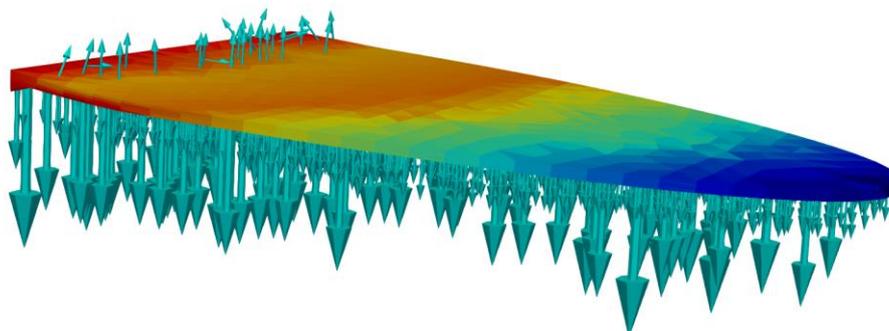
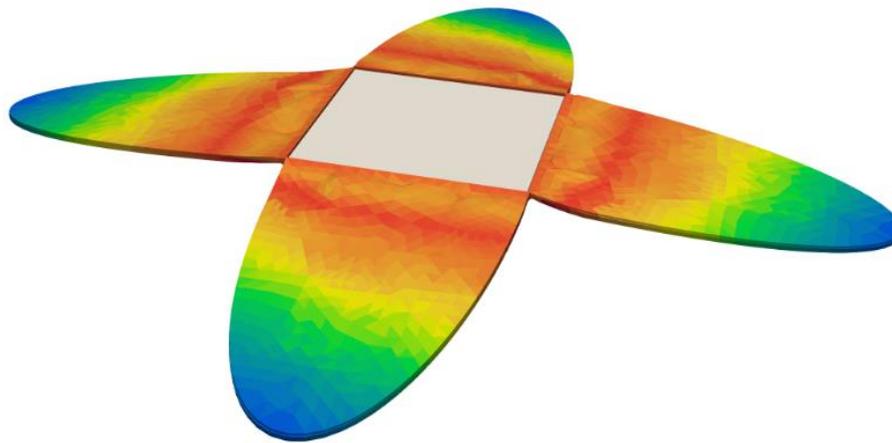


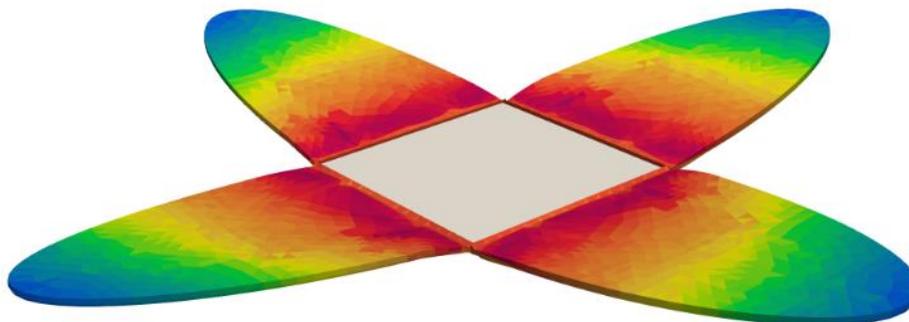
Figura 6.22. Campo eléctrico del actuador

Se puede representar la dirección y sentido del campo eléctrico mediante vectores. En la figura se aprecia que va en dirección vertical y sentido negativo, debido al gradiente del potencial eléctrico fijado como condición de contorno, ya que $\mathbf{E}(\varphi) = -\nabla\varphi$.

Realizando una composición con varios de estos actuadores, se pueden generar pinzas o “grippers” piezoeléctricos. Uno de estos grippers está representado en la siguiente figura junto con los desplazamientos verticales.



a)



b)



Figura 6.23. Gripper piezoeléctrico a) desde arriba b) desde abajo

6.5. Problema dinámico

El último de los problemas propuestos es, de nuevo, el sensor piezoeléctrico bidimensional del capítulo 6.2. Sin embargo, esta vez la condición de contorno del desplazamiento vertical en la cara superior es sinusoidal y, por tanto, se trata de un problema dinámico.

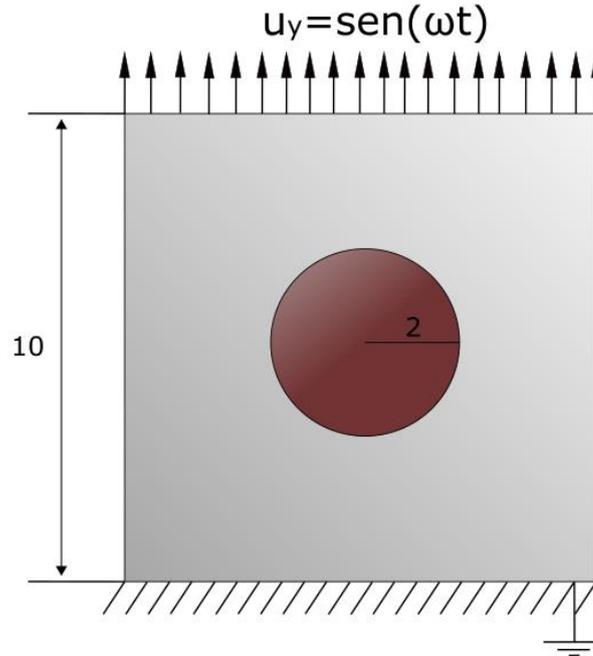


Figura 6.24. Problema dinámico

Para implementar este problema en FreeFEM, se ha de realizar una discretización temporal, tal y como se explicó en el capítulo anterior. Para el integrador Newmar-beta, los valores de sus parámetros serán, $\beta = 0,25$ y $\gamma = 0,5$. Para el problema estudiado, se estudiará un tiempo de 5s, y el intervalo entre instantes se ha establecido en 0,01s. Además, para implementar el bucle en FreeFEM se necesitan unas condiciones iniciales que, para este problema, son nulas $\mathbf{a}_0 = \mathbf{v}_0 = \mathbf{u}_0 = \mathbf{0}$. A la hora de la implementación, después de darle valores a todos los parámetros, se hace el bucle de la siguiente forma:

```
int counter=0;

solve Problem([u1,u2,phi],[v1,v2,alpha])=
int2d(Th)((rho/(beta*dt*dt))*[u1,u2]*[v1,v2])
-int2d(Th)((rho/(beta*dt*dt))*[up1,up2]*[v1,v2])
+int2d(Th)(stress(u1,u2,phi)*epsilonmatrix(v1,v2))
-int2d(Th)(desplazamiento(u1,u2,phi)*Campoelectrico(alpha))
```

```

+on(1,u1=0.0,u2=0.0,phi=0.0)
+on(3,u2=1*sin(10*counter*dt));

// Exportamos a fichero las energías
ofstream ff("Welastm.dat");
ofstream hh("Welaste.dat");
ofstream gg("Kinetic.dat");
ofstream tt("Wext.dat");

//Integrador temporal
for(real t=0;t<T;t+=dt)
{
// Predictor step for displacements and velocity
up1 = u1 + dt*vel1 + 0.5*dt*dt*(1-2*beta)*acc1;
up2 = u2 + dt*vel2 + 0.5*dt*dt*(1-2*beta)*acc2;
vp1 = vel1 + (1-gamma)*dt*acc1;
vp2 = vel2 + (1-gamma)*dt*acc2;
// Resolvemos el problema para el time step actual
Problem;
// Actualizamos valos de aceleraciones y velocidades
acc1 = 1/(beta*dt*dt)*(u1 - up1);
acc2 = 1/(beta*dt*dt)*(u2 - up2);
vel1 = vp1 + gamma/(beta*dt)*(u1 - up1);
vel2 = vp2 + gamma/(beta*dt)*(u2 - up2);
// Movemos la malla
real coef=1;
mesh th3=movemesh(Th,[x+coef*u1,y+coef*u2]); // malla deformada
sigmaVm = sqrt(stress(u1,u2,phi)[0]*stress(u1,u2,phi)[0]
+stress(u1,u2,phi)[3]*stress(u1,u2,phi)[3]
-stress(u1,u2,phi)[0]*stress(u1,u2,phi)[3]
+3*stress(u1,u2,phi)[1]*stress(u1,u2,phi)[1]);

// Calculamos las energías
Welastm = int2d(Th)(0.5*stress(u1,u2,phi)'*epsilononmatrix(u1,u2));

```

```

Welaste = -int2d(Th)(0.5*desplazamiento(u1,u2,phi)*Campoelectrico(phi));
Kinetic = int2d(Th)(0.5*rho*[vel1,vel2]*[vel1,vel2]);
Wext = int2d(Th)(rho*[acc1,acc1]*[u1,u2]) + Welastm + Welaste;

ff<<Welastm<<endl;
hh<<Welaste<<endl;
gg<<Kinetic<<endl;
tt<<Wext<<endl;

// Exportamos a paraview
savevtk("PRUEBA2dbeam_Time_step_"+n+"_"+counter+".vtk",th3,u1,u2,phi,sigmaV
m,[-dx(phi),-dy(phi)],counter);
counter++;
}

```

Como se observa, se inicializa un contador y se resuelve el problema con la formulación variacional del problema dinámico después de haber actualizado todas las variables con la información del instante anterior, aplicando las expresiones (6.3). En cuanto a los ficheros de energías, se han guardado para cada instante la energía potencial mecánica, la potencial eléctrica, la cinética y la producida por el desplazamiento en la frontera Dirichlet. Sus expresiones son, respectivamente:

$$\begin{aligned}
W_{elast,m} &= \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{u}) \, dV \\
W_{elast,e} &= -\frac{1}{2} \int_{\Omega} \mathbf{D} \cdot \mathbf{E}(\varphi) \, dV \\
W_{kinetic} &= \frac{1}{2} \int_{\Omega} \rho \cdot \mathbf{v}^2 \, dV \\
W_{ext} &= \int_{\Omega} \rho \mathbf{a} \cdot \mathbf{v} \, dV + W_{elast,m} + W_{elast,e}
\end{aligned} \tag{6.2}$$

Al guardar los ficheros de todas las energías y sumarlas para cada instante, se observa que la suma es constante por conservación de energía, y para este problema nula al ser todas las condiciones iniciales nulas. En la *Figura 6.25* se encuentra representada la suma de todas las energías y la energía producida por el desplazamiento impuesto en la frontera Dirichlet W_{ext} .

$$W_{elast,m} + W_{elast,e} + W_{kinetic} + W_{ext} = cte = 0 \tag{6.3}$$

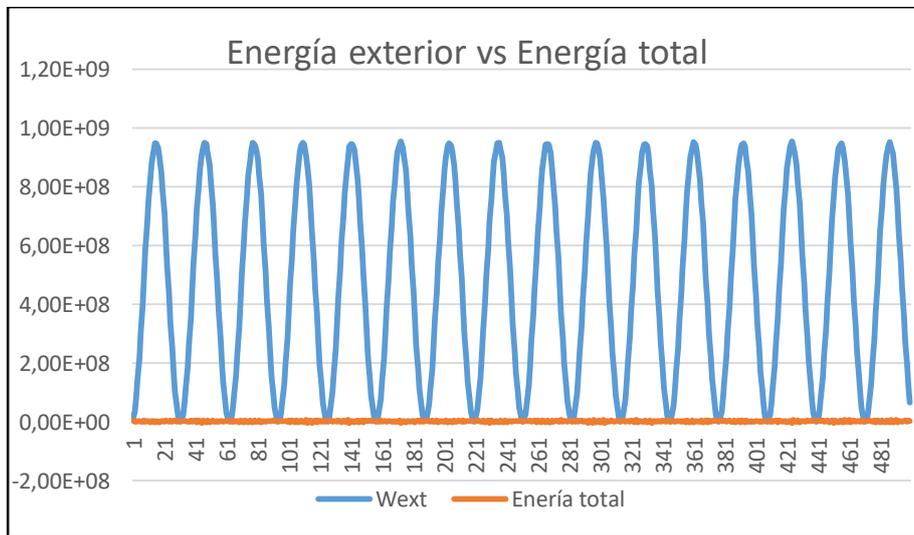


Figura 6.25. Energías en el problema mecánico

Se puede observar que la energía total es nula, aunque tiene algunas irregularidades debidas al error de redondeo computacional, pero sobre todo al integrador temporal, el cual conlleva algo de disipación. A continuación, se muestran algunos resultados para distintos puntos de la oscilación del caso de inclusión blanda.

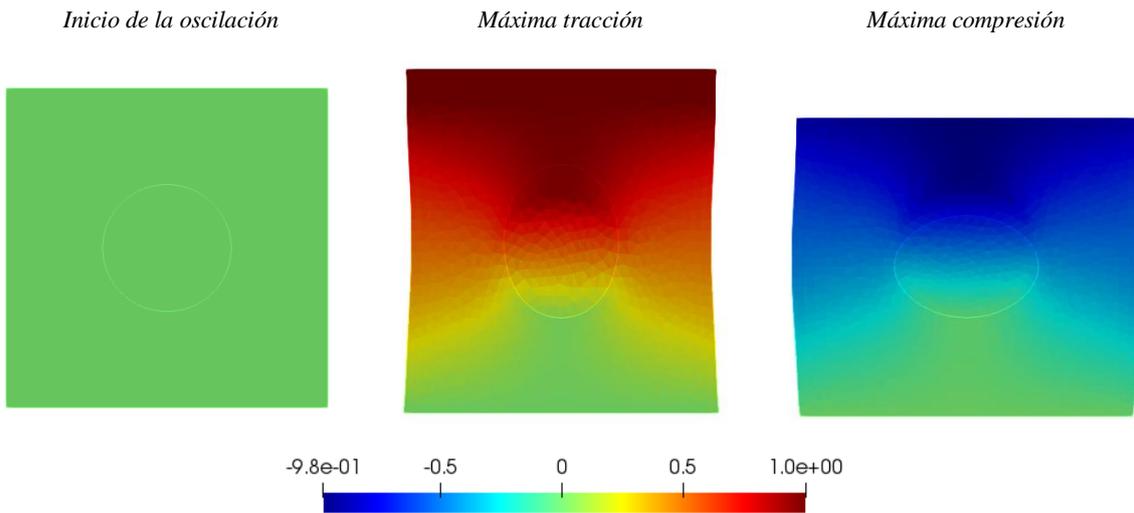


Figura 6.26. Desplazamiento vertical del problema mecánico

Se muestra en la figura el desplazamiento vertical en los tres estados más representativos del problema: Estado inicial, máxima tracción y máxima compresión. Se observa cómo en la cara inferior el desplazamiento es siempre nulo debido a que está empotrado.

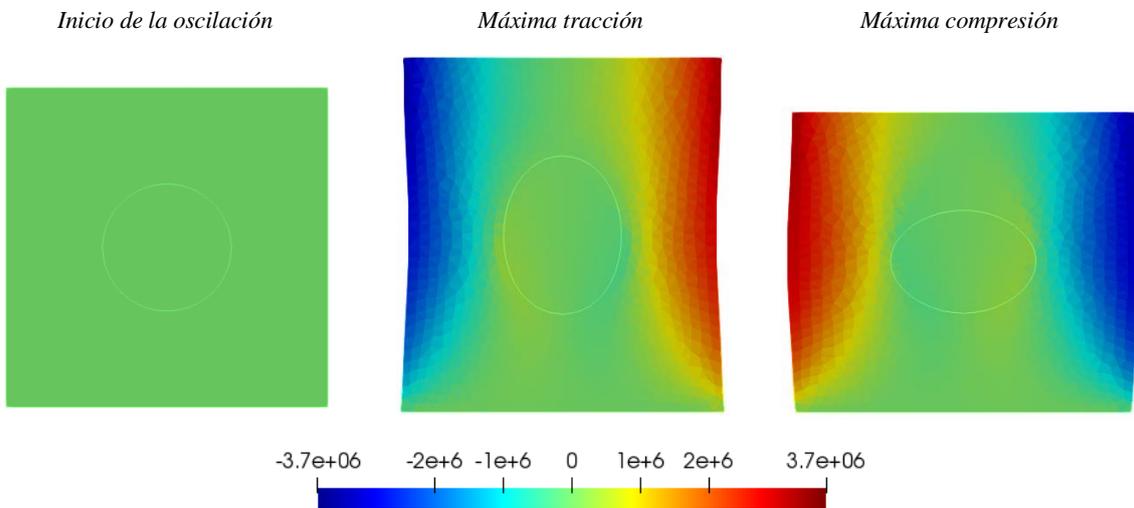


Figura 6.27. Potencial eléctrico del problema mecánico

En cuanto al potencial eléctrico, se puede apreciar en esta figura cómo al cambiar el sentido del esfuerzo mecánico también lo hace su respuesta eléctrica, es decir, que el potencial generado debido a la tracción es inverso al debido a compresión.

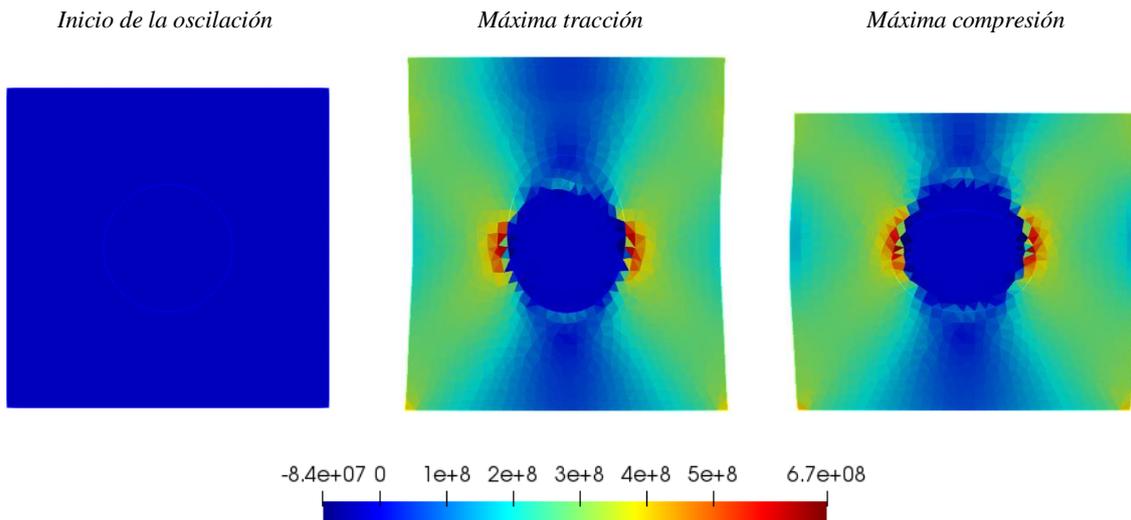


Figura 6.28. Tensión equivalente del problema mecánico

Por último, se muestra la tensión equivalente en los estados más representativos de la oscilación. De nuevo, hay que tener en cuenta que Paraview omite el valor nulo de la tensión equivalente al inicio de la oscilación para ajustar su escala. Se puede apreciar una distribución similar en la tensión equivalente de Von Mises para la tracción y la compresión. Esto es debido a que la fórmula no tiene en cuenta el sentido del esfuerzo. Sin embargo, en la bibliografía, a la hora de calcular la tensión admisible para un esfuerzo estático, el coeficiente de seguridad para los esfuerzos de tracción ha de ser más alto, ya que los materiales soportan mejor la compresión. En conclusión, en el análisis de la tensión equivalente no se tiene en cuenta el sentido del esfuerzo mecánico, pero sí se hace posteriormente a la hora de dimensionar un elemento o elegir un material aplicando un coeficiente de seguridad mayor o menor para el cálculo de la tensión admisible.

CAPÍTULO 7: OPTIMIZACIÓN TOPOLÓGICA DEL PROBLEMA MECÁNICO

7.1. Formulación del problema

El problema de optimización topológica consiste en, a partir de un espacio de diseño, ocupado por un material con leyes constitutivas y sujeto a condiciones de contorno Dirichlet y Neumann (*Figura 7.1*), encontrar la distribución óptima que debe ocupar el material para que se satisfaga una condición, que puede ser maximizar o minimizar algo. Para el problema mecánico de la figura, la ley constitutiva es la de elasticidad lineal, y se busca minimizar el desplazamiento producido por la fuerza \mathbf{P} , es decir, el producto $\mathbf{P} \cdot \mathbf{u}$



Figura 7.1. Espacio de diseño de optimización topológica

La mejor solución al problema planteado sería rellenar todo el espacio de diseño de material, pero en la práctica no existen estructuras macizas debido a costes, peso, etc. Por tanto, el problema está sujeto a una restricción de volumen, es decir:

$$\int_D \chi \, dV \leq f |D| \quad (7.1)$$

Donde $|D|$ es el volumen total del espacio de diseño, f es una fracción que adopta valores 0 y 1 y \mathcal{X} es una función cuyo valor es:

$$\mathcal{X} = \begin{cases} 1 & \text{donde el material está presente} \\ 0 & \text{donde no hay material} \end{cases} \quad (7.2)$$

El problema quedaría, por tanto, en encontrar la distribución de material tal que:

$$\begin{aligned} \min \mathbf{J} &= \mathbf{P} \cdot \mathbf{u} \\ \text{sujeto a } &\begin{cases} \int_D \mathcal{X} dV \leq f |D| \\ \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} & \text{en } D \\ \mathbf{u} = 0 & \text{en } \partial D_D \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} & \text{en } \partial D_N \end{cases} \end{aligned} \quad (7.3)$$

Siendo $\boldsymbol{\sigma}$ igual a:

$$\boldsymbol{\sigma} = \mathcal{X}(2\mu\boldsymbol{\varepsilon} + \lambda \text{tr}(\boldsymbol{\varepsilon})\mathbf{I}) = \begin{cases} 2\mu\boldsymbol{\varepsilon} + \lambda \text{tr}(\boldsymbol{\varepsilon})\mathbf{I} & \text{para } \mathcal{X} = 1 \\ 0 & \text{para } \mathcal{X} = 0 \end{cases} \quad (7.4)$$

La ecuación (7.4) que define el tensor de Cauchy en función de \mathcal{X} supone un problema numérico, ya que al haber elementos de la malla donde la rigidez es nula induce un muy alto número de condicionamiento de la matriz de rigidez necesaria para resolver el problema mediante el método de elementos finitos. El número de condicionamiento de una matriz simétrica se define de esta forma:

$$\text{cond } \mathbf{K} = \frac{\max(\text{eig}(\mathbf{K}))}{\min(\text{eig}(\mathbf{K}))} \quad (7.5)$$

Para resolver este problema, en las zonas del espacio de diseño donde no hay material se puede considerar que existe un material residual al que llamaremos vacío, cuyas propiedades son una muy pequeña fracción del material original. Por tanto, la ecuación (7.4) queda:

$$\sigma = \begin{cases} 2\mu\epsilon + \lambda \operatorname{tr}(\epsilon)I & \text{para } \mathcal{X} = 1 \\ \alpha(2\mu\epsilon + \lambda \operatorname{tr}(\epsilon)I) & \text{para } \mathcal{X} = 0 \end{cases} \quad (7.6)$$

De esta forma, y dando valores muy pequeños al factor de escalado, $\alpha = 10^{-6}$, se consigue un material que en la práctica es vacío, pero no altera el número de condicionamiento como para hacer el sistema irresoluble.

7.2. Formulación continua

La formulación de este problema, tal y como se ha presentado, es teóricamente correcta pero inabordable en la práctica. Esto es debido a que la función $\mathcal{X} \in \{0,1\}$ es una variable discreta y no se puede trabajar sobre formulación discreta con el método de elementos finitos. La solución es sustituir la variable discreta \mathcal{X} por una variable continua $\rho \in [0,1]$. Esto significa que se admiten densidades intermedias entre 0 y 1 para realizar los cálculos, aunque en la realidad sólo exista el material residual y el original, es decir, los valores 0 y 1 respectivamente. Por tanto, se va a forzar que al final de la optimización, el parámetro ρ tienda a estos valores. Esto se consigue mediante una interpolación de las propiedades, el llamado **método SIMP** (Solid Isotropic Material with Penalisation).

