

UNIVERSIDAD NACIONAL DEL LITORAL



# Desarrollo de un Método de Composición de Dominios tipo Chimera en Mecánica Computacional

Bruno Alberto Storti

FICH

FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS

INTEC

INSTITUTO DE DESARROLLO TECNOLÓGICO PARA LA INDUSTRIA QUÍMICA

CIMEC

CENTRO DE INVESTIGACIÓN DE MÉTODOS COMPUTACIONALES

sinc(i)

INSTITUTO DE INVESTIGACIÓN EN SEÑALES, SISTEMAS E INTELIGENCIA

COMPUTACIONAL

Tesis de Doctorado **2020**





**Doctorado en Ingeniería**  
mención mecánica computacional

Título de la obra:

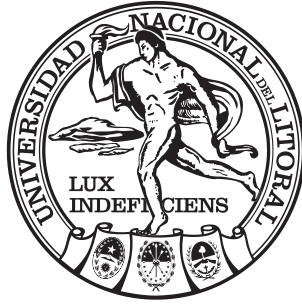
**Desarrollo de un Método de Composición  
de Dominios tipo Chimera en  
Mecánica Computacional**

Autor: Bruno Alberto Storti

Lugar: Santa Fe, Argentina

Palabras clave:

método quimera , mallas solapadas,  
mallas no coincidentes, optimización,  
esquema matrix-free, elementos finitos,  
optimización en paralelo, enjambre de partículas.



UNIVERSIDAD NACIONAL DEL LITORAL  
Facultad de Ingeniería y Ciencias Hídricas  
Instituto de Desarrollo Tecnológico para la Industria Química  
Centro de Investigación de Métodos Computacionales  
Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional

# Desarrollo de un Método de Composición de Dominios tipo Chimera en Mecánica Computacional

**Bruno Alberto Storti**

Tesis remitida al Comité Académico del Doctorado  
como parte de los requisitos para la obtención  
del grado de  
DOCTOR EN INGENIERÍA  
Mención Mecánica Computacional  
de la  
UNIVERSIDAD NACIONAL DEL LITORAL

**2020**

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas, Ciudad Universitaria,  
Paraje "El Pozo", S3000, Santa Fe, Argentina.





UNIVERSIDAD NACIONAL DEL LITORAL

Facultad de Ingeniería y Ciencias Hídricas

Instituto de Desarrollo Tecnológico para la Industria Química

Centro de Investigación de Métodos Computacionales

Instituto de Investigaciones en Señales, Sistemas e Inteligencia Computacional

# Desarrollo de un Método de Composición de Dominios tipo Chimera en Mecánica Computacional

**Bruno Alberto Storti**

**Lugar de trabajo:**

CIMEC

Centro de Investigación de Métodos Computacionales

Facultad de Ingeniería y Ciencias Hídricas

Universidad Nacional del Litoral

**Director:**

Jorge D'Elía      CIMEC

**Co-director:**

Luciano Garelli      CIMEC

**Jurado Evaluador:**

Dr. César Augusto Aguirre

Dpto. de Ciencias de la Tierra (DCT) - UNER

Dr. Sergio Elaskar

(IDIT (UNC-CONICET)) - UNC

Dra. Marian Gabriela Marcovecchio

INGAR (UTN-CONICET) - UNL

Dr. Mariano Vázquez

BSC-Centro Nacional de Supercomputación, España

**2020**





2020

Año del General  
Manuel Belgrano



**UNIVERSIDAD NACIONAL DEL LITORAL**  
**Facultad de Ingeniería y Ciencias Hídricas**

Santa Fe, 28 de Mayo de 2020.

Como miembros del Jurado Evaluador de la Tesis de Doctorado en Ingeniería titulada “Desarrollo de un Método de Composición de Dominios tipo Chimera en Mecánica Computacional”, desarrollada por el Ing. Bruno Alberto STORTI, en el marco de la Mención “Mecánica Computacional”, certificamos que hemos evaluado la Tesis y recomendamos que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería.

La aprobación final de esta disertación estará condicionada a la presentación de dos copias encuadernadas de la versión final de la Tesis ante el Comité Académico del Doctorado en Ingeniería.

.....  
Dr. César Augusto Aguirre

.....  
Dr. Sergio Elaskar

.....  
Dra. Marian Gabriela Marcovecchio

.....  
Dr. Mariano Vázquez

Santa Fe, 28 de Mayo de 2020.

Certificamos haber leído la Tesis, preparada bajo nuestra dirección en el marco de la Mención “Mecánica Computacional” y recomendamos que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería.

.....  
Dr. Luciano Garelli  
Codirector de tesis

.....  
Dr. Jorge D’Elía  
Director de Tesis

Universidad Nacional del Litoral  
Facultad de Ingeniería y  
Ciencias Hídricas

Secretaría de Posgrado

Ciudad Universitaria – C.C. 217  
Ruta Nacional N° 168 – Km 472,4  
(3000) Santa Fe – Argentina  
Tel: (54)(0342) 4575 233 / 245 / 246 – int. 103  
Fax: (54) (0342) 4575 224  
E-mail: posgrado@fich.unl.edu.ar

*Dr. José Luis Meacor*  
*DNI 14830760*  
*Secretario de Posgrado*  
*FICH.*



# Declaración legal del autor

Esta Tesis ha sido remitida como parte de los requisitos para la obtención del grado de Doctor ante la Universidad Nacional del Litoral y ha sido depositada en la Biblioteca de la Facultad de Ingeniería y Ciencias Hídricas para que esté disponible a sus lectores bajo las condiciones estipuladas por el Reglamento de la mencionada Biblioteca.

Citaciones breves de esta disertación son permitidas sin la necesidad de un permiso especial, en la suposición de que la fuente sea correctamente citada. Solicitudes de permiso para una citación extendida o para la reproducción de este manuscrito en un todo o en parte serán exigidas por el portador legal del derecho de propiedad intelectual de la misma.

Bruno Alberto Storti



# Agradecimientos

En primer término, quiero agradecer a las instituciones que me permitieron realizar mis estudios de doctorado, comenzando por el Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), el cual me brindó un lugar de trabajo para desarrollar las labores correspondientes, así como al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), a la Universidad Nacional del Litoral (UNL) y la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT), quienes proveyeron el financiamiento a través de la beca doctoral y sus proyectos. Además, quiero agradecer a la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional (UTN) y sus profesores, donde llevé a cabo mis estudios en Ingeniería, y tomé primer contacto con las simulaciones numéricas.

Quiero agradecer especialmente al profesor Alejandro Albanesi, por introducirme en el apasionante mundo de la Mecánica Computacional, permitiéndome participar de los proyectos de investigación realizados en el Laboratorio de Flujiometría (FLOW-UTN), y por el apoyo humano brindado en todo momento.

Deseo expresar mi agradecimiento a los directores de esta Tesis, los Dres. Luciano Garelli y Jorge D'Elía, por el apoyo que han brindado a este trabajo, por la libertad y respeto a mis sugerencias e ideas y por el rigor científico con que han orientado las mismas.

Agradezco también a todos los compañeros del CIMEC, de quienes nunca dejé de aprender, fueron mi soporte y con quienes tuve la suerte de generar un fuerte vínculo de amistad.

Un agradecimiento especial es también dado a toda la comunidad de software libre, que me ha provisto de incontable cantidad de herramientas las cuales utilicé en el desarrollo de esta Tesis y que utilizo a diario.

Finalmente, nada de esto hubiera sido posible sin el apoyo incondicional de mis amigos y mi familia, especialmente de mi viejo, con quien tengo la dicha de compartir la misma pasión por la Mecánica Computacional.



# Resumen

El método quimera se encuentra dentro de los Métodos de Composición de Dominios (DCM, por *Domain Composition Methods*). Los DCM consisten en obtener la solución de un problema de la mecánica computacional a partir de las formulaciones del mismo problema expresadas en varios subdominios. Uno de los objetivos principales del método quimera es evitar la dificultad que representa la generación de una malla conforme no estructurada en problemas que contienen cuerpos de formas geométricas intrincadas. En esta tesis se plantea el desarrollo e implementación de un método de composición de dominios tipo quimera para mallas superpuestas orientado a la resolución de problemas ingenieriles de optimización que impliquen cuerpos móviles, aprovechando cualidades inherentes del método como son la simplificación de la generación de mallas y el refinamiento local. Se plantean en primer lugar algoritmos para un acople algebraico de mallas estructuradas parcialmente solapadas. Esto permite analizar la convergencia de los *solvers* iterativos, y la convergencia en malla del método propuesto, como así también comparar los errores introducidos por el acople con respecto a una malla conforme. Posteriormente, se implementa el método propuesto en mallas no estructuradas parcial y completamente solapadas. A continuación, se introduce un interpolador de alto orden para no perder la convergencia espacial del método, como también un algoritmo para la detección de interfases de interpolación automático. En cuanto a la resolución del acople entre subdominios se propone un esquema tipo Dirichlet-Dirichlet para la interpolación, mientras que para la resolución de los sistemas lineales se utilizan *solvers* iterativos eficientes tal es de bi-gradientes conjugados estabilizado (BiCGStab, por *BiConjugate Gradient Stabilized*). Luego se introduce el optimizador pyOpt, un optimizador de licencia libre escrito en el lenguaje de programación Python, el cual es acoplado al método quimera para abordar problemas de optimización de transferencia térmica en plataformas de computación de alto desempeño (HPC, por *High Performance Computing*). Dentro de la plataforma de pyOpt se utilizarán tanto optimizadores basados en gradiente, como libres de gradiente. Finalmente, la robustez de las herramientas desarrolladas son validadas mediante su aplicación en casos ingenieriles, principalmente en optimización de transferencia térmica. La implementación final de los algoritmos se realiza en lenguaje C++, bajo los paradigmas de la Programación Orientada a Objetos (OOP, por *Object-Oriented Programming*), utilizando plataformas de HPC. Los códigos computacionales obtenidos son incorporados al entorno del código multifísica paralelo PETSc-FEM.





# Abstract

The Chimera scheme can be characterized as a Domain Composition Method (DCM). The main objective of DCMs is to find the solution to a problem of computational mechanics from the formulations of the same problem expressed on two or more subdomains. One of the main purposes of the Chimera method is to avoid the difficulty represented by the generation of a conformal unstructured mesh for problems containing complex bodies. In the present thesis, the development and implementation of a Chimera-like domain composition method for overlapping meshes applied to optimization problems in engineering addressing moving bodies are presented, taking advantage of inherent qualities of the method such as simplified mesh generation and local refinement. First, an algebraic-based coupling algorithm for structured meshes partially overlapped is introduced. This allows us to assess the convergence rate of the iterative solvers and the mesh convergence of the proposed coupling scheme, as well as to measure the error introduced by the coupling in regards to a conformal mesh. Afterward, the implementation of the methodology is extended to address partially and completely overlapping non-structured meshes. A high-order interpolation operator is developed to maintain the mesh convergence of the FEM method after the coupling. Additionally, an algorithm to properly detect the interpolation interfaces automatically is presented. Regarding the resolution of the coupled system between subdomains, a Dirichlet-Dirichlet scheme is proposed for the interpolation. Meanwhile, for the resolution of the resulting linear systems, efficient iterative solvers such, as the Bi-Conjugated Gradients Stabilized (BiCGStab), are used. Then, the pyOpt optimization framework is introduced, an open-source code based on the Python programming language, which is coupled to the Chimera method to address heat transfer optimization problems on High-Performance Computing (HPC) platforms. Both gradient-free and gradient-based optimizers implemented in pyOpt are used in this thesis. Finally, the applicability and robustness of the developed tools are assessed in several engineering cases, mainly in heat transfer optimization problems. The final implementations of the algorithms are made in the C++ programming language, under the paradigm of Object-Oriented Programming (OOP) to be used on HPC platforms. The computational codes obtained are incorporated into the environment of the parallel multiphysics code PETSc-FEM.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación y antecedentes . . . . .	1
1.1.1. Introducción al método quimera . . . . .	1
1.1.2. El método quimera como un método de composición de dominios . . . . .	2
1.2. Objetivos . . . . .	5
1.2.1. Objetivo general . . . . .	5
1.2.2. Objetivos particulares . . . . .	5
1.3. Estructura de la Tesis . . . . .	6
1.4. Publicaciones . . . . .	7
1.4.1. Publicaciones en revistas indizadas . . . . .	7
1.4.2. Publicaciones y presentaciones en congresos . . . . .	8
<b>2. Mallas 2D estructuradas parcialmente solapadas</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Descripción del método quimera . . . . .	9
2.2.1. Algoritmo de acople matrix-free . . . . .	11
2.2.2. Algoritmo BiCGStab . . . . .	12
2.2.3. Estimación del error producido por el método quimera . . . . .	13
2.3. Ejemplos numéricos . . . . .	14
2.3.1. Comparación del método quimera con una malla estructurada uniforme . . . . .	15
2.3.2. Comparación de una malla quimera con una malla conforme análoga . . . . .	19
2.4. Análisis de convergencia de las soluciones iterativas . . . . .	20
2.5. Conclusiones . . . . .	21
<b>3. Mallas 2D no estructuradas parc. y compl. solapadas</b>	<b>23</b>
3.1. Introducción . . . . .	23
3.2. Interpolación de alto orden . . . . .	23

3.2.1.	Esquema de interpolación . . . . .	23
3.2.2.	Ejemplo numérico. Error de interpolación . . . . .	26
3.3.	Ejemplos numéricos . . . . .	26
3.3.1.	Ejemplo numérico 1: Solución del problema de Poisson en dominios 2D parcialmente solapados . . . . .	26
3.3.2.	Ejemplo numérico 2: Solución del problema de Poisson en dominios 2D completamente solapados . . . . .	27
3.3.3.	Ejemplo numérico 3: condición de borde impuesta sobre la malla patch completamente solapada . . . . .	30
3.4.	Conclusiones . . . . .	31
<b>4.</b>	<b>Acoplamiento del método quimera con optimización</b>	<b>33</b>
4.1.	Introducción . . . . .	33
4.2.	Optimización combinada . . . . .	33
4.3.	Optimizadores en pyOpt . . . . .	34
4.3.1.	Optimizador ALPSO . . . . .	34
4.3.2.	Optimizador SLSQP . . . . .	37
4.4.	Paralelización del proceso de optimización . . . . .	38
4.4.1.	Distribución dinámica de procesos (DPM) . . . . .	38
4.5.	Tratamiento de múltiples dominios con el método quimera . . . . .	39
4.6.	Estrategia de optimización con el método quimera . . . . .	41
4.7.	Ejemplo test . . . . .	41
4.7.1.	Configuración del problema de optimización . . . . .	42
4.7.2.	Parámetros utilizados en la optimización . . . . .	43
4.7.3.	Análisis de la escalabilidad . . . . .	43
4.8.	Resultados y conclusiones . . . . .	43
<b>5.</b>	<b>Optimización del sistema de refrigeración de un álabe</b>	<b>45</b>
5.1.	Introducción . . . . .	45
5.2.	Descripción del experimento . . . . .	46
5.3.	Geometría del álabe y condiciones de carga . . . . .	46
5.4.	Descripción del problema de optimización . . . . .	47
5.4.1.	Transformación de los dominios de las variables . . . . .	48
5.5.	Convergencia en malla . . . . .	49
5.6.	Resultados . . . . .	49

5.7. Conclusiones . . . . .	50
<b>6. Optimización con el método quimera en dominios 3D</b>	<b>53</b>
6.1. Introducción . . . . .	53
6.2. Algoritmo de interpolación de alto orden para mallas 3D . . . . .	53
6.2.1. Operadores de interpolación . . . . .	54
6.2.2. Error de interpolación . . . . .	56
6.3. Reconocimiento automático de frontera de interpolación . . . . .	57
6.4. Convergencia y error del método quimera en 3D . . . . .	58
6.5. Test de optimización térmica 3D . . . . .	60
6.5.1. Configuración del problema . . . . .	60
6.5.2. Problema de optimización . . . . .	61
6.5.3. Resultados . . . . .	62
6.6. Optimización de la refrigeración de un molde de inyección . . . . .	63
6.6.1. Modelo matemático y físico del problema . . . . .	64
6.6.2. Generación de malla de tuberías según las variables de diseño . . . . .	65
6.6.3. Modelo matemático del problema de optimización . . . . .	68
6.6.4. Resultados . . . . .	68
6.7. Conclusiones . . . . .	71
<b>7. Conclusiones</b>	<b>73</b>
<b>Apéndices</b>	<b>79</b>
<b>A matrix-free Chimera approach based on D/D coupl. for DCM</b>	<b>81</b>
1. Introduction . . . . .	84
2. Numerical formulation . . . . .	87
2.1. Algebraic system splitting . . . . .	87
2.2. Iterative solution of the system . . . . .	88
2.3. Interpolation operators . . . . .	90
2.4. Interpolation error . . . . .	94
2.5. Chimera error estimation . . . . .	95
2.6. Degrees of freedom as a metrics to compare meshes . . . . .	95
3. Numerical examples . . . . .	96
3.1. Numerical example 1: Two-dimensional Poisson problem on partially overlapped meshes . . . . .	96

3.2.	Numerical example 2: Two-dimensional Poisson problem on completely overlapped meshes . . . . .	101
3.3.	Numerical example 3: Boundary condition imposed on patch mesh completely overlapped . . . . .	106
3.4.	Numerical example 4: Three-dimensional Laplace problem on completely overlapped domains . . . . .	110
4.	Conclusions and outlook. . . . .	114
<b>Optimization of a blade refrigeration using Chimera and HPC</b>		<b>121</b>
1.	Introduction . . . . .	124
2.	Numerical modeling and method . . . . .	127
2.1.	The Chimera method for the heat conduction equation . . . . .	127
2.2.	Algorithm of the Chimera scheme . . . . .	129
2.3.	Interpolation scheme . . . . .	131
2.4.	Treatment of the hole nodes on the background mesh . . . . .	133
2.5.	Mesh convergence and iterative solution . . . . .	134
3.	Optimization problems on overlapped grids . . . . .	137
3.1.	Hybrid optimization . . . . .	137
3.2.	Optimization algorithms . . . . .	138
3.3.	Parallelization of the optimization process . . . . .	139
3.4.	Optimization strategy . . . . .	139
3.5.	Numerical example . . . . .	140
4.	Vane optimization . . . . .	145
4.1.	Profile geometry and load conditions . . . . .	146
4.2.	Mathematical model of the optimization problem . . . . .	148
4.3.	Transformation of design variable domains . . . . .	149
4.4.	Grid independence analysis of background and patch meshes . . . . .	150
4.5.	Results . . . . .	151
5.	Conclusions . . . . .	154

# Índice de figuras

1.1. Métodos de descomposición de dominios vs métodos de composición de dominios <sup>1</sup> . . .	3
2.1. Descripción del método quimera . . . . .	10
2.2. Visualización del término fuente expresado en la Ec. (2.10), con $x_c = 0.75$ y $y_c = 0.5$ . . . . .	14
2.3. Mallas utilizadas para el análisis del error en la norma $L_2$ . . . . .	15
2.4. Solución (a) y error (b) obtenidos a la ecuación de Poisson con la fuente ubicada en la malla fina. . . . .	17
2.5. Solución (a) y error (b) obtenidos de la ecuación de Poisson con la fuente ubicada en la zona de solapamiento. . . . .	18
2.6. Solución (a) y error (b) obtenidos de la ecuación de Poisson con la fuente ubicada en la malla gruesa. . . . .	19
2.7. Imagen conceptual de las mallas quimera y de las mallas conformes análogas utilizadas para el presente estudio. . . . .	19
2.8. Historial de convergencia de BiCGStab para el método quimera y de CG para las mallas conformes y uniformes. . . . .	21
3.1. Ilustración de una interpolación de bajo orden lineal (a) y una interpolación de alto orden (b) para mallas unidimensionales parcialmente solapadas. . . . .	24
3.2. Esquema de interpolación para mallas no estructuradas . . . . .	25
3.3. Error RMS de la interpolación propuesta. . . . .	26
3.4. Ejemplos de las mallas utilizadas en el problema 2D de Poisson. Las tres mallas poseen similar cantidad de grados de libertad $N$ . . . . .	27
3.5. Dos dominios parcialmente solapados (Izq.), y dos dominios completamente solapados (Der.). . . . .	28
3.6. Detección automática topológica de la frontera de interpolación de la malla $\Omega_1$ . . . . .	29

3.7. Mallas empleadas: a) Malla conforme homogénea con tamaño de elemento casi constante en todo el dominio (OD homogénea); b) Malla conforme con refinamiento progresivo hacia el centro del dominio (OD refinada); c) Los elementos en rojo de la malla OD refinada son rotados y perturbados aleatoriamente para generar $\Omega_2$ ; d) Mallas solapadas generadas a partir de la malla OD refinada (malla quimera) . . . . .	30
3.8. Mallas empleadas: a) Malla OD homogénea con tamaño de elemento casi constante; b) Malla OD refinada con un refinamiento progresivo hacia $\Gamma_2$ ; c) Los elementos en rojo de la malla OD refinada son rotados ( $\vec{\omega}$ ) y perturbados aleatoriamente para así generar la malla $\Omega_2$ de método quimera ; d) Malla quimera regular formada por una malla no estructurada homogénea que cubre todo el dominio, y una malla estructurada refinada progresivamente hacia $\Gamma_2$ . . . . .	30
4.1. Algoritmo PSO en serie (a) y en paralelo (b). . . . .	36
4.2. Convergencia de la función objetivo obtenida por los optimizadores ALPSO y SLSQP en el ejemplo numérico. . . . .	40
4.3. Solución de la ecuación de Laplace obtenida por el método quimera con múltiples subdominios: (a) solución de $\phi$ en $\Omega_1$ y $\Omega_2$ , y la solución global, (b) solución del gradiente de $\phi$ en $\Omega_1$ y $\Omega_2$ , y la solución global. . . . .	41
4.4. Esquema de la malla de fondo $\Omega_1$ y de la malla <i>patch</i> $\Omega_2$ utilizadas en el problema de optimización. . . . .	42
4.5. Posición de los pasajes refrigerantes durante el proceso de optimización. . . . .	44
5.1. Configuración de los álabes durante los ensayos experimentales de Hylton et al. (1983) (Izq.). Ubicación de los instrumentos de medición sobre los álabes durante los ensayos (Der.) <sup>2</sup> . . . . .	46
5.2. Geometría del álabes C3X (Izq.), y datos del refrigerante (Der.). . . . .	47
5.3. Representación conceptual de la malla de fondo ( $\Omega_1$ ) y la malla <i>patch</i> $\Omega_2$ utilizada por el esquema quimera . . . . .	49
5.4. Comparación de los resultados obtenidos en la temperatura para: (a) disposición de referencia, (b) peor condición obtenida y (c) disposición optimizada. . . . .	50
6.1. Interpolación de alto orden sobre mallas 3D. . . . .	54
6.2. Error RMS de la interpolación propuesta en dominios 3D. . . . .	56
6.3. Reconstrucción de valores de la función de prueba $\phi(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$ impuesta en una malla gruesa ( $\Omega_1$ ), sobre una malla más fina ( $\Omega_2$ ). . . . .	57



6.4. Proceso generación de nodos ficticios sobre $\partial\Omega_2$ y detección de los nodos de interpolación sobre la malla de fondo $\Omega_1$ . Se consideran dos mallas completamente solapadas $\Omega_1$ y $\Omega_2$ . . . . .	58
6.5. Correcta detección de elementos en $\Omega_1$ (tetraedros rojos) intersectados por $\partial\Omega_2$ . . . . .	59
6.6. Elementos de $\Omega_1$ que son intersectados por $\partial\Omega_2$ en una sección de la Fig. 6.5. Se observa que al no utilizar nodos ficticios, nodos internos de $\Omega_1$ pueden quedar vinculados a nodos <i>huecos</i> , pudiendo introducir errores en la solución . . . . .	59
6.7. Dominios $\Omega_1$ y $\Omega_2$ del problema de optimización, donde $\Omega_1$ consiste en un cubo de longitud unitaria, mientras que $\Omega_2$ consiste en 4 esferas huecas, cuyo bordes internos $\Gamma_2$ tienen impuestas una condición tipo Dirichlet. Se visualizan también las mallas utilizadas para discretizar ambos dominios. . . . .	61
6.8. Convergencia de la función objetivo obtenida por el optimizador ALPSO. . . . .	62
6.9. Disposición optimizada de las esferas dentro del dominio cúbico: (a) espacio de búsqueda analizado por cada esfera y posición final obtenida; (b) temperatura obtenida en el dominio mediante la configuración optimizada. . . . .	63
6.10. Modelo esquemático del molde de inyección, de la pieza a fabricar y de los canales refrigerantes. . . . .	66
6.11. Vista superior y lateral de la geometría de los canales de refrigeración. . . . .	66
6.12. Construcción de la malla de las tuberías. . . . .	67
6.13. Detalles de las mallas utilizadas en el molde, pieza y canales. . . . .	67
6.14. Convergencia de la función objetivo obtenida por el optimizador ALPSO. . . . .	69
6.15. Parámetros de entrada del proceso de optimización y flujo térmico en los canales utilizando una representación con coordenadas paralelas. . . . .	69
6.16. Temperatura obtenida por la configuración óptima de los canales. . . . .	70
6.17. Gradiente térmico obtenido por la configuración óptima de los canales. . . . .	70
6.18. Temperatura en el molde y su gradiente térmico obtenidos en $\Omega_1$ y $\Omega_2$ presentada en una sección transversal. La continuidad de la temperatura y del gradiente térmico pueden ser observados. . . . .	71
A.1. Example of a square domain $\Omega$ split into two subdomains with an overlapped region $\Omega_O$ (left) and subsequently, the splitting of $\Omega_1$ nodes into groups. . . . .	87
A.2. Third order interpolation on an unstructured mesh. . . . .	90

A.3. Topological neighbours employed for the reconstruction of $\phi$ on the midside nodes of a boundary element. The system for the reconstruction is well-conditioned in (a) and (c), and ill-conditioned in (b), when using the neighbours of the first layer. When the neighbours on the second layer are also used in (b), the system becomes well-conditioned. . . . .	94
A.4. RMS error for the proposed interpolation. . . . .	94
A.5. Examples of the meshes used for the current numerical test. The OD refined mesh is generated from Chimera meshes by removing $\Omega_1$ overlapped elements and then using a Delaunay triangulation in the overlapping area. The OD homogeneous mesh has near constant element size. The three meshes have approximately the same $N$ . . . .	98
A.6. Mesh convergence of the solution (left) and its derivative (right) of the Poisson problem for a delta source term with $\bar{x}_c = [0.75, 0.5]$ , $\delta = 0.025$ and $r = 0.15$ evaluated on OD homogeneous, OD refined and partially overlapped Chimera meshes. . . . .	99
A.7. Solution of numerical example 1 solved on overlapping meshes showing good agreement on the overlapped area and overlapping meshes. Left: solution of both subdomains ( $\Omega_1$ and $\Omega_2$ patch together). Different element sizes of meshes and the overlapped region can be observed. Right: A detailed view of the overlapping zone showing the continuity of the solution between both subdomains. . . . .	99
A.8. Solution of numerical example 1 on two domains partially overlapped and projected over the planes $y = 0.5$ , $y = 0.35$ and $y = 0.1$ of the function (left) and its derivative on the x-axis (right), in contrast to the reference solution. The continuity of both the solution and its derivative over the overlapped area is remarkable. . . . .	100
A.9. Convergence of the iterative solvers for numerical example 1 on an OD homogeneous mesh, a Chimera mesh composed of two partially overlapped domains, and an OD refined mesh. A relative residual tolerance of $10^{-8}$ was used in all cases. Mesh sizes are depicted in Table A.1. . . . .	101
A.10. Automatic topological detection of interpolation boundary on background mesh: (a) domains $\Omega_1$ and $\Omega_2$ completely overlapped and interpolation boundaries $\Gamma_{I_1}$ and $\Gamma_{I_2}$ where the transmission of information occurs. (b) the different types of nodes of the interface detection procedure. . . . .	102

A.11.Examples of the meshes occupied for the current numerical test: a) A one-domain homogeneous mesh with nearly constant element size (OD homogeneous). b) A one-domain mesh with progressive refinement towards the domain centre (OD refined). c) Red elements of the OD refined mesh are rotated and randomly perturbed to generate the  $\Omega_2$  domain. d) The overlapped meshes generated from the OD refined mesh (Chimera mesh). . . . . 103

A.12.Mesh convergence of the solution (left) and its derivative (right) of the Poisson problem for a delta source term with  $\bar{x}_c = [0.5, 0.5]$ ,  $\delta = 0.015$  and  $r = 0.1$  evaluated on OD homogeneous, OD refined and fully overlapped Chimera meshes. . . . . 104

A.13.Solution of numerical example 2 for a delta source term (with parameters defined as depicted in Table A.2) solved on two completely overlapping meshes. Left: solution of both subdomains ( $\Omega_1$  and  $\Omega_2$  patch together). Rotation and distortion of  $\Omega_2$  elements can be observed. Right: A detailed view of the overlapping zone. The continuity of the solution between both subdomains can be observed. . . . . 104

A.14.Solution of numerical example 2 on two domains that are partially overlapped and projected over the planes  $y = 0.5$  and  $y = 0.35$  of the function (left) and its derivative on the x-axis (right), in contrast with the reference solution. The continuity of both the solution and its derivative can be observed, in addition to a good agreement with the reference solution. . . . . 105

A.15.Convergence of the iterative solvers for numerical example 2 on an OD homogeneous mesh, a Chimera mesh composed of two completely overlapped domains, and an OD refined mesh. A relative residual tolerance of  $10^{-8}$  was used in all cases. Mesh sizes are depicted in Table A.3. . . . . 106

A.16.Examples of the meshes occupied for the current numerical test: a) OD homogeneous mesh with nearly constant element size. b) OD refined mesh with its refinement towards  $\Gamma_2$ . c) The elements in red of the OD refined mesh are rotated ( $\vec{\omega}$ ) and randomly perturbed ( $\vec{\varepsilon}$ ) to generate the  $\Omega_2$  mesh of the Chimera mesh. d) Chimera regular mesh. A non-structured homogeneous mesh is used for the background, and a structured mesh is progressively refined towards  $\Gamma_2$  for the patch. . . . . 107

A.17.Convergence of the solution (left) and its derivative (right) of the Laplace problem evaluated on OD homogeneous, OD refined, Chimera and Chimera regular meshes. . 108

A.18.Solution of numerical example 3, stated in Table A.4, solved on a Chimera regular mesh (left), and a detail of the overlapping zone (right). The continuity of the solution in the overlapped area can be observed. . . . . 108

A.19. Solution of the Laplace problem for the Chimera regular mesh, comformed by two domains completely overlapped, projected over the planes  $y = 0.5$  and  $y = 0.35$  of the function (left) and its derivative on the x-axis (right), in contrast to the reference solution. . . . . 109

A.20. Convergence history of the iterative solvers for the Laplace problem for an OD homogeneous mesh, an OD refined mesh, a Chimera mesh generated from the OD refined mesh and a Chimera regular mesh, where  $\Omega_1$  and  $\Omega_2$  are generated independently. The case corresponds to that presented in Table A.5. . . . . 109

A.21. Automatic topological detection of interpolation boundary on background three-dimensional mesh. A detail of the section of two three-dimensional grids completely overlapped,  $\Omega_1$  and  $\Omega_2$ , in conjunction with the interface boundary  $\partial\Omega_2$ , is considered. 111

A.22. Examples of the meshes occupied for the three-dimensional numerical test: a) OD homogeneous mesh with nearly constant element size. b) OD refined mesh with its refinement towards  $\Gamma_2$ . c) The elements in red of the OD refined mesh are randomly rotated ( $\vec{\omega}$ ) and perturbed ( $\vec{\varepsilon}$ ) to generate the  $\Omega_2$  mesh of the Chimera mesh. d) Chimera regular mesh. A non-structured homogeneous mesh is used for the background, and a structured mesh is progressively refined towards  $\Gamma_2$  for the patch. . . . . 112

A.23. Convergence of the solution (left) and its derivative (right) of the three-dimensional Laplace problem evaluated on OD homogeneous, OD refined, Chimera and Chimera regular meshes. . . . . 113

A.24. Solution of numerical example 4 solved on a three-dimensional Chimera regular mesh (left), and its derivative (right). The continuity of the solution and its derivative in the overlapped area can be noted. . . . . 113

A.25. Solution of the Laplace problem for the three-dimensional Chimera regular mesh projected on the intersections of the plane  $z = 0.5$  with the planes  $y = 0.5$  and  $y = 0.35$ . Both the solution of  $\phi$  (left) and its derivative on the x-axis (right) are shown in contrast to the reference solution. . . . . 114

A.26. Convergence history of the iterative solvers for the three-dimensional Laplace problem for an OD homogeneous mesh, an OD refined mesh, a Chimera mesh generated from the OD refined mesh and a Chimera regular mesh. The case corresponds to that presented in Table A.6. . . . . 114

B.1. Splitting of domain  $\Omega$  (a) into two subdomains  $\Omega_{1,2}$  with an overlapping region ( $\Omega_0$ ) (b). . . . . 128

B.2. Splitting of the nodes of  $\Omega_1$  and  $\Omega_2$  into groups according to their location. . . . . 130

B.3. High-order interpolation algorithm. . . . . 131

B.4. Topological interpolation boundary detection. First, in (a), the interface elements of  $\Omega_1$  (green elements) are recognized from the  $I^2$  nodes and the fictitious nodes. Then, in (b), overlapped and interface nodes of  $\Omega_1$  are distinguished. . . . . 133

B.5. Conceptual illustration of the meshes used for analysis. . . . . 135

B.6. Error and convergence analysis: (a) mean error and standard deviation of the  $L_2$ -norm for ten runs, (b) convergence histories of the iterative solvers for meshes with approximately 9700 degrees of freedom. . . . . 136

B.7. Optimization strategy using the Chimera method. . . . . 140

B.8. Conceptual representation of the background and patch meshes for the optimization procedure of the plate. . . . . 142

B.9. Convergence of the objective function obtained by the ALPSO and SLSQP optimization solvers for the numerical example. . . . . 144

B.10. Positions of holes through the optimization procedure. (a), (b) and (c) correspond to the three first iterations of ALPSO, and (d) corresponds to the final solution obtained by the SLSQP optimizer. . . . . 144

B.11. Speed-up of ALPSO optimizer with dynamic process management (DPM) (left). Speed-up of SLSQP optimizer using parallel gradient calculation (PGC) (right). . . . . 145

B.12. C3X Vane geometry. Configuration used as a reference. . . . . 146

B.13. Temperature measured by thermocouples and cubic interpolation over the pressure and suction surface for the FEM analysis. . . . . 148

B.14. Transformation of hole coordinate system. . . . . 150

B.15. Conceptual representation of the background and patch meshes used by the Chimera scheme. . . . . 151

B.16. Actual overlapping meshes used for the optimization procedure. The background mesh is formed by 48,960 linear unstructured triangular elements, while the patch mesh is composed of a total of 12,800 linear structured triangular elements (1,280 elements around each passage). . . . . 152

B.17. Positions of the passage centers through the optimization process. . . . . 152

B.18. Convergence of the objective function obtained by the ALPSO and SLSQP optimization solvers for the numerical example. . . . . 153

B.19. Comparison of the temperature field obtained by the FEM results. . . . . 153

B.20. Comparison of the temperature gradient field obtained by the FEM results. . . . . 154



# Índice de tablas

2.1. Información de la malla quimera y de la malla conforme uniforme con sus respectivos errores obtenidos en la norma $L_2$ para el Caso 1. . . . .	16
2.2. Información de la malla quimera y de la malla conforme uniforme con sus respectivos errores obtenidos en la norma $L_2$ para el Caso 2. . . . .	17
2.3. Información de la malla quimera y de la malla conforme uniforme con sus respectivos errores obtenidos en la norma $L_2$ para el Caso 3. . . . .	18
2.4. Error en la norma $L_2$ obtenido con una malla quimera y por una malla análoga conforme. . . . .	20
2.5. Análisis de convergencia de las soluciones iterativas. . . . .	21
3.1. Parámetros del término de fuente delta utilizado en el ejemplo numérico 2. . . . .	29
5.1. Propiedades del fluido refrigerante dentro de los conductos. . . . .	48
6.1. Resultados del estudio de convergencia obtenidos por el interpolador de alto orden en mallas tridimensionales. . . . .	56
6.2. Parámetros físicos del refrigerante que circula por los canales en el molde por inyección. . . . .	65
6.3. Cotas superior e inferior establecidas para las variables del problema de optimización . . . . .	68
A.1. Summary of the results obtained for one case of the mesh convergence study of Fig. A.6 . . . . .	100
A.2. Delta source term parameters for numerical example 2. . . . .	103
A.3. Summary of the results obtained for one case of the mesh convergence study of Fig. A.12 . . . . .	105
A.4. The Laplace problem stated in a one-domain mesh and on a Chimera mesh conformed by two subdomains. . . . .	106
A.5. Summary of the results obtained for one case of the mesh convergence study of Fig. A.20. . . . .	110

A.6. Summary of the results obtained for one case of the mesh convergence study of Fig. A.26. . . . .	115
B.1. Optimizer parameters. . . . .	143
B.2. Coolant and passage characteristics. . . . .	147
B.3. Grid independence study. . . . .	151



# Capítulo 1

## Introducción

### 1.1. Motivación y antecedentes

#### 1.1.1. Introducción al método quimera

La idea principal del método quimera (Benek et al., 1985; Steger y Benek, 1987; Steger, 1991) es generar mallas independientes para cada uno de los componentes presentes en un dominio computacional de un problema de la mecánica computacional, y vincularlos mediante alguna estrategia de acople con el objetivo de obtener una solución única del problema modelado. Una de las finalidades principales del método quimera (Steger, 1991) es evitar la dificultad que representa la generación de una malla conforme no estructurada para problemas que contienen cuerpos de formas intrincadas, como por ejemplo el ala de un avión o el álabe de una turbina eólica (Andrew M. et al., 2009). El método consiste en utilizar una malla gruesa de fondo cubriendo todo el dominio computacional, y mallas más finas alrededor de los objetos. De esta forma la tarea de mallado se reduce en gran medida, manteniendo la precisión de los resultados como así también el costo computacional. En la bibliografía se puede observar que la combinación del esquema quimera con un método ALE (*Arbitrary Lagrangean Eulerian*) es una buena opción para la resolución de problemas de Interacción Fluido-Estructura (FSI por *Fluid Structure Interaction*) (Peter y Wolfgang A., 2006). Las principales aplicaciones del método quimera pueden clasificarse de la siguiente forma:

**Simplificación en la generación de mallas:** en épocas cuando a los malladores les resultaba imposible mallar geometrías intrincadas, se propuso el método quimera como una alternativa en la cual diferentes mallas eran generadas alrededor de los componentes dentro del dominio computacional en forma independiente (Steger y Benek, 1987; Zheng y Liou, 2003).

**Refinamiento local:** cuando se requiere mayor precisión de la solución en algunas zonas específicas del dominio computacional. Dicho refinamiento simplemente se logra posicionando una malla más fina (malla *patch*) sobre la malla original (Kao et al., 1994; Meakin, 1995).

**Cuerpos móviles:** el método quimera permite tratar con problemas en los cuales existen cuerpos en movimiento. En estos casos las mallas independientes se mueven como cuerpos rígidos y la solución debe ser re-acoplada cada vez que se desplaza. Esto puede resultar computacionalmente muy costoso pero permite mantener la capa límite y el refinamiento local alrededor de los cuerpos en forma natural.

**Optimización:** Esta es una aplicación directa del método quimera donde diferentes objetos, cada uno con su respectiva malla, se mueve libremente sobre el dominio computacional sin la necesidad de remallar. A partir de esto, diferentes técnicas de optimización pueden ser utilizadas para encontrar la configuración óptima de los componentes respecto a una cierta función objetivo (Nielsen et al., 2010).

Las principales ventajas de método quimera para las últimas dos aplicaciones respecto a un remallado completo del dominio se basa en el hecho de que las mallas alrededor de los objetos se mueven como cuerpos rígidos de una configuración o paso de tiempo a otra/o. El acople solamente afecta las proximidades de las interfases y, si las interfases se encuentran lo suficientemente lejos del cuerpo, entonces el impacto de dicho acople es mínimo. Cuando se considera el método quimera, la malla es dividida en una malla de fondo (*background mesh*) cubriendo todo el dominio computacional, y las mallas superpuestas (*patch meshes*) que rodean los diferentes componentes u objetos que están ubicados sobre la malla de fondo. Lo que generalmente se realiza antes de resolver el sistema algebraico es un pre-proceso destinado a crear la interfase de la malla de fondo en función de la ubicación de las mallas superpuestas. Esto generalmente se logra quitando elementos de la malla de fondo que se encuentran en la región de solapamiento y así obtener las interfases entre la malla de fondo y las mallas superpuestas. A este proceso se lo denomina comunmente en la bibliografía como *hole cutting*.

### 1.1.2. El método quimera como un método de composición de dominios

El método quimera se encuentra dentro de los **Métodos de Composición de Dominios** (DCM por *Domain Composition Methods*). Los DCM consisten en obtener la solución a un problema a partir de las formulaciones del mismo problema expresadas en varios subdominios (ver Fig. 1.1). Estos métodos tienen el objetivo opuesto a los Métodos de Descomposición de Dominio (DDM por *Domain Decomposition Methods*), los cuales consisten en dividir una malla conforme existente en submallas o subdominios para luego aplicar una estrategia de paralelización de la solución (Eguzkitza et al., 2015; Houzeaux et al., 2017). Al estar ambos basados en la misma metodología (ya que después de descomponer el problema los DDM deben componer la solución), en la bibliografía muchas veces

se los confunde. Estos DCM pueden ser divididos en tres grandes grupos: los DCM basados en malla, los DCM basados en ecuación, y los DCM basados en el sistema algebraico.

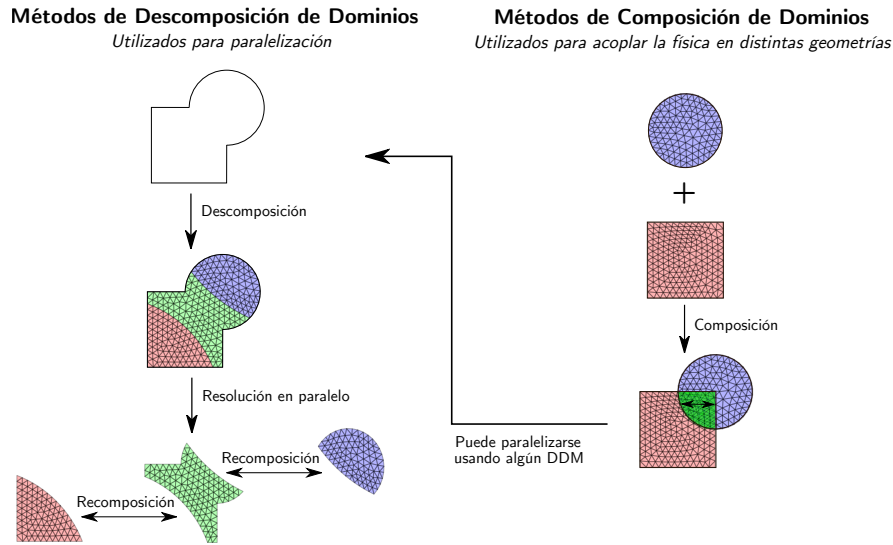


Figura 1.1: Métodos de descomposición de dominios vs métodos de composición de dominios <sup>1</sup>.

En los **DCM basados en malla** el acople entre subdominios se realiza a nivel de mallado, en la etapa de pre-proceso. Entre ellos se encuentra el método DRAGON (por *Direct Replacement of Arbitrary Grid Overlapping by Non-structured grid*, Liou y Kao, 1994), el cual genera una malla conforme final a partir de dos mallas estructuradas acopladas entre ellas mediante una malla no-estructurada. La malla no-estructurada de acople es generada después de realizar el *hole cutting*. Una estrategia más simple es el SSMUM (*Shear-Slip Mesh Update Method*, Kao et al., 1994) que se utiliza como una técnica de malla deslizante (*sliding-mesh*) para acoplar un dominio fijo a uno móvil de forma conforme. En este caso, los dominios se vinculan entre ellos mediante una capa de elementos, la cual se deforma si al menos uno de los dominios se encuentra en movimiento, pero si la deformación es muy grande el huelgo debe ser remallado. Otra alternativa es el *Mesh Matching* (Staten et al., 2010), cuyo objetivo es obtener una malla final conforme compuesta por hexahedros a partir de la generación de elementos hexaédricos entre las interfaces. Dentro de los métodos basados en manipulación de mallas se encuentra el HERMES (Eguzkitza et al., 2013; Eguzkitza Bazar, 2014) utilizado en mallas no conformes, el cual se basa en la creación de nuevas conectividades entre las mallas individuales pero sin la necesidad de agregar nodos en la interfase. El método se basa en la creación de nuevos elementos denominados como *elementos de extensión*, ya que su objetivo es extender un subdominio hacia el otro mediante la conexión de los nodos existentes en ambas mallas.

Por otro lado se encuentran los **DCM basados en ecuación**, los cuales pueden dividirse en tres grupos: los métodos de acople por restricción, los métodos libres de malla, y los métodos

<sup>1</sup>Imagen adaptada del trabajo de Houzeaux et al. (2017)

basados en la transmisión de condición. El primero de ellos acopla las soluciones obtenidas en los subdominios disjuntos restringiendo las ecuaciones de gobierno mediante ecuaciones de continuidad. Estas restricciones pueden ser impuestas por multiplicadores de Lagrange, o bien con métodos de penalización. Entre los más conocidos pueden distinguirse el método *mortar* (Belgacem y Maday, 1997; Cai et al., 1999), y el método FETI (*Finite Element Tearing and Interconnect*, Farhat y Roux, 1991; Stefanica y Klawonn, 1999). Otra opción utilizada en la bibliografía es la de los métodos libres de malla (o *mesh free*), los cuales se basan en definir conexiones de malla libre (*mesh-free connections*) en la interfase generada entre las mallas utilizando nuevos elementos denominados *meshless gluing elements*. El principal inconveniente de tales métodos es la determinación de la zona de influencia de los elementos de la interfase, ya que las funciones de forma de estos elementos no mantienen las mismas propiedades que las funciones de forma del FEM (por *Finite Element Method*) clásico. En el último método, el acople es logrado mediante la transmisión de las condiciones (en el contexto de FEM), o bien imponiendo la continuidad de los flujos (en el contexto de volúmenes finitos). En el contexto de elementos finitos, la transferencia de condiciones es la que permite la continuidad en las variables y en los flujos a través de las interfases. Estas condiciones pueden ser del tipo Dirichlet, o del tipo Neumann/Robin. Información detallada sobre estas estrategias puede encontrarse en Quarteroni y Valli (1999). Uno de los principales inconvenientes de los métodos que utilizan condiciones de borde de tipo Neumann o Robin, es una falta de equivalencia con la formulación de un solo dominio, y un posible deterioro de la convergencia en malla cuando las derivadas no son calculadas en forma correcta. Además, tales métodos no son lo suficientemente generales en la condición de borde de tipo Neumann, cuando la misma involucra coeficientes físicos que deben actualizarse paso a paso, por lo que además hay que escribir un *script* ad-hoc para calcularla en cada problema físico a acoplar.

Por último se encuentran los **DCM basados en el sistema algebraico**, en los cuales se basa esta tesis, que tratan la composición de la solución directamente en la matriz y en el lado derecho de la ecuación de los sistemas algebraicos individuales (Houzeaux et al., 2015). El hecho de trabajar el acople entre subdominios a nivel algebraico presenta dos principales ventajas: la primera es que la solución obtenida a partir de la formulación expresada en múltiples subdominios es equivalente a la obtenida en un solo dominio, y la segunda es que al plantear el acople a nivel algebraico de los sistemas permite ocultar la transferencia de información entre los subdominios dentro de las operaciones del *solver* iterativo, evitando posibles problemas de convergencia.

En esta tesis se propone un esquema quimera en el contexto del Método de Elementos Finitos con un acople de tipo Dirichlet/Dirichlet para mallas no estructuradas, no coincidentes, parcialmente o completamente solapadas, tanto en dos y tres dimensiones espaciales (2D y 3D, respectivamente).

La malla quimera se compondrá por dos o más mallas tradicionales solapadas con diferentes niveles de refinamiento, las cuales intercambian información entre ellas mediante fronteras de interpolación definidas en cada subdominio, y mediante la utilización de un algoritmo de interpolación de alto orden. La resultante del acople entre subdominios conlleva a que la matriz del sistema global pierda la simetría, sin embargo la solución puede ser obtenida mediante el uso del *solver* iterativo BiCGStab (*BiConjugate Gradient Stabilized*). De esta manera los requerimientos en cuanto a memoria y esfuerzos de cómputo para resolver el sistema de ecuaciones resultan similares a los requeridos para la resolución de un problema equivalente de un solo dominio, lo cual puede ser realizado en forma eficiente con el *solver* iterativo CG (*Conjugated Gradient*). También el acople puede ser planteado de manera *matrix-free*, en el cual se evita almacenar la matriz completa del sistema acoplado.