El SIMP se basa en trabajar con el parámetro ρ elevado a un exponente, creando un modelo de combinación convexa en el que las propiedades del material sean lo más próximas como sea posible a los valores extremos de 0 y 1. Esto es:

$$\sigma = \sigma^M \rho^3 + (1 - \rho^3) \sigma^V \quad (7.7)$$

siendo

$$\sigma^M = 2\mu\epsilon + \lambda \operatorname{tr}(\epsilon)I \quad \sigma^V = \alpha(2\mu\epsilon + \lambda \operatorname{tr}(\epsilon)I) \quad (7.8)$$

De esta forma, se verifican los casos límites de la expresión (7.6), y para casos intermedios en la densidad ρ el SIMP va a aproximar los valores de los parámetros físicos del material al extremo más cercano.

De forma complementaria al SIMP, se utiliza una herramienta de **proyección de densidades**. Esto quiere decir que no se va a trabajar directamente con ρ , sino con una proyección de la misma, $\hat{\rho} = \hat{\rho}(\rho)$, que es una función tangente hiperbólica:

$$\hat{\rho} = \frac{\tanh(\beta\eta) + \tanh(\beta(\rho - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))} \quad (7.9)$$

En esta expresión, el parámetro β controla la pendiente de la curva, y suele tomar valores de entre 2 y 64, mientras que η controla la ubicación del dentro de la misma sobre el eje de abscisas, tomando valores intermedios entre 0 y 1. Para unos valores de $\beta = 8$ y $\eta = 0,5$, se obtiene la curva:

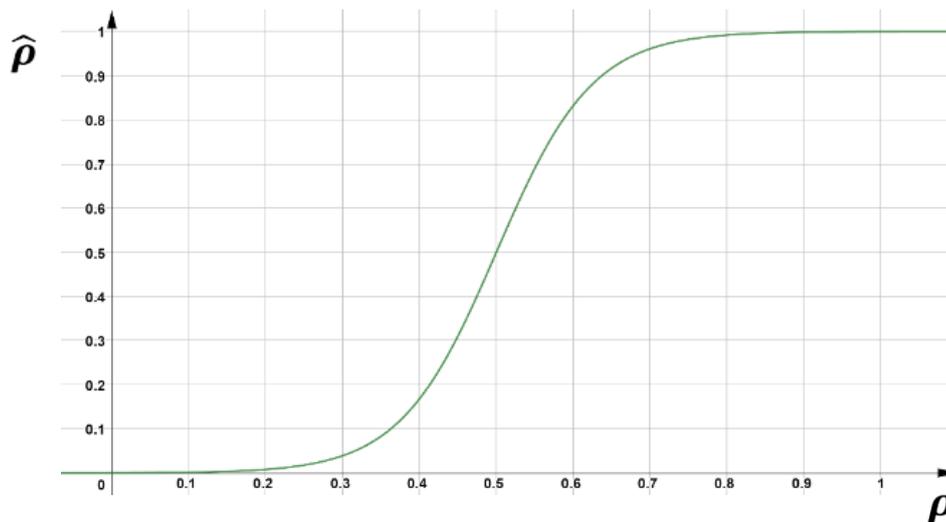


Figura 7.2. Función de proyección de ρ

Como se observa en la *Figura 7.2*, para valores intermedios de ρ , la función proyección aproxima los valores rápidamente a los extremos. Combinando el método SIMP (7.7) con la función proyección (7.9) se puede trabajar con la función continua ρ sin obtener valores intermedios, y por tanto el problema es equivalente al de la expresión (7.3), en la que solo hay valores extremos 0 y 1, y a la vez resoluble mediante el método de elementos finitos.

7.3. Filtrado de densidades

Todos los métodos explicados hasta ahora para la resolución del problema son insuficientes, ya que el problema está *ill-posed* (mal formulado). Esto quiere decir que, al representar la distribución de densidades obtenida, se obtendrá un patrón de ajedrez (*Figura 7.3*). en el que se encuentran nodos de material y de vacío adyacentes de forma alternante.



Figura 7.3. Patrón de ajedrez

Para solucionar esto y restaurar el problema, se aplica una técnica de **filtrado de densidades**. Para eso, se introduce el concepto de **densidad filtrada** $\tilde{\rho}(\rho)$, la cual se define como:

$$\tilde{\rho}(\mathbf{X}) = \int_D \rho(\mathbf{X}') \Delta(\|\mathbf{X} - \mathbf{X}'\|) d\mathbf{X}' \quad (7.10)$$

La función Δ , conocido como *kernel*, representa un promediado tal que la densidad filtrada $\tilde{\rho}(\mathbf{X})$ va a tener en cuenta el valor de la densidad ρ en \mathbf{X} , junto con los de sus puntos cercanos. De esta forma, para un radio de filtrado configurado se hará una media ponderada del valor de ρ . La *Figura 7.4*. representa un filtro de densidades con un radio de 2,5 elementos.

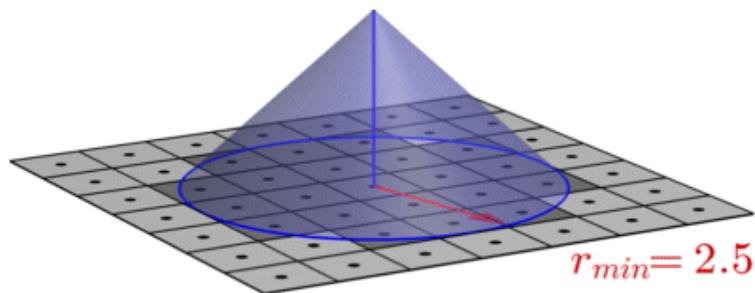


Figura 7.4. Filtro de densidades

Al aplicar este filtro, se evitan altos gradientes de densidades y, por tanto, se borra el patrón de ajedrez del problema, pudiendo obtener buenos resultados. Cabe mencionar que, forma alternativa al filtrado de densidades, se pueden utilizar otros tipos de filtros basados en PDEs, como la escrita a continuación.

$$W(\tilde{\rho}(\rho), \rho) = \int_D \tilde{\rho} \delta \rho dV + l^2 \int_D \nabla \tilde{\rho} \nabla \delta \rho dV - \int_D \rho \delta \rho dV = 0 \quad (7.11)$$

Donde l controla el radio de acción del filtro, y $W(\tilde{\rho}(\rho), \rho)$ es la forma débil de la PDE asociada al mismo, que es:

$$(\tilde{\rho} - \rho) - l^2 \Delta \tilde{\rho} = 0 \quad (7.12)$$

Para obtener la densidad filtrada, habría que resolver (7.11) implementándola como un problema de elementos finitos.

7.4. Formulación final y resolución del problema

Considerando todos los aspectos de los apartados anteriores, a partir del problema considerado inicialmente (7.3) se puede escribir la formulación matemática final del problema mecánico de optimización topológica, que consistiría en encontrar la distribución de densidades tal que:

$$\begin{aligned} \min J(\boldsymbol{\phi}, \rho) &= \mathbf{P} \cdot \mathbf{u} \text{ (en un punto dado)} \\ \text{sujeto a } &\left\{ \begin{array}{ll} \int_D \rho dV \leq f |D| & \\ \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, \hat{\rho}) = \mathbf{0} & \text{en } D \\ \mathbf{u} = 0 & \text{en } \partial D_D \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} & \text{en } \partial D_N \\ \rho \leq 1 & \\ \rho \geq 0 & \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \boldsymbol{\sigma}^V & \\ \hat{\rho} = \hat{\rho}(\tilde{\rho}(\rho)) & \end{array} \right. \end{aligned} \quad (7.13)$$

El problema (7.13) consiste, por tanto, en encontrar la distribución de densidades ρ tal que el desplazamiento en un punto dado (o varios) sea mínimo. El problema está

sujeto a la restricción de volumen, y debe cumplir las leyes constitutivas del material y la ecuación de equilibrio mecánico. El tensor de Cauchy $\boldsymbol{\sigma}$ está definido mediante el método SIMP en función del valor de $\hat{\rho}(\tilde{\rho}(\rho))$, que es la distribución de ρ a la que se le ha aplicado un filtro de densidades y posteriormente se ha proyectado. Por último, el parámetro ρ tiene como cotas superior e inferior 1 y 0 respectivamente.

Para la resolución del problema se crea un lagrangiano restringiendo la función objetivo con la satisfacción de la PDE de elasticidad lineal junto con sus condiciones de contorno y la restricción de volumen:

$$\begin{aligned} \mathcal{L}(\rho, \mathbf{u}, \mathbf{p}, \lambda) = & J(\mathbf{u}) + \int_D \mathbf{p} \cdot (\nabla \cdot \boldsymbol{\sigma} + \mathbf{f}) dV + \int_{\partial D_N} \mathbf{p} \cdot (\mathbf{t} - \boldsymbol{\sigma} \cdot \mathbf{n}) da \\ & + \lambda \left(\int_D \rho dV - f |D| \right) \end{aligned} \quad (7.14)$$

Haciendo integración por partes y teorema de la divergencia y separando términos, tal y como se hizo en capítulos anteriores, el lagrangiano queda como:

$$\mathcal{L} = \int_{\partial D_N} \mathbf{t} \cdot \mathbf{u} da - \int_D \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{p}) dV + \int_D \mathbf{p} \cdot \mathbf{f} dV + \int_{\partial D_N} \mathbf{p} \cdot \mathbf{t} da + G \quad (7.15)$$

donde $G = \lambda \left(\int_D \rho dV - f |D| \right)$.

En estas expresiones $J(\mathbf{u})$ es la función objetivo del problema, $\mathbf{p} \in H_0^1(D)$ es un multiplicador de Lagrange, el cual fuerza que se satisfaga la restricción asociada a la PDE, y λ es otro multiplicador de Lagrange que fuerza la restricción de volumen del último término. En base a este lagrangiano, se ignora el último término, que se satisface fácilmente en el código, y se toman la densidad ρ , el desplazamiento \mathbf{u} y el conocido como estado adjunto \mathbf{p} como variables independientes con el siguiente objetivo:

- 1) Tomar una distribución inicial de densidades homogénea de $\rho = f$, satisfaciéndose la restricción de volumen.
- 2) Ejecutar el algoritmo iterativo de optimización topológica:
 - a) Calcular \mathbf{u}

- b) *Calcular \mathbf{p}*
 - c) *Actualizar el nuevo valor de las densidades comprobando las restricciones de volumen y las cotas*
- 3) *Parar las iteraciones cuando la función objetivo J no decrezca más (criterio de convergencia).*

Aplicando el teorema de los multiplicadores de Lagrange, se calcularán los puntos críticos del lagrangiano \mathcal{L} con respecto de \mathbf{p} y \mathbf{u} . Para el punto crítico con respecto a \mathbf{p} se calcula la derivada direccional del mismo y se iguala a 0

$$D\mathcal{L}|\delta\mathbf{p}| = 0 \tag{7.16}$$

Partiendo de la expresión (7.15) y realizando la derivada direccional, se obtiene la forma débil del problema de equilibrio elástico, donde $\delta\mathbf{p}$ hace de función test:

$$D\mathcal{L}|\delta\mathbf{p}| = - \int_D \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u})) : \boldsymbol{\varepsilon}(\delta\mathbf{p}) dV + \int_D \delta\mathbf{p} \cdot \mathbf{f} dV + \int_{\partial D_N} \delta\mathbf{p} \cdot \mathbf{t} da = 0 \tag{7.17}$$

donde $\boldsymbol{\sigma} = \boldsymbol{\sigma}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \boldsymbol{\sigma}^V$. Esta ecuación de equilibrio obtenida a partir de la derivada direccional del lagrangiano con respecto del estado adjunto es conocida como **problema forward**, y a partir del mismo se obtendrá \mathbf{u} en todo el dominio.

De forma similar se calcula el punto crítico del lagrangiano con respecto de \mathbf{u} :

$$D\mathcal{L}|\delta\mathbf{p}| = 0 \tag{7.18}$$

Donde la función objetivo, el término $J(\mathbf{u})$, se escribe como una condición de contorno Neumann.

$$J(\mathbf{u}) = \int_{\partial D_N} \mathbf{t} \cdot \mathbf{u} da \tag{7.19}$$

Haciendo ahora la derivada direccional del lagrangiano (7.15) queda:

$$D\mathcal{L}|\delta\mathbf{u}| = \int_{\partial D_N} \mathbf{t} \cdot \delta\mathbf{u} \, da - \int_D \boldsymbol{\varepsilon}(\mathbf{p}) : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}(\mathbf{u})} \boldsymbol{\varepsilon}(\delta\mathbf{u}) \, dV = 0 \quad (7.20)$$

En la expresión (7.20) se encuentra el tensor de elasticidad $\mathbf{C} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}$, y se puede demostrar para un problema lineal que $\boldsymbol{\varepsilon}(\mathbf{p}) : \mathbf{C} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{p}))$, por tanto la expresión se puede reescribir como:

$$D\mathcal{L}|\delta\mathbf{u}| = \int_{\partial D_N} \mathbf{t} \cdot \delta\mathbf{u} \, da - \int_D \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{p})) : \boldsymbol{\varepsilon}(\delta\mathbf{u}) \, dV = 0 \quad (7.21)$$

Esta expresión, la derivada direccional del lagrangiano con respecto al desplazamiento, es conocida como **problema adjunto**, y a partir de él se obtiene el multiplicador de Lagrange \mathbf{p} para todo el dominio.

Si se observan ambas derivadas direccionales del lagrangiano, (7.18) y (7.23), al trabajar sobre un problema sin fuerzas volumétricas, se aprecia que son idénticas, cambiando \mathbf{u} por \mathbf{p} y viceversa. Es decir, que el problema es **auto-adjunto** porque el problema adjunto coincide con el problema forward y, por tanto, $\mathbf{p} = \mathbf{u}$. Esto significa que para este problema no es necesario calcular el estado adjunto, ya que será siempre igual a la incógnita del problema. Si el problema no fuera auto-adjunto, habría que resolver tanto el problema forward como el problema adjunto, creando un espacio de elementos finitos en FreeFEM para el problema forward y otro para el adjunto.

El siguiente paso es calcular una **dirección de descenso** evaluando la sensibilidad del Lagrangiano con respecto de la variable de diseño ρ . Esto es, estudiar cómo cambia el Lagrangiano al variar ρ , y esto se consigue haciendo la derivada direccional del mismo con respecto a incrementos de densidad, $D\mathcal{L}|\Delta\rho|$.

Cabe mencionar que se estudia la disminución del lagrangiano porque está compuesto por la función objetivo junto con la ecuación de equilibrio elástico y la

restricción de volumen, siendo ambos términos nulos por definición. Es decir, que al disminuir el valor del lagrangiano también lo está haciendo la función objetivo $J(\mathbf{u})$.

Para calcular la dirección de descenso, se puede escribir ρ^{k+1} como:

$$\rho^{k+1} = \rho^k + \Delta\rho \tag{7.22}$$

Además, como $\mathcal{L} = \mathcal{L}(\rho)$, se puede expresar $\mathcal{L}(\rho^{k+1})$ como su desarrollo de Taylor de primer orden:

$$\mathcal{L}(\rho^{k+1}) \approx \mathcal{L}(\rho) + \frac{\partial \mathcal{L}}{\partial \rho} \Delta\rho \tag{7.23}$$

Reordenando los términos, se llega a la expresión:

$$\Delta\mathcal{L} = \mathcal{L}(\rho^{k+1}) - \mathcal{L}(\rho) \approx \frac{\partial \mathcal{L}}{\partial \rho} \Delta\rho \tag{7.24}$$

Como se quiere minimizar la función objetivo J , y por tanto \mathcal{L} , se busca que $\Delta\mathcal{L} < 0$. Para conseguirlo, se puede definir $\Delta\rho$ como:

$$\Delta\rho = -\epsilon \frac{\partial \mathcal{L}}{\partial \rho} \tag{7.25}$$

De esta forma, $\Delta\mathcal{L}$ va a ser siempre negativo, ya que es igual a número negativo multiplicando por un número que, al estar elevado al cuadrado, va a ser siempre positivo:

$$\Delta\mathcal{L} = -\epsilon \left(\frac{\partial \mathcal{L}}{\partial \rho} \right)^2 \tag{7.26}$$

En la ecuación (7.27), $\frac{\partial \mathcal{L}}{\partial \rho}$ es la dirección de descenso calculada a partir del lagrangiano, y ϵ es el llamado *step size*, cuyo valor es dado por el algoritmo, y cuantifica cuánto desciende el mismo en la dirección $\frac{\partial \mathcal{L}}{\partial \rho}$ para cada iteración. Cuanto mayor sea el *step size*, más rápido decrecerá la función objetivo y el programa necesitará menos iteraciones para llegar a la distribución de densidades óptima. Sin embargo, ϵ debe ser

pequeño, ya que en la ecuación (7.25) se ha hecho un desarrollo de Taylor de primer orden, una aproximación que requiere de cambios pequeños en la distribución de densidades. Deberá haber, por tanto, un compromiso entre la rapidez del programa y la validez de las hipótesis consideradas para que $\Delta\mathcal{L}$ sea negativo.

Para calcular la dirección de descenso se parte de la expresión:

$$\mathcal{L} = \int_{\partial D_N} \mathbf{t} \cdot \mathbf{u} \, da - \int_D \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \hat{\rho}) : \boldsymbol{\varepsilon}(\mathbf{p}) \, dV + \int_{\partial D_N} \mathbf{p} \cdot \mathbf{t} \, da \quad (7.27)$$

Como $\hat{\rho} = \hat{\rho}(\tilde{\rho}(\rho))$, derivando con respecto ρ y aplicando la regla de la cadena:

$$\begin{aligned} D\mathcal{L}|\Delta\rho| &= -D \left(\int_D \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}), \hat{\rho}) : \boldsymbol{\varepsilon}(\mathbf{p}) \, dV \right) |\Delta\rho| \\ &= -D \left(\int_D [\boldsymbol{\sigma}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \boldsymbol{\sigma}^V] : \boldsymbol{\varepsilon}(\mathbf{p}) \, dV \right) |\Delta\rho| \\ &= - \int_D \left[3\hat{\rho}^2 \frac{\partial \hat{\rho}}{\partial \tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial \rho} \Delta\rho \boldsymbol{\sigma}^M - 3\hat{\rho}^2 \frac{\partial \hat{\rho}}{\partial \tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial \rho} \Delta\rho \boldsymbol{\sigma}^V \right] : \boldsymbol{\varepsilon}(\mathbf{p}) \, dV \end{aligned} \quad (7.28)$$

Finalmente, la dirección de descenso queda como:

$$D\mathcal{L}|\Delta\rho| = - \int_D \left[3\hat{\rho}^2 \frac{\partial \hat{\rho}}{\partial \tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial \rho} \Delta\rho \boldsymbol{\sigma}^M - 3\hat{\rho}^2 \frac{\partial \hat{\rho}}{\partial \tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial \rho} \Delta\rho \boldsymbol{\sigma}^V \right] : \boldsymbol{\varepsilon}(\mathbf{p}) \, dV \quad (7.29)$$

Esta dirección de descenso será la que se introduzca en el método de optimización, y éste escogerá el step size ϵ para entonces actualizar el nuevo valor de la densidad ρ en cada elemento de la malla. Para resolver el problema es necesario calcular los términos de la regla de la cadena $(\frac{\partial \hat{\rho}}{\partial \tilde{\rho}}, \frac{\partial \tilde{\rho}}{\partial \rho})$, lo cual no es trivial.

El primer término que se va a calcular es $\frac{\partial \hat{\rho}}{\partial \tilde{\rho}}$. Sabiendo que $\hat{\rho}$ se define como:

$$\hat{\rho} = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\rho} - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))} \quad (7.30)$$

Como la derivada de una función tangente hiperbólica es:

$$f(x) = \tanh(x) \quad f'(x) = 1 - \tanh^2(x) \quad (7.31)$$

El primero de los términos quedaría, aplicando la regla de la cadena:

$$\frac{\partial \hat{\rho}}{\partial \tilde{\rho}} = \frac{1 - \beta(\tanh(\beta(\tilde{\rho} - \eta)))^2}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))} \quad (7.32)$$

Para obtener $\frac{\partial \tilde{\rho}}{\partial \rho}$ hay que partir de la PDE del filtro de densidades visto anteriormente (ecuación 7.11). Ese filtro define $\tilde{\rho}$ como función implícita de ρ . Si derivamos la expresión por tanto con respecto a ρ :

$$\int_D \frac{\partial \tilde{\rho}}{\partial \rho} \delta \rho dV + l^2 \int_D \nabla \left(\frac{\partial \tilde{\rho}}{\partial \rho} \right) \nabla \delta \rho dV - \int_D \delta \rho dV = 0 \quad (7.33)$$

La ecuación obtenida es la solución de la forma débil asociada a la siguiente PDE:

$$\left(\frac{\partial \tilde{\rho}}{\partial \rho} \right) - l^2 \Delta \left(\frac{\partial \tilde{\rho}}{\partial \rho} \right) = 1 \quad (7.34)$$

Por tanto, el término $\frac{\partial \tilde{\rho}}{\partial \rho}$ se obtendrá como solución de la forma débil (7.33) que habrá que implementar como problema y resolver por elementos finitos en FreeFEM. Por lo tanto, para calcular la dirección de descenso habrá que implementar la ecuación (7.29), sustituyendo la (7.32) y la solución al problema (7.33) que se habrá resuelto anteriormente.

Una vez hechos todos los cálculos, ya se podrían implementar en FreeFEM para su resolución mediante el siguiente algoritmo:

- 1) *Tomar una distribución inicial de densidades homogénea de $\rho = f$, satisfaciéndose la restricción de volumen.*
- 2) *Algoritmo iterativo de optimización:*
 - a) *Filtrado de densidades $\tilde{\rho}$*
 - b) *Proyectado de densidades $\hat{\rho}$*
 - c) $\boldsymbol{\sigma} = \boldsymbol{\sigma}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \boldsymbol{\sigma}^V$
 - d) *Problema Forward para hallar \mathbf{u}*
 - e) *Problema adjunto: $\mathbf{p} = \mathbf{u}$*
 - f) *Cálculo de sensibilidad $\frac{\partial \mathcal{L}}{\partial \rho} \left(\frac{\partial \hat{\rho}}{\partial \tilde{\rho}}, \frac{\partial \tilde{\rho}}{\partial \rho} \right)$*
 - g) *Actualización del valor de las densidades*
- 3) *Parar las iteraciones cuando la función objetivo J no decrezca más (criterio de convergencia).*

7.5. Filtrado de sensibilidades

De forma alternativa al filtro de densidades, algunos autores prefieren filtrar las sensibilidades, cuya implementación algorítmica es la siguiente:

- 1) *Tomar una distribución inicial de densidades homogénea de $\rho = f$, satisfaciéndose la restricción de volumen.*
- 2) *Algoritmo iterativo de optimización:*
 - a) $\boldsymbol{\sigma} = \boldsymbol{\sigma}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \boldsymbol{\sigma}^V$
 - b) *Problema Forward para hallar \mathbf{u}*
 - c) *Problema adjunto: $\mathbf{p} = \mathbf{u}$*
 - d) *Cálculo de sensibilidad $\frac{\partial \mathcal{L}}{\partial \rho}$*
 - e) *Filtrado de sensibilidad $\frac{\partial \mathcal{L}}{\partial \rho}$*
 - f) *Actualización del valor de las densidades*
- 3) *Parar las iteraciones cuando la función objetivo J no decrezca más (criterio de convergencia).*

De esta forma, ni se filtran ni se proyectan las densidades, sino que se obtiene una sensibilidad “mala” a partir de las densidades sin filtrar, y se le aplica un filtro para la actualización del valor de las densidades, haciendo más sencillo el cálculo de la sensibilidad y disminuyendo los pasos previos. Este filtro está basado en PDEs, de forma similar al filtro de densidades (7.11). Para la implementación del problema mecánico se utilizará este tipo de filtro, mientras que para el problema piezoeléctrico se usará filtro de densidades. De esta forma se apreciará el funcionamiento y comportamiento de ambos.