En el desarrollo de esta tesis también se propone un nuevo enfoque para abordar problemas ingenieriles de optimización de transferencia de calor mediante la utilización del esquema quimera, en conjunto con optimizadores basados en población, y los basados en gradientes. Adicionalmente se hace uso de plataformas de computación de alto desempeño (HPC por *High Performance Computing*) para aminorar los tiempos de cómputo requeridos durante el proceso de optimización.

## 1.2. Objetivos

### 1.2.1. Objetivo general

Desarrollo e implementación de un método de composición de dominios tipo quimera orientado a la solución numérica de problemas ingenieriles de optimización con cuerpos móviles, aprovechando cualidades inherentes del método tales como la simplificación de la generación de mallas y un refinamiento adaptativo local.

### 1.2.2. Objetivos particulares

Como objetivos particulares dentro del alcance de esta tesis se proponen:

- **Implementar el MC para mallas estructuradas parcialmente solapadas.** Desarrollar un algoritmo de acople orientado al MC con mallas estructuradas parcialmente solapadas mediante la utilización de `Octave` (Eaton et al., 2015), `PETSc-FEM` (Storti et al., 2007), y las librerías `Eigen` (Guennebaud et al., 2010) en 2D. Para el acople se desarrollará un interpolador de alto orden tipo Dirichlet-Dirichlet.
- **Ampliar la implementación del MC para la aplicación en mallas no estructuradas, parcialmente y completamente solapadas para múltiples subdominios.** Esto permi-

tirá utilizar el método en geometrías de formas más intrincadas y establecer las bases para hacer uso del esquema propuesto en problemas de optimización.

- **Acoplamiento del MC con algoritmos de optimización.** Estudio y utilización del optimizador de código libre `pyOpt` (Perez et al., 2012) para la solución numérica de problemas no-lineales con restricciones. A continuación, solución numérica de problemas de transferencia térmica mediante la utilización del método quimera , tanto con el optimizador basado en población ALPSO (Sedlaczek y Eberhard, 2006), como con el optimizador basado en gradiente SLSQP (Kraft y Schnepper, 1989).
- **Paralelización del esquema de optimización.** Para disminuir los tiempos de cómputo y así poder abordar problemas de ingeniería de mayor tamaño y en cantidad de variables, se extenderá la implementación del MC para ser utilizado junto con códigos de optimización mediante cómputo en paralelo. Para dicho fin, se utilizará la librería `mpi4py` (Dalcin et al., 2011) de manera que se pueda ejecutar el MC acoplado a los optimizadores en el cluster `Seshat` (Seshat, 2018) que se encuentran en las instalaciones de CIMEC.
- **Implementación del MC a problemas en dominios tridimensionales (3D).** Ampliación del esquema de acople con subdominios tridimensionales (3D). Desarrollo e implementación de un algoritmo de interpolación de alto orden, como también de un algoritmo de detección de interfases para mallas con dominios en tres dimensiones (3D). Finalmente, aplicación de la propuesta en problemas ingenieriles.

### 1.3. Estructura de la Tesis

La presente tesis se encuentra estructurada bajo el formato de *Tesis por compilación*. En base a esto, en el cuerpo principal de la Tesis se presenta una descripción global de la temática a ser abordada, mientras que información más detallada y precisa son referenciadas a los trabajos publicados, adjuntos en los Anexos. En ellos se podrán encontrar los desarrollos llevados a cabo para abordar cada problema, y así justificar los resultados presentados en esta tesis. Es importante aclarar que, si bien algunos resultados de problemas adicionales planteados en el cuerpo principal de la presente tesis son complementarios a los resultados obtenidos en los Anexos, no obstante, son considerados importantes para un desarrollo ordenado y completo de la exposición. De manera análoga, tanto las formulaciones como las imágenes de los Anexos pueden encontrarse repetidas en el cuerpo principal de la tesis con el fin de priorizar lo anteriormente enunciado.

En el Capítulo 2 se introduce el acople algebraico propuesto del esquema quimera y su solución numérica en forma iterativa. Se estudian algunos test numéricos sobre mallas estructuradas parcial-

mente solapadas en 2D evaluando error, convergencia en malla y convergencia del *solver* iterativo. Este Capítulo está parcialmente vinculado al Anexo A.

En el Capítulo 3 se presenta el interpolador de alto orden propuesto utilizado para el acople entre subdominios, y se introduce un algoritmo para la detección de interfases de interpolación para dominios completamente solapados. Se estudian algunos casos numéricos sobre mallas no estructuradas parcialmente y completamente solapadas evaluando error, convergencia en malla y convergencia del *solver* iterativo. Este Capítulo está íntegramente vinculado al Anexo A.

En el Capítulo 4 se introduce la plataforma de optimización de pyOpt junto a los optimizadores utilizados en esta tesis. Se presenta también una extensión del MC para abordar múltiples dominios. Se muestra la solución de un problema simple de optimización de transferencia de calor mediante cómputo en paralelo, donde se evalúa el *speedup*, la convergencia, y el desempeño de los optimizadores utilizados. Este Capítulo está parcialmente vinculado al Anexo B.

En el Capítulo 5 se presenta una aplicación directa del MC aplicado a un problema de ingeniería con formas intrincadas, en el cual se optimiza la configuración del sistema de refrigeración del álabe de una turbina de gas para mejorar su desempeño bajo cargas térmicas. Este Capítulo está íntegramente vinculado al Anexo B.

En el Capítulo 6 se presenta la ampliación del MC para tratar mallas solapadas en dominios tridimensionales (3D). Se resuelve un test numérico de optimización de transferencia de calor. Por último se resuelve un problema ingenieril tridimensional (3D) para optimizar la disposición de los canales de refrigeración de un molde de inyección. Este Capítulo está parcialmente vinculado al Anexo A.

Finalmente, en el Capítulo 7 se exponen las conclusiones obtenidas a partir de los resultados logrados.

## 1.4. Publicaciones

A continuación se listan las publicaciones que ha realizado el autor de la presente tesis, incluyendo solamente aquellas que derivaron directamente de la realización de la misma.

### 1.4.1. Publicaciones en revistas indizadas

- Storti B., Garelli L., Storti M., D'Elía J. *A matrix-free Chimera approach based on Dirichlet-Dirichlet coupling for domain composition purposes*. En *Computers and Mathematics with Applications*, 2020. doi.org/10.1016/j.camwa.2020.01.021.
- Storti B., Garelli L., Storti M., D'Elía J. *Numerical optimization of a blade cooling passa-*

*ges configuration using a Chimera approach and parallel computing.* En *Finite Elements in Analysis and Design*, 2020. doi.org/10.1016/j.finel.2020.103423.

#### 1.4.2. Publicaciones y presentaciones en congresos

- B. Storti, L. Garelli, M. Storti, J. D'Elía. *A 3D implementation of a Chimera scheme applied to heat transfer optimization problems.* En el XXIV Congreso sobre Métodos Numéricos y sus Aplicaciones (ENIEF 2019), Santa Fe, Argentina.
- B. Storti, L. Garelli, M. Storti, J. D'Elía. Optimización del sistema de refrigeración de un álabe utilizando técnicas de mallas solapadas y computación en paralelo. En *Mecánica Computacional Vol. XXXVI*, págs. 1971-1991, actas del XII Congreso Argentino de Mecánica Computacional (MECOM).
- L. Garelli, B. Storti, M. Storti, J. D'Elía. *A Chimera Method Based on Dirichlet-Dirichlet Coupling Applied to Moving Boundary Problems.* En el XIII Congreso Mundial en Mecánica Computacional (WCCM) / II Congreso Panamericano en Mecánica Computacional (PANACM). Nueva York, Estados Unidos, 2018.
- B. Storti, L. Garelli, M. Storti, J. D'Elía. *A Coupling Strategy for a Chimera Method Applied to Thermal Conduction Optimization Problems.* En el XXIII Congreso de Métodos Numéricos y sus Aplicaciones (ENIEF), Córdoba, Argentina, 2017.
- B. Storti, L. Garelli, M. Storti, J. D'Elía. *A Chimera Method Based On Dirichlet-Dirichlet Coupling and Pasting Penalization Operators.* En *Mecánica Computacional*, vol. XXXIV, págs. 3161-3175, actas del XXII Congreso de Métodos Numéricos y sus Aplicaciones (ENIEF) 2016)



## Capítulo 2

# El método quimera en mallas estructuradas parcialmente solapadas

### 2.1. Introducción

El objetivo de este Capítulo es presentar un esquema de acople algebraico tipo quimera para mallas estructuradas parcialmente solapadas no coincidentes en el contexto del método de elementos finitos donde cada malla posee distintos niveles de discretización. La metodología está basada en la transferencia de la solución numérica de una malla hacia el borde de la otra mediante condiciones de borde de tipo Dirichlet, y en la utilización de operadores de interpolación. La principal ventaja de la presente metodología es que la topología de las mallas no cambia durante la solución de los sistemas lineales resultantes. Otra ventaja es que la solución al sistema acoplado entre ambos subdominios puede ser hecha en forma iterativa con una tasa de convergencia similar a la que se obtendría de la solución del mismo problema pero planteado en un solo dominio. Resultados del error en la norma  $L_2$  de la solución global obtenida, convergencia en malla del acople y convergencia de solver iterativo son presentados en el presente Capítulo.

### 2.2. Descripción del método quimera

Con el fin de simplificar la presentación del método, vamos a considerar dos subdominios  $\Omega_1$  y  $\Omega_2$  en 2D, con una pequeña región solapada entre ellos  $\Omega_O$  (ver Fig. 2.1a). Entonces sea el dominio  $\Omega = \Omega_1 \cup \Omega_2$ , donde queremos resolver la ecuación de Poisson dada por:

$$\begin{aligned}\Delta\phi &= -f && \text{en } \Omega; \\ \phi &= \bar{\phi} && \text{en } \Gamma = \partial\Omega\end{aligned}\tag{2.1}$$

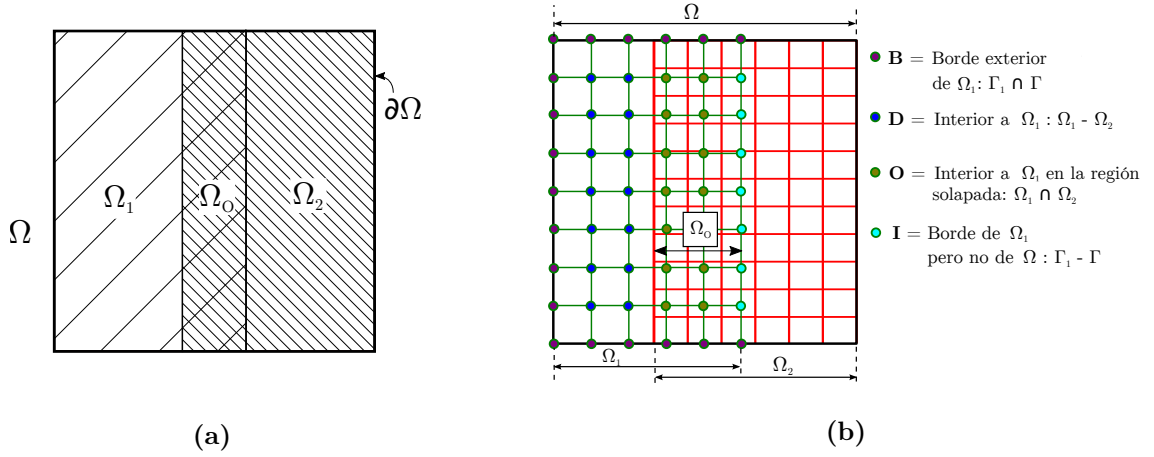


Figura 2.1: Descripción del método quimera .

Se considera una malla para cada subdominio pero que no sean coincidentes. Ahora vamos a considerar la solución del problema en el dominio  $\Omega_1$ . Una discretización estándar por elementos finitos nos lleva a la formulación matricial

$$\mathbf{Ax} = \mathbf{b} \quad (2.2)$$

Por otra parte, los nodos de la malla pueden ser divididos en cuatro grupos (ver Fig. 2.1b):

- El grupo **B** que contiene los nodos del borde del dominio  $\Omega_1$  y que también pertenecen a la frontera de  $\Omega$ .
- El grupo **D** compuesto por los nodos interiores a  $\Omega_1$  que no se encuentran en la región solapada,  $\Omega$ .
- El grupo **O** también compuesto por los nodos interiores a  $\Omega_1$  pero solo aquellos que se encuentran sobre la región solapada.
- El grupo **I** el cual contiene los nodos en el borde de  $\Omega_1$  pero que no están contenidos en la frontera de  $\Omega$ . Estos nodos deben estar contenidos en el subdominio  $\Omega_2$ .

A partir de esta clasificación podemos agrupar los nodos internos de los grupos **O** y **D** en un solo grupo que llamaremos **Z**, el cual contiene todos los nodos interiores.

Luego, al dividir los nodos en grupos, el sistema lineal queda separado de la siguiente forma:

$$\begin{bmatrix} \mathbf{A}_{BB} & \mathbf{A}_{BZ} & \mathbf{A}_{BI} \\ \mathbf{A}_{ZB} & \mathbf{A}_{ZZ} & \mathbf{A}_{ZI} \\ \mathbf{A}_{IB} & \mathbf{A}_{IZ} & \mathbf{A}_{II} \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_Z \\ \mathbf{x}_I \end{bmatrix} = \begin{bmatrix} \mathbf{b}_B \\ \mathbf{b}_Z \\ \mathbf{b}_I \end{bmatrix} \quad (2.3)$$

Del sistema planteado, los valores de  $\mathbf{x}_B$  son conocidos debido a las condiciones de borde establecidas. Si conociéramos los valores en los nodos de la interfase  $\mathbf{x}_I$ , entonces tendríamos un problema definido con condiciones de borde de tipo Dirichlet en toda la frontera, pudiendo encontrar la solución al

sistema resolviendo:

$$\mathbf{A}_{ZZ}\mathbf{x}_Z = \mathbf{b}_Z - \mathbf{A}_{ZB}\mathbf{x}_B + \mathbf{A}_{ZI}\mathbf{x}_I \quad (2.4)$$

Pero como los valores en los nodos de la interfase  $\mathbf{x}_I$  no son conocidos, debemos interpolarlos de los valores en los nodos interiores al dominio  $\Omega_2$ . De esta forma se obtiene un sistema lineal completo en el interior de ambos dominios, como se muestra en la Ec. (2.5). Desde este momento usaremos los supra-índices para referirnos a cada subdominio.

$$\begin{aligned} \mathbf{A}_{ZZ}^1\mathbf{x}_Z^1 &= \mathbf{b}_Z^1 - \mathbf{A}_{ZB}^1\mathbf{x}_B + \mathbf{A}_{ZI}^1\Pi_{I_{1,2}}\mathbf{x}^2, \\ \mathbf{A}_{ZZ}^2\mathbf{x}_Z^2 &= \mathbf{b}_Z^2 - \mathbf{A}_{ZB}^2\mathbf{x}_B + \mathbf{A}_{ZI}^2\Pi_{I_{2,1}}\mathbf{x}^1, \end{aligned} \quad (2.5)$$

De la Ec. (2.5), el operador  $\Pi_{I_{1,2}}$  es un operador de proyección que interpola valores de un vector  $\mathbf{x}^2$  a los nodos de la interfase  $I^1$ . Estos operadores realizan la proyección de una malla hacia el borde de la otra mediante interpolaciones de alto orden. Si bien estos operadores de interpolación de alto orden son utilizados en los ejemplos del presente Capítulo, el algoritmo para llevar a cabo dicha interpolación se describe en forma detallada en el Capítulo 3 para un mayor orden de la tesis. Se debe tener en cuenta que, para poder realizar dicha interpolación, debe haber al menos una capa de elementos en la región solapada. De esta forma la interpolación involucra solamente los nodos de los grupos  $B^2$  y  $Z^2$ , y no los del grupo  $I^2$ . De esta forma el sistema lineal de la Ec. (2.5) puede ser resuelto para los nodos interiores  $\mathbf{x}_Z^1$  y  $\mathbf{x}_Z^2$ .

### 2.2.1. Algoritmo de acople matrix-free

El esquema de la Sección 2.2 puede plantearse como un operador lineal que puede ser utilizado por un *solver* iterativo del tipo *matrix-free* como ser el *Generalized Minimal RESidual* (GMRES), o por el *BiConjugate Gradient Stabilized method* (BiCGStab). El sistema que queda planteado no es ni simétrico ni definido positivo, a pesar de que el problema original sí lo es. Sin embargo experiencias numéricas, que serán presentadas en el transcurso de esta tesis, muestran que puede ser resuelto con BiCGStab con tasas de convergencia similares a las que se pueden obtener para un problema de malla conforme (sin solapamiento) resuelto con Gradiente Conjugado (CG). La Ec. (2.5) puede ser re-escrita en forma de residuos como:

$$\begin{aligned} \mathbf{r}_Z^1(\mathbf{x}_Z^1, \mathbf{x}_Z^2) &= \mathbf{A}_{ZZ}^1\mathbf{x}_Z^1 - \mathbf{b}_Z^1 + \mathbf{A}_{ZB}^1\mathbf{x}_B - \mathbf{A}_{ZI}^1\Pi_{I_{1,2}}\mathbf{x}^2, \\ \mathbf{r}_Z^2(\mathbf{x}_Z^1, \mathbf{x}_Z^2) &= \mathbf{A}_{ZZ}^2\mathbf{x}_Z^2 - \mathbf{b}_Z^2 + \mathbf{A}_{ZB}^2\mathbf{x}_B - \mathbf{A}_{ZI}^2\Pi_{I_{2,1}}\mathbf{x}^1. \end{aligned} \quad (2.6)$$

El algoritmo para calcular los residuos con los valores interiores de ambos subdominios se muestra a continuación.

---

**Algoritmo 1:** Cálculo de residuos

---

**Entrada:**  $\mathbf{x}$ **Salida** :  $\mathbf{r} = f(\mathbf{x})$ 

- 1: Extraer  $\mathbf{x}_Z^1, \mathbf{x}_Z^2$  de  $\mathbf{x} = [\mathbf{x}_Z^1; \mathbf{x}_Z^2]$
  - 2: Interpolar  $\mathbf{x}_Z^2$  en los nodos  $I^1$
  - 3: Calcular el residuo  $\mathbf{r}_Z^1$  en  $Z^1$  de acuerdo a la Ec. (2.6) (línea 1).
  - 4: Interpolar  $\mathbf{x}_Z^1$  en los nodos  $I^2$
  - 5: Calcular el residuo  $\mathbf{r}_Z^2$  en  $Z^2$  de acuerdo a la Ec. (2.6) (línea 2).
  - 6: Combinar  $\mathbf{r} = [\mathbf{r}_Z^1; \mathbf{r}_Z^2]$
- 

Se puede decir que el esquema planteado para resolver el sistema lineal es *matrix-free* ya que, al observar los puntos 3 y 5 del Algoritmo 1, sólo se necesitan ensamblar las matrices de los sistemas individuales de cada subdominio, evitando así el ensamblado del dominio completo en una sola matriz. Mayores detalles acerca de la implementación *matrix-free* del algoritmo son detallados en el Anexo A.

**2.2.2. Algoritmo BiCGStab**

El algoritmo BiCGStab es un método iterativo basado en los subespacios de Krylov, y que es utilizado para la solución de sistemas lineales no simétricos (Van der Vorst, 1992). Dicho método es una variante del algoritmo de gradientes biconjugados (BiCG), y se caracteriza por lograr una mejor tasa de convergencia que el algoritmo original de BiCG u otros algoritmos similares como, por ejemplo, el de gradientes conjugados cuadrático (CGS por *Conjugated Gradient Squared*). En el Algoritmo 2 se detallan los pasos a seguir en el algoritmo de BiCGStab (Kelley, 1995).

Al analizar este algoritmo se puede observar que la matriz  $A$  no necesita ser formada ni almacenada, es decir, no se requiere almacenar sus coeficientes en forma explícita sino que se puede obtener la matriz mediante operaciones de productos matriz-vector. A ciertas implementaciones de los métodos basados en el espacio de Krylov se las llama *matrix-free* por esta razón. Este enfoque es muy conveniente cuando la matriz del sistema es muy grande, ya que manipularla o almacenarla podría requerir mucha memoria y tiempo de cómputo. Entonces, basta con implementar el Algoritmo 1 en una función que compute los residuos de ambos subdominios y ser pasada como argumento al algoritmo de BiCGStab para que el sistema quede planteado de manera *matrix-free*.

En cuanto al costo computacional por iteración del algoritmo BiCGStab respecto a CG, tal y como se explica en el Anexo A, es mayor debido a que BiCGStab tiene que almacenar tres vectores adicionales que CG. Adicionalmente, por cada iteración, BiCGStab tiene que realizar un producto

**Algoritmo 2:**  $\text{bicgstab}(x, b, A, \epsilon, kmax)$ 

- 
1.  $r = b - Ax$ ,  $\hat{r}_0 = \hat{r} = r$ ,  $\rho_0 = \alpha = \omega = 1$ ,  $v = p = 0$ ,  $k = 0$ ,  $\rho_1 = \hat{r}_0^T r$
  2. Mientras  $\|r\|_2 > \epsilon \|b\|_2$  y  $k < kmax$  ejecutar:
    - a)  $k = k + 1$
    - b)  $\beta = (\rho_k / \rho_{k-1})(\alpha / \omega)$
    - c)  $p = r + \beta(p - \omega v)$
    - d)  $v = Ap$
    - e)  $\alpha = \rho_k / (\hat{r}_0^T v)$
    - f)  $s = r - \alpha v$ ,  $t = As$
    - g)  $\omega = t^T s / \|t\|_2^2$ ,  $\rho_{k+1} = -\omega \hat{r}_0^T t$
    - h)  $x = x + \alpha p + \omega s$
    - i)  $r = s - \omega t$
- 

matriz-vector y dos productos escalares más que CG. Entonces, a medida que el problema crece en tamaño, las operaciones de producto matriz-vector predominan en el problema. Por lo que, si el problema es lo suficientemente grande, entonces el costo computacional por iteración de BiCGStab puede llegar a ser el doble al de una iteración de CG. Esto es importante de remarcar ya que, si bien la estrategia de acoplar dos o más subdominios para obtener una solución global del sistema implica un costo adicional de cómputo, este se puede estimar con antelación para que el mismo no resulte excesivamente superior.

### 2.2.3. Estimación del error producido por el método quimera

Como resultado de la descomposición del dominio  $\Omega$  en dos subdominios,  $\Omega_1$  y  $\Omega_2$ , cada subdominio tendrá su respectivo error en la norma  $L_2$  de la solución debido al distinto nivel de discretización en cada uno de ellos. A modo de poder comparar el error producido por el método respecto al producido por mallas conformes equivalentes, definimos a continuación un error resultante de la malla quimera que se compondrá de la suma del error en ambos subdominios. Para esto, primero definimos el error en  $\Omega_1$  que vendrá dado por:

$$\|u_{\text{ref}} - u_{h_1}\|_{(L_2, \Omega_1)} = \sqrt{\sum_{j=1}^{N_j^1} \frac{1}{N_i^1} \left\{ \sum_{i=1}^{N_i^1} (u_{\text{ref}}(i) - u_{h_1}(i))^2 \right\}} w_{h_1}(j) \quad (2.7)$$

donde  $u_{h_1}$  representa la solución obtenida del acople en  $\Omega_1$ ,  $u_{\text{ref}}$  es la solución utilizada de referencia, obtenida en una malla uniforme y estructurada compuesta por  $2048^2$  elementos e interpolada en  $\Omega_1$ ,  $N_j^1$  representa el número total de elementos de  $\Omega_1$ ,  $N_i^1$  representa el número total de nodos del

elemento  $j$ , mientras que  $w_{h_1}$  es el área del elemento  $j$ . De la misma manera se puede definir el error en el subdominio  $\Omega_2$  como:

$$\|u_{\text{ref}} - u_{h_2}\|_{(L^2, \Omega_2)} = \sqrt{\sum_{j=1}^{N_j^2} \frac{1}{N_i^2} \left\{ \sum_{i=1}^{N_i^2} (u_{\text{ref}}(i) - u_{h_2}(i))^2 \right\} w_{h_2}(j)} \quad (2.8)$$

donde  $u_{h_2}$  representa la solución obtenida del acople en  $\Omega_2$ , mientras que  $u_{\text{ref}}$  es la misma solución de la malla compuesta por  $2048^2$  e interpolada en  $\Omega_2$ . Con el objetivo de comparar los resultados obtenidos por el método quimera en múltiples subdominios en contraste con los resultados que uno obtendría con una malla conforme uniforme y estructurada, se define el error en  $\Omega$  para la solución con el método quimera como:

$$\|u_{\text{ref}} - u_h\|_{(L^2, \Omega)} = \sqrt{\|u_{\text{ref}} - u_{h_1}\|_{(L^2, \Omega_1)}^2 + \|u_{\text{ref}} - u_{h_2}\|_{(L^2, \Omega_2)}^2} \quad (2.9)$$

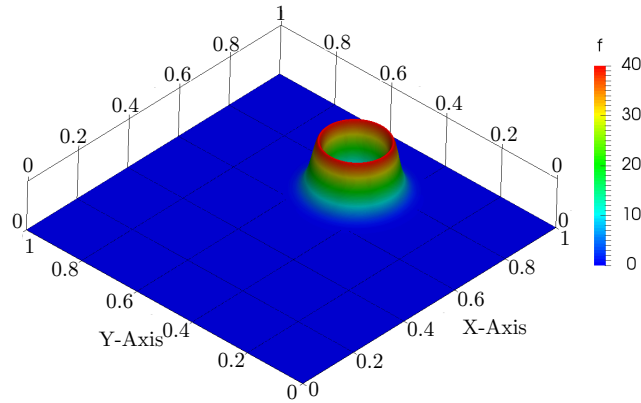
Debido a que en la zona de solapamiento el error se encontraría doblemente computado, para solventar esto se tiene en cuenta el error computado sólo sobre la malla más fina en dicha región.

### 2.3. Ejemplos numéricos

Para evaluar el comportamiento del acople en **mallas estructuradas parcialmente solapadas** se considera la solución al problema de Poisson en dos dimensiones con el siguiente término fuente:

$$\begin{aligned} f &= \frac{\delta}{\cosh((r - 0.2)/\delta)^2}, \\ r &= \sqrt{(x - x_c)^2 + (y - y_c)^2}, \\ \delta &= 0.025; \end{aligned} \quad (2.10)$$

Esto representa un término fuente concentrado en el círculo  $r = 0.2$ , con su centro posicionado en



**Figura 2.2:** Visualización del término fuente expresado en la Ec. (2.10), con  $x_c = 0.75$  y  $y_c = 0.5$ .

$(x_c, y_c)$ . El interés de este caso es en la gran curvatura de la solución debida a un término fuente altamente concentrado.

El problema es resuelto en dos dominios, cada uno con una extensión de 0.525 en el eje  $x$ , es decir:

$$\begin{aligned}\Omega_1 &= [0, 0.525] \times [0, 1], \\ \Omega_2 &= [0.475, 1] \times [0, 1],\end{aligned}\tag{2.11}$$

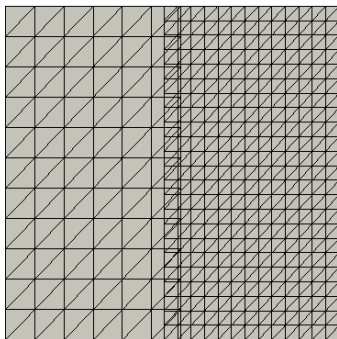
La región solapada queda definida entonces como:

$$\Omega_1 \cap \Omega_2 = [0.475, 0.525] \times [0, 1]$$

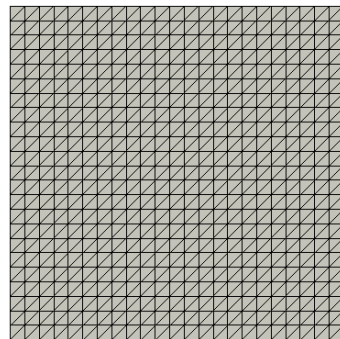
Las mallas son cartesianas homogéneas en ambos dominios, con  $N_{1y} = 11$  elementos en la dirección  $y$  para  $\Omega_1$ , y con  $N_{2y} = 23$  para  $\Omega_2$ . De esta manera las mallas resultantes no son coincidentes. El tamaño de las mallas en la dirección  $x$  son escogidos de tal forma que la relación de aspecto de los elementos permanezca constante. Luego las mallas son refinadas en pasos proporcionales a  $\sqrt{2}$ , es decir, la malla en el paso de refinamiento  $n + 1$  es proporcional a  $1/\sqrt{2}$  respecto a la malla en el paso  $n$ .

### 2.3.1. Comparación del método chimera con una malla estructurada uniforme

Para los ejemplos numéricos se consideran tres casos. En el Caso 1 se tiene el centro de la fuente dispuesto sobre  $\Omega_2$ , es decir, en la malla fina. En el Caso 2, la fuente se encuentra dispuesta en el centro del dominio computacional, o sea, sobre la región de solapamiento. Mientras que en el Caso 3, la fuente se encuentra posicionada sobre la malla más gruesa. Para cada ejemplo se analiza la convergencia de la metodología propuesta resolviendo el problema en mallas solapadas, en contraste con un esquema típico de FEM resuelto en un solo dominio (ver Fig. 2.3).



(a) Malla Chimera.



(b) Malla uniforme estructurada.

**Figura 2.3:** Mallas utilizadas para el análisis del error en la norma  $L_2$ .

### 2.3.1.1. Caso 1: fuente contenida en la región de la malla más fina

En este caso se resuelve la ecuación de Poisson (Ec. (2.1)) para un término fuente singular (Ec. (2.10)) con su centro localizado en  $x_c = 0.75$  y  $y_c = 0.5$ , donde la región total solapada es de 0.05. El interés de este test numérico es evaluar si el error en la norma  $L_2$  obtenido en la malla quimera se asemeja a aquel obtenido por una malla uniforme cuyo tamaño de elemento es igual al tamaño de elemento de  $\Omega_2$  de las mallas solapadas o algún error es introducido durante el acople. En la Tabla 2.1 pueden visualizarse los datos de las mallas obtenidos para 8 niveles de refinamiento, como así también el valor del error en la norma  $L_2$  para cada caso.

ref	Chimera					Uniforme			
	Malla			Error $L_2$		Malla			Error $L_2$
	$[N_{1y}, N_{2y}]$	$[h_{1y}, h_{2y}]$	Elem	$[\Omega_1, \Omega_2]$	$\Omega$	$N_{x,y}$	$h$	Elem	$\Omega$
1.00	[11 , 23]	[9.09e-02 , 4.34e-02]	730	[1.51e-02 , 1.21e-02]	2.44e-02	23	4.34e-02	1058	7.50e-03
1.41	[16 , 33]	[6.25e-02 , 3.03e-02]	1476	[1.41e-03 , 3.27e-03]	3.71e-03	33	3.03e-02	2178	1.35e-03
2.00	[23 , 47]	[4.34e-02 , 2.12e-02]	2948	[6.29e-05 , 3.59e-04]	3.68e-04	47	2.12e-02	4418	5.96e-04
2.82	[32 , 66]	[3.12e-02 , 1.51e-02]	5708	[5.40e-05 , 1.42e-04]	1.53e-04	66	1.51e-02	8712	1.51e-04
4.00	[45 , 93]	[2.22e-02 , 1.07e-02]	11274	[1.36e-05 , 7.26e-05]	7.41e-05	93	1.07e-02	17298	7.43e-05
5.65	[63 , 131]	[1.58e-02 , 7.63e-03]	22362	[7.48e-06 , 3.65e-05]	3.74e-05	131	7.63e-03	34322	3.75e-05
8.00	[89 , 185]	[1.12e-02 , 5.40e-03]	44626	[3.87e-06 , 1.81e-05]	1.86e-05	185	5.40e-03	68450	1.87e-05
11.31	[125 , 261]	[8.00e-03 , 3.83e-03]	88536	[2.01e-06 , 9.08e-06]	9.33e-06	261	3.83e-03	136242	9.39e-06

**Tabla 2.1:** Información de la malla quimera y de la malla conforme uniforme con sus respectivos errores obtenidos en la norma  $L_2$  para el Caso 1.

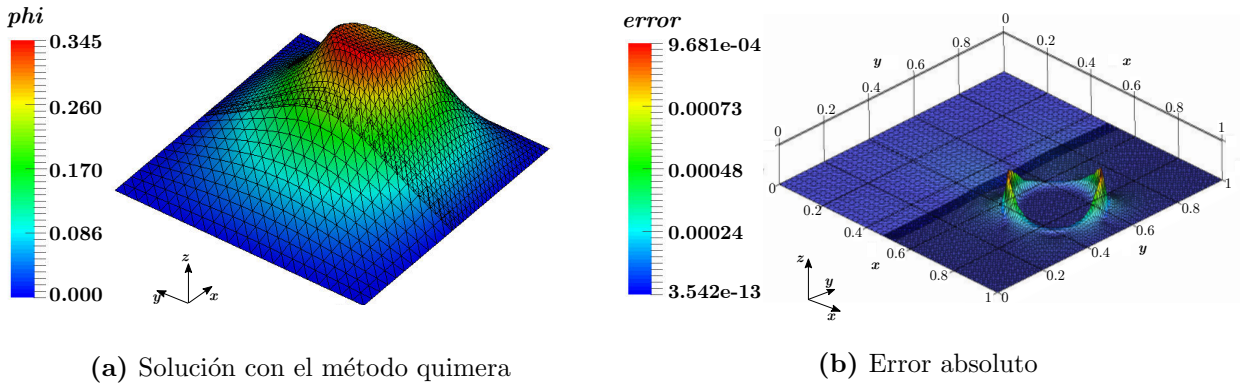
De los resultados puede observarse que, cuando los mayores gradientes de la solución se encuentran en la malla fina, a partir del tercer nivel de refinamiento, el error obtenido por el método quimera es muy cercano al obtenido utilizando una malla uniforme estructurada cuyos elementos tienen el tamaño de la malla fina de las mallas solapadas. Puede concluirse también que el esquema mantiene el orden de convergencia  $O(h^2)$ .

Tanto la solución del campo  $\phi$  como el error absoluto obtenido en las mallas solapadas para el Caso 1 se muestran en la Fig. 2.4. Se puede observar claramente de la figura la continuidad lograda de la solución en la región de solapamiento.

### 2.3.1.2. Caso 2: término fuente posicionado sobre la zona de solapamiento

En el presente test numérico se resuelve la Ec. (2.1) con el término fuente expresado por la Ec. (2.10), cuyo centro se encuentra posicionado en  $x_c = 0.5$  e  $y_c = 0.50$ , sobre la región solapada. El





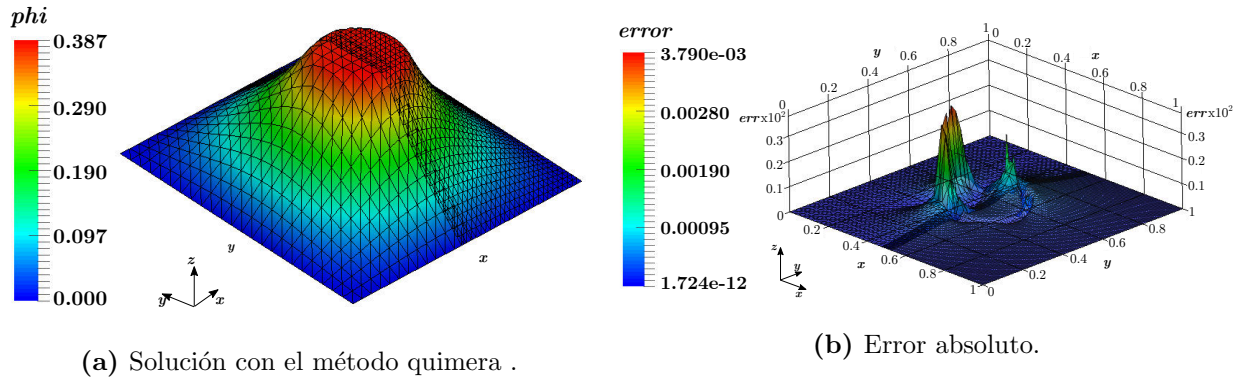
**Figura 2.4:** Solución (a) y error (b) obtenidos a la ecuación de Poisson con la fuente ubicada en la malla fina.

principal propósito de este ejemplo es verificar la convergencia del método, especialmente cuando los mayores gradientes de la solución se encuentren en la región de interpolación. Tanto los datos de las mallas que se utilizaron para el experimento, como los errores computados en la malla chimera y en la malla uniforme se presentan en la Tabla 2.2. Puede verse de los resultados obtenidos que el

	Chimera					Uniforme			
	Malla		Elem	Error $L_2$		Malla		Error $L_2$	
ref	$[N_{1y}, N_{2y}]$	$[h_{1y}, h_{2y}]$		$[\Omega_1, \Omega_2]$	$\Omega$	$N_{x,y}$	$h$	Elem	$\Omega$
1.00	[11, 23]	[9.09e-02, 4.34e-02]	730	[1.43e-02, 1.65e-02]	2.98e-02	23	4.34e-02	1058	9.44e-03
1.41	[16, 33]	[6.25e-02, 3.03e-02]	1476	[6.63e-03, 2.22e-03]	7.05e-03	33	3.03e-02	2178	3.54e-03
2.00	[23, 47]	[4.34e-02, 2.12e-02]	2948	[2.27e-03, 1.56e-03]	2.89e-03	47	4.34e-02	4418	1.27e-03
2.82	[32, 66]	[3.12e-02, 1.51e-02]	5708	[2.03e-03, 1.15e-03]	2.39e-03	66	1.51e-02	8712	1.65e-04
4.00	[45, 93]	[2.22e-02, 1.07e-02]	11274	[5.95e-04, 3.37e-04]	7.11e-04	93	1.07e-02	17298	7.42e-05
5.65	[63, 131]	[1.58e-02, 7.63e-03]	22362	[1.26e-04, 5.96e-05]	1.43e-04	131	7.63e-03	34322	3.79e-05
8.00	[89, 185]	[1.12e-02, 5.40e-03]	44626	[5.68e-05, 2.23e-05]	6.25e-05	185	5.40e-03	68450	1.89e-05
11.31	[125, 261]	[8.00e-03, 3.83e-03]	88536	[2.73e-05, 7.07e-06]	2.85e-05	261	3.83e-03	136242	9.50e-06

**Tabla 2.2:** Información de la malla chimera y de la malla conforme uniforme con sus respectivos errores obtenidos en la norma  $L_2$  para el Caso 2.

orden de convergencia se mantiene, sin embargo, el error en la norma  $L_2$  obtenido por el método chimera es ligeramente mayor que el obtenido por una malla refinada uniforme con tamaño de elemento igual al de la malla fina. Esto era de esperarse ya que, como puede observarse en la Fig. 2.5b, gran parte del error es introducido por la malla gruesa del método chimera .



**Figura 2.5:** Solución (a) y error (b) obtenidos de la ecuación de Poisson con la fuente ubicada en la zona de solapamiento.

### 2.3.1.3. Case 3: término fuente contenido en la malla gruesa

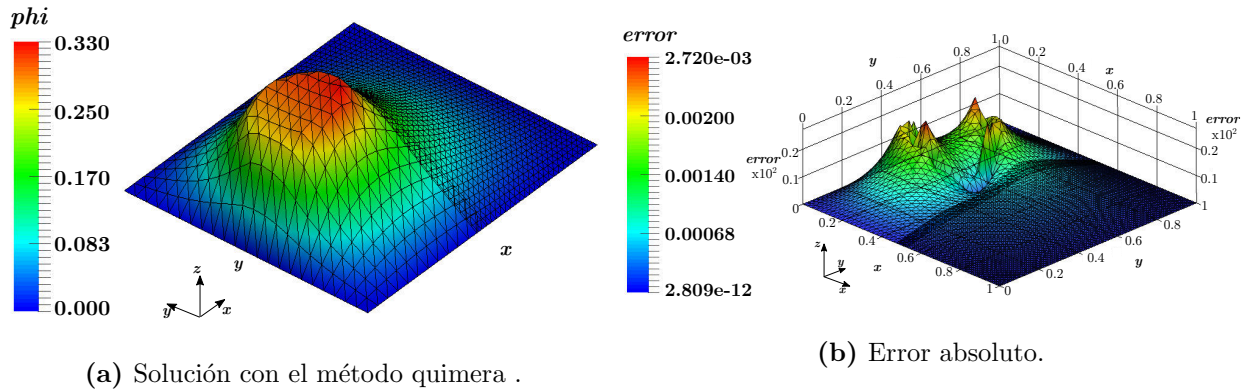
En este ejemplo numérico se resuelve nuevamente la ecuación de Poisson con un término de fuente singular (ver Ec. (2.1)) cuyo centro se encuentra posicionado en  $x_c = 0.25$  e  $y_c = 0.5$ , es decir, sobre la malla gruesa. Se mantiene la longitud de solapamiento de 0.05.

Tanto la información de las mallas utilizadas como los errores computados pueden visualizarse en la Tabla 2.3.

ref	Chimera					Uniforme			
	Malla		Elem	Error $L_2$		Malla		Error $L_2$	
	$[N_{1y}, N_{2y}]$	$[h_{1y}, h_{2y}]$		$[\Omega_1, \Omega_2]$	$\Omega$	$N_{x,y}$	$h$	Elem	$\Omega$
1.00	[11 , 23]	[9.09e-02 , 4.34e-02]	730	[2.08e-02 , 1.80e-02]	9.09e-02	11	4.34e-02	1058	3.67e-02
1.41	[16 , 33]	[6.25e-02 , 3.03e-02]	1476	[5.10e-03 , 1.10e-03]	5.26e-03	16	6.25e-02	2178	1.01e-02
2.00	[23 , 47]	[4.34e-02 , 2.12e-02]	2948	[4.94e-03 , 1.94e-03]	5.40e-03	23	4.34e-02	4418	7.50e-03
2.82	[32 , 66]	[3.12e-02 , 1.51e-02]	5708	[1.29e-03 , 1.57e-04]	1.30e-03	32	3.12e-02	8712	2.28e-03
4.00	[45 , 93]	[2.22e-02 , 1.07e-02]	11274	[8.86e-04 , 3.10e-04]	9.56e-04	45	2.22e-02	17298	9.27e-04
5.65	[63 , 131]	[1.58e-02 , 7.63e-03]	22362	[1.77e-04 , 3.29e-05]	1.81e-04	63	1.58e-02	34322	2.05e-04
8.00	[89 , 185]	[1.12e-02 , 5.40e-03]	44626	[8.10e-05 , 9.04e-06]	8.17e-05	89	1.12e-02	68450	8.10e-05
11.31	[125 , 261]	[8.00e-03 , 3.83e-03]	88536	[4.10e-05 , 4.75e-06]	4.13e-05	125	8.00e-03	136242	4.12e-05

**Tabla 2.3:** Información de la malla quimera y de la malla conforme uniforme con sus respectivos errores obtenidos en la norma  $L_2$  para el Caso 3.

Puede observarse de los resultados presentados en la Tabla 2.3 que cuando los mayores gradientes de la solución se encuentran en la malla gruesa entonces, a partir del tercer grado de refinamiento, el error obtenido por el método quimera es muy similar a aquel que se obtiene utilizando una malla uniforme estructurada cuyos elementos tienen el tamaño de la malla gruesa de las mallas solapadas.

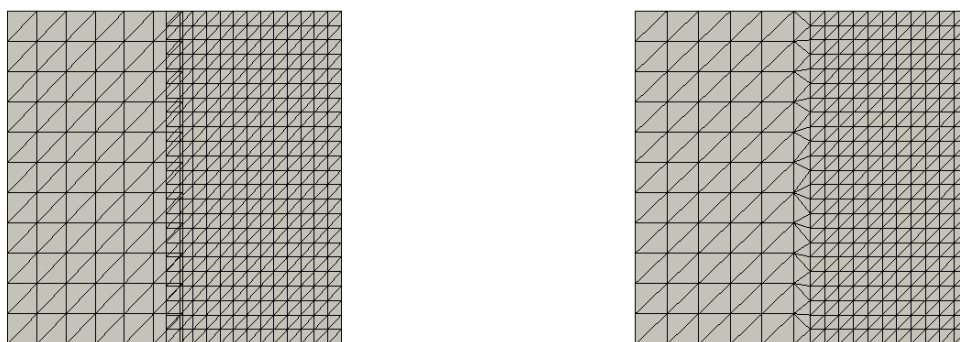


**Figura 2.6:** Solución (a) y error (b) obtenidos de la ecuación de Poisson con la fuente ubicada en la malla gruesa.

También se puede concluir que el orden de convergencia es  $O(h^2)$ . Tanto la solución como el error absoluto obtenido en las mallas solapadas para el Caso 3 pueden verse en la Fig. 2.6. Se puede observar claramente de la figura la continuidad lograda de la solución en la región de solapamiento.

### 2.3.2. Comparación de una malla quimera con una malla conforme análoga

Un estudio de convergencia complementario se realiza a continuación comparando los resultados obtenidos por el esquema propuesto, en contraste con los resultados obtenidos con una malla conforme lo más parecida a las mallas solapadas (Fig. 2.7). La malla conforme se generó utilizando el software de uso libre **SALOME** (Ribes y Caremoli, 2007.). Se analizaron ocho niveles de refinamiento y se compararon ambos resultados. Para este caso se posicionó el término fuente sobre la malla fina. Los resultados son presentados en la Tabla 2.4. Una vez obtenida la solución en la malla



(a) Malla para el método quimera .

(b) Malla conforme análoga.

**Figura 2.7:** Imagen conceptual de las mallas quimera y de las mallas conformes análogas utilizadas para el presente estudio.

conforme, los valores en los nodos son interpolados sobre la solución de referencia para computar el error en la norma  $L_2$ .

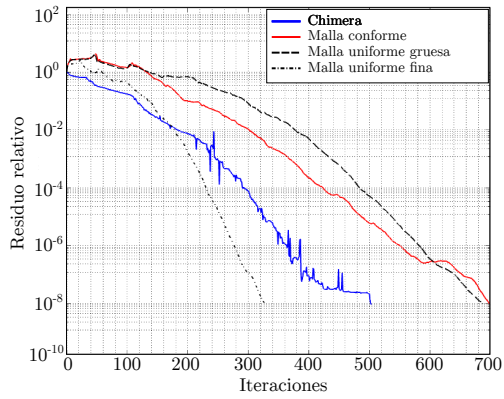
ref	Malla Chimera		Malla conforme	
	Elementos	Error $L_2$	Elementos	Error $L_2$
1.00	730	2.445e-02	650	4.8696e-03
1.41	1476	3.717e-03	1361	2.1124e-03
2.00	2948	3.689e-04	2736	5.7245e-04
2.82	5708	1.537e-04	5214	2.8146e-04
4.00	11274	7.418e-05	11604	1.2195e-04
5.65	22362	3.742e-05	20596	6.9399e-05
8.00	44626	1.862e-05	46202	3.1310e-05
11.31	88536	9.331e-06	81114	1.7930e-05

**Tabla 2.4:** Error en la norma  $L_2$  obtenido con una malla quimera y por una malla análoga conforme.

Como puede visualizarse en la Tabla 2.4, el error obtenido por el esquema quimera es ligeramente menor que el obtenido por la malla conforme equivalente. Este resultado muestra conveniente el uso del esquema quimera ya que permite resolver un problema mediante la descomposición del mismo en dos subdominios, aliviando los esfuerzos de mallado, y aún así obteniendo resultados precisos comparables a los obtenidos por una malla análoga conforme.

## 2.4. Análisis de convergencia de las soluciones iterativas

La convergencia de los *solvers* iterativos CG y BiCGStab fue estudiada para el Caso 1. Se hizo especial énfasis en este caso ya que en la aplicación a un problema práctico, generalmente se desea que los mayores gradientes de la solución se sitúen sobre la malla más fina. El estudio se llevó a cabo utilizando mallas solapadas donde la malla gruesa se encuentra conformada por  $125 \times 64$  elementos que luego son divididos en triángulos, mientras que la malla fina por  $261 \times 144$  elementos, para luego conformarlos también en triángulos. El esquema fue resuelto con BiCGStab. También, a modo de comparación, se comparó la convergencia de una malla conforme adaptativa compuesta por una cantidad de elementos triangulares similar a la de la malla quimera, resolviendo el sistema con CG. Adicionalmente, el mismo problema se resolvió con dos mallas conformes uniformes, una fina y una gruesa, las cuales se conforman por  $261 \times 261$  y  $125 \times 125$  elementos cuadrangulares respectivamente, que luego son divididos en triángulos. Estas mallas fueron resueltas también con CG, donde la tolerancia definida para ambos *solvers* es de  $1e-08$ .