CAPÍTULO 8: IMPLEMENTACIÓN DE OPTIMIZACIÓN TOPOLÓGICA DEL PROBLEMA MECÁNICO

En este capítulo se explicarán los aspectos más importantes del código desarrollado en freeFEM, y se expondrán 4 ejemplos sencillos de optimización topológica para el problema mecánico, variando las condiciones de contorno y la función objetivo para la misma geometría. El código completo del primer ejemplo se encuentra en el apartado ANEXOS, mientras que para el resto de ejemplos está la creación de sus respectivas mallas, que es lo único que varía.

8.1. Código genérico

Después de cargar el paquete “iovtk” para exportar los datos a Paraview, el primer paso del código es la creación de la malla (*Figura 8.1*). Se ha hecho de forma que se puedan cambiar las condiciones de contorno o afinar la malla fácilmente para la realización de distintos ejemplos.

```
//Preproceso: geometria, malla, parametros del modelo

real L = 4;

real W = 1;

int loaded = 1, free = 2, fixed = 3;

border b1(t=-0.025*W,0.025*W){x=0.5*L;y=t;label=loaded;}

border b2(t=0.025*W,0.5*W){x=0.5*L;y=t;label=free;}

border b3(t=0.5*L,-0.5*L){x=t;y=0.5*W;label=free;}

border b4(t=0.5*W,-0.5*W){x=-0.5*L;y=t;label=fixed;}

border b5(t=-0.5*L,0.5*L){x=t;y=-0.5*W;label=free;}

border b6(t=-0.5*W,-0.025*W){x=0.5*L;y=t;label=free;}
```

```
//Malla

int s = 2;

mesh Th=buildmesh(b1(2*s) + b2(24*s) + b3(200*s) + b4(50*s) + b5(200*s) +
b6(24*s));
```

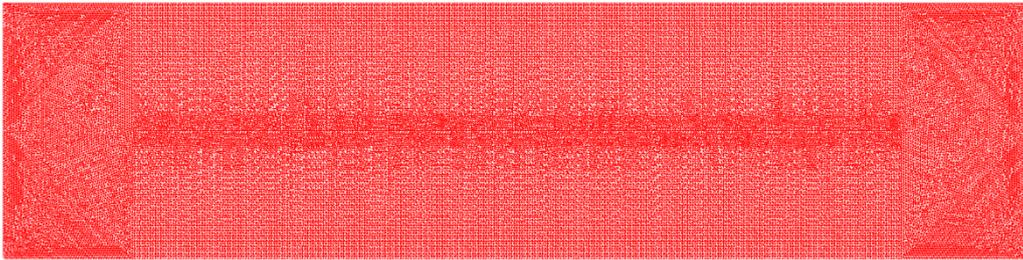


Figura 8.1. Malla del problema de optimización topológica mecánico

A continuación, se crea el espacio de elementos finitos para la solución y función test, así como para las densidades (que en el código se han llamado theta) y sensitividades:

```
//Espacios de elementos finitos

fespace Vh(Th,P1); //elementos de Lagrange P1

Vh ux,uy,vx,vy;

fespace Vh1(Th,P1);

Vh1 theta, thetaold, thetanew, dtheta; //densidades

Vh1 sens, sensn, sensv; //sensitividades
```

El siguiente paso es introducir los parámetros de la optimización, así como los parámetros mecánicos del modelo. En esta parte del código, cabe destacar que “volfrac” es la restricción de volumen, p es el exponente del método SIMP, el cual irá actualizándose con las iteraciones, y alpha controla el radio de filtrado, que en este caso será de sensitividades. En cuanto a los parámetros mecánicos, epsilon es el coeficiente de escalado para la densidad en el vacío.

```

// parametros de la optimización

real volfrac = 0.5;

real p = 1.0;

real chg = 1;

real alpha = (W/20)^2.; //controla la length-scale del filtro de Helmholtz

int i = 0;

int Imax = 100;

// parametros mecánicos modelo

real nu=0.28; //coeficiente de Poisson del acero

real E=1e9;

real mu = E/(2*(1+nu));

real lambda = 2.*mu*nu/(1.-2.*nu);

real epsilon = 1e-6;

real fy= -1.0; //carga distribuida

real fx= 0.0; //carga distribuida

```

En cuanto a los macros, se incluyen todos los macros necesarios para escribir el problema mecánico, así como los términos del SIMP y la sensibilidad:

```

macro IdentityMatrix [1.0, 0.0, 0.0, 1.0] //EOM

macro Trace(a) (a[0] + a[3]) //EOM

macro epsilonmatrix(u1,u2) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dy(u1) + dx(u2)),
dy(u2)] //EOM

macro stress0(u1,u2)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2)) +2.
*mu*epsilonmatrix(u1,u2))//EOM

macro stressM(u1,u2)(theta^p*stress0(u1,u2)) //EOM

```

```

macro stressV(u1,u2)((1.0-theta^p)*epsilon*stress0(u1,u2)) //EOM

macro stress (u1,u2)(stressM(u1,u2) + stressV(u1,u2))//EOM

macro sensivityGauss(u1,u2)(-p*theta^(p-1)*(stressM(u1,u2) -
stressV(u1,u2))*epsilonmatrix(u1,u2))//EOM

```

El resto del código es el bucle de optimización. Primero se comienza inicializando la densidad con la restricción de volumen y entonces comienza el bucle. En él, se actualiza el contador y el exponente del método SIMP, se resuelve el problema Forward (el adjunto no hace falta al ser el problema auto-adjunto), se calcula y se filtra la sensibilidad mediante el llamado *Filtro de Helmholtz* y se aplica el criterio de optimalidad para calcular el step size y obtener la nueva distribución de densidades. Después se calcula la función objetivo, se guarda el archivo.vtk y se repite el bucle.

```

theta = volfrac;

// Optimization loop
while (chg > 1e-3 && i<=Imax){

    // incrementamos iteración de optimización
    i=i+1;

    //vieja densidad == nueva (la de la iteración anterior)

    thetaold=theta;

    //Update of penalising exponent

    if (i>20){

        p = min(3.0,p*1.008);

    }

    cout <<"exponent = " << p <<endl;

```

```

//Forward (FEM) problem

solve
FEMProblem([ux,uy],[vx,vy])=int2d(Th)((stress(ux,uy)*epsilonmatrix(vx,vy)))
- int1d(Th,loaded)(fx*vx+fy*vy)+on(fixed,ux=0,uy=0);

//Adjoint equation

// No es necesaria, problema auto-adjunto

//Sensitivity

sens = -sensitivityGauss(ux,uy);

//Regularised sensitivity through the application of a Helmholtz's filter

solve smoothing(sensn,sensv)=
int2d(Th)(alpha*(dx(sensn)*dx(sensv)+dy(sensn)*dy(sensv))+sensn*sensv)-
int2d(Th)(sens*sensv);

//OC update

real l1 = 0; real l2= 10000; real move= 0.05;

while((l2-l1)/(l2+l1)>(1e-4)&& l2>1e-40){

real lmid=0.5*(l2+l1);

thetaneu = max(0., max(theta-move,min(1.,min(theta+move,theta*(max(1e-
10,sensn/lmid))^0.3)))));

if((int2d(Th)(thetaneu)-int2d(Th)(volfrac))>0){

l1= lmid;}

else{

l2=lmid;}

}

theta = thetaneu;

```

```

// Objective functions and visualization

real J = int1d(Th,loaded)(fx*ux + fy*uy);

real vol = int2d(Th)(theta)/int2d(Th)(1.);

cout <<"iter = " << i << "; comp = " <<J<<"; vol = " <<vol<< ";change = " << chg<<".----
-----" <<endl;

savevtk("Benchmark_1_" + i + ".vtk", Th, theta, i);

}

```

8.2. Benchmark 1

El primero de los problemas realizados es un benchmark empotrado en una de sus caras y sometido a una carga puntual tal y como se indica en la *Figura 8.2*.



Figura 8.2. Benchmark 1

Se ha configurado para este ejemplo una restricción de volumen de 0,5. En la figura 8.3 se muestra cómo varía la distribución de densidades en el espacio de diseño con las iteraciones del bucle de optimización.

Iteración 0



Iteración 20



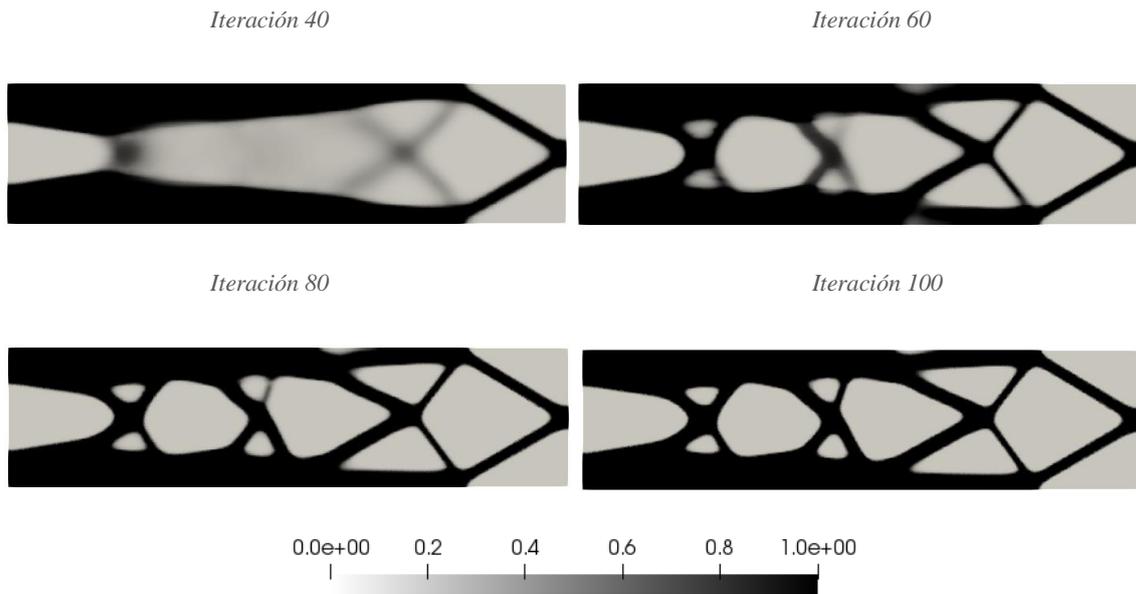


Figura 8.3. Evolución en la distribución de densidades del benchmark 1

Como se puede observar en la figura, la densidad comienza siendo igual a 0,5 en todos los puntos para cumplir con la restricción de volumen, pero ya en la primera iteración se vislumbra la forma que empieza a tener la distribución de densidades. En la iteración 20 se está formando la punta característica en estos problemas de optimización topológica, tras la cual se forman dos cruces a partir de la iteración 40, y a partir de la 60 se empieza a formar una tercera, acabando de definirse plenamente el diseño en la iteración 100.

8.3. Benchmark 2

En este caso, el benchmark está sujeto a desplazamientos en un punto a ambos lados del mismo, y la carga se le aplica en el punto central de la cara inferior.

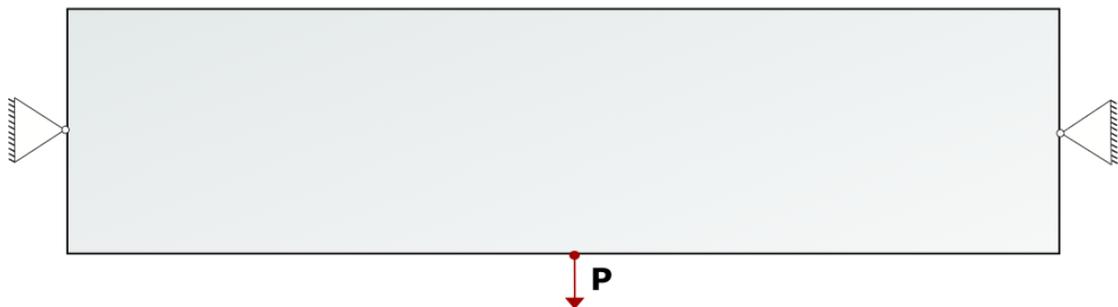


Figura 8.4. Benchmark 2

En esta ocasión, se ha considerado una restricción de volumen de 0,2. La evolución en la distribución de densidades es:

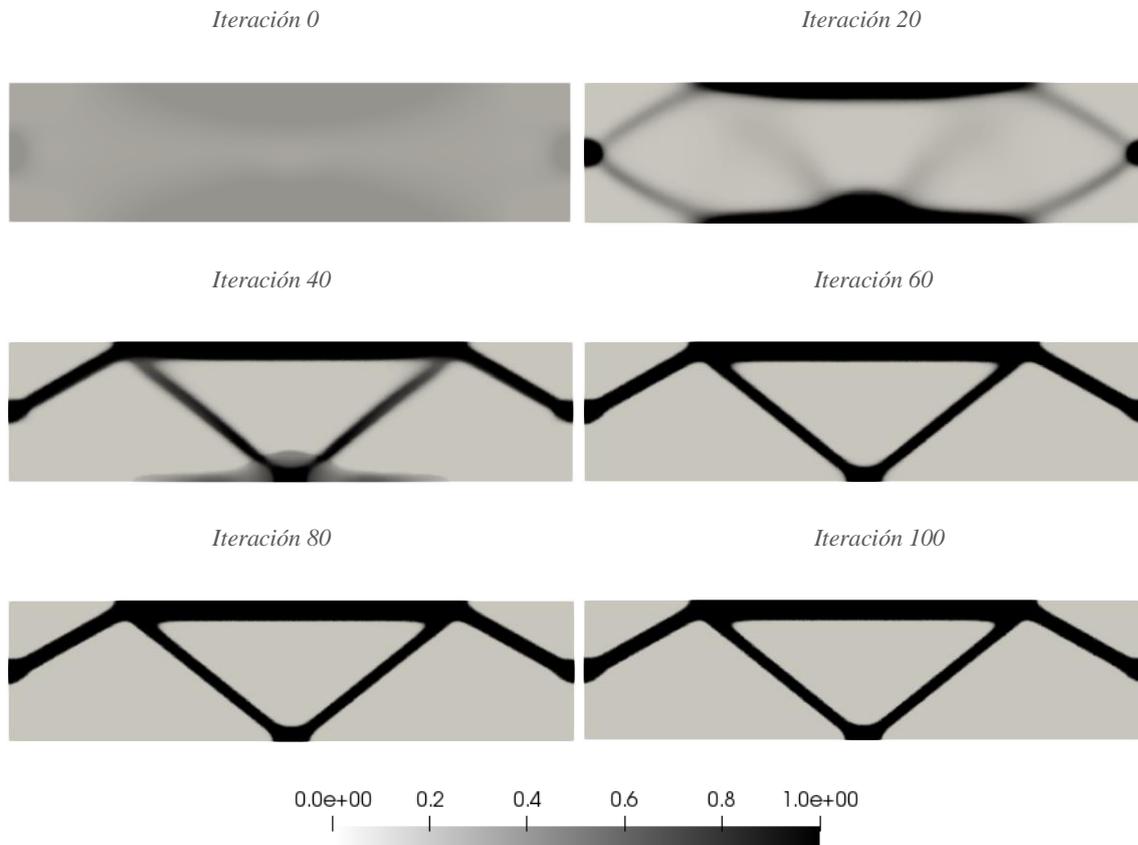


Figura 8.5. Evolución en la distribución de densidades del benchmark 2

Como la restricción de volumen para este caso es muy pequeña, el algoritmo encuentra rápidamente la distribución de densidades óptima. En la iteración 20 se empieza a observar el patrón que sigue el criterio, pero conforme avanza el bucle se va borrando en la parte inferior para terminar de formarse sólo la de arriba, junto con la forma triangular del centro. A partir de la iteración 60 la distribución es prácticamente óptima y los cambios son casi inapreciables.

8.4. Benchmark 3

Para este problema se restringen los desplazamientos horizontales en la cara izquierda, mientras que se restringen los desplazamientos verticales en la esquina inferior derecha. La carga está aplicada en la esquina superior izquierda.



Figura 8.6. Benchmark 3

La restricción de volumen es de 0,5 y la evolución en la distribución de densidades queda:

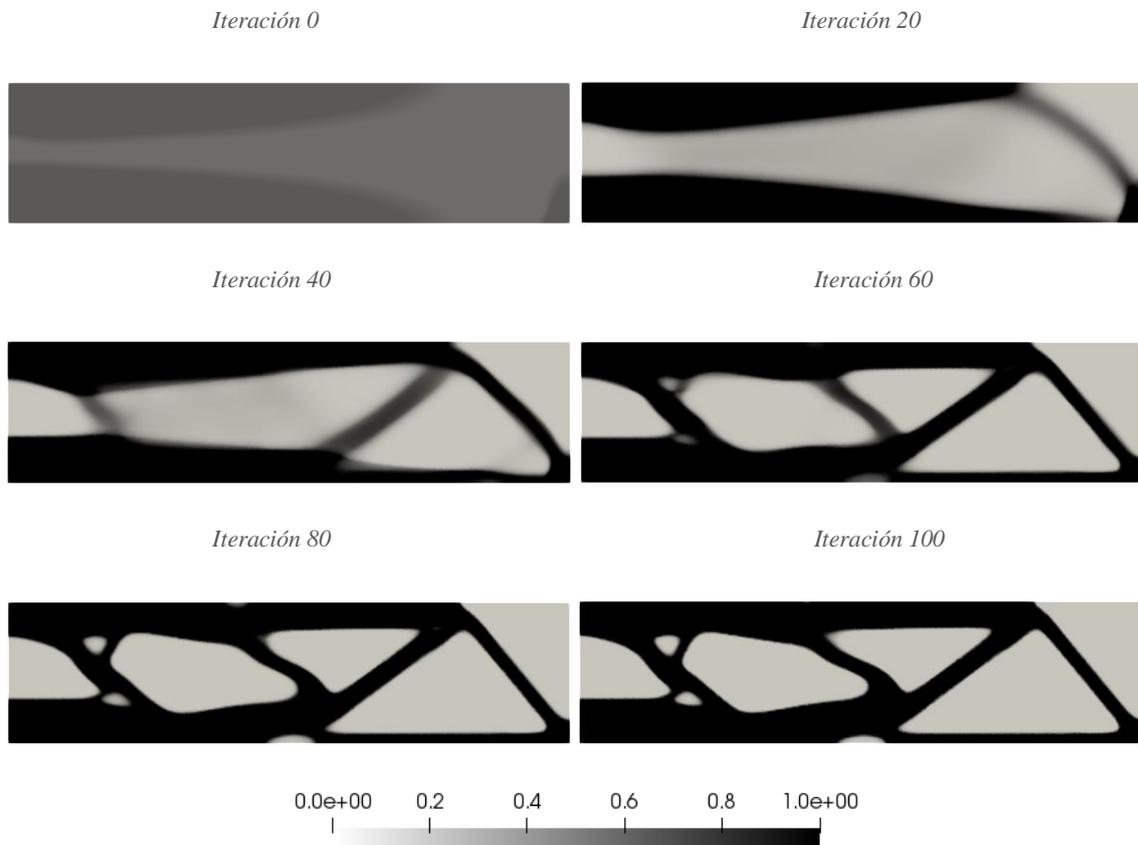


Figura 8.7. Evolución en la distribución de densidades del benchmark 3

En este caso la distribución de densidades es asimétrica debido a que las condiciones de contorno también lo son. Por esta razón no se generan unas cruces claras como en el primero de los problemas, aunque sí se obtiene un patrón algo similar. En la iteración 20 se empieza a formar la punta, y a partir de la 40 la forma triangular y la línea transversal de la izquierda. A partir de la iteración 60 se empieza a dibujar la cruz a partir de la línea transversal, y aparece otra línea, generando otro patrón triangular.

8.5. Benchmark 4

Por último, se analiza este problema en el que hay dos cargas implicadas, situadas en la esquina superior izquierda y en el centro de la cara superior. En cuanto a los apoyos, toda la cara izquierda tiene restringidos los desplazamientos horizontales, mientras que la esquina inferior derecha está fijada frente a desplazamientos.

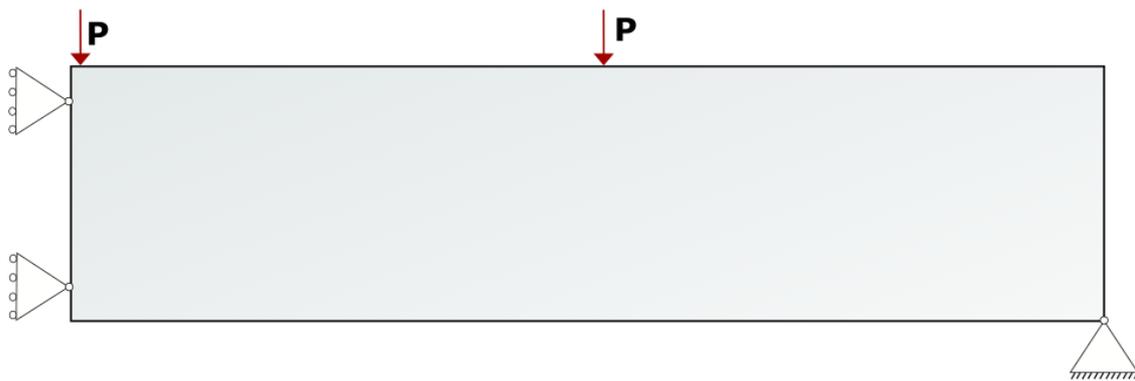


Figura 8.8. Benchmark 4

La restricción de volumen es, 0,5 de nuevo, quedando la siguiente distribución de densidades:

Iteración 0



Iteración 20



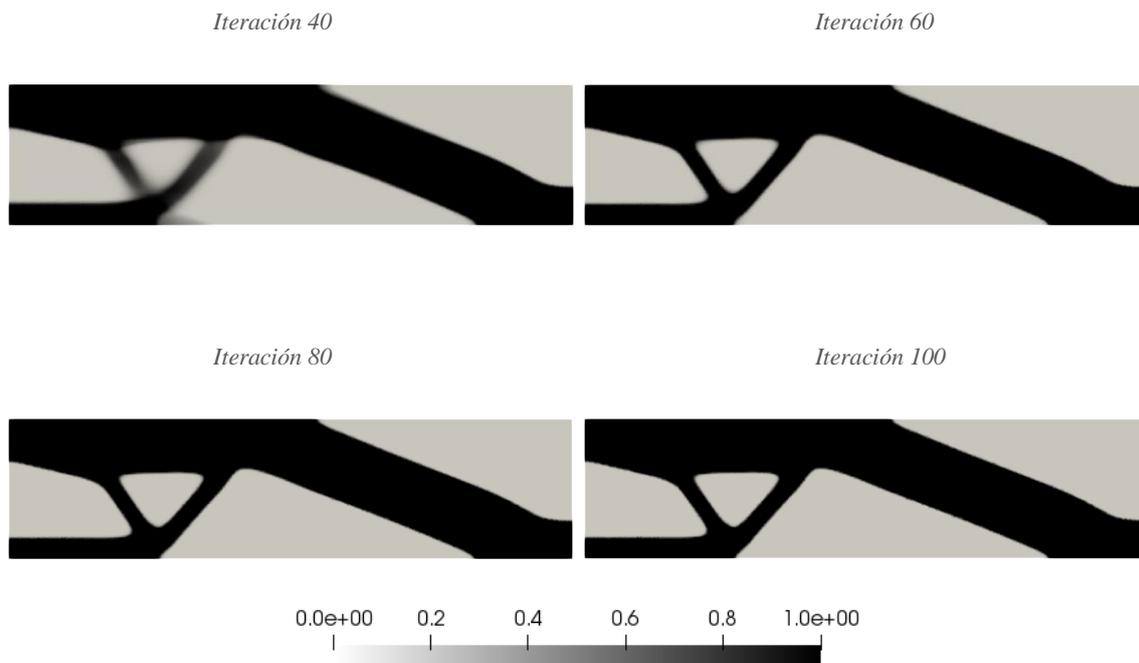


Figura 8.9. Evolución en la distribución de densidades del benchmark 4

Se aprecia cómo el material se distribuye especialmente donde se encuentran las condiciones de contorno, tanto apoyos como fuerzas impuestas. En la iteración 20 ya se ha formado el patrón principal que une las zonas donde se aplican las fuerzas con el apoyo que restringe los desplazamientos verticales y horizontales. Además, se empieza a vislumbrar la aparición de un patrón triangular que une la cara superior e inferior del espacio de diseño, la cual se termina de formar en torno a la iteración 60. Los cambios hasta la iteración 100 son casi inapreciables, ya que se alcanza la distribución de densidades óptima bastante rápido.