**Figura 2.8:** Historial de convergencia de BiCGStab para el método quimera y de CG para las mallas conformes y uniformes.

Malla	Elementos	Solver	Iter.	Error
Quimera	88536	BiCGStab	503	9.33e-06
Conforme	81114	CG	699	1.79e-05
Gruesa	31250	CG	327	8.00e-03
Fina	136242	CG	687	9.39e-06

**Tabla 2.5:** Análisis de convergencia de las soluciones iterativas.

Como puede observarse en la Fig. 2.8, un acople de Quimera resuelto con BiCGStab presenta una buena tasa de convergencia, alcanzando la tolerancia establecida en 503 iteraciones, mientras que la malla adaptativa conforme alcanza dicha tolerancia en 699 iteraciones. Además, el número total de iteraciones que requiere el acople del método quimera para alcanzar la convergencia se mantiene menor que el requerido por la malla fina uniforme, obteniendo asimismo prácticamente un mismo error.

## 2.5. Conclusiones

En este primer Capítulo se presenta tanto la formulación matemática de un acople tipo quimera para mallas parcialmente solapadas en 2D, como también un algoritmo para resolver el sistema de ecuaciones resultante en forma iterativa de manera *matrix-free*. El error resultante, contemplando el error producido en ambos subdominios, es definido con el fin de comparar el error producido por el método quimera respecto al obtenido por soluciones sobre mallas conformes. El esquema se analizó mediante la solución de la ecuación de Poisson en dominios bidimensionales en tres casos. Se mostró que la solución obtenida por el acople mantiene una buena convergencia y la precisión en los resultados, incluso si los mayores gradientes se encuentran sobre la malla más gruesa o en la región de solapamiento. Se hizo especial énfasis en el primer caso ya que el principal objetivo del método es posicionar la malla más fina sobre la zona de altos gradientes. Entonces, en dicho caso se llevó a cabo un estudio para comparar la solución obtenida por la presente metodología, respecto a una malla conforme equivalente. Del tercer nivel de refinamiento en adelante, los resultados obtenidos por el método quimera presentaron menor error que aquel obtenido por la malla conforme análoga.

Adicionalmente se llevó a cabo un estudio de convergencia de los *solvers* iterativos, utilizando

BiCGStab para resolver el esquema de mallas solpadas, donde se compararon los resultados con los obtenidos por CG en mallas conformes. El esquema quimera probó ser bien definido, mostrando buenas tasas de convergencia, y requiriendo menor número de iteraciones para converger que una malla conforme equivalente.

## Capítulo 3

# El método quimera en mallas no estructuradas parcialmente y completamente solapadas

### 3.1. Introducción

Con el fin de poder utilizar el método quimera para abordar problemas con geometrías de formas complicadas y posteriormente poder utilizarlo en problemas de optimización, en este Capítulo se extiende la implementación del algoritmo propuesto para el caso de mallas no coincidentes y no estructuradas, tanto parcialmente como totalmente solapadas. También se explica detalladamente un algoritmo de interpolación de alto orden propuesto en esta tesis con el fin de conservar el orden de convergencia espacial del método y garantizar que el error en la norma  $L_2$  se asemeje al esperado por una malla análoga conforme. Adicionalmente, se introduce un algoritmo de detección de fronteras de interpolación utilizado en la interpolación de datos en casos con mallas completamente solapadas. Al final de este Capítulo se describen diversos experimentos numéricos evaluando la calidad de la solución obtenida mediante la utilización del esquema quimera, como así también las tasas de convergencia obtenidas por los *solvers* iterativos utilizados. En el Anexo A se exponen con mayor profundidad los conceptos y resultados desarrollados en el presente Capítulo.

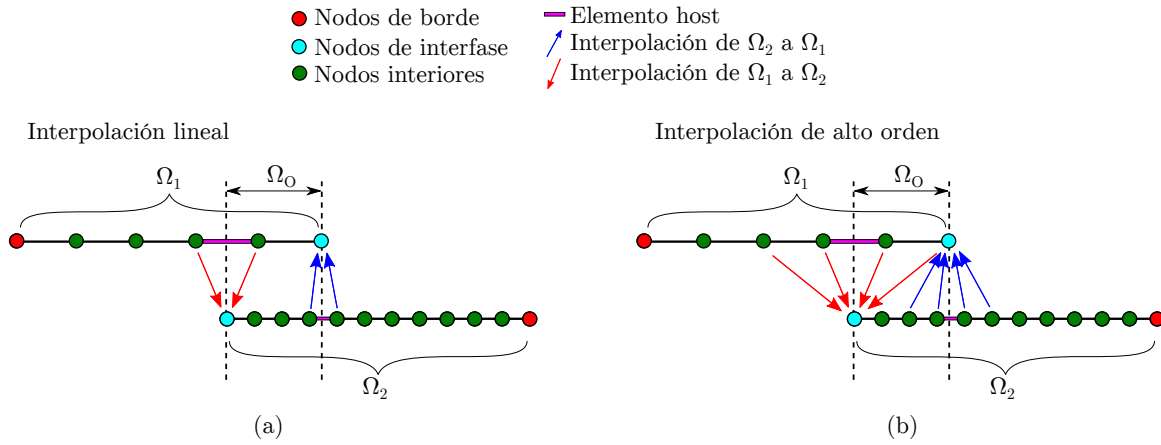
### 3.2. Interpolación de alto orden

#### 3.2.1. Esquema de interpolación

El orden de convergencia espacial de la solución está directamente afectado por el esquema de interpolación utilizado según se pudo verificar con diversos experimentos numéricos (ver el Anexo

A). Con el fin de preservar el orden de convergencia que posee el método de elementos finitos en mallas no estructuradas utilizando elementos lineales, es necesaria una interpolación de tercer orden. A continuación se describe un esquema de interpolación utilizando operadores de Lagrange implementado y utilizado en esta tesis.

A modo esquemático, se muestra en la Fig. 3.1 tanto el proceso de interpolación de bajo orden como el de alto orden en mallas unidimensionales parcialmente solapadas. En el caso de una interpolación de bajo orden lineal (ver Fig. 3.1a) los únicos nodos que participan de la interpolación de los valores de los nodos internos (nodos verdes) al nodo de la interfase de la malla recíproca, son los nodos extremos del elemento *host*, mientras que en la interpolación de alto orden propuesta en esta tesis (ver Fig. 3.1b), no sólo se utilizan dichos nodos, sino también los valores de algunos o varios de sus vecinos topológicos.



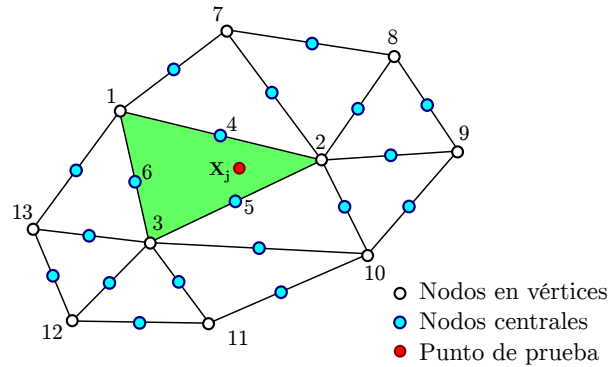
**Figura 3.1:** Ilustración de una interpolación de bajo orden lineal (a) y una interpolación de alto orden (b) para mallas unidimensionales parcialmente solapadas.

### 3.2.1.1. Descripción del esquema de interpolación de alto orden

Tanto la formulación matemática del esquema de interpolación propuesto para mallas en dominios 2D, como también el algoritmo utilizado para su implementación, se encuentran explicados de manera detallada en el Anexo A. A continuación se describe de manera global el procedimiento del esquema de interpolación propuesto.

Para la descripción del algoritmo de interpolación se asume que existe una malla  $\Omega_1$  compuesta de elementos triangulares lineales no estructurados (ver Fig. 3.2), y que se conocen los valores de una función suave  $\phi$  en los vértices de los nodos (círculos blancos en la Fig. 3.2). Ahora suponemos que existen ciertos puntos de prueba  $\mathbf{x}_j$  pertenecientes al dominio  $\Omega_2$  en los cuales se requiere conocer su valor a partir de una interpolación de tercer orden de los valores de  $\phi$ . Esto requiere que el error de la interpolación sea  $O(h^3)$ , siendo  $h$  el tamaño de elemento típico de la malla  $\Omega_1$ .





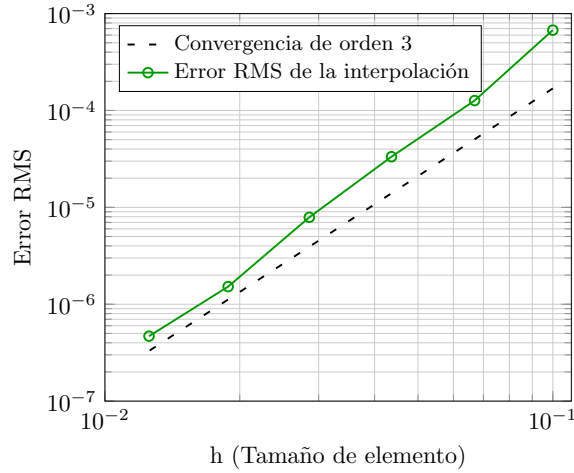
**Figura 3.2:** Esquema de interpolación para mallas no estructuradas

La interpolación se basa en reconstruir los valores de la función  $\phi$  en los nodos centrales (en color cian en la Fig. 3.2), y luego hacer una interpolación lagrangiana  $P_2$ . Para realizar dicha reconstrucción de valores en los nodos centrales, se utiliza un ajuste por mínimos cuadrados a los nodos de los vértices junto con sus vecinos topológicos. Volviendo al punto de prueba  $\mathbf{x}_j$  en la Fig. 3.2, para realizar una interpolación sobre dicho nodo (que se encuentra dentro del elemento coloreado en verde), se requieren los valores en los nodos centrales 4, 5 y 6, como así también los valores de los nodos en los vértices 1, 2 y 3. Para reconstruir el valor de  $\phi$  en el nodo central 5, por ejemplo, se realiza un ajuste por mínimos cuadrados utilizando los valores en los vecinos topológicos de los nodos 2 y 3, siendo éstos los nodos del 1 al 13 (descontando los nodos centrales).

Como la reconstrucción no puede imponerse sobre el total de vecinos topológicos, ya que el sistema quedaría sobredeterminado, los parámetros son elegidos de forma que la reconstrucción en los nodos sea la que mejor se ajusta a la función  $\phi$  en términos de mínimos cuadrados. Sin embargo, para la reconstrucción en los nodos que se encuentran en los extremos de la arista (nodos 2 y 3 en el ejemplo para la reconstrucción del nodo central 5) se impone el valor exacto. Esto se lleva a cabo mediante el uso de los multiplicadores de Lagrange, cuya formulación se encuentra detallada en el Anexo A.

Finalmente se obtiene un sistema para obtener los parámetros a ser utilizados en la reconstrucción del valor del nodo central 5. Este sistema generalmente es regular y bien condicionado, sin embargo puede darse el caso que esto no sea así. En el Anexo A se trata un ejemplo en el cual la frontera de interpolación de  $\Omega_2$  intersecta elementos de borde de  $\Omega_1$ . En ese ejemplo, la primera capa de vecinos topológicos no es suficiente ya que el sistema resulta mal condicionado. Este caso es también considerado por el algoritmo propuesto y lo soluciona agregando más capas de nodos para realizar el ajuste.

### 3.2.2. Ejemplo numérico. Error de interpolación



**Figura 3.3:** Error RMS de la interpolación propuesta.

Con el fin de evaluar el esquema de interpolación propuesto sobre un dominio cuadrado  $\Omega = [0, 1] \times [0, 1]$ , se definen mallas no estructuradas con número creciente de nodos compuestas por elementos triangulares lineales. Luego se escoge aleatoriamente dentro del dominio un número dado de puntos de prueba ( $N = 100$ ), se calcula la función de prueba  $\phi(x, y) = \sin(\pi x) \sin(\pi y)$  en los nodos de la malla y se interpolan los valores en los puntos de prueba. El error cuadrático medio (RMS) es calculado como:

$$\epsilon_{\text{RMS}}^2 = \frac{1}{N} \sum_{j=1}^N |\tilde{\phi}_j - \phi_j|^2 \quad (3.1)$$

De la Fig. 3.3 se puede concluir que el orden de convergencia obtenido es  $O(h^3)$ . El interpolador se implementó utilizando el lenguaje de programación C++, haciendo uso de las librerías Eigen (Guennebaud et al., 2010).

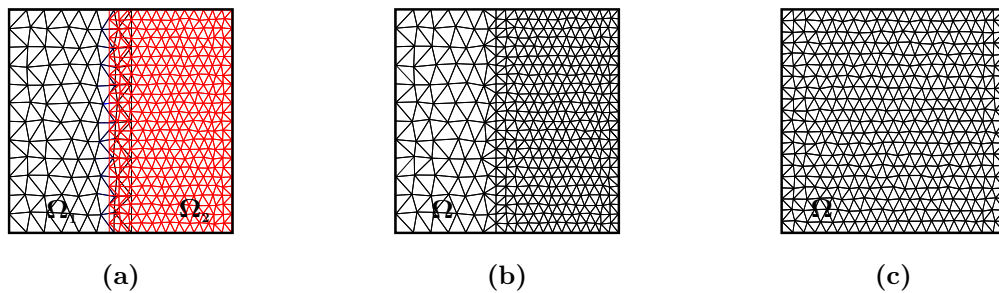
## 3.3. Ejemplos numéricos

### 3.3.1. Ejemplo numérico 1: Solución del problema de Poisson en dominios 2D parcialmente solapados

Este ejemplo numérico es similar a uno desarrollado anteriormente en el Capítulo previo, en el cual se posiciona una fuente singular sobre la zona de la malla fina y la solución obtenida se compara con soluciones obtenidas en mallas conformes. Pero en este caso las mallas utilizadas en ambos subdominios no son estructuradas, y además se introduce una perturbación aleatoria a los nodos con el fin de considerar el caso más general posible durante el acople.

En este caso los valores que definen la fuente quedan definidos como:  $r = 0.15$ ,  $\delta = 0.025$ , y

$[0.75, 0.5]$  para los valores de  $x_c$  e  $y_c$ , respectivamente. Las mallas son generadas en forma cartesiana con  $N_{1y} = 11$  elementos en la dirección  $y$  para  $\Omega_1$  y  $N_{2y} = 23$  para  $\Omega_2$ , de forma que no sean coincidentes en la zona de solapamiento. Una vez generadas, en cada simulación se aplica una perturbación aleatoria a los nodos mediante un desplazamiento aleatorio en ellos. Las mallas en ambos subdominios se van refinando progresivamente para realizar un estudio de convergencia. Un esquema de las mallas generadas se puede observar en la Fig. 3.4. Los resultados obtenidos por la malla quimera (Fig. 3.4a) son comparados con dos mallas conformes con similar cantidad de grados de libertad: una malla conforme análoga a la malla quimera (Fig. 3.4b), y otra malla conforme homogénea con tamaño de elemento constante en todo el dominio (Fig. 3.4c).



**Figura 3.4:** Ejemplos de las mallas utilizadas en el problema 2D de Poisson. Las tres mallas poseen similar cantidad de grados de libertad  $N$ .

En el Anexo A se presentan los resultados obtenidos del presente ejemplo numérico. En la Fig. A.6 se puede observar la convergencia en malla y los errores en la norma  $L_2$  obtenidos tanto para la solución, como para su gradiente. En las Figs. A.7 y A.8 puede observarse la solución del campo  $\phi$  y de su gradiente, en las cuales se destaca la continuidad obtenida en la región de solapamiento. Mientras que en la Fig. A.9 y en Tabla A.1 se presentan los resultados de convergencia de los *solvers* iterativos utilizados para cada caso.

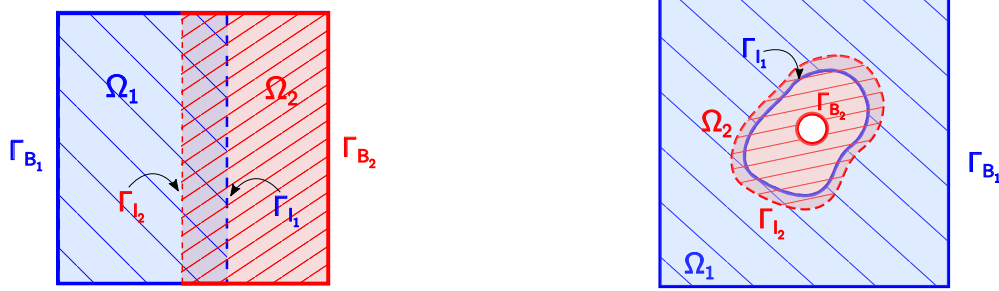
Conclusiones generales de los resultados obtenidos en este ejemplo numérico y de los tratados a continuación, se enuncian al final del presente Capítulo.

### 3.3.2. Ejemplo numérico 2: Solución del problema de Poisson en dominios 2D completamente solapados

En este ejemplo numérico, a diferencia de los ejemplos tratados anteriormente, la malla fina que discretiza el dominio  $\Omega_2$  se encuentra completamente solapada sobre la malla gruesa de fondo que discretiza el dominio  $\Omega_1$ . Esta técnica permite la generación de las mallas de  $\Omega_1$  y  $\Omega_2$  en forma independiente ya sea para posicionar la malla fina sobre la zona donde se encuentran los mayores gradientes de la solución, o representar una frontera (objeto) dentro del dominio de fondo.

Se debe remarcar que cuando las mallas se encuentran parcialmente solapadas, la detección de

las fronteras de interpolación es una tarea sencilla debido a que los nodos pertenecen al borde de cada dominio, por lo tanto son fáciles de distinguir (ver Fig. 3.5a). En contraste, cuando la malla del dominio  $\Omega_2$  se encuentra totalmente solapada sobre  $\Omega_1$  (ver Fig. 3.5b), los nodos de la interfase de  $\Omega_1$  no son fáciles de distinguir, por lo que en esta tesis se propone la estrategia de detección topológica que se describe a continuación.



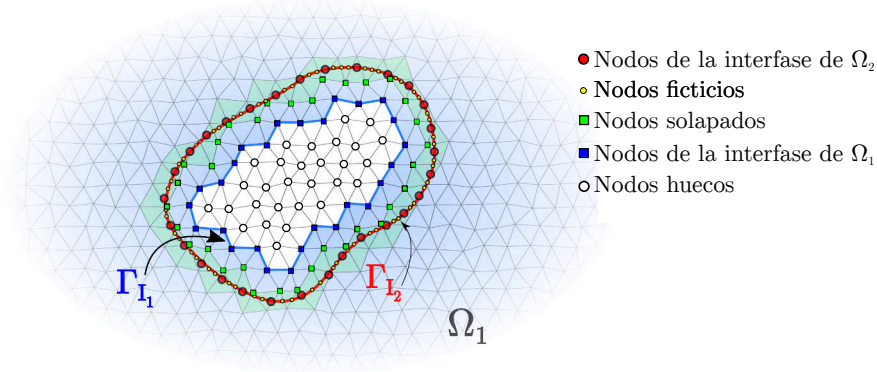
**Figura 3.5:** Dos dominios parcialmente solapados (Izq.), y dos dominios completamente solapados (Der.).

### 3.3.2.1. Detección topológica de las interfaces de interpolación

Si bien el procedimiento para la detección de la interfase de interpolación para mallas completamente solapadas se describe en detalle en el Apéndice A, no obstante, a continuación se enuncian de manera global los pasos seguidos. El algoritmo para la detección de los nodos de la interfase del dominio  $\Omega_1$  consiste en los siguientes pasos:

- Primero se deben detectar los elementos de la interfase de  $\Omega_1$  (elementos en verde en la Fig. 3.6), los cuales son intersectados por la frontera  $\partial\Omega_2$ . Para lograrlo, se generan nodos ficticios (nodos amarillos en la Fig. 3.6) entre los nodos que conforman  $\partial\Omega_2$  (nodos rojos).
- Luego se realiza una búsqueda mediante un algoritmo *quadtree* para ubicar los elementos de la malla de fondo donde se encuentran los nodos de borde y los ficticios.
- Se corrobora cuáles son los nodos que se encuentran dentro de la zona de solapamiento (nodos coloreados en verde).
- Se buscan sus nodos vecinos mediante una búsqueda topológica, y sólo aquellos que se encuentren dentro de la zona solapada serán tratados como nodos de la interfase de  $\Omega_1$  (nodos azules). Estos nodos estarían ubicados en la segunda capa.

Los nodos de la interfase de  $\Omega_1$  también son referidos en la literatura como *fringe nodes* (Eguzkitza et al., 2013; Steger y Benek, 1987; Steger, 1983). En los experimentos numéricos de aquí en adelante siempre se utilizarán los nodos ubicados en la tercera capa, a menos que se indique lo contrario.



**Figura 3.6:** Detección automática topológica de la frontera de interpolación de la malla  $\Omega_1$ .

### 3.3.2.2. Descripción del problema

En este experimento numérico se utiliza el mismo dominio computacional 2D, donde se resuelve el problema de Poisson (ver Ec. (2.1)). La malla quimera está compuesta por una malla gruesa cuadrada  $\Omega_1 = [0, 1] \times [0, 1]$  que cubre completamente el dominio, y otra malla circular más fina  $\Omega_2$  de radio 0.2 situada en el centro del dominio y completamente solapada. Se utiliza el mismo término fuente que el utilizado en el ejemplo numérico 1 del presente Capítulo, con la salvedad que en este caso los parámetros de dicha fuente vienen dados por:

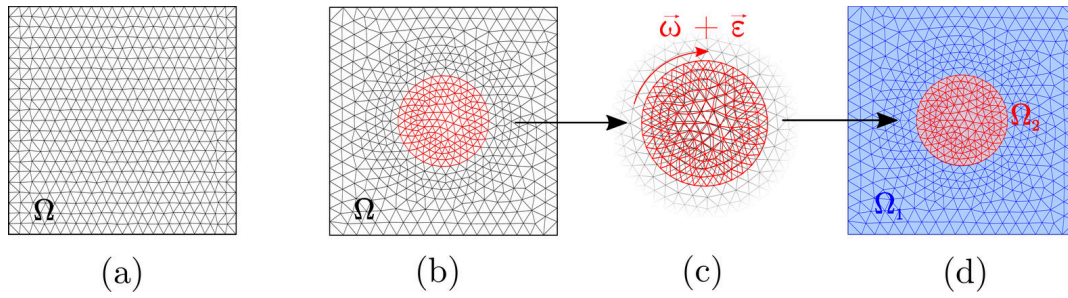
$[\mathbf{x}_c, \mathbf{y}_c]$	$\delta$	$\mathbf{r}$
$[0.5, 0.5]$	0.015	0.1

**Tabla 3.1:** Parámetros del término de fuente delta utilizado en el ejemplo numérico 2.

De esta forma el término fuente se encuentra aún más concentrado en el centro que en el caso anterior.

En la Fig. 3.7 se muestran las mallas utilizadas para el ejemplo numérico. Nuevamente, se tiene: (i) una malla conforme homogénea (*OD homogénea* en la Fig. 3.7a), la cual posee un tamaño de elemento aproximadamente constante; (ii) una malla conforme refinada hacia el centro del dominio (*OD refinada* de la Fig. 3.7b), lugar donde se encuentran los mayores gradientes; y (iii) una malla quimera (ver Fig. 3.7d), cuyo dominio  $\Omega_2$  surge de rotar y perturbar los elementos de la malla OD refinada que se encuentran en un círculo de radio 0.2 (ver Fig. 3.7c). El principal objetivo de este ejemplo es verificar que la metodología propuesta de detección de interfase en mallas completamente solapadas no introduzca error en la solución.

En el Anexo A se presentan los resultados obtenidos en este ejemplo. En la Fig. A.12 se puede observar la convergencia en malla y los errores en la norma  $L_2$  obtenidos tanto para la solución, como para su gradiente. En las Figs. A.13 y A.14 puede observarse la solución del campo  $\phi$  y de su



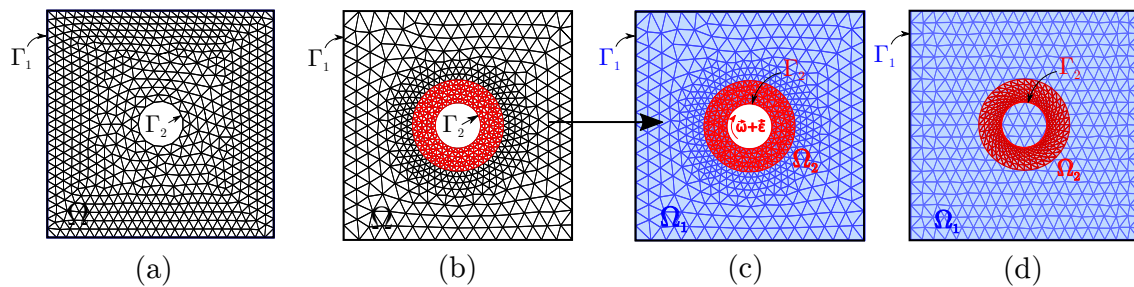
**Figura 3.7:** Mallas empleadas: a) Malla conforme homogénea con tamaño de elemento casi constante en todo el dominio (OD homogénea); b) Malla conforme con refinamiento progresivo hacia el centro del dominio (OD refinada); c) Los elementos en rojo de la malla OD refinada son rotados y perturbados aleatoriamente para generar  $\Omega_2$ ; d) Mallas solapadas generadas a partir de la malla OD refinada (malla quimera).

gradiente. En la Fig. A.15 y en Tabla A.3 se presentan los resultados de convergencia de los *solvers* iterativos utilizados para cada caso.

### 3.3.3. Ejemplo numérico 3: condición de borde impuesta sobre la malla patch completamente solapada

En este ejemplo numérico se considera el mismo dominio cuadrado, aunque en este caso con un hueco de radio 0.1 centrado en el dominio. El interés de este caso es que los mayores gradientes de la solución vendrán dados por la condición de borde definida sobre un objeto (en este caso sobre el círculo), y no por un término fuente.

En este caso se resuelve la ecuación de Laplace sobre las cuatro mallas mostradas en la Fig. 3.8. El problema en cada subdominio de la malla quimera con sus respectivas condiciones de borde se encuentra planteado en la Tabla A.4 del Anexo A.



**Figura 3.8:** Mallas empleadas: a) Malla OD homogénea con tamaño de elemento casi constante; b) Malla OD refinada con un refinamiento progresivo hacia  $\Gamma_2$ ; c) Los elementos en rojo de la malla OD refinada son rotados ( $\bar{\omega}$ ) y perturbados aleatoriamente para así generar la malla  $\Omega_2$  de método quimera; d) Malla quimera regular formada por una malla no estructurada homogénea que cubre todo el dominio, y una malla estructurada refinada progresivamente hacia  $\Gamma_2$ .

El principal objetivo del presente caso es analizar el desempeño del MC cuando se impone una condición de borde sobre la malla  $\Omega_2$ . Esto habilitaría la introducción de cuerpos dentro del dominio

lo que permitiría abordar posteriormente problemas de optimización. También tiene como objetivo analizar el desempeño de la malla quimera regular (Fig. 3.8) ya que, para aplicar el MC a problemas de optimización (a ser abordados en los siguientes Capítulos de esta tesis), la idea es generar una malla de fondo cubriendo todo el dominio computacional con una malla no estructurada, mientras que para los objetos cuya disposición se desee optimizar, en caso de ser posible, sean mallados con una malla estructurada refinada hacia la zona de mayores gradientes.

En el Anexo A se presentan los resultados de convergencia en malla y error en la norma  $L_2$  (ver A.17), como también la visualización de la solución (Figs. A.18 y A.19), y los resultados del estudio de convergencia de los *solvers* iterativos (Fig. A.20, y Tabla A.5).

### 3.4. Conclusiones

El esquema quimera propuesto aplicado a mallas no coincidentes y no estructuradas fue presentado en este Capítulo. También se introdujo un algoritmo de interpolación de alto orden utilizado para la transferencia de información entre mallas y que se explicó en detalle. El esquema se evaluó en tres ejemplos numéricos para dominios en dos dimensiones, tanto parcialmente como completamente solapados. En uno de los casos, las condiciones de borde se imponen tanto en la malla fina, como también en la gruesa. Los resultados muestran que la convergencia de segundo orden se mantiene en las mallas no estructuradas, y se obtienen buenas tasas de convergencia con el *solver* iterativo BiCGStab. Adicionalmente se pudo observar la continuidad tanto en la solución como la de su gradiente en la zona de solapamiento. Respecto al rendimiento computacional de la solución de los sistemas matriciales, en todos los casos analizados, el *solver* iterativo utilizado en el problema acoplado requirió aproximadamente entre 25-37% menos iteraciones comparado a una malla análoga conforme resuelta con CG. Esto sugiere que los requerimientos en cuanto a memoria y al costo computacional de la solución requeridos por el esquema quimera sean similares a los obtenidos por el problema resuelto en un dominio conforme. Adicionalmente, los resultados obtenidos en el último ejemplo numérico muestran un muy buen desempeño del método, en cuánto a error y convergencia del *solver* iterativo, cuando es utilizado en una malla no estructurada de fondo cubriendo todo el dominio, y una malla estructurada alrededor del objeto. Esto resulta muy conveniente para su empleo en problemas de optimización, que será el tema a tratar en el siguiente Capítulo.





## Capítulo 4

# Acoplamiento del método quimera con optimización

### 4.1. Introducción

Como se mencionó en la introducción de esta tesis, diversos problemas de la mecánica computacional pueden aprovechar las ventajas del método quimera. La optimización para la configuración de un sistema dado en ese contexto es una aplicación directa donde una cantidad dada de objetos, cada uno con su respectiva malla, conformando la malla *patch*, se mueven libremente sobre una malla de fondo que cubre enteramente el dominio computacional, en busca de la mejor configuración respecto a una cierta función objetivo. La principal ventaja de la utilización del método quimera en problemas de optimización en mecánica computacional es que el esquema permite grandes desplazamientos relativos de los objetos dentro del dominio durante la minimización de una función objetivo sin la necesidad de remallar el dominio computacional, como así también mantener la calidad de la solución. En este Capítulo introduciremos el código abierto de optimización *pyOpt* (Perez et al., 2012) utilizado conjuntamente durante el desarrollo de esta tesis, y explicaremos brevemente algunos de sus *solvers* utilizados. Los mismos pueden diferenciarse en dos grandes grupos: optimizadores basados en gradiente y optimizadores libres de gradiente. Se describe la estrategia seguida en esta tesis en la cual se combina la utilización de ambos tipos de optimizadores, y también se estudia su desempeño en plataformas de HPC.

### 4.2. Optimización combinada

La mayoría de los problemas de optimización en mecánica computacional poseen espacios de diseños no convexos, con múltiples mínimos locales (Hillier, 2012; Bazaraa et al., 2013; Ryoo y

Sahinidis, 1995). Debido a esto, la utilización de optimizadores basados en gradiente se encuentra restringida en tales problemas, ya que la solución final que se obtiene es muy dependiente de la solución inicial utilizada. Si la solución inicial de prueba empleada en el problema de optimización se encuentra alejada del mínimo global, es muy probable que la solución final obtenida por el optimizador caiga en un mínimo local, pudiendo estar distante de una solución aceptable. En esta tesis la estrategia utilizada para abordar el problema mencionado se basa en los siguientes pasos:

- Primero se utiliza un optimizador evolutivo libre de gradiente para encontrar un posible mínimo global.
- Luego, la solución encontrada se utiliza como punto de partida para un optimizador basado en gradiente y así mejorar la solución del problema de optimización.

Esta estrategia es comúnmente referida en la literatura como optimización combinada u optimización híbrida, en la cuál se aprovechan las ventajas de diversos optimizadores (Parwani et al., 2014; Siavashi et al., 2017; Abolhasani et al., 2015). Para este fin, la estructura de pyOpt, el cual se encuentra implementado bajo el paradigma de Programación Orientada a Objetos (POO), posee clases que nos permiten pasar en forma directa la solución de un optimizador para luego ser utilizada como punto inicial de otro. Los detalles de dicha implementación y cómo es utilizada en esta tesis se encuentra detallada en el Anexo B.

### 4.3. Optimizadores en pyOpt

#### 4.3.1. Optimizador ALPSO

En esta tesis se utilizó el optimizador basado en población ALPSO (por *Augmented Lagrangian Particle Swarm Optimizer*) para acercarse a un mínimo global. El algoritmo de ALPSO desarrollado por Sedlaczek y Eberhard (2006) es utilizado en la solución de problemas no suaves con restricciones. El algoritmo utiliza multiplicadores de Lagrange para resolver problemas de optimización sujetos a restricciones de igualdad y desigualdad. Al estar basado en el método PSO (*Particle Swarm Optimizer*, Eberhart y Kennedy, 1995), es más factible encontrar el mínimo global de la función objetivo. El algoritmo se basa en el comportamiento social de bandadas de aves o cardúmenes de peces en busca de alimento mediante la cooperación social. Cada partícula del enjambre se encuentra definida por su posición (ver Ec. (4.1)) y su velocidad (ver Ec. (4.2)), y cada una constituye una solución del problema. Al comienzo de la optimización cada partícula evalúa la función objetivo partiendo de una posición inicial aleatoria. Luego, las partículas se desplazan por el espacio de búsqueda siguiendo reglas simples basadas en su posición y velocidad. A través del proceso de optimización cada partícula actualiza su velocidad y posición en base a dos valores principales, el

valor  $p_{\text{best}}$ , que refiere al mejor valor obtenido por la partícula hasta ese momento, y  $g_{\text{best}}$ , que refiere al mejor valor obtenido por todas las partículas de la población. En base a esto, la posición de la partícula se actualiza de la siguiente forma:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \quad (4.1)$$

$$\mathbf{v}_i^{k+1} = \omega \mathbf{v}_i^k + c_1 r_1 (\mathbf{x}_i^{p_{\text{best}},k} - \mathbf{x}_i^k) + c_2 r_2 (\mathbf{x}_i^{g_{\text{best}},k} - \mathbf{x}_i^k) \quad (4.2)$$

donde  $\omega$  es el factor de inercia,  $v_i^k$  es la velocidad de la partícula en la iteración  $k$ ,  $\mathbf{x}_i^k$  es la posición de la partícula en el espacio de búsqueda en la iteración  $k$ ,  $r_1$  y  $r_2$  son valores aleatorios reales en el intervalo (0,1), mientras que  $c_1$  y  $c_2$  son los llamados parámetros cognitivos y social que definen en qué medida la dirección que toma la partícula se encuentra afectada por los valores obtenidos por el enjambre. Si bien estos son los parámetros más importantes que rigen el comportamiento del algoritmo PSO, también se necesitan otros parámetros adicionales. Dichos parámetros adicionales se describen en detalle en el Anexo B de la presente tesis, junto con los criterios seguidos para una correcta elección de los mismos. En este caso, el algoritmo de optimización PSO puede describirse de la siguiente forma:

#### 1. Inicialización

- a) Definición de las constantes  $k_{\text{max}}$ ,  $c_1$ ,  $c_2$ ,  $\omega_1$ ,  $\omega_2$ ,  $v_{\text{init}}$ , y  $v_{\text{max}}$ .
- b) Inicializar los contadores  $k = 0$ ,  $i = 0$ .
- c) Inicializar aleatoriamente las posiciones de las partículas  $\mathbf{x}_i^k$ , para  $i = 1, \dots, p$ .
- d) Inicializar las velocidades de las partículas  $\mathbf{v}_i^k$  con  $v_{\text{init}}$ , para  $i = 1, \dots, p$ .
- e) Evaluar la función objetivo  $f_i^0$  con  $\mathbf{x}_i^0$ , para  $i = 1, \dots, p$ .
- f) Definir  $f_i^{\text{best}} = f_i^0$ ,  $\mathbf{x}_i^{p_{\text{best}}} = \mathbf{x}_i^0$ .
- g) Definir  $f_g^{\text{best}}$  con el mejor  $f_i^{\text{best}}$ .

#### 2. Optimización

- a) Actualizar la velocidad de las partículas  $\mathbf{v}_i^{k+1}$  según la Ec. (4.2).
- b) Actualizar la posición de las partículas  $\mathbf{x}_i^{k+1}$  según la Ec. (4.1).
- c) Evaluar la función objetivo  $f_i^k$  con las coordenadas  $\mathbf{x}_i^k$ .
- d) Si  $f_i^k \leq f_i^{\text{best}}$ , entonces  $f_i^{\text{best}} = f_i^k$  y  $\mathbf{x}_i^{p_{\text{best}}} = \mathbf{x}_i^k$ .
- e) Si  $f_i^k \leq f_g^{\text{best}}$ , entonces  $f_g^{\text{best}} = f_i^k$  y  $\mathbf{x}_i^{g_{\text{best}}} = \mathbf{x}_i^k$ .
- f) Si la condición de parada se satisface, entonces ir al punto 3.
- g) Incrementar  $i$ . Si  $i > N_{SW}$ , entonces incrementar  $k$  y setear  $i = 1$ .
- h) Ir a 2(a).

#### 3. Finalizar la optimización. Reportar resultados obtenidos.

Cuando el algoritmo de PSO opera en forma paralela en los distintos procesadores, el Paso 2 del algoritmo del PSO presenta las siguientes modificaciones:

## 2. Optimización

- a) Actualizar la velocidad de las partículas  $\mathbf{v}_i^{k+1}$  según la Ec. (4.2).
- b) Actualizar la posición de las partículas  $\mathbf{x}_i^{k+1}$  según la Ec. (4.1).
- c) **Evaluar la función objetivo  $f_i^k$  en forma paralela** con las coordenadas  $\mathbf{x}_i^k$ .
- d) Si  $f_i^k \leq f_i^{best}$ , entonces  $f_i^{best} \leq f_i^k$  y  $\mathbf{x}_i^{pbest} = \mathbf{x}_i^k$ .
- e) Si  $f_i^k \leq f_g^{best}$ , entonces  $f_g^{best} \leq f_i^k$  y  $\mathbf{x}_i^{gbest} = \mathbf{x}_i^k$ .
- f) Si la condición de parada se satisface, entonces ir al punto 3.
- g) Incrementar  $k$ .
- h) Ir a 2(a).

En la Fig. 4.1 se visualizan los algoritmos antes mencionados para una mejor comprensión.

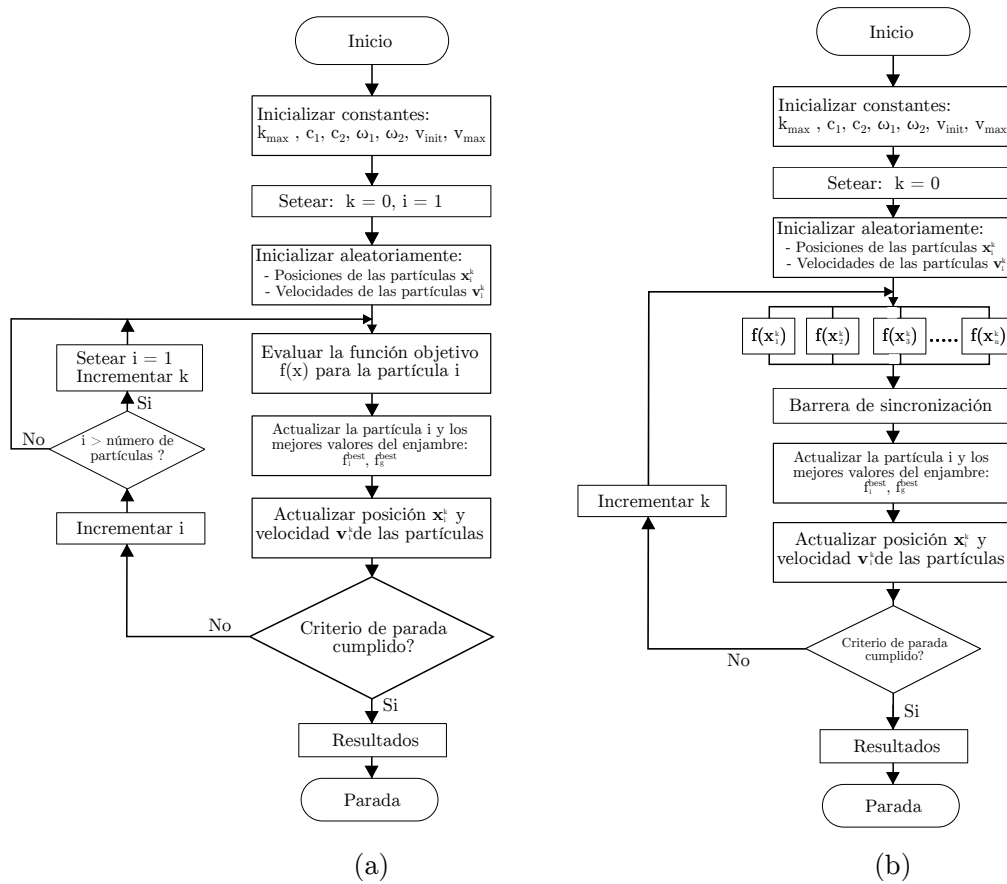


Figura 4.1: Algoritmo PSO en serie (a) y en paralelo (b).

#### 4.3.1.1. La optimización PSO con restricciones y el método de multiplicador de Lagrange aumentado

Debido a que en la práctica muchos problemas de ingeniería contienen tanto restricciones de igualdad como de desigualdad, el algoritmo PSO se encuentra limitado en este aspecto. Debido a esto surge la alternativa de usar el algoritmo ALPSO, el cual es una extensión del algoritmo de PSO

que permite lidiar con dichas restricciones mediante la introducción de una función de penalización que mapea la función objetivo original  $f(\mathbf{x})$  a una pseudo función objetivo  $\psi(\mathbf{x})$  de la siguiente forma:

$$\psi(\mathbf{x}) = f(\mathbf{x}) + r_p \phi(\mathbf{x}) \quad (4.3)$$

Una de las funciones de penalización más comunes es la penalización cuadrática. Por ejemplo, para restricciones de igualdad, la función cuadrática de penalización esta dada por:

$$\phi(\mathbf{x}) = g(\mathbf{x}) \cdot g(\mathbf{x}) \quad (4.4)$$

Sin embargo, la precisión de la solución obtenida depende directamente del valor del factor de penalización  $r_p$ . Si dicho valor es infinito, la solución tiende a la exacta. En forma alternativa, se puede plantear una función de penalización más elaborada mediante el uso del método de multiplicador de lagrange aumentado. Mediante su utilización, una penalización finita es suficiente. La función de Lagrange generalizada puede escribirse de la siguiente manera:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m_e} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{m_i} \lambda_{(j+m_e)} h_j(\mathbf{x}) \quad (4.5)$$

La solución a la ecuación original del problema de optimización restringido es un punto estacionario de la Ec. (4.5) dando valores adecuados de  $\lambda$ . Sin embargo, esto no garantiza que estemos en la presencia de un punto estacionario mínimo. Para remediarlo, las funciones de Lagrange aumentadas consisten en un tercer término que transforma el punto estacionario en un mínimo mediante el uso de una penalización cuadrática

$$L(\mathbf{x}, \lambda, r_p) = f(\mathbf{x}) + \sum_{i=1}^{m_e+m_i} \lambda_i \theta_i(\mathbf{x}) + \sum_{j=1}^{m_e+m_i} r_{p_i} \theta_i^2(\mathbf{x}) \quad (4.6)$$

siendo:

$$\theta_i(\mathbf{x}) = \begin{cases} g_i(\mathbf{x}) & i = 1, \dots, m_e \\ \text{máx} \left[ h_{(i-m_e)}(\mathbf{x}), \frac{-\lambda_i}{2r_{p,i}} \right] & i = m_e + 1, \dots, m_e + m_i \end{cases}$$

En esta tesis no se profundizará más sobre la metodología seguida por el optimizador ALPSO, y se referencia al lector interesado al artículo de [Jansen y Perez \(2011\)](#) para una mayor comprensión del método y de sus posibles aplicaciones.

### 4.3.2. Optimizador SLSQP

Para el refinamiento de la solución del proceso de optimización se utilizó el algoritmo SLSQP (*Sequential Least Squares Quadratic Programming* [Kraft y Schnepfer, 1989](#)). El optimizador SLSQP,

de la familia de los algoritmos de programación secuencial cuadrática, es una modificación del *solver* implementado por Lawson y Hanson (1995). El algoritmo utiliza el método de Han-Powell quasi-Newton con una actualización BFGS de la matriz-B (Scales, 1985), y una función test L1 en el algoritmo de búsqueda lineal. Para calcular los gradientes dos opciones se encuentran implementadas: con diferencias finitas, y con paso complejo, y que pueden ser calculadas en serie o en paralelo.

#### 4.4. Paralelización del proceso de optimización

En problemas de optimización en mecánica computacional tales que la física involucrada es complicada y existe un gran número de variables de diseño, los algoritmos evolutivos requieren usualmente un gran número de evaluaciones de la función objetivo para converger. Esto implica un gran costo computacional si se tiene en cuenta que cada evaluación implica una simulación por elementos finitos. También, en el caso de utilizar optimizadores basados en gradiente, el costo computacional requerido para evaluar dichos gradientes puede resultar muy elevado. Entonces para disminuir el tiempo requerido en resolver un problema de optimización complicado resulta fundamental el uso de HPC (*High Performance Computing*) en *clusters* de computadoras. Por eso en esta tesis se hace uso de las utilidades implementadas en *pyOpt* para un cómputo en paralelo de la optimización. Los optimizadores de *pyOpt* logran esto mediante el uso del paquete de acceso libre *mpi4py* (Dalcin et al., 2011), y a través del uso del administrador de tareas en plataformas HPC SLURM (Jette et al., 2002). Para el optimizador ALPSO, la utilidad DPM (*Dynamic Process Management*) es utilizada, donde las evaluaciones de la función objetivo son realizadas en forma paralela realizando asignaciones de procesos dinámicos. Mientras que para el optimizador SLSQP, la utilidad PGC (*Parallel Gradient Calculation*) es usada con el fin de paralelizar la resolución del gradiente de la función objetivo.

##### 4.4.1. Distribución dinámica de procesos (DPM)

Existen dos maneras de distribuir cada evaluación de las partículas individuales en una misma iteración en los procesadores disponibles: mediante el alojamiento estático de los procesos, y el alojamiento dinámico. En el alojamiento estático, cada procesador tiene asignado un número dado de partículas en cada iteración de antemano. La principal desventaja de esta implementación es que, si alguno de los dos siguientes escenarios se presentan, la eficiencia de paralelizado puede verse muy deteriorada (Schutte et al., 2004):

1. El tamaño del enjambre no es un múltiplo entero del número de procesadores.

2. El tiempo de análisis de la función objetivo depende de la posición de la partícula.

Cualquiera de estos dos escenarios puede resultar en que algunos o la mayoría de los procesadores queden ociosos al acercarse al final de cada iteración del proceso de optimización. Entonces, al encontrarse las cargas de los procesadores desbalanceadas, a medida que se utilizan más unidades de procesadores en el proceso de optimización, la eficiencia de paralelismo puede verse deteriorada en gran medida. Por otro lado, en una distribución dinámica de los procesos implementada en *pyOpt*, se almacenan dinámicamente las evaluaciones de la función objetivo de cada partícula. Esto quiere decir que, en cada iteración, la tarea correspondiente a evaluar la función objetivo de cada partícula es distribuida siguiendo un algoritmo del tipo *master-esclavo*. En dicho algoritmo, el nodo *master* tiene la función de enviar la posición de la partícula a cada procesador que se encuentre libre. Una vez enviada la posición de la partícula, el procesador esclavo evalúa la función objetivo y las restricciones de dicha partícula, y luego envía los resultados de vuelta al nodo *master*. Este proceso se repite hasta que todas las partículas del enjambre de una iteración dada son evaluadas. Si bien esta implementación requiere de un procesador exclusivamente para el manejo del alojamiento y la comunicación de las tareas, a su vez, evita el comportamiento ocioso del resto de los procesadores lo cual se refleja en una mejor eficiencia de paralelismo a medida que se aumenta la cantidad de nodos utilizados.

El esquema quimera fue implementado en conjunto a los algoritmos de optimización en el *cluster Seshat* (Seshat, 2018), el cual se encuentra localizado en las instalaciones del instituto CIMEC donde se lleva a cabo esta tesis. Este *cluster* está compuesto por un total de 69 nodos, donde cada uno consiste de un procesador Intel(R) Xeon(R) CPU E5-1620 v2 @ 3.70GHz (1 CPU x 4 procesadores), 16GB de RAM DDR3 1600 MHz y conexión InfiniBand. En el Anexo B se encuentran los detalles de las llamadas a los optimizadores para hacer uso de sus implementaciones en paralelo.