CAPÍTULO 9: OPTIMIZACIÓN TOPOLÓGICA DEL PROBLEMA ELECTRO-MECÁNICO

9.1. Formulación y resolución del problema

Para abordar el problema de optimización topológica en piezoeléctricos, se puede partir de la formulación del problema en el contexto mecánico del capítulo 7 y cambiar las leyes constitutivas correspondientes. En este caso, para el benchmark de la *Figura 9.1*, el problema podría consistir, por ejemplo, en minimizar el desplazamiento producido por la fuerza P , o lo que es lo mismo, maximizar los desplazamientos en sentido contrario. En este caso, además de las condiciones de contorno mecánicas, también hay potencial eléctrico impuesto en las caras superior e inferior.

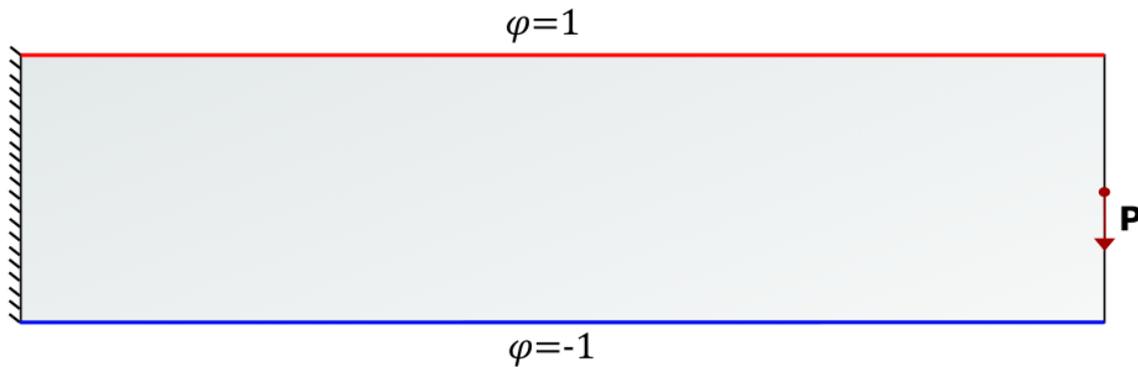


Figura 9.1. Espacio de diseño de optimización topológica en piezoeléctricos

El problema, por tanto, consistirá en encontrar la distribución de densidades óptima para minimizar una función objetivo, en un espacio de diseño ocupado por un material con leyes constitutivas piezoeléctricas y con una restricción de volumen dada y sometido a condiciones de contorno mecánicas y eléctricas.

$$\begin{aligned}
& \min J(\mathbf{u}) = \mathbf{P} \cdot \mathbf{u} \text{ (en un punto dado)} \\
& \text{sujeto a } \left\{ \begin{array}{ll} \int_D \rho \, dV \leq f |D| & \\ \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{0} & \text{en } D \\ \nabla \cdot \mathbf{D}(\varphi) = 0 & \text{en } D \\ \mathbf{u} = 0 & \text{en } \partial D_{Du} \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} & \text{en } \partial D_{Nu} \\ \varphi = \varphi^* & \text{en } \partial D_{D\varphi} \\ \mathbf{D} \cdot \mathbf{n} = 0 & \text{en } \partial D_{N\varphi} \\ 0 \leq \rho \leq 1 & \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \boldsymbol{\sigma}^V & \\ \mathbf{D} = \mathbf{D}^M \hat{\rho}^3 + (1 - \hat{\rho}^3) \mathbf{D}^V & \\ \hat{\rho} = \hat{\rho}(\tilde{\rho}(\rho)) & \end{array} \right. \quad (9.1)
\end{aligned}$$

El exponente utilizado en el método SIMP del problema piezoeléctrico puede ser diferente para el problema eléctrico y el problema mecánico, aunque en este caso sea en ambos igual a 3. De nuevo, para la resolución de este problema se definirá un lagrangiano que contiene la función objetivo, las leyes constitutivas mecánicas y eléctricas que se van a forzar nulas mediante los estados adjuntos mecánico \mathbf{p} y eléctrico α y la restricción de volumen $G = \lambda \left(\int_D \rho \, dV - f |D| \right)$.

$$\mathcal{L}(\rho, \mathbf{u}, \varphi, \mathbf{p}, \alpha, \lambda) = J(\mathbf{u}) - \left[\int_D \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{p}) \, dV - \int_{\partial D_N} \mathbf{p} \cdot \mathbf{t} \, da - \int_D \mathbf{D} \cdot \mathbf{E}(\alpha) \, dV \right] + G \quad (9.2)$$

Haciendo la derivada direccional del lagrangiano con respecto a los estados adjuntos mecánico y eléctrico respectivamente, se obtiene de nuevo la forma débil de los respectivos problemas, que van a constituir el problema forward piezoeléctrico:

$$\begin{aligned}
D\mathcal{L}|\delta\mathbf{p}| &= - \int_D \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta\mathbf{p}) \, dV + \int_{\partial D_N} \delta\mathbf{p} \cdot \mathbf{t} \, da = 0 \\
D\mathcal{L}|\delta\alpha| &= \int_D \mathbf{D} \cdot \mathbf{E}(\delta\alpha) \, dV = 0 \quad (9.3)
\end{aligned}$$

A partir de la implementación de (9.3) se obtendrán los desplazamientos y el potencial en todos los puntos del espacio de diseño. A continuación, y debido a que este

problema no es auto-adjunto, hay que obtener el problema adjunto mecánico derivando con respecto al desplazamiento, y el problema adjunto eléctrico derivando con respecto al potencial. Empezando con la derivada con respecto al desplazamiento:

$$D\mathcal{L}|\delta\mathbf{u}| = DJ|\delta\mathbf{u}| - \int_D \boldsymbol{\varepsilon}(\mathbf{p}): D\boldsymbol{\sigma}|\delta\mathbf{u}|dV + \int_D \mathbf{E}(\alpha) \cdot D\mathbf{D}|\delta\mathbf{u}|dV = 0 \quad (9.4)$$

Como el problema es lineal, se puede demostrar:

$$D\boldsymbol{\sigma}|\delta\mathbf{u}| = \boldsymbol{\sigma}(\delta\mathbf{u}) \qquad D\mathbf{D}|\delta\mathbf{u}| = \mathbf{D}(\delta\mathbf{u}) \quad (9.5)$$

Por lo tanto, la ecuación (9.4) se puede reescribir como:

$$D\mathcal{L}|\delta\mathbf{u}| = DJ|\delta\mathbf{u}| - \int_D \boldsymbol{\varepsilon}(\mathbf{p}): \boldsymbol{\sigma}(\delta\mathbf{u}, \varphi)dV + \int_D \mathbf{E}(\alpha) \cdot \mathbf{D}(\delta\mathbf{u}, \varphi)dV = 0 \quad (9.6)$$

Haciendo ahora la derivada direccional del lagrangiano con respecto al potencial eléctrico se obtiene la segunda ecuación del problema adjunto.

$$D\mathcal{L}|\delta\varphi| = - \int_D \boldsymbol{\varepsilon}(\mathbf{p}): D\boldsymbol{\sigma}|\delta\varphi|dV + \int_D \mathbf{E}(\alpha) \cdot D\mathbf{D}|\delta\varphi|dV = 0 \quad (9.7)$$

Nuevamente, por ser el problema lineal, se puede reescribir la ecuación como:

$$D\mathcal{L}|\delta\mathbf{u}| = - \int_D \boldsymbol{\varepsilon}(\mathbf{p}): \boldsymbol{\sigma}(\mathbf{u}, \delta\varphi)dV + \int_D \mathbf{E}(\alpha) \cdot \mathbf{D}(\mathbf{u}, \delta\varphi)dV = 0 \quad (9.8)$$

El problema adjunto queda por tanto compuesto por las ecuaciones (9.6) y (9.8), y habrá que sumarlas para introducirlas en el código en FreeFEM.

9.2. Cálculo de la sensibilidad con filtrado de densidades

Para el problema piezoeléctrico, donde se le añade al lagrangiano la ecuación constitutiva del desplazamiento eléctrico, la dirección de descenso queda:

$$\begin{aligned}
 D\mathcal{L}|\Delta\rho| = & - \int_D \left[3\hat{\rho}^2 \frac{\partial\hat{\rho}}{\partial\tilde{\rho}} \frac{\partial\tilde{\rho}}{\partial\rho} \Delta\rho \boldsymbol{\sigma}^M - 3\hat{\rho}^2 \frac{\partial\hat{\rho}}{\partial\tilde{\rho}} \frac{\partial\tilde{\rho}}{\partial\rho} \Delta\rho \boldsymbol{\sigma}^V \right] : \boldsymbol{\varepsilon}(\boldsymbol{p}) dV \\
 & + \int_D \left[3\hat{\rho}^2 \frac{\partial\hat{\rho}}{\partial\tilde{\rho}} \frac{\partial\tilde{\rho}}{\partial\rho} \Delta\rho \boldsymbol{D}^M - 3\hat{\rho}^2 \frac{\partial\hat{\rho}}{\partial\tilde{\rho}} \frac{\partial\tilde{\rho}}{\partial\rho} \Delta\rho \boldsymbol{D}^V \right] : \boldsymbol{E}(\boldsymbol{\alpha}) dV
 \end{aligned} \tag{9.9}$$

Donde las expresiones $(\frac{\partial\hat{\rho}}{\partial\tilde{\rho}}, \frac{\partial\tilde{\rho}}{\partial\rho})$, se han calculado en el capítulo 7 , en las ecuaciones (7.32) y (7.33)

CAPÍTULO 10: IMPLEMENTACIÓN DE OPTIMIZACIÓN TOPOLÓGICA DEL PROBLEMA ELECTRO-MECÁNICO

El problema piezoeléctrico que se ha resuelto en este capítulo es el de la *Figura 9.1*, cuya geometría es la misma que para los ejemplos mecánicos. Su condición de contorno mecánica es estar empotrado en la cara izquierda, y las eléctricas son un potencial positivo en la cara superior y otro negativo en la cara inferior. Se busca maximizar el desplazamiento vertical negativo, o lo que es lo mismo, minimizar el positivo, en el punto **P**. A este punto se le ha añadido una fuerza (muelle) que se opone a la deformación y se han estudiado 4 casos: Sin muelle, muelle blando, muelle intermedio y muelle duro. Además, se estudiará mediante otro caso la influencia del vector de polarización y se resolverán los casos de maximización de desplazamiento horizontal, tanto de tracción como de compresión.

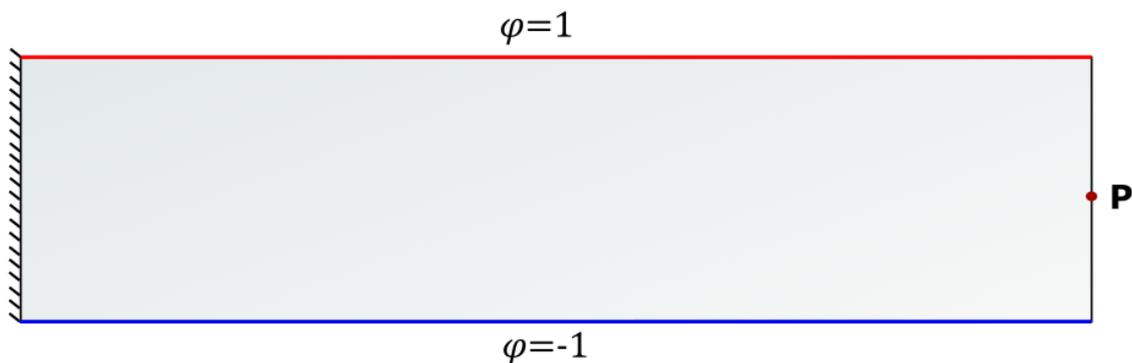


Figura 10.1. Espacio de diseño de optimización topológica en piezoeléctricos

10.1. Código genérico

Al igual que para el caso mecánico, después de cargar el paquete “iovtk” el primer paso es la creación de la malla. La geometría es la misma, pero se añaden dos etiquetas para las caras con carga positiva y negativa:

```
//Preproceso: geometria, malla, parametros del modelo
```

```
real L = 4;
```

```
real W = 1;
```

```

int loaded = 1, free = 2, fixed = 3, charged1 = 4, charged2 = 5;

border b1(t=-0.025*W,0.025*W){x=0.5*L;y=t;label=loaded;}

border b2(t=0.025*W,0.5*W){x=0.5*L;y=t;label=free;}

border b3(t=0.5*L,-0.5*L){x=t;y=0.5*W;label=charged1;}

border b4(t=0.5*W,-0.5*W){x=-0.5*L;y=t;label=fixed;}

border b5(t=-0.5*L,0.5*L){x=t;y=-0.5*W;label=charged2;}

border b6(t=-0.5*W,-0.025*W){x=0.5*L;y=t;label=free;}

//Malla

int s = 1;

mesh Th=buildmesh(b1(2*s) + b2(24*s) + b3(200*s) + b4(50*s) + b5(200*s) +
b6(24*s));

```

El siguiente paso es introducir todos los parámetros de la optimización, así como los parámetros materiales.

```

// parametros de la optimización

real volfrac = 0.5;

real pm = 3.0; // exponente SIMP mecánico

real pe = 3.0; // exponente SIMP eléctrico

real chg = 1;

real beta = 2.0; // Parámetro de proyección

real eta = 0.5; // Parámetro de proyección

real alpha = (W/20)^2.; // lenght-scale del filtro de densidades

int i = 0;

int Imax = 300;

// parametros de materiales

real permittivity=4*8.8e-12;

```

```

real nu=0.28; //coeficiente de Poisson del acero

real E=1e0;

real mu = E/(2*(1+nu));

real lambda = 2.*mu*nu/(1.-2.*nu);

real alpha1 = 0.001;

real alpha2 = 0.001;

real epsilon = 1e-6; // Parámetro de escalado para el material vacío

real fy= -1.0; //carga distribuida

real fx= 0.0; //carga distribuida

real spring= 1e-1; //muelle

```

A continuación, se generan los espacios de elementos finitos para la resolución de los problemas correspondientes:

```

//Espacios de elementos finitos

espace Vh1(Th,P1); //elementos de Lagrange P1

Vh1 vx, vy, vphi; //funciones test

Vh1 ux, uy, phi; //soluciones del problema piezo

Vh1 px, py; //estado adjunto mecánico

Vh1 q; //estado adjunto eléctrico

Vh1 vx, vy, vphi; //funciones test

Vh1 theta, thetaold, thetanew, dtheta; //densidades

Vh1 sens, sensn; //sensitividades

Vh1 thetaf, thetap, thetav;

Vh1 DthetapDthetaf, DthetafDtheta;

```

En cuanto a las macros, se introducirán todos los necesarios para escribir las leyes constitutivas del material, así como los términos del método SIMP, parte de la sensibilidad y la función objetivo.

```

macro IdentityMatrix [1.0, 0.0, 0.0, 1.0] //EOM

macro Trace(a) (a[0] + a[3]) //EOM

macro epsilonmatrix(u1,u2) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dy(u1) + dx(u2)),  
dy(u2)] //EOM

macro Campoelectrico(phi) [-dx(phi),-dy(phi)]//EOM

macro N [1,0]//EOM

macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[1]*B[0], A[1]*B[1]] //EOM

macro mult (A,b)[A[0]*b[0]+A[1]*b[1],A[2]*b[0]+A[3]*b[1]]//EOM

macro stress0 (u1,u2,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2)) +2.  
*mu*epsilonmatrix(u1,u2) - alpha1*(Campoelectrico(phi))*N)*IdentityMatrix -  
alpha2/2*(DyadicProduct(Campoelectrico(phi),N) +  
DyadicProduct(N,Campoelectrico(phi))))//EOM

macro stressM(u1,u2,phi)(thetap^pm*stress0(u1,u2,phi)) //EOM

macro stressV(u1,u2,phi)((1.0-thetap^pm)*epsilon*stress0(u1,u2,phi)) //EOM

macro stress (u1,u2,phi)(stressM(u1,u2,phi) + stressV(u1,u2,phi))//EOM

macro desplazamiento0(u1,u2,phi)(epsilon*Campoelectrico(phi) +  
alpha1*Trace(epsilonmatrix(u1,u2))*N + alpha2*mult(epsilonmatrix(u1,u2),N)  
)//EOM

macro desplazamientoM(u1,u2,phi)(thetap^pe*desplazamiento0(u1,u2,phi))  
//EOM

macro desplazamientoV(u1,u2,phi)((1.0-  
thetap^pe)*epsilon*desplazamiento0(u1,u2,phi)) //EOM

macro desplazamiento (u1,u2,phi)(desplazamientoM(u1,u2,phi) +  
desplazamientoV(u1,u2,phi))//EOM

```

```

macro sensitivityGauss(u1,u2,phi,p1,p2,q)(-pm*thetap^(pm-1)*(stressM(u1,u2,phi) -
stressV(u1,u2,phi))*epsilonmatrix(p1,p2) + pe*thetap^(pe-
1)*(desplazamientoM(u1,u2,phi) -
desplazamientoV(u1,u2,phi))*Camptoelectrico(q))//EOM

macro Objective(u1,u2) (u2) //EOM

macro DObjective(u1,u2) (1) //EOM

```

El resto del código es el bucle de optimización. Primero se inicializa la distribución de densidades con la restricción de volumen, se crea un fichero .dat para guardar los valores de la función objetivo en cada iteración y se comienza el bucle. En él, se actualiza la densidad y se la somete al filtrado y proyectado, para después resolver el problema forward y el adjunto y calcular la sensibilidad aplicando la regla de la cadena debido a que hay filtrado y proyectado de densidades. Después, se implementa el criterio de optimalidad para encontrar el step size adecuado y se calcula y guarda el valor de la función objetivo.

```

// Inicialización de densidades

theta = volfrac;

// Exportamos a fichero la función objetivo J

ofstream ff("Jmuelle3.dat");

i=0.;

// Optimization loop

while (chg > 1e-3 && i<=Imax){

    // incrementamos iteración de optimización

    i=i+1;

    //vieja densidad == nueva (la de la iteración anterior)

    thetaold=theta;

```

```

//Filtro de densidades

solve smoothing(thetaf,thetav)=
int2d(Th)(alpha*(dx(thetaf)*dx(thetav)+dy(thetaf)*dy(thetav))+thetaf*thetav)-
int2d(Th)(theta*thetav);

thetap = (tanh(beta*eta) + tanh(beta*(thetaf-eta)))/(tanh(beta*eta) + tanh(beta*(1-
eta)));

//Forward (FEM) problem

solve
ForwardProblem([ux,uy,phi],[vx,vy,vphi])=int2d(Th)(stress(ux,uy,phi)*epsilonmatri
x(vx,vy))

-int2d(Th)(desplazamiento(ux,uy,phi)*Campoelctrico(vphi))
+int1d(Th,loaded)(spring*uy*vy)
+on(charged1,phi=1.) + on(charged2,phi=-1.) +
on(fixed,ux=0,uy=0);

//Adjoint problem

solve Adjoint([px,py,q],[vx,vy,vphi])= int1d(Th,loaded)(DObjective(ux,uy)*vy)

-int2d(Th)(stress(px,py,q)*epsilonmatrix(vx,vy))
+
int2d(Th)(desplazamiento(px,py,q)*Campoelctrico(vphi))

-int1d(Th,loaded)(spring*py*vy)
+on(charged1,q=0.) + on(charged2,q=-0.) +
on(fixed,px=0,py=0.);

//Sensitivity with respect to projected and filtered density

sens = -sensivityGauss(ux,uy,phi,px,py,q);

//Chain rule

DthetapDthetaf = beta*(1-(tanh(beta*(thetaf-eta)))^2)/(tanh(beta*eta) +
tanh(beta*(1-eta)));

```

```

solve smoothingderivative(DthetafDtheta,thetav)=
int2d(Th)(alpha*(dx(DthetafDtheta)*dx(thetav)+dy(DthetafDtheta)*dy(thetav))+Dthet
afDtheta*thetav)-int2d(Th)(thetav);

    sensn = sens*DthetapDthetaf*DthetafDtheta;

//OC update (criterio de optimalidad)

real l1 = 0; real l2= 10000; real move= 0.05;

while((l2-l1)/(l2+l1)>(1e-4)&& l2>1e-40){

real lmid=0.5*(l2+l1);

thetanew = max(0., max(theta-move,min(1.,min(theta+move,theta*(max(1e-
10,sensn/lmid))^0.3))));

if((int2d(Th)(thetanew)-int2d(Th)(volfrac))>0){

l1= lmid;}

else{

l2=lmid;}

}

theta = thetanew;

// Objective functions and visualization

real Jmuelle3 = int1d(Th,loaded)(Objective(ux,uy));

ff<<Jmuelle3<<endl;

real vol = int2d(Th)(theta)/int2d(Th)(1.);

cout <<"-----"<<endl;

cout <<"iter = "<< i << "; Objective Function = "<<Jmuelle3<<"; vol = " <<vol<<
";change = "<< chg<<".-----"<<endl;

savevtk("TO_piezo_Muelle3_" + i + ".vtk",Th,thetap,i);

}

```

10.2. Caso sin muelle

Para este caso, con una restricción de volumen de 0.5, se han realizado 70 iteraciones, ya que al no haber resistencia la distribución de densidades es mala, y los resultados son los siguientes:

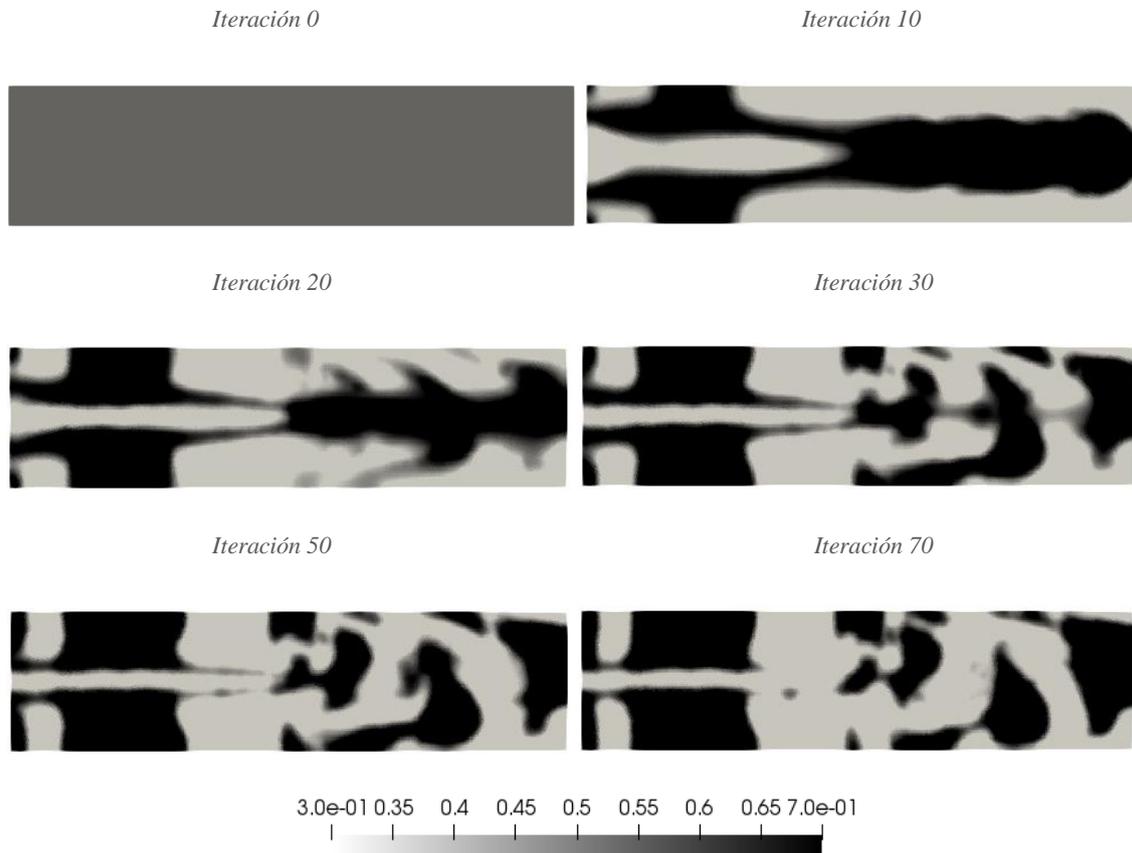


Figura 10.2. Evolución en la distribución de densidades del caso sin muelle

Se puede observar que la distribución de densidades está desconectada, lo que es debido a que no hay ninguna carga que se oponga al movimiento. Además, para el problema piezoeléctrico se ha implementado el filtro de densidades y la proyección de las mismas en lugar del filtro de sensibilidades, lo que produce que las densidades calculadas no sean tan extremas (aunque en la práctica sí, ya que se usa el método SIMP). Es por eso que se ha ajustado la escala de valores para una mejor visualización de los resultados.

En la *Figura 10.3* se observa la evolución de la función objetivo con las iteraciones.

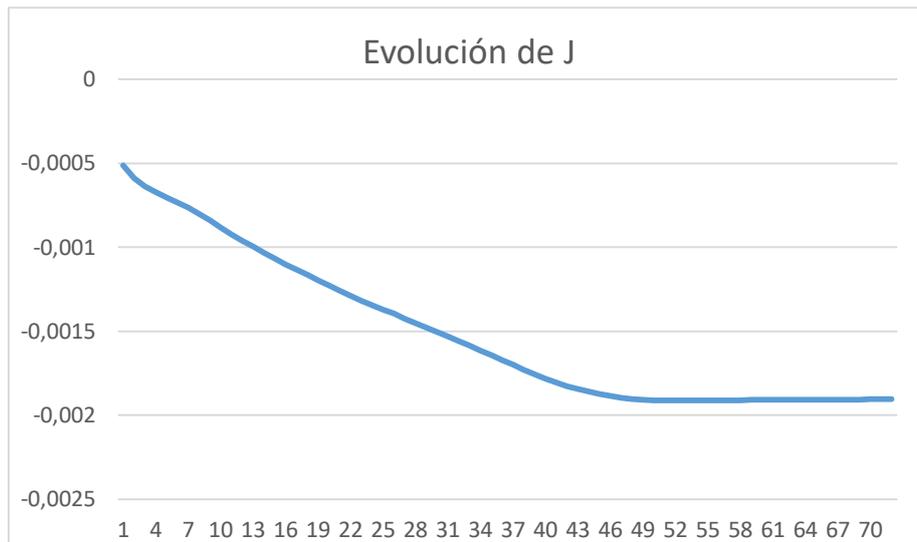
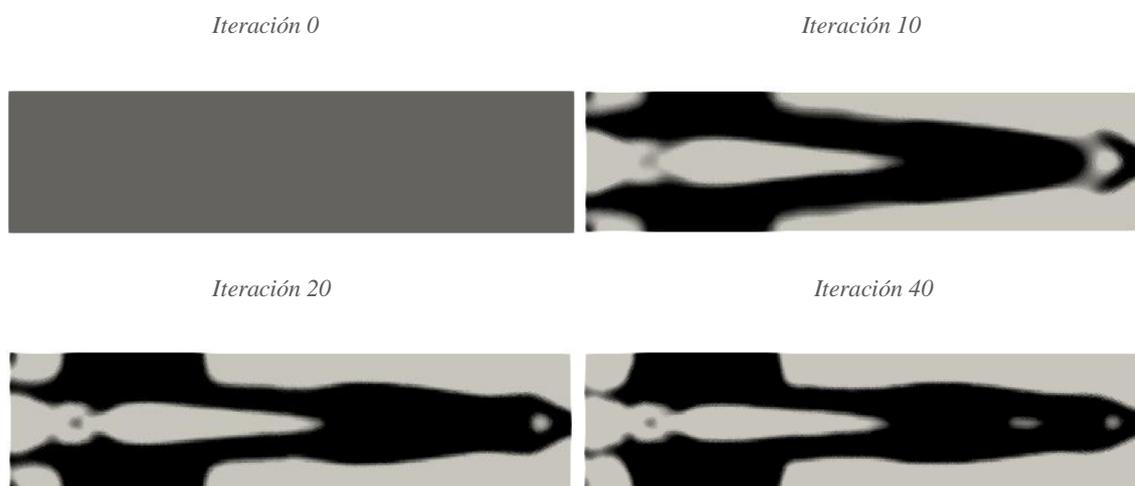


Figura 10.3. Evolución en la distribución de densidades del caso sin muelle

En cuanto a la evolución de la función objetivo, se observa cómo el programa de optimización topológica cumple su cometido, disminuyéndola desde la distribución de densidades inicial, hasta que alcanza la distribución óptima y, por tanto, los valores de J quedan constantes.

10.3. Caso muelle blando

Para este problema, así como para el resto de este capítulo, se realizan 300 iteraciones y la restricción de volumen es de 0.5. La distribución de densidades cambia muy rápido al inicio y después se estabiliza, por lo que se mostrarán saltos pequeños al inicio y grandes después.



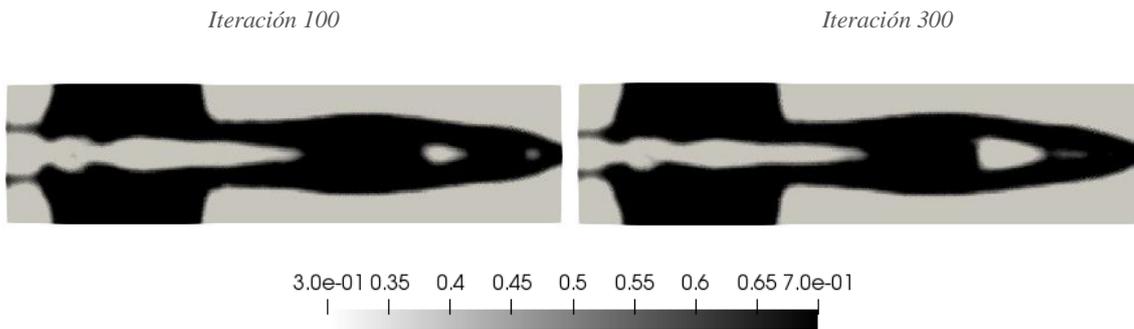


Figura 10.4. Evolución en la distribución de densidades del caso de muelle intermedio

Para el caso del muelle blando ya hay cierta consistencia en la distribución de densidades, la cual en las primeras iteraciones avanza muy rápido, y después suceden pequeños cambios en la misma. Se observa cómo rápidamente se genera la distribución de densidades fusiforme, la cual se mantiene a lo largo de todas las iteraciones. En la iteración 40 se observa que empieza a aparecer un agujero, el cual alcanza su máximo tamaño al final del bucle.

En la *Figura 10.5* se observa la evolución de la función objetivo con las iteraciones.

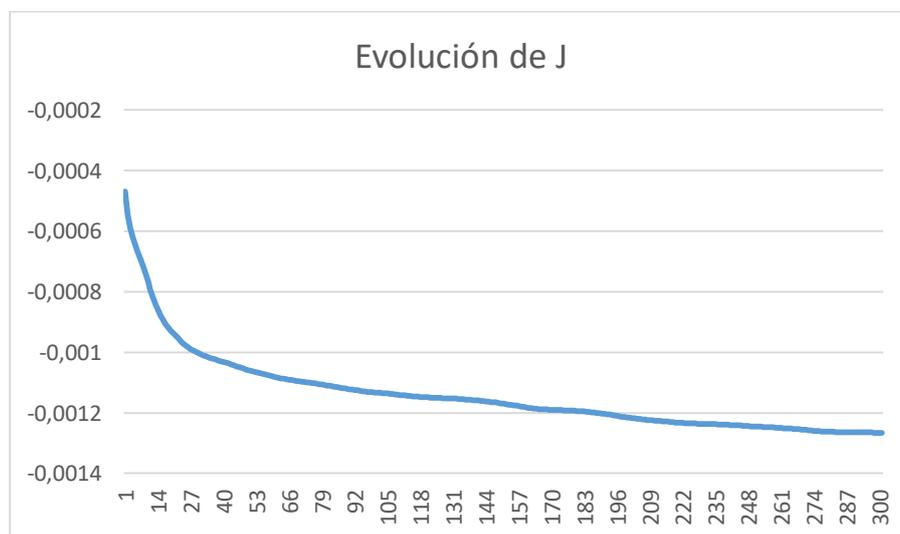


Figura 10.5. Evolución de J en el caso de muelle blando

De nuevo es en las primeras iteraciones, conforme aparece la distribución fusiforme, cuando mayor es el descenso de la función objetivo. Sin embargo, al haber una fuerza de resistencia ejercida por el muelle, en este caso la función objetivo no se estanca,

sino que sigue decreciendo lentamente hasta la última iteración mientras evoluciona la distribución de densidades.

10.4. Caso muelle intermedio

De nuevo se van a realizar 300 iteraciones, mostrándose la distribución de densidades especialmente en las primeras, ya que es donde mayores cambios ocurren.

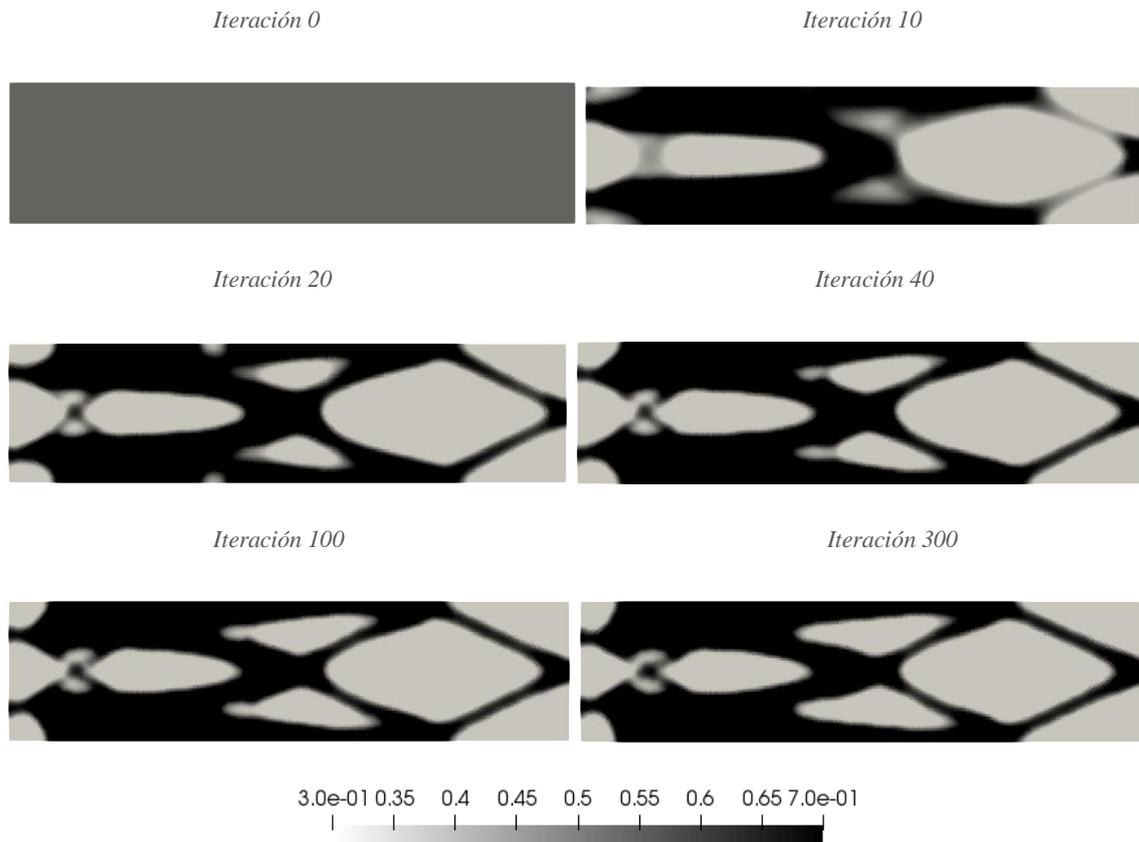


Figura 10.6. Evolución en la distribución de densidades del caso de muelle intermedio

Para este caso se forman las cruces características que veíamos en el caso mecánico, aunque la transición es mucho más rápida y algo distinta. En vez de formarse por partes la que será la distribución final, aparece desde el principio y en las siguientes iteraciones se ajusta ligeramente.

En la *Figura 10.7* se observa la evolución de la función objetivo con las iteraciones.

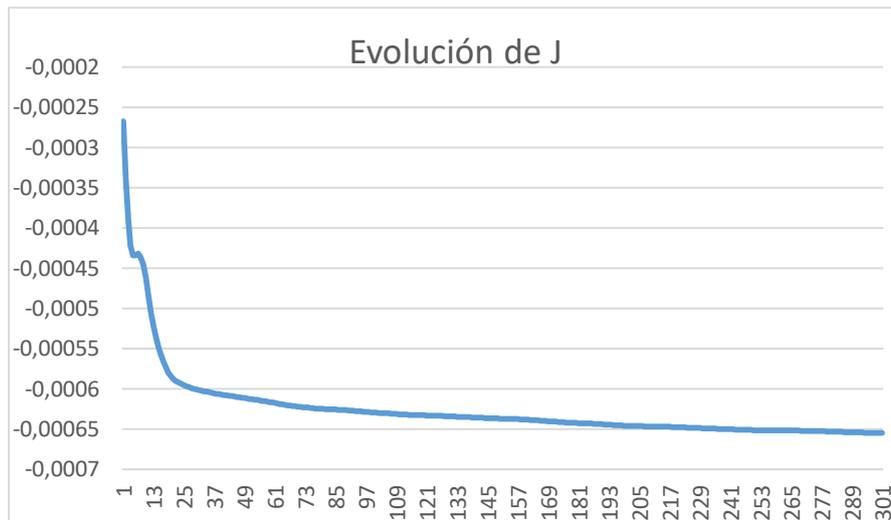
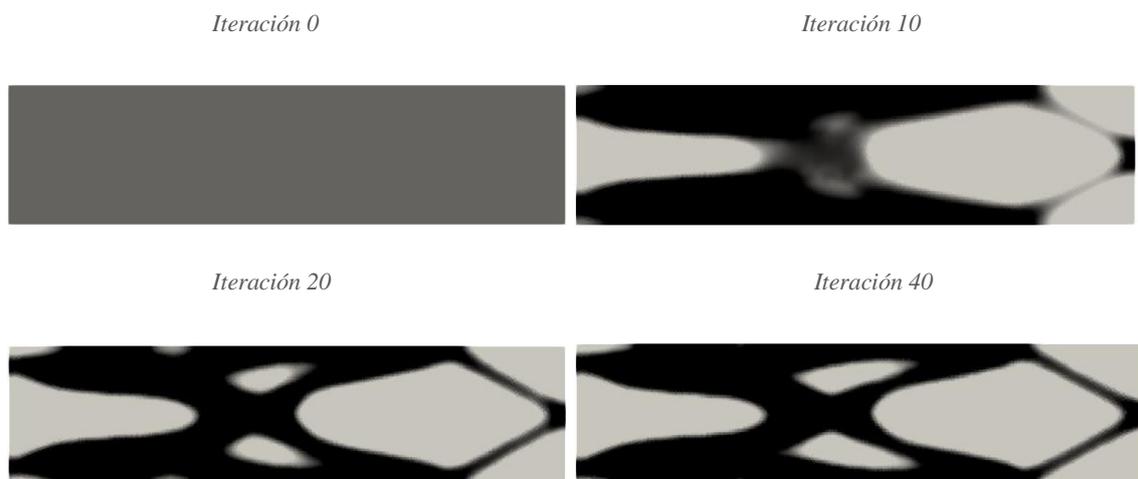


Figura 10.7. Evolución de J en el caso de muelle intermedio

En cuanto a la evolución de la función objetivo, se observa cómo hay un pequeño estancamiento al inicio, el cual es debido a los cambios en la distribución de densidades, los cuales son muy acusados en estas primeras iteraciones. Se aprecia que hasta la iteración 20, donde la distribución queda prácticamente definida, el step size es muy alto, ya que la función objetivo decrece rápidamente. A partir de esa iteración los cambios son mucho más pequeños, decreciendo J rápidamente hasta el final del bucle.

10.5. Caso muelle duro

De nuevo, para 300 iteraciones, se muestran los resultados del caso con muelle duro.



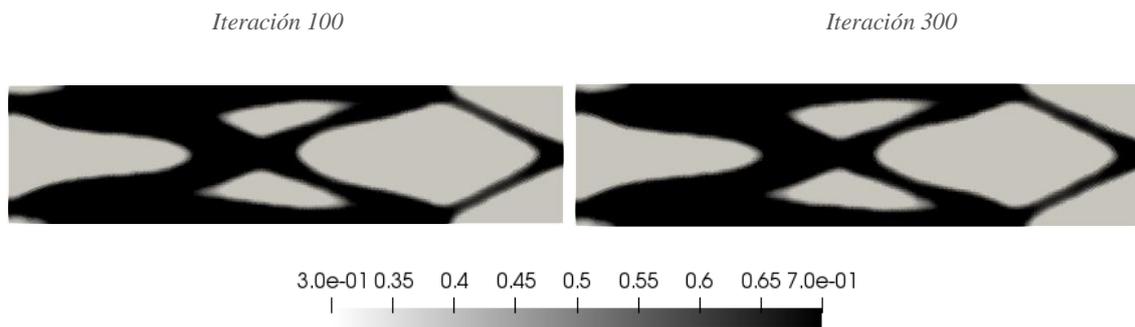


Figura 10.8. Evolución en la distribución de densidades del caso de muelle duro

La distribución en este caso es parecida a la del muelle intermedio. Se aprecia que en la iteración 10 ya está bastante avanzada, y para la iteración 20 ya está prácticamente formada. A partir de ahí, con las iteraciones suceden pequeños cambios que hacen decrecer levemente la función objetivo. De nuevo se observa el patrón típico de punta y cruces.

En la *Figura 10.9* se observa la evolución de la función objetivo con las iteraciones.

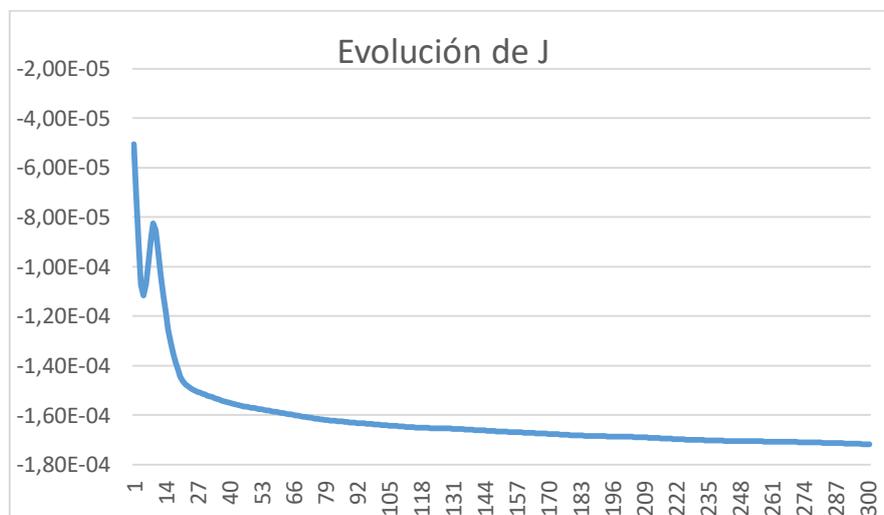


Figura 10.9. Evolución de J en el caso de muelle duro

En la evolución de la función objetivo vuelve a ser evidente la anomalía en las primeras iteraciones, propias de las primeras transiciones en la distribución de densidades, y que se hace más evidente al ser mayor la fuerza de resistencia. A parte de eso, de nuevo J decrece rápidamente en las primeras 20 iteraciones, para después evolucionar lentamente en concordancia con la distribución de densidades.

Para comprobar la coherencia con la función objetivo, en la siguiente figura se muestra como resultado la malla deformada para la distribución de densidades final (rojo) junto con la malla original (azul). Nótese que la deformación está ampliada con un factor de escalado de 20 para que los resultados sean visibles, al igual que en el resto en este capítulo.

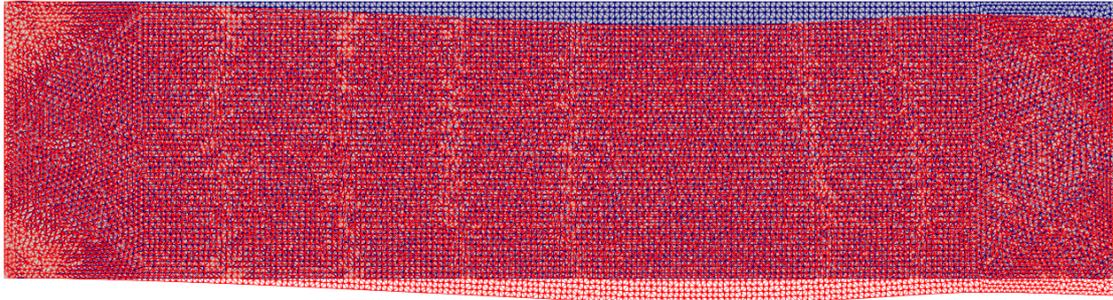


Figura 10.10. Malla deformada para el caso de muelle duro

Como se puede comprobar, se satisface un desplazamiento vertical negativo en el punto en el que se ha minimizado la función objetivo, y le acompaña el resto del espacio de diseño. Se nota también la influencia del muelle de resistencia incluido, ya que el desplazamiento es mayor a mitad del cuerpo y se mitiga en el punto objetivo. Si se retira el muelle en la iteración final, se puede representar la malla deformada para esta distribución de densidades sin muelle.