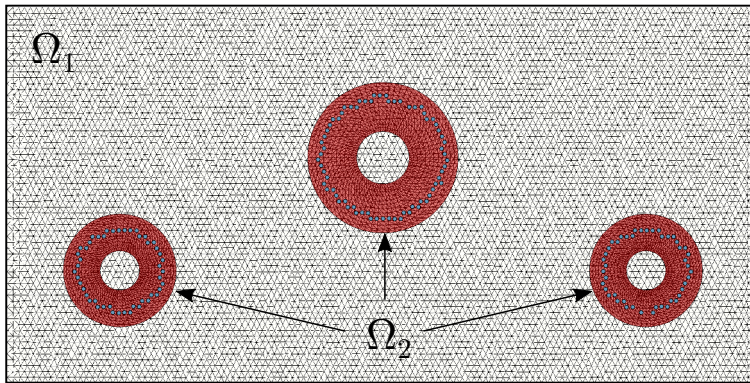
## 4.5. Tratamiento de múltiples dominios con el método quimera

Para abordar problemas de optimización en los cuales múltiples objetos se mueven dentro de un dominio computacional en la búsqueda de una configuración óptima, el algoritmo planteado y utilizado en los capítulos anteriores debe ser extendido para tratar con más de dos dominios. Para lograr esto, existen dos posibilidades:

1. Si los objetos dentro del dominio computacional se encuentran lo suficientemente alejados y no interactúan entre sí, es decir, sólo interactúan con la malla de fondo  $\Omega_1$ , entonces las múltiples mallas que rodean los objetos pueden ser condensadas en una misma malla  $\Omega_2$  y no se requieren modificaciones adicionales en la implementación;
2. Si los objetos interactúan entre sí, como puede ser el caso de dos objetos muy próximos uno con el otro, o más de dos mallas superpuestas con

distintos grados de refinamiento, entonces se debe establecer un orden jerárquico para definir la interpolación entre ellos. Esta implementación no es tan directa, por lo que deben realizarse mayores modificaciones en el código.

Debido a que el desarrollo de la presente tesis está orientada en el empleo del método quimera en la solución numérica de problemas de optimización, en donde en la mayoría de los casos se conoce con antelación que la solución óptima no corresponde a una disposición tan cercana de los objetos, se optó por el primer enfoque.

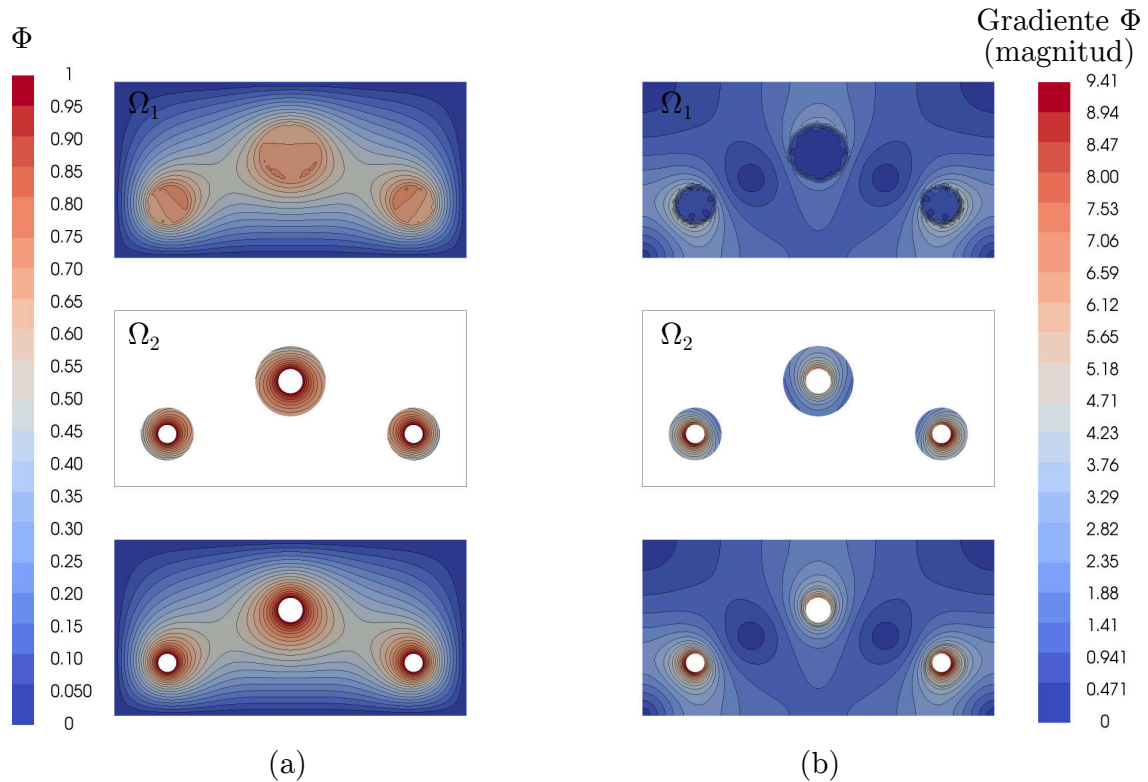


**Figura 4.2:** Convergencia de la función objetivo obtenida por los optimizadores ALPSO y SLSQP en el ejemplo numérico.

Para lograr condensar las mallas de los múltiples objetos en una sola, solo es necesario tomar las conectividades y coordenadas de los nodos de la malla de uno de los objetos como base, luego adicionar las coordenadas de los nodos y las conectividades de la malla de un segundo objeto a la malla base, con la precaución de reenumerar los nodos. Este proceso se debe repetir con la totalidad de las mallas de los objetos.

A modo de ejemplo, en la Fig. 4.2 se puede observar la malla de un dominio  $\Omega_2$  la cual contiene la malla de tres objetos, donde solamente se trabajó en la malla de solo uno, y las restantes se logran a partir del desplazamiento de las coordenadas de sus nodos, ya sea para trasladar el objeto o modificar sus dimensiones. También se puede observar de la Fig. 4.2 cómo el algoritmo de reconocimiento de interfase de interpolación identifica correctamente los nodos de la malla de fondo (nodos celestes) a ser utilizados en la interpolación. En la Fig. 4.3 se visualiza la solución al problema de Laplace con condiciones de borde de tipo Dirichlet homogéneas en la frontera de  $\Omega_1$ , mientras que los objetos poseen una condición de borde de tipo Dirichlet igual a la unidad en su borde interno. Se puede visualizar claramente la continuidad de la solución lograda tanto para  $\phi$  (ver Fig. 4.3a), como para su gradiente (ver Fig. 4.3b).





**Figura 4.3:** Solución de la ecuación de Laplace obtenida por el método quimera con múltiples subdominios: (a) solución de  $\phi$  en  $\Omega_1$  y  $\Omega_2$ , y la solución global, (b) solución del gradiente de  $\phi$  en  $\Omega_1$  y  $\Omega_2$ , y la solución global.

## 4.6. Estrategia de optimización con el método quimera

En el Anexo B, en la Fig. B.7 se encuentra explicado en forma detallada el procedimiento de optimización mediante el uso del método quimera. Del procedimiento se observa una etapa de preproceso, en el cuál tanto las geometrías del dominio computacional  $\Omega_1$ , como la de los objetos a ser optimizados  $\Omega_2$  (si la geometría de los objetos no cambia a lo largo del proceso de optimización), son generadas y malladas sólo una vez. También se distingue de la Fig. B.7 la etapa del método quimera, en el cual se detectan las interfases de interpolación, el cómputo de los operadores de interpolación y el subsecuente análisis por FEM del sistema acoplado.

## 4.7. Ejemplo test

Previo a emplear el esquema propuesto en un problema ingenieril, a modo de evaluar la convergencia y eficiencia que se logra en un problema de optimización en mecánica computacional, se resuelve un problema simple aunque no trivial de optimización con el método quimera, el cual consiste en minimizar la temperatura media de un dominio cuadrangular de dos dimensiones mediante la optimización de la disposición de ocho pasajes refrigerantes que se encuentran dentro del dominio conformando la malla *patch*. Los pasajes refrigerantes parten de una posición completamente

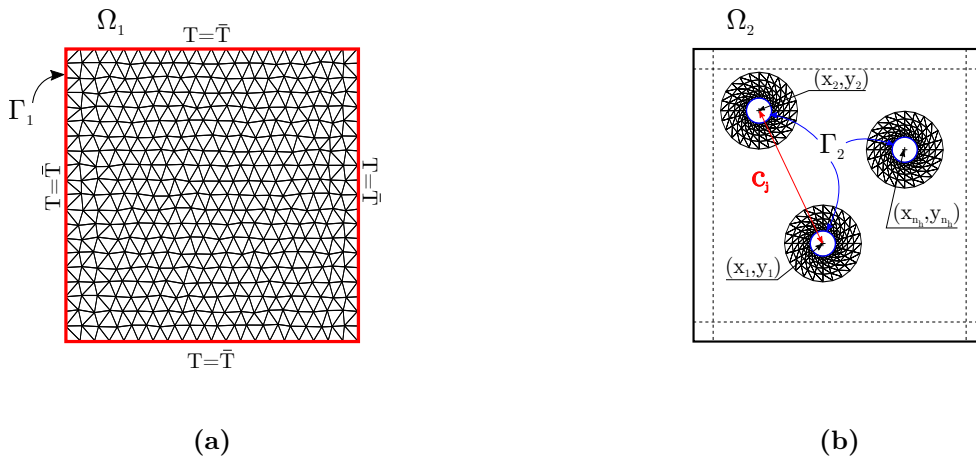
aleatoria.

#### 4.7.1. Configuración del problema de optimización

En el problema se resuelve la ecuación de calor en una placa de dos dimensiones. El valor de la conductividad se considera constante igual a la unidad para ambos subdominios. El borde de la placa tiene impuesta una condición de borde de tipo Dirichlet igual a  $100$  [°C], mientras que en los pasajes refrigerantes tienen impuesta una condición de borde tipo Robin, con un coeficiente de transferencia de  $100$  [ $\text{W m}^{-2} \text{°C}^{-1}$ ], siendo la temperatura del refrigerante de  $0$  [°C] (ver Fig. 4.4). Entonces el problema de optimización queda planteado como

$$\begin{aligned} \min_x \quad & f(\bar{x}) = T_{\text{media}} \\ \text{sujeto a} \quad & g_j(\bar{x}) = c_j - c_{\min} \leq 0 \quad j = 1, \dots, m \\ & x_{i_L} \leq x_i \leq x_{i_U}, \quad i = 1, \dots, n \end{aligned} \quad (4.7)$$

De la Ec. (4.7),  $x_{i_L}$  y  $x_{i_U}$  representan las cotas mínimas y máximas que pueden adquirir las variables. En este caso se adoptaron los valores  $0.1$  y  $0.9$  para  $x_{i_L}$  y  $x_{i_U}$  respectivamente. Estas restricciones fueron adoptadas de manera que los centros de los conductos puedan moverse solamente dentro del dominio cuadrado  $[0.1, 0.9] \times [0.1, 0.9]$ . Estas restricciones fueron impuestas de forma tal que la malla *patch*  $\Omega_2$  permanezca siempre dentro de la malla de fondo. Adicionalmente se restringe la distancia mínima entre pasajes con el fin de evitar la superposición de la malla *patch* (ver Fig. 4.4b). En este ejemplo un total de  $16$  variables de diseño son condensadas en un vector (ver Ec. (4.8)) que representan la posición de los ocho huecos.



**Figura 4.4:** Esquema de la malla de fondo  $\Omega_1$  y de la malla *patch*  $\Omega_2$  utilizadas en el problema de optimización.

$$\bar{x} = \left[ \underbrace{x_1, y_1}_{\text{Hole 1}}, \underbrace{x_2, y_2}_{\text{Hole 2}}, \dots, \underbrace{x_{n_h}, y_{n_h}}_{\text{Hole } n_h} \right] \quad (4.8)$$

### 4.7.2. Parámetros utilizados en la optimización

Los parámetros utilizados en el algoritmo de optimización del enjambre de partículas pueden influir directamente en el desempeño y eficacia del proceso de optimización para un problema dado. Para la elección de los parámetros más adecuados a utilizar en el proceso de optimización se realizó una búsqueda minuciosa en la literatura donde se analizan los efectos que los mismos tienen sobre el proceso de convergencia del algoritmo (Jiang et al., 2007; Shi y Eberhart, 1999; Xueyan y Zheng, 2015). En el Anexo B se detallan los parámetros utilizados durante la optimización con el método quimera, y que resultaron los más adecuados para los problemas abordados en esta tesis.

### 4.7.3. Análisis de la escalabilidad

Tanto el *solver* de enjambre de partículas como el optimizador basado en gradiente fueron utilizados en forma paralela en el *cluster*. A partir del problema de optimización se realizó un estudio de escalabilidad basado en la ley de Amdahl (Hill y Marty, 2008). De los resultados presentados en la Fig. B.11a se puede observar que al utilizar el alojamiento dinámico de procesos implementado en el algoritmo ALPSO se puede obtener un *speedup* muy cercano al lineal. Mientras que para el optimizador SLSQP, al poderse paralelizar sólo el cálculo del gradiente en el proceso de optimización, y al tratarse esta porción de aproximadamente el 77 % del tiempo total de cómputo, el *speedup* teórico se aleja del lineal. Se pudo observar de la Fig. B.11b que cuando la cantidad de variables de diseño es un múltiplo entero de la cantidad de procesadores, la escalabilidad se acerca a la teórica. Mientras que cuando esta condición no se cumple, el aumento de procesadores no muestra prácticamente mejora en los tiempos de optimización.

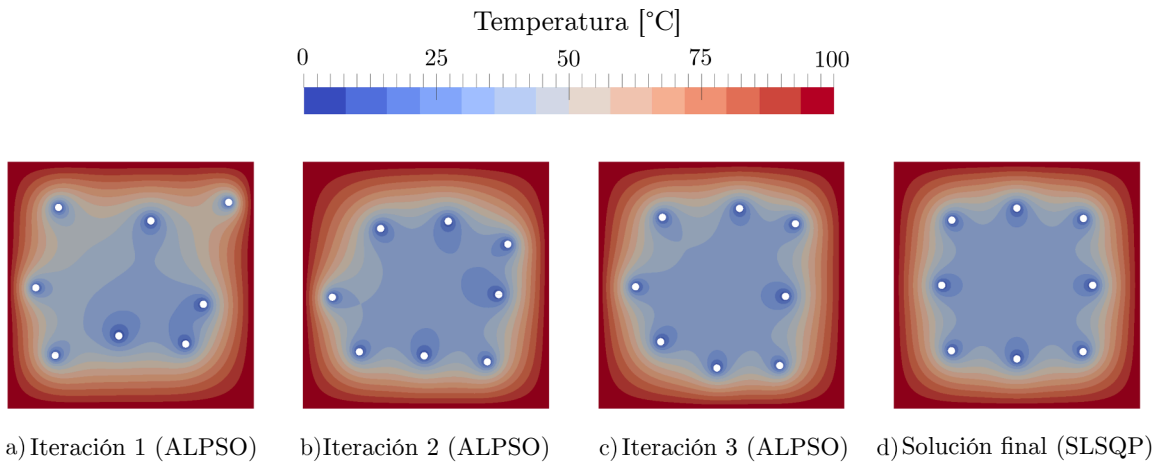
## 4.8. Resultados y conclusiones

En este Capítulo se introdujo la plataforma de optimización *pyOpt* utilizada en el desarrollo de la presente tesis. También se introdujeron algunos de los *solvers* de optimización utilizados implementados en ella, junto con la metodología seguida para resolver un problema de optimización mediante el uso del esquema de mallas solapadas. A fin de evaluar el desempeño de la metodología propuesta, se llevó a cabo la optimización de la disposición de una serie de pasajes refrigerantes dentro de una placa calentada en sus bordes con el fin de disminuir su temperatura media. Si bien no se trata de un problema complicado, resulta de gran interés para ser utilizado a modo de caso *benchmark* de la metodología propuesta.

La peor disposición obtenida por el optimizador en las primeras iteraciones presentó un valor en la temperatura media del dominio de 68.16 [°C], mientras que para la disposición final optimizada,

se consiguió una temperatura media en el dominio de  $58.51$  [°C]. La disposición optimizada de los pasajes refrigerantes de la Fig. 4.5d presenta tres particularidades:

- La distancia entre cada hueco refrigerante y su vecino más próximo permanece constante.
- La disposición final de los pasajes es simétrica con respecto a las diagonales del dominio y a las líneas  $x = 0.5$  e  $y = 0.5$ .
- Los huecos se ubican próximos al borde de la placa, lugar donde se encuentra la mayor temperatura del dominio.



**Figura 4.5:** Posición de los pasajes refrigerantes durante el proceso de optimización.

## Capítulo 5

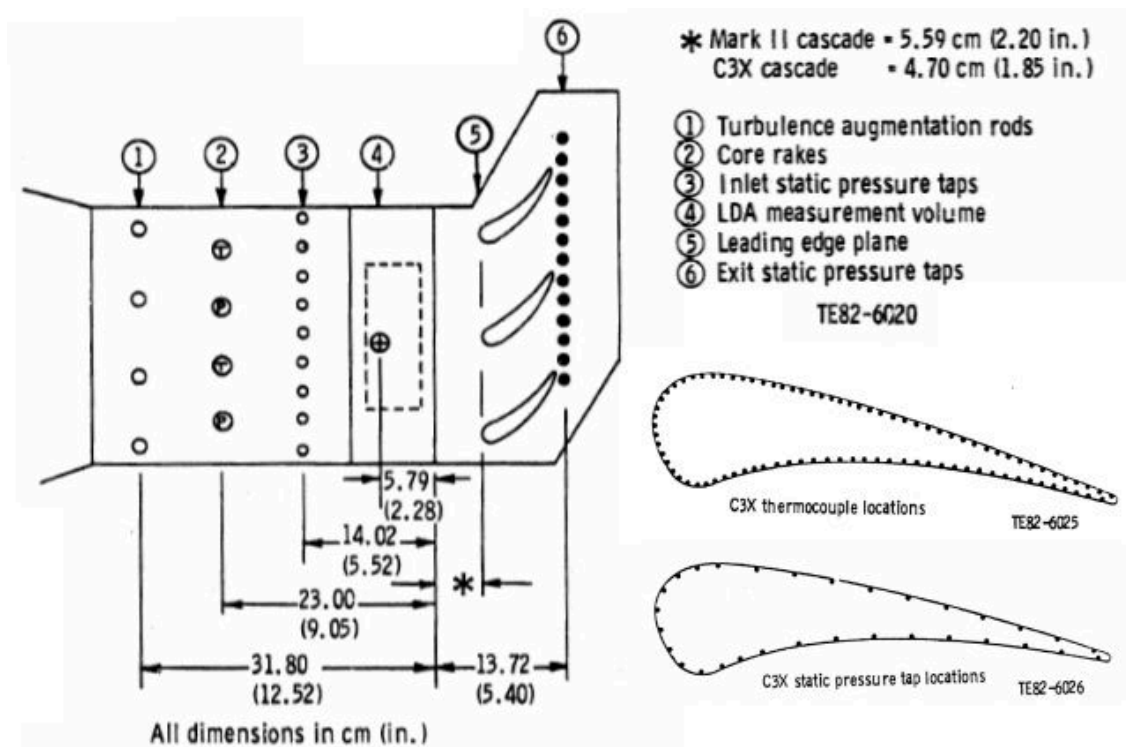
# Aplicación del método quimera para una optimización del sistema de refrigeración de un álabe

### 5.1. Introducción

Como se mencionó en capítulos anteriores, el método quimera tiene propiedades que lo hacen propicio para abordar ciertos problemas de optimización en mecánica computacional. En este Capítulo la metodología desarrollada es aplicada a un problema ingenieril de optimización. Concretamente, se presenta su utilización en conjunto con optimizadores de *pyOpt* para mejorar la disposición relativa de los conductos de refrigeración del álabe de una turbina de gas con el fin de minimizar la temperatura media del perfil y así disminuir las solicitaciones de origen térmico que se originan. Como es conocido en termodinámica, el aumento del rendimiento de una turbina de gas depende casi estrictamente de la temperatura máxima de los gases de combustión. Dicha temperatura se encuentra restringida por la resistencia termomecánica de los materiales de sus álabes. Para abordar este problema se propone un esquema de mallas solapadas tipo quimera aplicado a la optimización de la conducción térmica de un álabe con el objetivo de minimizar su temperatura media. El método quimera consiste en una malla de fondo fija, cubriendo todo el dominio, y mallas más finas rodeando pasajes refrigerantes que se mueven libremente sobre el perfil. La estrategia de optimización consiste en utilizar un algoritmo basado en población para acercarse a un probable mínimo global, y luego refinar la solución utilizando un algoritmo basado en gradiente. Tanto el cómputo de los gradientes como las evaluaciones de la función objetivo son realizados utilizando computación distribuida.

## 5.2. Descripción del experimento

Hylton et al. (1983) trabajaron en un experimento realizado en los laboratorios de la NASA en los cuales disponían de un arreglo de tres álabes de una turbina sometidos a distintas condiciones de trabajo (ver Fig. 5.1). La superficie de los álabes se encontraba instrumentada por aproximadamente 80 termocuplas y 30 orificios para la medición de la presión estática. Todo el instrumental se encontraba colocado sobre el plano medio de los álabes. Además, cada tubo refrigerante se encontraba también instrumentado con termocuplas y orificios de presión estática dispuestos en su entrada y salida. Los valores de los flujos máscicos para cada tubo fueron obtenidos utilizando placas orificios.

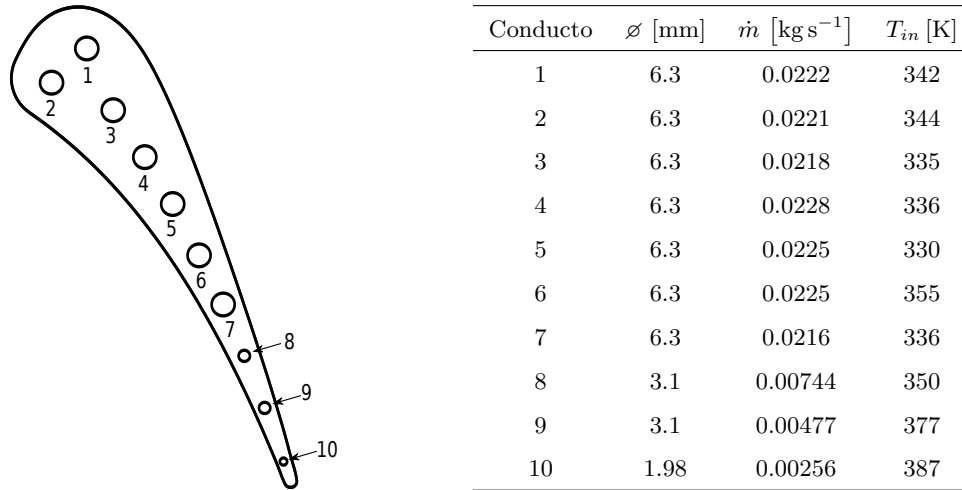


**Figura 5.1:** Configuración de los álabes durante los ensayos experimentales de Hylton et al. (1983) (Izq.). Ubicación de los instrumentos de medición sobre los álabes durante los ensayos (Der.)<sup>2</sup>.

## 5.3. Geometría del ábabe y condiciones de carga

Dos geometrías diferentes fueron analizadas durante las experiencias en la NASA. En esta tesis se escogió el ábabe C3X para su optimización. Los parámetros de funcionamiento tales como la distribución de la temperatura sobre la superficie del perfil, el flujo máscico, y la temperatura del refrigerante, fueron elegidos de la experiencia *RUN 157*. Tanto la geometría del ábabe, como

<sup>2</sup>Imagen del trabajo de Hylton et al. (1983).



**Figura 5.2:** Geometría del álabe C3X (Izq.), y datos del refrigerante (Der.).

así también las condiciones del fluido refrigerante para cada pasaje se encuentran detalladas en la Fig. 5.2. Debido a que resolver un análisis de Transferencia de Calor Conjugado (CHT) para cada evaluación de la función objetivo sería computacionalmente muy costoso, en el siguiente estudio se realizan algunas suposiciones simplificadoras. El fluido refrigerante se asume que presenta un comportamiento estacionario, es decir, sus propiedades físicas no varían en las simulaciones numéricas. También el flujo másico y la temperatura son consideradas constantes y uniformes en cada pasaje. En base a esto, en este problema se propone modelizar la transferencia de calor hacia el fluido refrigerante mediante una condición de borde convectiva. Para esto, el coeficiente pelicular sobre la superficie de los pasajes se estima a través del número de Nusselt dado por la correlación de Dittus-Boelter. El desarrollo detallado para estimar dicho coeficiente en cada conducto se puede observar en el Apéndice B. En la Tabla 5.1 se detallan las condiciones físicas del fluido refrigerante para cada conducto calculadas en esta tesis. Respecto a la condición de borde sobre el perfil, se utiliza una temperatura fija sobre su superficie de acuerdo a datos experimentales obtenidos por las termocuplas dispuestas sobre el álabe durante los experimentos desarrollados en laboratorio. Para definir adecuadamente la temperatura impuesta sobre todos los nodos de la superficie de la malla del álabe, se realizó una interpolación cúbica de los valores obtenidos en forma experimental (ver Fig. B.13).