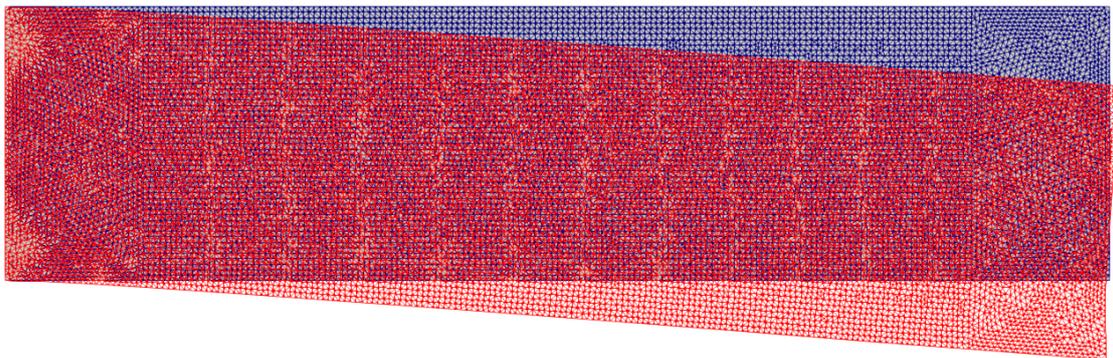


Figura 10.11. Malla deformada para el caso de muelle duro quitando el muelle

10.6. Influencia del vector de polarización piezoeléctrica

Con el objetivo de evaluar la influencia del vector de polarización o dirección preferencial N , se ha repetido el ejemplo del muelle duro rotando este vector 90° , siendo ahora $N = (0,1)$. Se han realizado 300 iteraciones, y los resultados están representados en la siguiente figura:

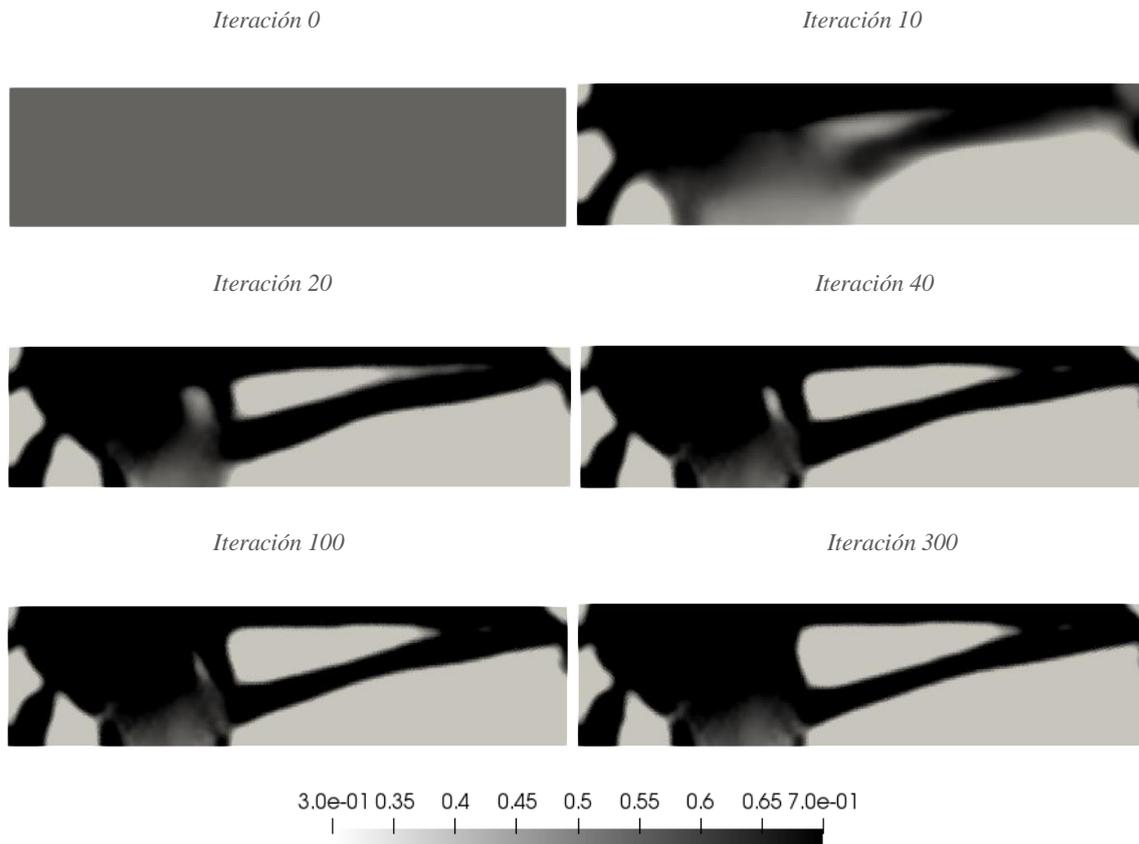


Figura 10.12. Evolución en la distribución de densidades del caso de muelle duro y $N = (0,1)$

Como se aprecia en la figura, la distribución de densidades para este caso no tiene nada que ver con la anterior, ya que el comportamiento del material cambia drásticamente para la misma geometría al modificar el vector de polarización. Además, la distribución resultante tiene densidades intermedias. Para solventar este problema en un modelo multifísico se puede variar el exponente del método SIMP para una de las ecuaciones de comportamiento, en este caso la eléctrica. Sin embargo, esta vez no es necesario ya que el vector de polarización $N = (1,0)$ da mejores resultados en la función objetivo, por lo que este diseño se descartaría.

Como ha pasado en el resto de casos, la evolución en la distribución de densidades es muy rápida al principio, creándose el patrón principal en las primeras iteraciones, y siendo los cambios mucho más ligeros a partir de la iteración 20. En la siguiente figura se muestra la evolución de la función objetivo.

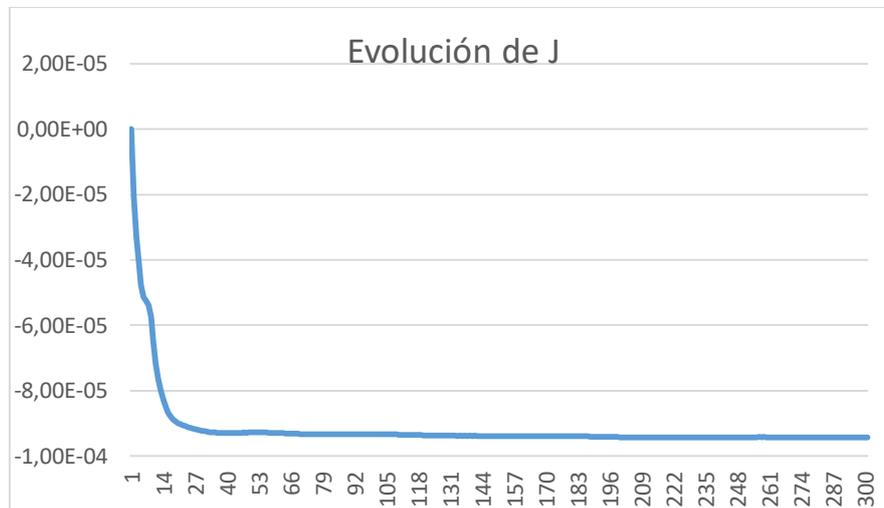


Figura 10.13. Evolución de J en el caso de muelle duro y $N = (0,1)$

Como se ha comentado anteriormente, y acorde con la evolución de la distribución de densidades, es en las primeras iteraciones cuando la función objetivo baja drásticamente, para después estancarse. Además, se puede observar cómo se obtienen mejores valores para el otro vector de polarización (*Figura 10.9*), ya que el objetivo es obtener el valor más negativo posible, para así maximizar el desplazamiento vertical en sentido negativo.

En conclusión, el vector de polarización piezoeléctrica o dirección preferencial es una herramienta más a tener en cuenta para que, desde el punto de vista del diseño, se obtengan los resultados óptimos.

10.7. Problema de compresión

Para este problema, el objetivo es minimizar los desplazamientos horizontales positivos, o lo que es lo mismo, maximizar la compresión, en el punto **P** de la *Figura 10.1*. Para ello, simplemente hay que modificar la función objetivo en el código, así como el muelle, que esta vez se aplicará de forma horizontal (muelle duro), oponiéndose al desplazamiento. Se ha tomado un vector de polarización $\mathbf{N} = (1,0)$ y realizado 300 iteraciones del bucle de optimización, obteniéndose los siguientes resultados:

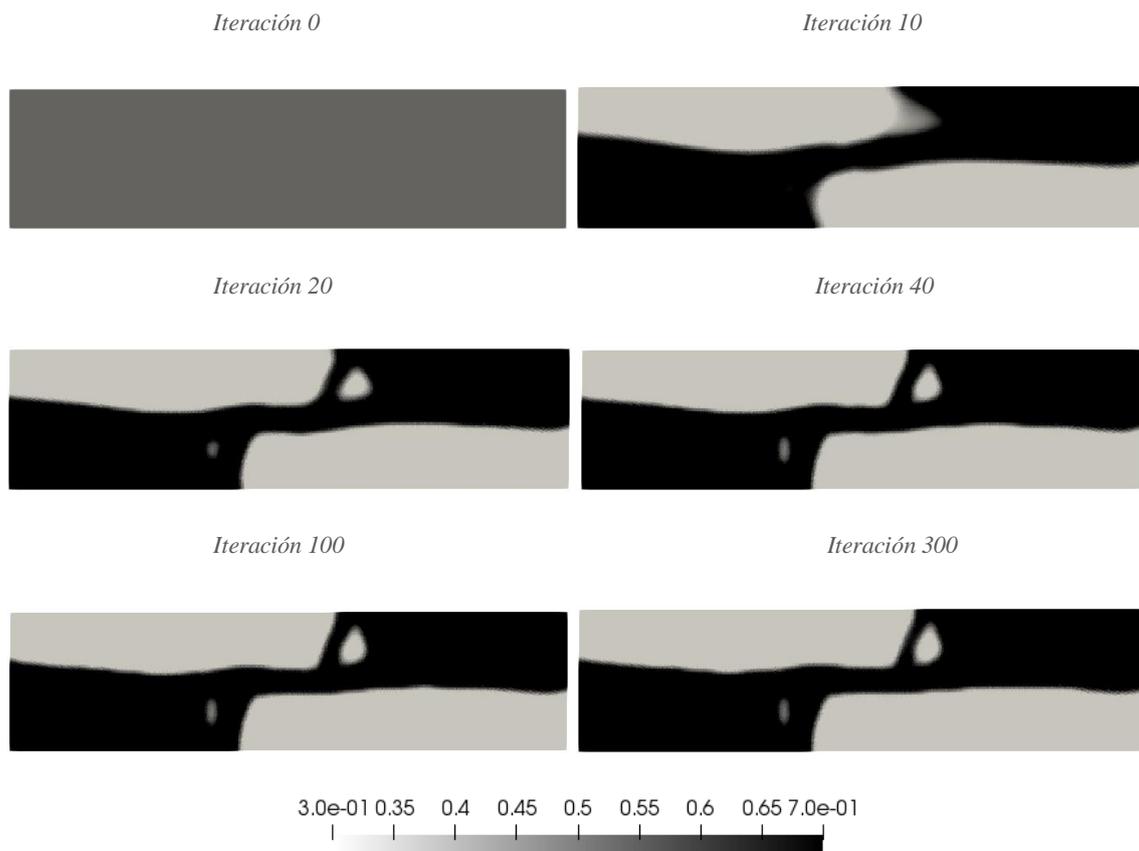


Figura 10.14. Evolución en la distribución de densidades del problema de compresión

En este caso, la distribución de densidades es prácticamente óptima a las pocas iteraciones, siendo ya visible en la iteración 10 y abriéndose los agujeros antes de la 20. Desde esa iteración los cambios son sutiles hasta el final de bucle. En la siguiente figura se aprecia la evolución de la función objetivo:

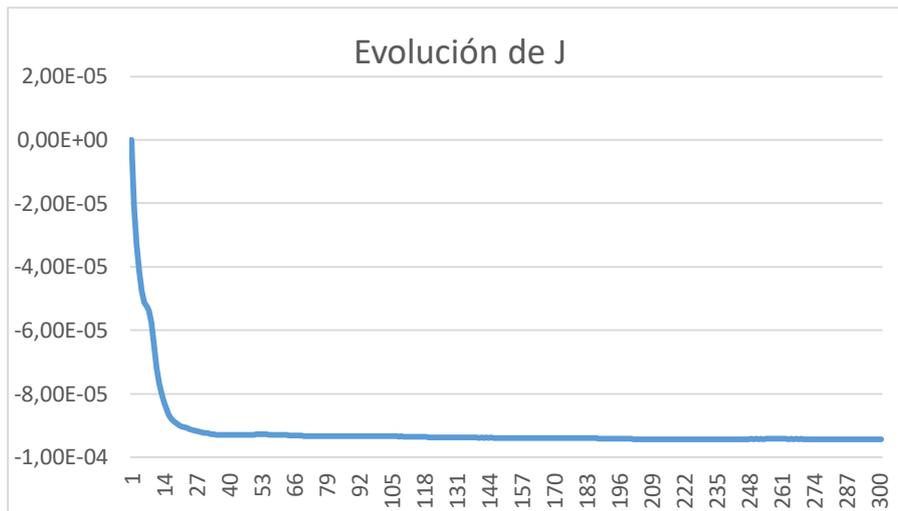


Figura 10.15. Evolución de J en el problema de compresión

De nuevo se aprecia en la figura que los cambios más importantes suceden en las primeras iteraciones, acorde a la evolución en las densidades. A partir de ahí la función objetivo se estabiliza y decrece muy suavemente.

Como en el capítulo 10.5, se va a comprobar la compresión mediante la representación de la malla deformada para la distribución óptima de densidades, junto con la malla sin deformar.

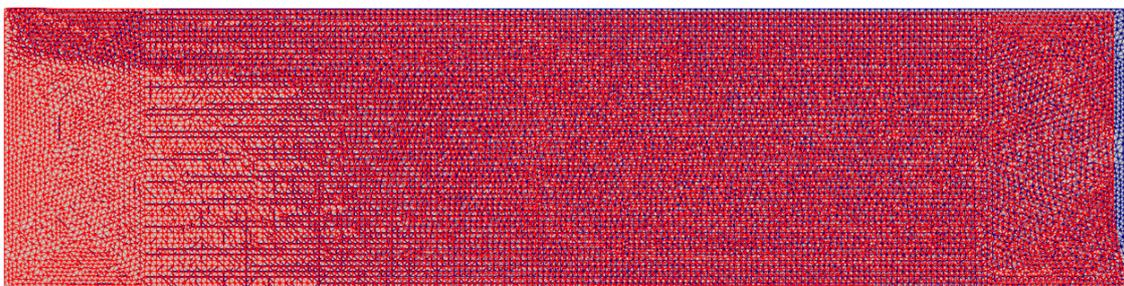


Figura 10.16. Malla deformada para problema de compresión

Se aprecia claramente la compresión en el punto objetivo, mientras que en el empotramiento casi no hay deformación, y por eso se solapan ambas mallas. Como apunte, también se observa que la compresión es más acusada donde se encuentra el material en la distribución de densidades, quedando la esquina inferior derecha casi sin deformar.

10.8. Problema de tracción

De forma análoga al problema de compresión, esta vez se va a buscar minimizar los desplazamientos horizontales negativos, es decir, maximizar la tracción, en el punto **P** de la *Figura 10.1*. Cambiando el signo de la función objetivo del problema de compresión, y realizando de nuevo 300 iteraciones, se obtiene:

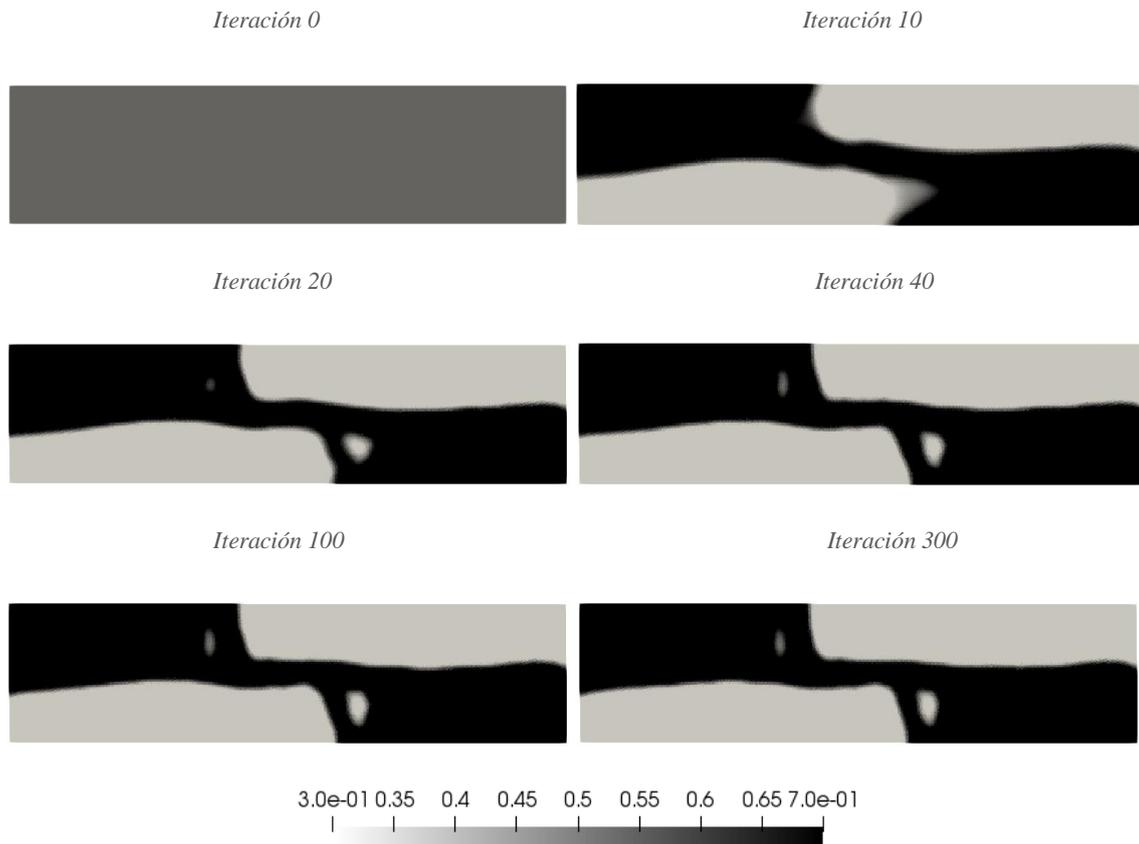


Figura 10.17. Evolución en la distribución de densidades del problema de tracción

Se puede observar cómo la distribución de densidades formada es la opuesta del problema de compresión, lo cual cobra sentido teniendo en cuenta las condiciones de contorno concretas, y que estamos maximizando desplazamientos en direcciones opuestas. En cuanto a la evolución de la distribución, así como la de la función objetivo, se comportan de la misma forma, tal y como se aprecia en la siguiente figura:

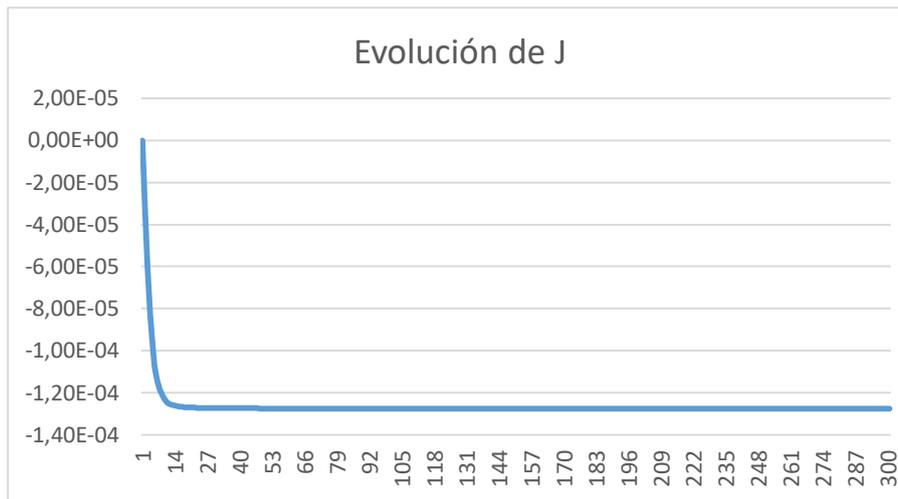


Figura 10.18. Evolución de J en el problema de tracción

Nuevamente se ve que la función objetivo decrece rápidamente en las primeras iteraciones para estabilizarse y descender suavemente después.

De nuevo, se muestra la malla deformada con la distribución de densidades óptima junto a la malla sin deformar.

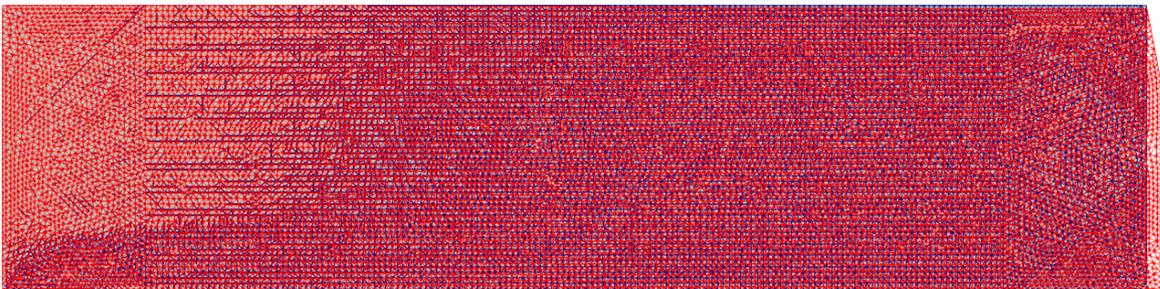


Figura 10.19. Malla deformada para el problema de tracción

En este problema sucede algo análogo al de compresión. La tracción se produce sobre todo cerca de la cara en la que se encuentra el punto objetivo, siendo menor cerca del empotramiento, donde se solapan las mallas. De la misma forma, la mayor tracción se produce en la zona donde hay material en la distribución de densidades, mientras que donde hay vacío, esquina superior derecha, no se produce apenas.

CAPÍTULO 11: CONCLUSIONES

Durante la realización de este trabajo se han desarrollado los conocimientos requeridos para la formulación y posterior resolución de un problema multifísico, en este caso el de condiciones de contorno para materiales piezoeléctricos. Para ello se han introducido los conceptos básicos de mecánica de continuo, así como el problema de elasticidad lineal para después plantear el electro-mecánico como extensión del mismo. Para la implementación de estos conceptos teóricos, se ha aprendido el funcionamiento del método de elementos finitos, la formulación variacional del problema fuerte de la piezoelectricidad y la discretización espacial de la misma, expresando las incógnitas como el producto de una matriz de rigidez y un término independiente, comprendiendo así el funcionamiento del software FreeFEM. Además, se ha introducido la discretización temporal para problemas dinámicos sin disipación.

Aplicando todos los conocimientos adquiridos y los procedimientos descritos, se han resuelto problemas de condiciones de contorno en piezoeléctricos, emulando las aplicaciones reales de sensores y actuadores en diferentes geometrías bidimensionales y tridimensionales y mostrando resultados coherentes e interesantes desde el punto de vista ingenieril, como el gripper piezoeléctrico mostrado en la *Figura 11.1*. También se ha implementado un problema dinámico y mostrado sus resultados en los puntos críticos del ciclo, así como la conservación de energías al no ser disipativo.

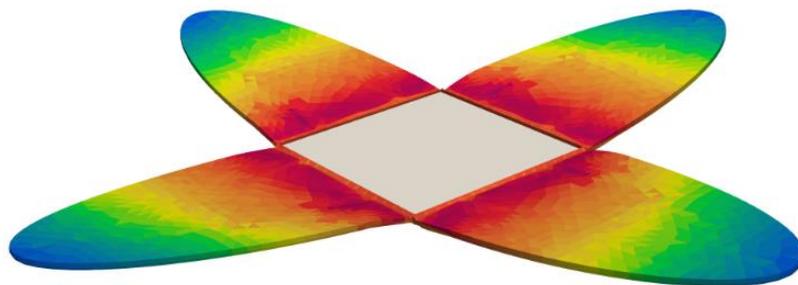


Figura 11.1. Gripper piezoeléctrico

En la segunda parte del trabajo se ha entendido lo que es la optimización topológica, sus aplicaciones y su importancia en la ingeniería para obtener diseños eficientes, y se han adquirido todos los conocimientos necesarios para su implementación: desde la formulación primigenia de una distribución discreta de material, hasta la formulación continua y la aplicación del método SIMP, junto con la proyección y filtrado de densidades (o la alternativa de filtrado de sensibilidades), y el cálculo de los problemas

forward y adjunto y la dirección de descenso para el problema mecánico y electro-mecánico.

Posteriormente, y aplicando todos los conceptos estudiados, se ha implementado el problema de optimización topológica en el contexto mecánico en FreeFEM, resolviendo una serie de benchmarks sencillos con diferentes condiciones de contorno mediante el método de filtrado de sensibilidades. El problema consiste en minimizar el desplazamiento producido por una fuerza puntual, y se obtienen resultados coherentes y concordantes con la literatura existente. En la *Figura 11.2* se muestran las distribuciones de densidades óptimas para dos de los benchmarks estudiados.

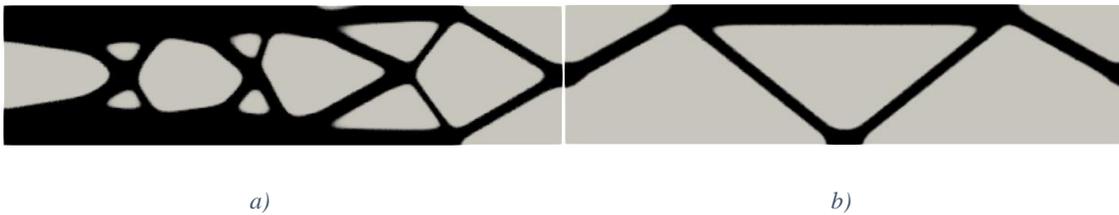


Figura 11.2. Distribución óptima de densidades para distintas solicitaciones mecánicas

Finalmente, enlazando los conocimientos adquiridos en la primera y segunda parte del trabajo, se ha implementado el problema de optimización topológica para materiales piezoeléctricos, combinando condiciones de contorno mecánicas y eléctricas y esta vez usando el filtro y proyección de densidades. El problema consiste en maximizar el desplazamiento en un punto de un actuador, el cual está sujeto a un muelle que se opone a su movimiento. Se ha repetido el problema para distintas durezas del muelle y cambiando la función objetivo (tracción y compresión), y se ha evaluado la influencia del vector de polarización, el cual será un factor determinante para el diseño. En la *Figura 11.3* se muestra la distribución de densidades óptima para dos de los casos estudiados.

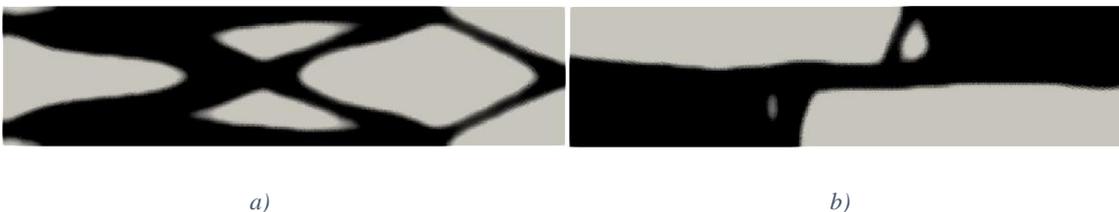


Figura 11.3. Distribución óptima de densidades para distintas solicitaciones electro-mecánicas

En conclusión, los objetivos propuestos para la realización de este trabajo se han cumplido satisfactoriamente. Como se ha podido observar, la extensión y profundidad conceptual de la optimización topológica es notable. Debido a esto, a partir de este trabajo se puede continuar la línea de investigación para mejorar y afinar los programas de

optimización topológica desarrollados para software libre, consiguiendo una implementación más eficiente explorando las capacidades de FreeFEM de implementación en paralelo. De esta forma, y disponiendo de la capacidad computacional necesaria, se podrán resolver problemas de geometrías tridimensionales mucho más complejas tanto para el problema mecánico como para el piezoeléctrico, llegando a desarrollar más aplicaciones reales en ingeniería. Otra futura extensión de este trabajo sería la del diseño de metamateriales piezoeléctricos y mecánicos, como los que presentan coeficiente de Poisson negativo.

BIBLIOGRAFÍA

- [1] Ortigosa, R., Martínez-Frutos, J., Gil, A.J. et al. A new stabilisation approach for level-set based topology optimisation of hyperelastic materials. *Struct Multidisc Optim* 60, 2343–2371 (2019).
- [2] R. Ortigosa, D. Ruiz, A.J. Gil, A. Donoso, J.C. Bellido. A stabilisation approach for topology optimisation of hyperelastic structures with the SIMP method, *Computer Methods in Applied Mechanics and Engineering*, Volume 364, 2020
- [3] R. Ortigosa, J. Martínez-Frutos, D. Ruiz, A. Donoso, J. C. Bellido. Density-based topology optimization considering nonlinear electromechanics, *Structural and Multidisciplinary Optimization*, Volume 64, 657-680,2021.
- [4] Allaire G, Jouve F, Toader. A Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics* 194(1):363–393 (2004)
- [5] G. Allaire. *Conception optimale de structures*, Springer, 2007.
- [6] G. Allaire. *Numerical Analysis and Optimization: An introduction to Mathematical Modelling and Numerical Simulation (Numerical Mathematics and Scientific Computation)*, Oxford Science Publications, 2007.
- [7] R. Font, F. Periago. The Finite Element Method with FreeFem++ for beginners. *The Electronic Journal of Mathematics and Technology*, Volume 7, Number 4

ANEXOS

Anexo 1: Sensor piezoeléctrico bidimensional

```
load"iovtk" //PARA IMPORTAR DATOS EN PARAVIEW

//PREPROCESO: GEOMETRIA, MALLA Y PARAMETROS FISICOS

border a1(t=0,10){x=t;y=0;label=1;}; // arista de abajo
border a2(t=0,10){x=10;y=t;label=2;}; // arista de la derecha
border a3(t=0,10){x=10-t;y=10;label=3;}; // arista de arriba
border a4(t=0,10){x=0;y=10-t;label=4;}; // arista de la izquierda
border interior(t=0,2*pi){x=5+2*cos(t);y=5+2*sin(t);}; // frontera interior

mesh Th=buildmesh(a1(30)+a2(30)+a3(30)+a4(30)+interior(40)); //malla

// PARAMETROS DEL MODELO
real f=0; //fuerza por unidad de volumen
real ro=0; //carga eléctrica por unidad de volumen
real epsilon= 4*8.8e-12; //Permitividad eléctrica

fespace Ph(Th,P0); //Espacio de elementos de Lagrange P0 para definir la región
Ph reg=region;

int ninterior=reg(5,5); //región interior del círculo
int nexterior=reg(9,9); //región exterior del círculo
cout<<"ninterior = "<<ninterior<<"nexterior= "<<nexterior<<endl;

Ph lambda=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de lambda según
la región
Ph mu=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de mu según la
región
Ph alpha1=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha1 según la
región
```

Ph alpha2=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha2 según la región

//PROCESO: CALCULO DE LA SOLUCION

fespace Vh(Th,P1); //elementos de Lagrange P1

Vh u1,u2,phi,v1,v2,alpha,sigmaVm;

//MACROS

macro IdentityMatrix [1.0, 0.0, 0.0, 1.0] //EOM

macro Trace(a) (a[0] + a[3]) //EOM

macro epsilonmatrix(u1,u2) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dy(u1) + dx(u2)), dy(u2)] //EOM

macro Campoelectrico(phi) [-dx(phi),-dy(phi)]//EOM

macro N [1,0]//EOM

macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[1]*B[0], A[1]*B[1]] //EOM

macro mult (A,b)[A[0]*b[0]+A[1]*b[1],A[2]*b[0]+A[3]*b[1]]//EOM

macro stress (u1,u2,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2)) +2.*mu*epsilonmatrix(u1,u2) - alpha1*(Campoelectrico(phi)*N)*IdentityMatrix - alpha2*(DyadicProduct(Campoelectrico(phi),N) + DyadicProduct(N,Campoelectrico(phi))))//EOM

macro desplazamiento(u1,u2,phi)(epsilon*Campoelectrico(phi) + alpha1*Trace(epsilonmatrix(u1,u2))*N + alpha2*mult(epsilonmatrix(u1,u2),N))//EOM

solve

Problem([u1,u2,phi],[v1,v2,alpha])=int2d(Th)(stress(u1,u2,phi)*epsilonmatrix(v1,v2))
-int2d(Th)(desplazamiento(u1,u2,phi)*Campoelectrico(alpha))
+on(1,u1=0.0,u2=0.0,phi=0.0)
+on(3,u2=1.0);

//POSTPROCESO: ANALISIS DE LOS RESULTADOS

```
sigmaVm =  
sqrt(stress(u1,u2,phi)[0]*stress(u1,u2,phi)[0]+stress(u1,u2,phi)[3]*stress(u1,u2,phi)[3]  
-stress(u1,u2,phi)[0]*stress(u1,u2,phi)[3]+3*stress(u1,u2,phi)[1]*stress(u1,u2,phi)[1]);
```

```
real coef=5; //”lupa” para ver el desplazamiento
```

```
mesh th3=movemesh(Th,[x+u1*coef,y+u2*coef]); //malla deformada
```

```
savevtk("Piezo1Caso1.vtk",Th,u1,u2,phi,sigmaVm,[-dx(phi),-dy(phi)]);
```

```
savevtk("Piezo1deformadoCaso1.vtk",th3,u1,u2,phi,sigmaVm,[-dx(phi),-dy(phi)]);
```

Anexo 2: Sensor piezoeléctrico tridimensional

```
load "msh3" //Para la malla
load "TetGen" //Para la malla
load"iovtk" //para importar datos en Paraview
include "MeshSurface.idp"

//GENERACIÓN DE LA MALLA

real hs = 0.15;
int[int] N = [15, 15, 15];
real [int,int] B = [[-1, 1], [-1, 1], [-1, 1]];
int [int,int] L = [[1, 2], [3, 4], [5, 6]]; //labels derecha, izquierda, front, back, abajo, arriba
mesh3 ThH = SurfaceHex(N, B, L, 1);
mesh3 ThS = Sphere(0.5, hs, 7, 1); //label 7
mesh3 ThHS = ThH + ThS;
real voltet = (hs^3)/6.;
cout << "voltet = " << voltet << endl;
real[int] domain = [0, 0, 0, 1, voltet, 0, 0, 0.7, 2, voltet];
mesh3 Th = tetg(ThHS, switch="pqaAAYYQ", nbofregions=2, regionlist=domain);

//ASIGNACIÓN DE PARÁMETROS

real f=0; //fuerza por unidad de volumen
real ro=0; //carga eléctrica por unidad de volumen
real epsilon= 4*8.8e-12; //Permitividad eléctrica

fespace Ph(Th,P0); //Espacio de elementos de Lagrange P0 para definir la región
Ph reg=region;
plot(reg);

int ninterior=reg(0,0,0); //región interior de la esfera
```

```

int nexterior=reg(0.9,0.9,0.9); //región exterior de la esfera

Ph lambda=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de lambda según
la región

Ph mu=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de mu según la
región

Ph alpha1=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha1 según la
región

Ph alpha2=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha2 según la
región

//PROCESO: CALCULO DE LA SOLUCION
fespace Vh(Th,P1); //elementos de Lagrange P1
Vh u1,u2,u3,phi,v1,v2,v3,alpha,sigmaVm;

//MACROS
macro IdentityMatrix [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0] //EOM
macro Trace(a) (a[0] + a[4] + a[8]) //EOM
macro epsilonmatrix(u1,u2,u3) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dz(u1) + dx(u3)),
0.5*(dy(u1) + dx(u2)), dy(u2),
0.5*(dz(u2) + dy(u3)), 0.5*(dz(u1) + dx(u3)),0.5*(dz(u2) + dy(u3)), dz(u3)] //EOM
macro Campoelectrico(phi) [-dx(phi),-dy(phi),-dz(phi)]//EOM
macro N [0,0,1]//EOM
macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[0]*B[2],
A[1]*B[0],A[1]*B[1],A[1]*B[2],A[2]*B[0],A[2]*B[1],A[2]*B[2]] //EOM
macro mult (A,b)[A[0]*b[0]+A[1]*b[1]+A[2]*b[2], A[3]*b[0]+A[4]*b[1]+A[5]*b[2],
A[6]*b[0]+A[7]*b[1]+A[8]*b[2]]//EOM
macro stress (u1,u2,u3,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2,u3))
+2.*mu*epsilonmatrix(u1,u2,u3) - alpha1*(Campoelectrico(phi)*N)*IdentityMatrix -
alpha2*(DyadicProduct(Campoelectrico(phi),N) +
DyadicProduct(N,Campoelectrico(phi))))//EOM
macro desplazamiento(u1,u2,u3,phi)(epsilon*Campoelectrico(phi) +
alpha1*Trace(epsilonmatrix(u1,u2,u3))*N + alpha2*mult(epsilonmatrix(u1,u2,u3),N)
)//EOM

```

```

solve
Problem([u1,u2,u3,phi],[v1,v2,v3,alpha])=int3d(Th)(stress(u1,u2,u3,phi)*epsilonmatrix
(v1,v2,v3))

-int3d(Th)(desplazamiento(u1,u2,u3,phi)*Campoelectrico(alpha))
+on(5,u1=0.0,u2=0.0,u3=0.0,phi=0.0)
+on(6,u3=0.2);

```

//POSTPROCESO: ANALISIS DE LOS RESULTADOS

```

real coef=5; //lupa, para ver el desplazamiento (el real es tan peque que ni se nota)
mesh3 Thm=movemesh3(Th,transfo=[x+coef*u1,y+coef*u2,z+coef*u3]); //malla
deformada

```

```

sigmaVm = sqrt(stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[0]
+stress(u1,u2,u3,phi)[4]*stress(u1,u2,u3,phi)[4]
+stress(u1,u2,u3,phi)[8]*stress(u1,u2,u3,phi)[8]
-stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[4]
-stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[8]
-stress(u1,u2,u3,phi)[4]*stress(u1,u2,u3,phi)[8]
+3*(stress(u1,u2,u3,phi)[1]*stress(u1,u2,u3,phi)[1]
+stress(u1,u2,u3,phi)[2]*stress(u1,u2,u3,phi)[2]
+stress(u1,u2,u3,phi)[5]*stress(u1,u2,u3,phi)[5]));

```

```

savevtk("Piezo13Dcaso1.vtk",Th,u1,u2,u3,phi,sigmaVm,[-dx(phi),-dy(phi),-dz(phi)]);
savevtk("Piezo13Ddeformadocaso1.vtk",Thm,u1,u2,u3,phi,sigmaVm,[-dx(phi),-
dy(phi),-dz(phi)]);

```

Anexo 2.1: MeshSurface

```
load "msh3"
load "medit"
func mesh3 SurfaceHex(int[int] & N,real[int,int] &B ,int[int,int] & L,int orientation)
{
    real x0=B(0,0),x1=B(0,1);
    real y0=B(1,0),y1=B(1,1);
    real z0=B(2,0),z1=B(2,1);

    int nx=N[0],ny=N[1],nz=N[2];

    mesh Thx = square(ny,nz,[y0+(y1-y0)*x,z0+(z1-z0)*y]);
    mesh Thy = square(nx,nz,[x0+(x1-x0)*x,z0+(z1-z0)*y]);
    mesh Thz = square(nx,ny,[x0+(x1-x0)*x,y0+(y1-y0)*y]);

    int[int] refx=[0,L(0,0)],refX=[0,L(0,1)]; // Xmin, Ymax faces labels renumbering
    int[int] refy=[0,L(1,0)],refY=[0,L(1,1)]; // Ymin, Ymax faces labels renumbering
    int[int] refz=[0,L(2,0)],refZ=[0,L(2,1)]; // Zmin, Zmax faces labels renumbering

    mesh3 Thx0 = movemesh23(Thx,transfo=[x0,x,y],orientation=-orientation,label=refx);
    mesh3 Thx1 =
    movemesh23(Thx,transfo=[x1,x,y],orientation=+orientation,label=refX);
    mesh3 Thy0 =
    movemesh23(Thy,transfo=[x,y0,y],orientation=+orientation,label=refy);
    mesh3 Thy1 = movemesh23(Thy,transfo=[x,y1,y],orientation=-orientation,label=refY);
    mesh3 Thz0 = movemesh23(Thz,transfo=[x,y,z0],orientation=-orientation,label=refz);
    mesh3 Thz1 =
    movemesh23(Thz,transfo=[x,y,z1],orientation=+orientation,label=refZ);
    mesh3 Th= Thx0+Thx1+Thy0+Thy1+Thz0+Thz1;
```

```

    return Th;
}

func mesh3 Sphere(real R,real h,int L,int orientation)
{
    mesh Th=square(10,20,[x*pi-pi/2,2*y*pi]); //  $\frac{-\pi}{2}, \frac{-\pi}{2}$ 
    // a parametrization of a sphere
    func f1 =cos(x)*cos(y);
    func f2 =cos(x)*sin(y);
    func f3 = sin(x);
    // partiel derivative
    func f1x=sin(x)*cos(y);
    func f1y=-cos(x)*sin(y);
    func f2x=-sin(x)*sin(y);
    func f2y=cos(x)*cos(y);
    func f3x=cos(x);
    func f3y=0;
    // the metric on the sphere  $M = DF^t DF$ 
    func m11=f1x^2+f2x^2+f3x^2;
    func m21=f1x*f1y+f2x*f2y+f3x*f3y;
    func m22=f1y^2+f2y^2+f3y^2;

    func perio=[[4,y],[2,y],[1,x],[3,x]]; // to store the periodic condition

    real hh=h/R;// hh mesh size on unite sphere
    real vv= 1/square(hh);
    Th=adaptmesh(Th,m11*vv,m21*vv,m22*vv,IsMetric=1,periodic=perio);
    Th=adaptmesh(Th,m11*vv,m21*vv,m22*vv,IsMetric=1,periodic=perio);
    Th=adaptmesh(Th,m11*vv,m21*vv,m22*vv,IsMetric=1,periodic=perio);
}

```

```
Th=adaptmesh(Th,m1 1*vv,m21*vv,m22*vv,IsMetric=1,periodic=perio);
int[int] ref=[0,L];

mesh3 ThS=
movemesh23(Th,transfo=[f1*R,f2*R,f3*R],orientation=orientation,refface=ref);
return ThS;
}
```

Anexo 3: Actuador piezoeléctrico tridimensional

```
load"gmsh";
load"msh3";
load"iovtk" //para importar datos en Paraview

//PREPROCESO: GEOMETRIA, MALLA Y PARAMETROS FISICOS

mesh3 Th =gmshload3("untitled2muyafinada.msh2");

// PARAMETROS DEL MODELO

real lambda=1e9; //coeficiente de Lamé
real mu=1e9; //coeficiente de Lamé
real f=0; //fuerza por unidad de volumen
real ro=0; //carga eléctrica por unidad de volumen
real epsilon= 4*8.8e-12; //Permitividad eléctrica
real alpha1= 0.1; // coeficiente piezoeléctrico
real alpha2= 0.1; //coeficiente piezoeléctrico

//PROCESO: CALCULO DE LA SOLUCION

fespace Vh(Th,P1); //elementos de Lagrange P1
Vh u1,u2,u3,phi,v1,v2,v3,alpha,sigmaVm;

//MACROS
macro IdentityMatrix [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0] //EOM
macro Trace(a) (a[0] + a[4] + a[8]) //EOM
macro epsilonmatrix(u1,u2,u3) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dz(u1) + dx(u3)),
0.5*(dy(u1) + dx(u2)), dy(u2),
0.5*(dz(u2) + dy(u3)), 0.5*(dz(u1) + dx(u3)),0.5*(dz(u2) + dy(u3)), dz(u3)] //EOM
```

```

macro Campoelectrico(phi) [-dx(phi),-dy(phi),-dz(phi)]//EOM
macro N [0,0,1]//EOM
macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[0]*B[2],
A[1]*B[0],A[1]*B[1],A[1]*B[2],A[2]*B[0],A[2]*B[1],A[2]*B[2]] //EOM
macro mult (A,b)[A[0]*b[0]+A[1]*b[1]+A[2]*b[2], A[3]*b[0]+A[4]*b[1]+A[5]*b[2],
A[6]*b[0]+A[7]*b[1]+A[8]*b[2]]//EOM
macro stress (u1,u2,u3,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2,u3))
+2.*mu*epsilonmatrix(u1,u2,u3) - alpha1*(Campoelectrico(phi)*N)*IdentityMatrix -
alpha2*(DyadicProduct(Campoelectrico(phi),N) +
DyadicProduct(N,Campoelectrico(phi))))//EOM
macro desplazamiento(u1,u2,u3,phi)(epsilon*Campoelectrico(phi) +
alpha1*Trace(epsilonmatrix(u1,u2,u3))*N + alpha2*mult(epsilonmatrix(u1,u2,u3),N)
)//EOM

```

```

solve
Problem([u1,u2,u3,phi],[v1,v2,v3,alpha])=int3d(Th)(stress(u1,u2,u3,phi)*epsilonmatrix
(v1,v2,v3))
-int3d(Th)(desplazamiento(u1,u2,u3,phi)*Campoelectrico(alpha))
+on(2,u1=0.0,u2=0.0,u3=0.0)
+on(3,phi=10000000)
+on(4,phi=0);

```

real coef=100; //lupa, para ver el desplazamiento

```

mesh3 Thm=movemesh3(Th,transfo=[x+coef*u1,y+coef*u2,z+coef*u3]); //malla
deformada

```

//POSTPROCESO: ANÁLISIS DE LOS RESULTADOS

```

sigmaVm = sqrt(stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[0]
+stress(u1,u2,u3,phi)[4]*stress(u1,u2,u3,phi)[4]
+stress(u1,u2,u3,phi)[8]*stress(u1,u2,u3,phi)[8]
-stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[4]

```

```
-stress(u1,u2,u3,phi)[0]*stress(u1,u2,u3,phi)[8]
-stress(u1,u2,u3,phi)[4]*stress(u1,u2,u3,phi)[8]
+3*(stress(u1,u2,u3,phi)[1]*stress(u1,u2,u3,phi)[1]
+stress(u1,u2,u3,phi)[2]*stress(u1,u2,u3,phi)[2]
+stress(u1,u2,u3,phi)[5]*stress(u1,u2,u3,phi)[5]));
```

```
savevtk("Piezo2.vtk",Th,u1,u2,u3,phi,sigmaVm,[-dx(phi),-dy(phi),-dz(phi)]);
```

```
savevtk("Piezo2deformado.vtk",Thm,u1,u2,u3,phi,sigmaVm,[-dx(phi),-dy(phi),-dz(phi)]);
```

Anexo 3.1: Script GMSH

```
// Gmsh project created on Tue Apr 13 20:56:42 2021
```

```
//+
```

```
Point(1) = {-0, 4, -0, 1.0};
```

```
//+
```

```
Point(2) = {-0, 0, 0, 1.0};
```

```
//+
```

```
Point(3) = {0, 2, -0, 1.0};
```

```
//+
```

```
Point(4) = {4, 2, -0.2, 1.0};
```

```
//+
```

```
Point(5) = {6, 2, -0, 1.0};
```

```
//+
```

```
Point(6) = {-2, 2, 0.1, 1.0};
```

```
//+
```

```
Point(7) = {-4, 2, 0.2, 1.0};
```

```
//+
```

```
Ellipse(1) = {5, 3, 4, 2};
```

```
//+
```

```
Ellipse(2) = {5, 3, 4, 1};
```

```
//+
```

```
Line(3) = {1, 2};
```

```
//+
```

```
Curve Loop(1) = {3, -1, 2};
```

```
//+
```

```
Plane Surface(1) = {1};
```

```
//+
```

```
Recursive Delete {
```

```
  Point{4};
```

```
}
```

```
//+
Extrude {0, 0, 0.08} {
  Surface{1};
}
//+
Extrude {0, 0, -0.08} {
  Surface{1};
}
//+
Recursive Delete {
  Point{7}; Point{6};
}
//+
Physical Volume(1) = {2, 1};
//+
Physical Surface(2) = {28, 11};
//+
Physical Surface(3) = {1};
//+
Physical Surface(4) = {37};
```

Anexo 4: Problema dinámico

```
load"iiovtk" //para importar datos en Paraview

//PREPROCESO: GEOMETRIA, MALLA Y PARAMETROS FISICOS

border a1(t=0,10){x=t;y=0;label=1;}; // arista de abajo
border a2(t=0,10){x=10;y=t;label=2;}; // arista de la derecha
border a3(t=0,10){x=10-t;y=10;label=3;}; // arista de arriba
border a4(t=0,10){x=0;y=10-t;label=4;}; // arista de la izquierda
border interior(t=0,2*pi){x=5+2*cos(t);y=5+2*sin(t);}; // frontera interior

mesh Th=buildmesh(a1(30)+a2(30)+a3(30)+a4(30)+interior(40)); //malla

// PARAMETROS DEL MODELO

real f=0; //fuerza por unidad de volumen
real ro=0; //carga eléctrica por unidad de volumen
real epsilon= 4*8.8e-12; //Permitividad eléctrica

fespace Ph(Th,P0); //Espacio de elementos de Lagrange P0 para definir la región
Ph reg=region;

int ninterior=reg(5,5); //región interior del círculo
int nexterior=reg(9,9); //región exterior del círculo

cout<<"ninterior = "<<ninterior<<"nexterior= "<<nexterior<<endl;

Ph lambda=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de lambda según
la región
```

```

Ph mu=1e6*(region==ninterior)+1e9*(region==nexterior); //valor de mu según la
región
Ph alpha1=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha1 según la
región
Ph alpha2=0*(region==ninterior)+0.001*(region==nexterior); //valor de alpha2 según la
región
Ph rho=1*(region==ninterior)+1000*(region==nexterior); //valor de rho según la región

```

```

real T=5; //tiempo final
real dt=0.01; //paso de tiempo en discretización
int n = T/dt;
real beta = 0.25;
real gamma = 0.5;
real Welastm;
real Welaste;
real Kinetic;
real Wext;

```

```
//PROCESO: CALCULO DE LA SOLUCION
```

```

fespace Vh(Th,P2); //elementos de Lagrange P2
Vh u1,u2,phi,v1,v2,alpha,sigmaVm;
Vh up1, up2;
Vh vel1, vel2;
Vh vp1, vp2;
Vh acc1, acc2;

```

```
//Condiciones iniciales
```

```

acc1=0;
acc2=0;
vel1=0;

```

```

vel2=0;
u1=0;
u2=0;

//macros
macro IdentityMatrix [1.0, 0.0, 0.0, 1.0] //EOM
macro Trace(a) (a[0] + a[3]) //EOM
macro epsilonmatrix(u1,u2) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dy(u1) + dx(u2)),
dy(u2)] //EOM
macro Campoelectrico(phi) [-dx(phi),-dy(phi)]//EOM
macro N [1,0]//EOM
macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[1]*B[0], A[1]*B[1]] //EOM
macro mult (A,b)[A[0]*b[0]+A[1]*b[1],A[2]*b[0]+A[3]*b[1]]//EOM
macro stress (u1,u2,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2)) +2.
*mu*epsilonmatrix(u1,u2) - alpha1*(Campoelectrico(phi)*N)*IdentityMatrix -
alpha2*(DyadicProduct(Campoelectrico(phi),N) +
DyadicProduct(N,Campoelectrico(phi))))//EOM
macro desplazamiento(u1,u2,phi)(epsilon*Campoelectrico(phi) +
alpha1*Trace(epsilonmatrix(u1,u2))*N + alpha2*mult(epsilonmatrix(u1,u2),N) )//EOM

int counter=0;
solve Problem([u1,u2,phi],[v1,v2,alpha])=
int2d(Th)((rho/(beta*dt*dt))*[u1,u2]*[v1,v2])
-int2d(Th)((rho/(beta*dt*dt))*[up1,up2]*[v1,v2])
+int2d(Th)(stress(u1,u2,phi)*epsilonmatrix(v1,v2))
-int2d(Th)(desplazamiento(u1,u2,phi)*Campoelectrico(alpha))
+on(1,u1=0.0,u2=0.0,phi=0.0)
+on(3,u2=1*sin(10*counter*dt));

// Exportamos a fichero las energías
ofstream ff("Welastm.dat");
ofstream hh("Welaste.dat");

```

```

ofstream gg("Kinetic.dat");
ofstream tt("Wext.dat");

//Integrador temporal
for(real t=0;t<T;t+=dt)
{
// Predictor step for displacements and velocity
up1 = u1 + dt*vel1 + 0.5*dt*dt*(1-2*beta)*acc1;
up2 = u2 + dt*vel2 + 0.5*dt*dt*(1-2*beta)*acc2;
vp1 = vel1 + (1-gamma)*dt*acc1;
vp2 = vel2 + (1-gamma)*dt*acc2;
// Resolvemos el problema para el time step actual
Problem;
// Actualizamos valos de aceleraciones y velocidades
acc1 = 1/(beta*dt*dt)*(u1 - up1);
acc2 = 1/(beta*dt*dt)*(u2 - up2);
vel1 = vp1 + gamma/(beta*dt)*(u1 - up1);
vel2 = vp2 + gamma/(beta*dt)*(u2 - up2);
// Movemos la malla
real coef=1;
mesh th3=movemesh(Th,[x+coef*u1,y+coef*u2]); // malla deformada
plot(th3);
sigmaVm = sqrt(stress(u1,u2,phi)[0]*stress(u1,u2,phi)[0]
+stress(u1,u2,phi)[3]*stress(u1,u2,phi)[3]
-stress(u1,u2,phi)[0]*stress(u1,u2,phi)[3]
+3*stress(u1,u2,phi)[1]*stress(u1,u2,phi)[1]);

// Calculamos las energías
Welastm = int2d(Th)(0.5*stress(u1,u2,phi)*epsilonmatrix(u1,u2));
Welaste = -int2d(Th)(0.5*desplazamiento(u1,u2,phi)*Campoelectrico(phi));
Kinetic = int2d(Th)(0.5*rho*[vel1,vel2]'*[vel1,vel2]);
Wext = int2d(Th)(rho*[acc1,acc1]'*[u1,u2]) + Welastm + Welaste;

ff<<Welastm<<endl;

```

```
hh<<Welaste<<endl;
gg<<Kinetic<<endl;
tt<<Wext<<endl;

// Exportamos a paraview

savevtk("PRUEBA2dbeam_Time_step_"+n+"_"+counter+".vtk",th3,u1,u2,phi,sigmaV
m,[-dx(phi),-dy(phi)],counter);
counter++;
}
```

Anexo 5: TO Benchmark 1

```
load"iiovtk" //para importar datos en Paraview
//Preproceso: geometria, malla, parametros del modelo
real L = 4;
real W = 1;
int loaded = 1, free = 2, fixed = 3;

border b1(t=-0.025*W,0.025*W){x=0.5*L;y=t;label=loaded;}
border b2(t=0.025*W,0.5*W){x=0.5*L;y=t;label=free;}
border b3(t=0.5*L,-0.5*L){x=t;y=0.5*W;label=free;}
border b4(t=0.5*W,-0.5*W){x=-0.5*L;y=t;label=fixed;}
border b5(t=-0.5*L,0.5*L){x=t;y=-0.5*W;label=free;}
border b6(t=-0.5*W,-0.025*W){x=0.5*L;y=t;label=free;}

//Malla
int s = 2;
mesh Th=buildmesh(b1(2*s) + b2(24*s) + b3(200*s) + b4(50*s) + b5(200*s) +
b6(24*s));

//Espacio elementos finitos para la solución y densides
fespace Vh(Th,P1); //elementos de Lagrange P1
Vh ux,uy,vx,vy;

fespace Vh1(Th,P1);
Vh1 theta, thetaold, thetanew, dtheta; //densidades
Vh1 sens, sensn, sensv; //sensitivities

// parametros de la optimización
real volfrac = 0.5;
real p = 1.0;
real chg = 1;
//real alpha = 0.002;
```

```

real alpha = (W/20)^2.; //controla la length-scale del filtro de Helmholtz
int i = 0;
int Imax = 100;

// parametros mecánicos modelo
real nu=0.28; //coeficiente de Poisson del acero
real E=1e9;
real mu = E/(2*(1+nu));
real lambda = 2.*mu*nu/(1.-2.*nu);
real epsilon = 1e-6;
real fy= -1.0; //carga distribuida
real fx= 0.0; //carga distribuida
real sqrt2=sqrt(2.); //constante para ahorrar tiempo de calculo

//Make initial design
theta = volfrac;
//mu = theta^p*(mu1-mu0)+mu0;
//lambda = theta^p*(lambda1-lambda0)+lambda0;

//macros para la formulacion variacional
macro IdentityMatrix [1.0, 0.0, 0.0, 1.0] //EOM
macro Trace(a) (a[0] + a[3]) //EOM
macro epsilonmatrix(u1,u2) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dy(u1) + dx(u2)),
dy(u2)] //EOM
macro stress0(u1,u2)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2)) +2.
*mu*epsilonmatrix(u1,u2))/EOM
macro stressM(u1,u2)(theta^p*stress0(u1,u2)) //EOM
macro stressV(u1,u2)((1.0-theta^p)*epsilon*stress0(u1,u2)) //EOM
macro stress (u1,u2)(stressM(u1,u2) + stressV(u1,u2))/EOM
macro sensivityGauss(u1,u2)(-p*theta^(p-1)*(stressM(u1,u2)
stressV(u1,u2))*epsilonmatrix(u1,u2))/EOM

```

```

//Vh1 thechange;
real thechange;
// Optimization loop
while (chg > 1e-3 && i<=Imax){
    // incrementamos iteración de optimización
    i=i+1;
    //vieja densidad == nueva (la de la iteración anterior)
    thetaold=theta;
    //Update of penalising exponent
    if (i>20){
        p = min(3.0,p*1.008);
    }
    cout <<"exponent = " << p <<endl;

//Forward (FEM) problem
    solve
    FEMProblem([ux,uy],[vx,vy])=int2d(Th)((stress(ux,uy)*epsilonmatrix(vx,vy)))
        - int1d(Th,loaded)(fx*vX+fy*vY)+on(fixed,ux=0,uy=0);

//Adjoint equation
    // No se necesita al ser el problema auto-adjunto
//Sensitivity
    sens = -sensitivityGauss(ux,uy);

//Regularised sensitivity through the application
//of a Helmholtz's filter
    solve
    smoothing(sensn,sensv)=
    int2d(Th)(alpha*(dx(sensn)*dx(sensv)+dy(sensn)*dy(sensv))+sensn*sensv)-
    int2d(Th)(sens*sensv);

```

```

//OC update
real l1 = 0; real l2= 10000; real move= 0.05;
while((l2-l1)/(l2+l1)>(1e-4)&& l2>1e-40){
real lmid=0.5*(l2+l1);
thetaneu = max(0., max(theta-move,min(1.,min(theta+move,theta*(max(1e-
10,sensn/lmid))^0.3))));
if((int2d(Th)(thetaneu)-int2d(Th)(volfrac))>0){
l1= lmid;}
else{
l2=lmid;}
}

theta = thetaneu;

// Objective functions and visualization
real J = int1d(Th,loaded)(fx*ux + fy*uy);
real vol = int2d(Th)(theta)/int2d(Th)(1.);
plot(theta,fill=1);
cout <<"iter = "<< i << "; comp = "<<J<<"; vol = " <<vol<< ";change ="<< chg<<".----
-----"<<endl;
savevtk("Benchmark_1_" + i + ".vtk",Th,theta,i);

}

```

Anexo 6: TO Benchmark 2 (malla)

```
real L = 4;
real W = 1;
int loaded = 1, free = 2, fixed = 3;

border b1(t=-0.025*W,0.025*W){x=0.5*L;y=t;label=fixed;}
border b2(t=0.025*W,0.5*W){x=0.5*L;y=t;label=free;}
border b3(t=0.5*L,-0.5*L){x=t;y=0.5*W;label=free;}
border b4(t=0.5*W,0.025*W){x=-0.5*L;y=t;label=free;}
border b5(t=0.025*W,-0.025*W){x=-0.5*L;y=t;label=fixed;}
border b6(t=-0.025*W,-0.5*W){x=-0.5*L;y=t;label=free;}
border b7(t=-0.5*L,-0.01*L){x=t;y=-0.5*W;label=free;}
border b8(t=-0.01*L,0.01*L){x=t;y=-0.5*W;label=loaded;}
border b9(t=0.01*L,0.5*L){x=t;y=-0.5*W;label=free;}
border b10(t=-0.5*W,-0.025*W){x=0.5*L;y=t;label=free;}

//Malla
int s = 2;
mesh Th=buildmesh(b1(2*s) + b2(24*s) + b3(200*s) + b4(24*s) + b5(2*s) + b6(24*s)+
b7(98*s)+ b8(4*s)+ b9(98*s)+ b10(23*s));
```

Anexo 7: TO Benchmark 3 (malla)

```
real L = 4;
real W = 1;
int loaded = 1, free = 2, fixed1 = 3, fixed2 = 4;

border b1(t=-0.5*W,0.5*W){x=0.5*L;y=t;label=free;}
border b2(t=0.5*L,-0.495*L){x=t;0.5*W;label=free;}
border b3(t=-0.495*L,-0.5*L){x=t;y=0.5*W;label=loaded;}
border b4(t=0.5*W,-0.5*W){x=-0.5*L;y=t;label=fixed1;}
border b5(t=-0.5*L,0.495*L){x=t;y=-0.5*W;label=free;}
border b6(t=0.495*L,0.5*L){x=t;y=-0.5*W;label=fixed2;}

//Malla
int s = 2;
mesh Th=buildmesh(b1(50*s) + b2(198*s) + b3(2*s) + b4(50*s) + b5(198*s) +
b6(2*s));
```

Anexo 8: TO Benchmark 4 (malla)

```
real L = 4;
real W = 1;
int loaded = 1, free = 2, fixed1 = 3, fixed2 = 4;

border b1(t=-0.5*W,0.5*W){x=0.5*L;y=t;label=free;}
border b2(t=0.5*L,0.0025*L){x=t;0.5*W;label=free;}
border b3(t=0.0025*L,-0.0025*L){x=t;0.5*W;label=loaded;}
border b4(t=-0.0025*L,-0.495*L){x=t;0.5*W;label=free;}
border b5(t=-0.495*L,-0.5*L){x=t;y=0.5*W;label=loaded;}
border b6(t=0.5*W,-0.5*W){x=-0.5*L;y=t;label=fixed1;}
border b7(t=-0.5*L,0.495*L){x=t;y=-0.5*W;label=free;}
border b8(t=0.495*L,0.5*L){x=t;y=-0.5*W;label=fixed2;}

//Malla
int s = 1;
mesh Th=buildmesh(b1(50*s) + b2(99*s) + b3(1*s) + b4(99*s) + b5(1*s) + b6(50*s)+
b7(198*s)+ b8(2*s));
```

Anexo 9: TO piezoeléctrico

```
load"iiovtk" //para importar datos en Paraview

//Preproceso: geometría, malla, parametros del modelo

real L = 4;

real W = 1;

int loaded = 1, free = 2, fixed = 3, charged1=4, charged2=5;

border b1(t=-0.025*W,0.025*W){x=0.5*L;y=t;label=loaded;} //región pequeña donde
prescribir la objective function

border b2(t=0.025*W,0.5*W){x=0.5*L;y=t;label=free;}

border b3(t=0.5*L,-0.5*L){x=t;y=0.5*W;label=charged1;}

border b4(t=0.5*W,-0.5*W){x=-0.5*L;y=t;label=fixed;}

border b5(t=-0.5*L,0.5*L){x=t;y=-0.5*W;label=charged2;}

border b6(t=-0.5*W,-0.025*W){x=0.5*L;y=t;label=free;}

//Malla

int s = 1;

mesh Th=buildmesh(b1(2*s) + b2(24*s) + b3(200*s) + b4(50*s) + b5(200*s) +
b6(24*s));

// parametros de la optimización

real volfrac = 0.5;

real pm = 3.0; //penalización interpolación tensor de tensiones sigma

real pe = 3.0; //penalización interpolación desplazamiento eléctrico

real beta = 2.0; //parámetro para el proyectado de densidades

real eta = 0.5; //parámetro para el proyectado de densidades

real chg = 1;

real alpha = (W/20)^2.; //controla la length-scale del filtro de Helmholtz

int i = 0;

int Imax = 300;

// parametros de materiales
```

```

real permittivity= 4*8.8e-12; //Permitividad eléctrica
real nu=0.28; //coeficiente de Poisson del acero
real E=1e0;
real mu = E/(2*(1+nu));
real lambda = 2.*mu*nu/(1.-2.*nu);
real alpha1=0.001;
real alpha2=0.001;
real epsilon = 1e-6; //para definir el vacío escalado con respecto al material
real fy= -1.0; //carga distribuida
real fx= 0.0; //carga distribuida
real spring = 1e-1;

// Espacios FEM
fespace Vh1(Th,P1); //elementos de Lagrange P1
Vh1 vx, vy, vphi; //test functions del problema
Vh1 ux, uy, phi; //solution fields of the problem
Vh1 px, py; //mechanical adjoint state
Vh1 q; //electrical adjoint state

Vh1 theta, thetaold, thetanew, dtheta; //densidades
Vh1 sens, sensn; //sensitivities
Vh1 thetaf, thetap, thetav;
Vh1 DthetapDthetaf, DthetafDtheta;

// Macros usadas
macro IdentityMatrix [1.0, 0.0, 0.0, 1.0] //EOM
macro Trace(a) (a[0] + a[3]) //EOM
macro epsilonmatrix(u1,u2) [dx(u1), 0.5*(dy(u1) + dx(u2)), 0.5*(dy(u1) + dx(u2)),
dy(u2)] //EOM
macro Campoelectrico(phi) [-dx(phi),-dy(phi)]//EOM
macro N [1,0]//EOM

```

```

macro DyadicProduct(A,B) [A[0]*B[0], A[0]*B[1], A[1]*B[0], A[1]*B[1]] //EOM
macro mult (A,b)[A[0]*b[0]+A[1]*b[1],A[2]*b[0]+A[3]*b[1]]//EOM
macro stress0 (u1,u2,phi)(lambda*IdentityMatrix*Trace(epsilonmatrix(u1,u2)) +2.
*mu*epsilonmatrix(u1,u2) - alpha1*(Campoelectrico(phi)*N)*IdentityMatrix -
alpha2/2*(DyadicProduct(Campoelectrico(phi),N) +
DyadicProduct(N,Campoelectrico(phi))))//EOM
macro stressM(u1,u2,phi)(thetap^pm*stress0(u1,u2,phi)) //EOM
macro stressV(u1,u2,phi)((1.0-thetap^pm)*epsilon*stress0(u1,u2,phi)) //EOM
macro stress (u1,u2,phi)(stressM(u1,u2,phi) + stressV(u1,u2,phi))//EOM
macro desplazamiento0(u1,u2,phi)(epsilon*Campoelectrico(phi) +
alpha1*Trace(epsilonmatrix(u1,u2))*N + alpha2*mult(epsilonmatrix(u1,u2),N) )//EOM
macro desplazamientoM(u1,u2,phi)(thetap^pe*desplazamiento0(u1,u2,phi)) //EOM
macro desplazamientoV(u1,u2,phi)((1.0-
thetap^pe)*epsilon*desplazamiento0(u1,u2,phi)) //EOM
macro desplazamiento (u1,u2,phi)(desplazamientoM(u1,u2,phi) +
desplazamientoV(u1,u2,phi))//EOM
macro sensitivityGauss(u1,u2,phi,p1,p2,q)(-pm*thetap^(pm-1)*(stressM(u1,u2,phi) -
stressV(u1,u2,phi))*epsilonmatrix(p1,p2) + pe*thetap^(pe-
1)*(desplazamientoM(u1,u2,phi) -
desplazamientoV(u1,u2,phi))*Campoelectrico(q))//EOM
macro Objective(u1,u2) (u2) //EOM
macro DObjective(u1,u2) (1) //EOM

```

```
// Inicialización de densidades
```

```
theta = volfrac;
```

```
// Exportamos a fichero la función objetivo J
```

```
ofstream ff("Jmuelle3.dat");
```

```
i=0.;
```

```
real thechange;
```

```
// Optimization loop
```

```
while (chg > 1e-3 && i<=Imax){
```

```
    // incrementamos iteración de optimización
```

```
    i=i+1;
```

```

//vieja densidad == nueva (la de la iteración anterior)

thetaold=theta;

//Filtro de densidades y proyección

solve
int2d(Th)(alpha*(dx(thetaf)*dx(thetav)+dy(thetaf)*dy(thetav))+thetaf*thetav)-
int2d(Th)(theta*thetav);

thetap = (tanh(beta*eta) + tanh(beta*(thetaf-eta)))/(tanh(beta*eta) + tanh(beta*(1-
eta)));

cout <<"mechanical exponent = " << pm <<endl;

cout <<"electric exponent = " << pe <<endl;

cout <<"beta = " << beta <<endl;

//Forward (FEM) problem

solve
ForwardProblem([ux,uy,phi],[vx,vy,vphi])=int2d(Th)(stress(ux,uy,phi)*epsilonmatrix(
vx,vy))
-int2d(Th)(desplazamiento(ux,uy,phi)*Campoelectrico(vphi))
+int1d(Th,loaded)(spring*uy*vy)
+on(charged1,phi=1.) + on(charged2,phi=-1.) +
on(fixed,ux=0,uy=0);

//Adjoint problem

solve Adjoint([px,py,q],[vx,vy,vphi])= int1d(Th,loaded)(DObjective(ux,uy)*vy)
-int2d(Th)(stress(px,py,q)*epsilonmatrix(vx,vy))
+ int2d(Th)(desplazamiento(px,py,q)*Campoelectrico(vphi))
-int1d(Th,loaded)(spring*py*vy)
+on(charged1,q=0.) + on(charged2,q=-0.) +
on(fixed,px=0,py=0.);

//Sensitivity with respect to projected and filtered density

sens = -sensitivityGauss(ux,uy,phi,px,py,q);

//Chain rule

DthetapDthetaf = beta*(1-(tanh(beta*(thetaf-eta)))^2)/(tanh(beta*eta) + tanh(beta*(1-
eta)));

```

```

solve smoothingderivative(DthetafDtheta,thetav)=
int2d(Th)(alpha*(dx(DthetafDtheta)*dx(thetav)+dy(DthetafDtheta)*dy(thetav))+Dtheta
fDtheta*thetav)-int2d(Th)(thetav);

sensn = sens*DthetapDthetaf*DthetafDtheta;

//OC update (creiterio de optimalidad)
real l1 = 0; real l2= 10000; real move= 0.05;
while((l2-l1)/(l2+l1)>(1e-4)&& l2>1e-40){
real lmid=0.5*(l2+l1);
thetanew = max(0., max(theta-move,min(1.,min(theta+move,theta*(max(1e-
10,sensn/lmid))^0.3))));

if((int2d(Th)(thetanew)-int2d(Th)(volfrac))>0){
l1= lmid;}
else{
l2=lmid;}
}

theta = thetanew;

// Objective functions and visualization
real Jmuelle3 = int1d(Th,loaded)(Objective(ux,uy));
ff<<Jmuelle3<<endl;
real vol = int2d(Th)(theta)/int2d(Th)(1.);
plot(thetap,fill=1);
cout <<"-----" <<endl;
cout <<"iter = " << i << "; Objective Function = " <<Jmuelle3<<"; vol = " <<vol<<
";change = " << chg<<".-----" <<endl;
savevtk("TO_piezo_Muelle3_" + i + ".vtk",Th,thetap,i);
}

```