## 5.4. Descripción del problema de optimización

El problema de optimización de la configuración de los pasajes refrigerantes del álabe puede plantearse matemáticamente de la siguiente forma:

Conducto	L/∅	Re	$k_f$ [W m <sup>-1</sup> K <sup>-1</sup> ]	$\lambda$ [Pa sec]	Pr	Nu	$h$ [W m <sup>-2</sup> K <sup>-1</sup> ]
1	12.10	211057	2.1258e-05	0.0310067	0.768	376.15	1851.30
2	12.10	209198	2.13504e-05	0.0311414	0.768	373.50	1846.23
3	12.10	210478	2.09324e-05	0.0305318	0.768	375.32	1818.94
4	12.10	219643	2.09792e-05	0.0306000	0.768	388.34	1886.23
5	12.10	219699	2.06978e-05	0.0301896	0.768	388.42	1861.32
6	12.10	208078	2.18537e-05	0.0318756	0.768	371.90	1881.66
7	12.10	208082	2.09792e-05	0.0306000	0.768	371.90	1806.39
8	24.58	141301	2.16259e-05	0.0315433	0.768	272.87	2776.55
9	24.58	85787	2.28372e-05	0.0333101	0.768	183.05	1966.96
10	38.48	70730	2.32745e-05	0.0339479	0.768	156.86	2689.51

**Tabla 5.1:** Propiedades del fluido refrigerante dentro de los conductos.

$$\begin{aligned}
& \underset{\bar{x}}{\text{mín}} && T_{\text{media}}(\bar{x}) \\
\text{sujeto a} && g_j(\bar{x}) = c_j - c_{\min} \geq 0, && j = 1, \dots, m_e, \\
&& \nabla T(\bar{x}) \leq \nabla T_{\max}, && j = m_e + 1, \dots, m, \\
&& x_{i_L} \leq x_i \leq x_{i_U}, && i = 1, \dots, n
\end{aligned} \tag{5.1}$$

La función objetivo a minimizar es la temperatura media del perfil  $T_{\text{media}}$  y, como restricción, el gradiente de la temperatura no debe superar un valor específico  $\nabla T_{\max}$ . Como las tensiones termomecánicas de un componente dependen directamente de los gradientes térmicos, para asegurar valores aceptables de dichas sollicitaciones en el álabo, los gradientes térmicos se imponen como una restricción en el problema de optimización. Para los conductos 1 y 2, es decir, para  $i = 1, 2$ , las variables  $r_i$  y  $\theta_i$  corresponden a la posición de sus centros en coordenadas polares del dominio transformado (ver Fig. B.14). Mientras que para el resto de los conductos, las variables  $x_i$  e  $y_i$  corresponden a las coordenadas cartesianas del centro del pasaje  $i$ , para  $i = 3, 4, \dots, 10$ . Las cotas máximas y mínimas  $x_{i_U}$  y  $x_{i_L}$  que puede tomar cada variable vendrán dadas por la transformación de coordenadas explicada en la siguiente sección.

#### 5.4.1. Transformación de los dominios de las variables

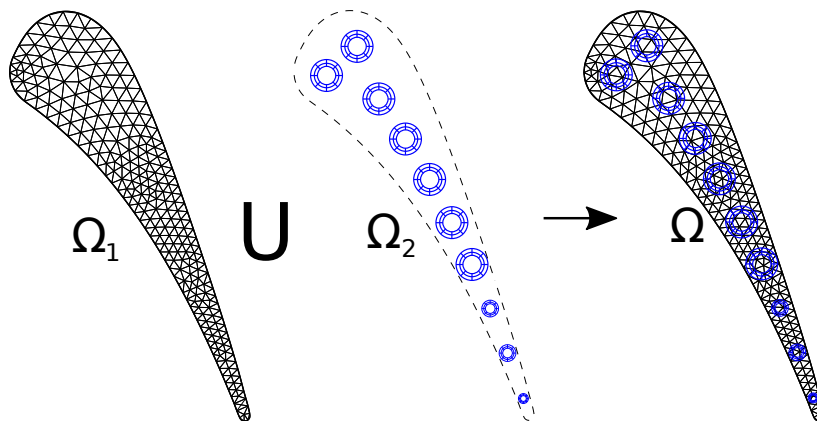
A modo de restringir el movimiento de los tubos en la optimización, se realiza una transformación de coordenadas para cada variable de diseño similar a la realizada por Wang et al. (2015), donde el centro de cada pasaje tiene permitido moverse dentro de cierta región. Esta transformación tiene dos principales objetivos, por un lado, restringir el movimiento de los conductos a modo de evitar la superposición entre ellos y, por otro lado, restringir el desplazamiento de los mismos de manera



que no exceda el dominio computacional delimitado por el álabe. En el Anexo B se detallan las transformaciones realizadas para cada subdominio.

## 5.5. Convergencia en malla

Una vez configurado el modelo de FEM, se realizó un estudio de convergencia en malla. Para este estudio se utilizó una temperatura homogénea en la superficie del perfil de 600 [K]. No se utilizaron las temperaturas obtenidas en las termocuplas para evitar que la interpolación introduzca errores en el estudio de convergencia. Se analizaron distintas mallas no estructuradas con un número creciente de elementos triangulares, variando de 5600 a 77000 celdas. La malla de fondo del perfil se genera usando la plataforma libre SALOME (Ribes y Caremoli, 2007.), mientras que la malla superpuesta de los pasajes se genera por un script ad-hoc. El número total de celdas se obtiene de la suma total de elementos de la malla de fondo y elementos de la malla *patch*. Para cada caso, la malla *patch* estaba compuesta por aproximadamente la quinta parte de celdas respecto a la malla de fondo. A modo de representación se puede ver un esquema de las mallas utilizadas en la Fig. 5.3 que, al superponerlas y resolver, se tiene la solución a un único sistema.

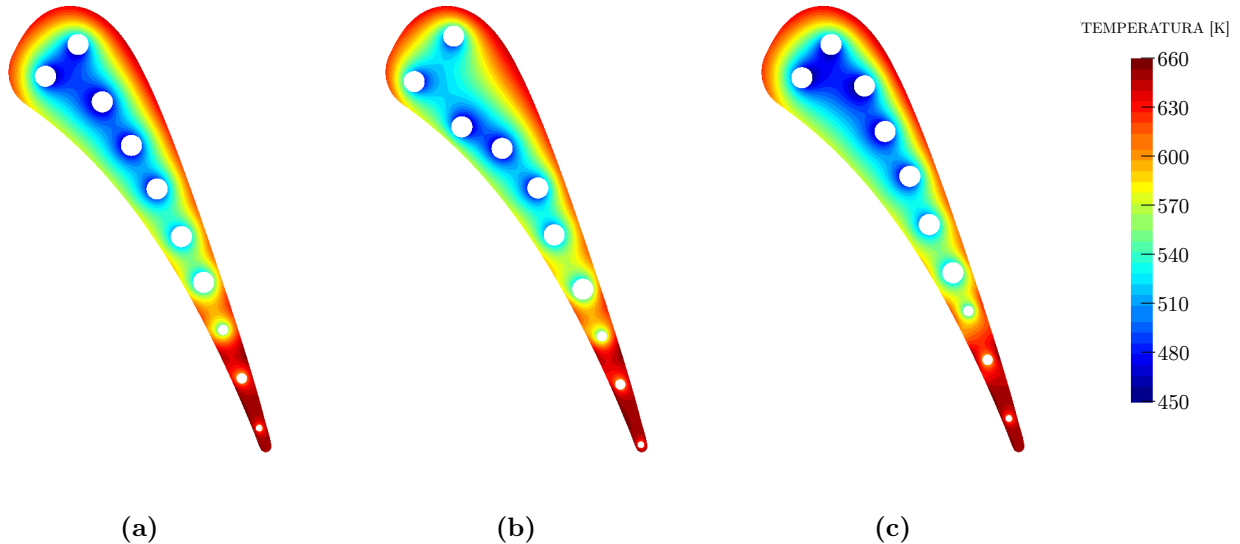


**Figura 5.3:** Representación conceptual de la malla de fondo ( $\Omega_1$ ) y la malla *patch*  $\Omega_2$  utilizada por el esquema quimera .

## 5.6. Resultados

Un total de aproximadamente mil evaluaciones de la función objetivo fueron realizadas por el optimizador cubriendo prácticamente el dominio computacional completo (ver Fig. B.17). De los resultados de la convergencia de la optimización presentados en el Anexo B puede apreciarse que, para las primeras evaluaciones partiendo de posiciones aleatorias efectuadas por el optimizador ALPSO, la peor disposición de los conductos refrigerantes analizada induce una temperatura media

del álabe de 566 [K] (ver Fig. B.19b). A medida que el optimizador converge a la solución, la temperatura media del álabe alcanza un mínimo de 558.17 [K], mejorando ya en este punto el valor de temperatura media obtenida por la disposición de referencia de 560.81 [K] (ver Fig. B.19a). Al iniciar el refinamiento de la solución de optimización mediante el optimizador SLSQP, se logran mínimos en la temperatura media del álabe de 557.90 [K] después de pocas iteraciones (ver Fig. B.19c).



**Figura 5.4:** Comparación de los resultados obtenidos en la temperatura para: (a) disposición de referencia, (b) peor condición obtenida y (c) disposición optimizada.

## 5.7. Conclusiones

En este Capítulo se presentó la aplicación del esquema quimera en conjunto con los optimizadores de la plataforma *pyOpt* para abordar el problema de optimización de la configuración de los conductos refrigerantes del álabe de una turbina de gas. El esquema permite la generación de la malla del perfil y de los conductos refrigerantes en forma independiente, reduciendo así los esfuerzos de mallado mientras que la calidad de la solución se mantiene. Como caso test se utilizó la geometría del álabe C3X. Tanto las condiciones de contorno como las propiedades del álabe y del aire refrigerante fueron obtenidas de datos experimentales. En base a esta información se estimó para cada conducto un coeficiente pelicular que permitió modelar la transferencia térmica desde el perfil mediante una condición de borde convectiva. Cada malla de los conductos podía moverse libremente dentro de una zona delimitada en el álabe en búsqueda de la configuración que minimice la temperatura media del componente. La solución obtenida del proceso de optimización logró mejorar el desempeño térmico de la solución de referencia, reduciendo en 3[K] la temperatura media mínima

del perfil. La estrategia de optimización utilizando el método quimera y técnicas de paralelismo presentó buenos resultados respecto a técnicas de remallado tradicionales utilizadas comúnmente para abordar este tipo de problemas.



## Capítulo 6

# Optimización con el método quimera en dominios tridimensionales

### 6.1. Introducción

Hasta este punto, todos los ejemplos numéricos y aplicaciones del método quimera tratados en esta tesis estuvieron orientados a la solución de problemas en dominios con dos dimensiones. Su extensión a dominios tridimensionales permite tratar problemas de mayor complejidad y con mayor cantidad de elementos, aprovechando las ventajas del acople *matrix-free* y la paralelización del problema de optimización. En este Capítulo, se extiende la implementación del método para poder abordar problemas de optimización en tres dimensiones. Se tratarán cuestiones como ser un algoritmo de interpolación de alto orden, reconocimiento de la interfase de interpolación de la malla de fondo, estudio de la convergencia en malla y de la convergencia del *solver* iterativo. Adicionalmente, se resuelve un ejemplo numérico de optimización de transferencia de calor a modo de evaluar la implementación del esquema con el optimizador de enjambre de partículas. Finalmente, se utiliza la metodología propuesta para una solución de un problema ingenieril en el cual se busca optimizar la refrigeración de un molde de inyección, y así lograr una mayor productividad de piezas inyectadas.

### 6.2. Algoritmo de interpolación de alto orden para mallas 3D

Como se mostró en los ejemplos numéricos presentados en el Anexo A, la convergencia espacial de la solución obtenida resultado del acople es afectada por el esquema de interpolación. De los ejemplos presentados en dos dimensiones se pudo observar que, para conservar el orden de convergencia de FEM, es decir  $\mathcal{O}(h^2)$ , se requiere una interpolación de tercer orden. Si bien en el Anexo A se explica

en forma detallada el algoritmo de interpolación de alto orden propuesto en esta tesis para mallas en dos dimensiones, algunos detalles de la implementación para mallas tridimensionales se detallan a continuación.

### 6.2.1. Operadores de interpolación

Se considera que una malla compuesta de tetraedros cubre un dominio  $\Omega_1$  y que los valores de una función suave  $\phi$  son conocidos en los vértices de los elementos. Ahora se supone que existen ciertos puntos de prueba los cuales se encuentran sobre la frontera de  $\Omega_2$  y se requiere obtener una interpolación de tercer orden de  $\phi$  en ellos. La Fig. 6.1 muestra la intersección de un nodo de prueba  $x_j$  con un elemento de la malla  $\Omega_1$ .

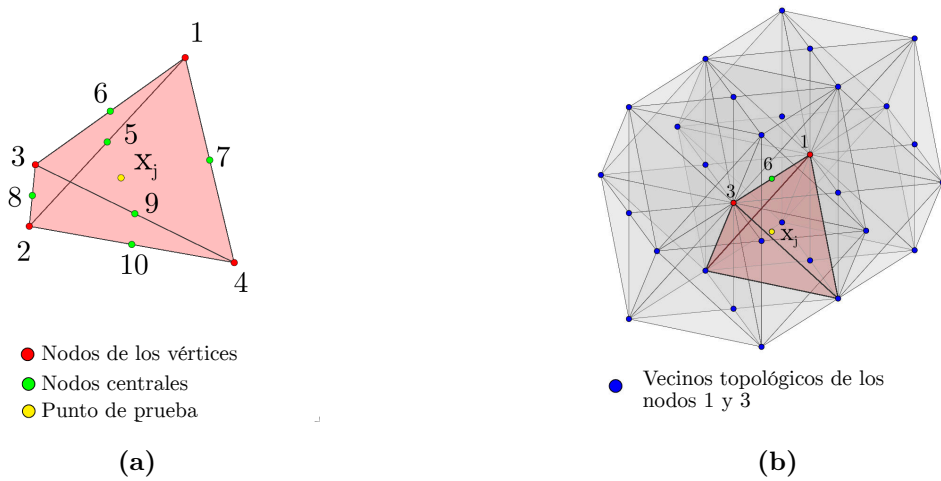


Figura 6.1: Interpolación de alto orden sobre mallas 3D.

Como ya se mencionó anteriormente, los valores de  $\phi$  en los vértices del elemento donde se encuentra el vértice de prueba  $x_j$  (nodos en rojo 1-4 de la Fig. 6.1a) se asumen conocidos y se busca reconstruir los valores en los centros de aristas del tetraedro en rojo (nodos en verde 5-10 de la Fig. 6.1a), ya que para obtener la interpolación de alto orden sobre un nodo interior al elemento, los valores de  $\phi$  en los nodos 1-10 son requeridos. De manera análoga al esquema de interpolación en 2D, para reconstruir los valores en los nodos centrales se realiza un ajuste por mínimos cuadrados utilizando los valores en los nodos de los vértices, así como también el de sus vecinos topológicos. Entonces, suponiendo que se quiere reconstruir el valor del nodo 6 de la Fig. 6.1a, se debe realizar un ajuste por cuadrados mínimos utilizando los valores en los nodos de los vértices 1 y 3, así como también el de sus vecinos topológicos marcados en azul en la Fig. 6.1b. Para una malla de triángulos en 2D, si el punto de prueba se posiciona sobre un elemento central y no de borde de la malla  $\Omega_1$ , generalmente la cantidad de vecinos topológicos alcanza los 10 nodos, mientras que en una malla 3D de elementos tetraédricos, la cantidad de vecinos es tres veces mayor alcanzando los 30 nodos

aproximadamente. Luego, sea  $\tilde{\phi}$  la reconstrucción de  $\mathcal{O}(h^3)$ :

$$\tilde{\phi}(x, y, z) = a + bx + cy + dz + ex^2 + fy^2 + gz^2 + hxy + ixz + jyz. \quad (6.1)$$

Si es aplicada a todos los nodos vecinos, la reconstrucción queda planteada como:

$$\tilde{\phi} = \Psi \mathbf{a}. \quad (6.2)$$

siendo

$$\tilde{\phi} = \begin{bmatrix} \tilde{\phi}_1 \\ \vdots \\ \tilde{\phi}_m \end{bmatrix}, \in \mathbb{R}^m \quad \mathbf{a} = \begin{bmatrix} a \\ b \\ \vdots \end{bmatrix}, \in \mathbb{R}^{n_a} \quad \Psi = \begin{bmatrix} 1 & x_1 & y_1 & z_1 & x_1^2 & \cdots & y_1 z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & y_m & z_m & x_m^2 & \cdots & y_m z_m \end{bmatrix}, \in \mathbb{R}^{m \times n_a} \quad (6.3)$$

donde  $m$  es el número total de nodos que participan en la reconstrucción, es decir, los nodos de los vértices con todos sus vecinos topológicos (generalmente  $m \approx 30$ ), y  $n_a = 10$  es el número de parámetros utilizados en la reconstrucción de  $\tilde{\phi}(x, y, z)$ . Debido a que la reconstrucción no puede ser impuesta en la totalidad de los  $m$  nodos, pues el sistema quedaría sobredeterminado, la elección del vector de coeficientes  $\mathbf{a}$  se realiza de manera que la reconstrucción resultante se ajuste lo mejor posible por mínimos cuadrados a los valores de  $\phi$ . Se debe tener en cuenta que los valores de los nodos de los vértices, es decir de los nodos 1 y 3, son impuestos en forma exacta. Para lograr esto se utilizan multiplicadores de Lagrange y si bien el procedimiento junto a su formulación son los mismos que los seguidos en el caso 2D detallado en el Anexo A, aquí son repetidos para una mejor presentación de la presente tesis.

Entonces, para comenzar, se separa el total de los nodos en dos grupos disjuntos: el grupo  $E$  que contiene a los nodos de los vértices, es decir  $E = \{1, 3\}$ , y el grupo  $N$  el cual contiene el total de los nodos utilizados para la reconstrucción  $M$ , con excepción de los nodos del grupo  $E$ , es decir que  $N = M - E$ . De esta forma los parámetros de la reconstrucción se determinan de forma tal que el ajuste sea exacto en los nodos de los vértices y a su vez sean óptimos en el resto de los vecinos topológicos, es decir que:

$$\mathbf{a} = \operatorname{argmin} \sum_N |\tilde{\phi}_k - \phi_k|^2, \quad (6.4)$$

sujeto a las restricciones

$$\tilde{\phi}_k = \phi_k, \quad \text{for } k \in E. \quad (6.5)$$

Luego, utilizando multiplicadores de Lagrange, el funcional lagrangiano a ser optimizado se plantea como:

$$\mathcal{L}(\mathbf{a}, \boldsymbol{\lambda}) = \|\Psi_N \mathbf{a} - \phi_N\|^2 - \boldsymbol{\lambda}^T (\Psi_E \mathbf{a} - \phi_E) \quad (6.6)$$

Entonces, las ecuaciones a optimizar son:

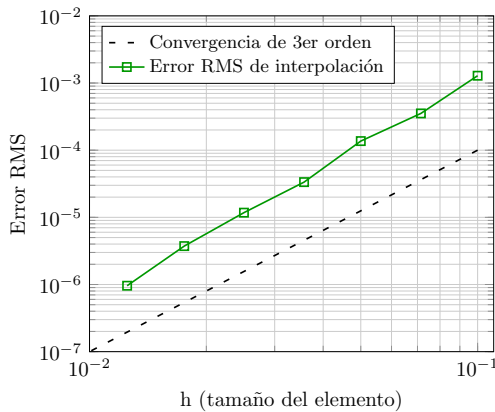
$$\underbrace{\begin{bmatrix} (\Psi_N^T \Psi_N) & \Psi_E^T \\ \Psi_E & \mathbf{0} \end{bmatrix}}_{\mathbf{L}} \begin{bmatrix} \mathbf{a} \\ \lambda \end{bmatrix} = \begin{bmatrix} \Psi_N^T \phi_N \\ \phi_E \end{bmatrix}. \quad (6.7)$$

Al resolver el sistema planteado se obtienen los parámetros  $\mathbf{a}$  a ser utilizados para la reconstrucción del nodo central. El proceso se debe repetir para la reconstrucción del resto de los nodos centrales 5, 7, 8, 9 y 10.

### 6.2.2. Error de interpolación

Con el objetivo de evaluar la convergencia de la interpolación propuesta en mallas no estructuradas tridimensionales, se realizó un estudio similar al realizado en el Anexo A para evaluar el interpolador de dos dimensiones. Con dicho fin, se generaron mallas no estructuradas con un número creciente de tetraedros en un dominio cúbico de longitud unitaria. Luego, se generan en forma aleatoria  $N = 1000$  puntos de prueba distribuidos dentro del dominio y la función de prueba  $\phi(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$  es computada en los nodos de los tetraedros que luego es interpolada en los puntos de prueba. De la misma manera que para las evaluaciones de las interpolaciones sobre mallas 2D, el error RMS se computa como:

$$\epsilon_{\text{RMS}}^2 = \frac{1}{N} \sum_{j=1}^N |\tilde{\phi}_j - \phi_j|^2 \quad (6.8)$$



**Figura 6.2:** Error RMS de la interpolación propuesta en dominios 3D.

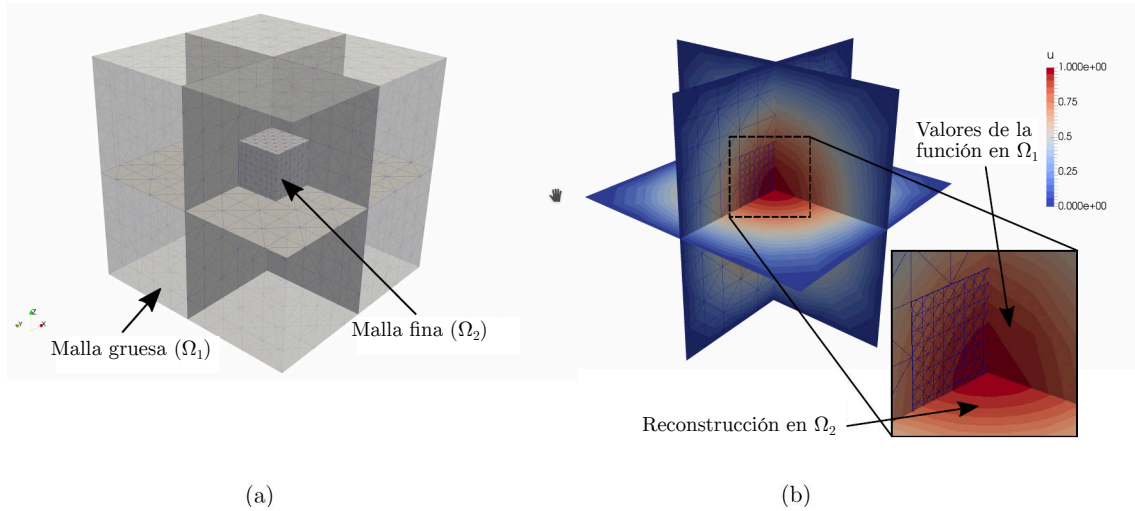
Malla	h	Nodos	Elementos	$\epsilon_{\text{RMS}}^2$
1	0.100	1331	5000	1.2844e-03
2	0.071	3375	13720	3.5271e-04
3	0.050	9261	40000	1.3658e-04
4	0.035	24839	109760	3.3460e-05
5	0.025	68921	320000	1.1744e-05
6	0.017	195112	925965	3.7306e-06
7	0.012	531441	2560000	9.5846e-07

**Tabla 6.1:** Resultados del estudio de convergencia obtenidos por el interpolador de alto orden en mallas tridimensionales.

En la Tabla 6.1 se muestran los datos de las mallas utilizadas, mientras que en la Fig. 6.2 se puede apreciar que el orden de convergencia obtenido por el esquema es efectivamente  $O(h^3)$ . También se puede observar en la Fig. 6.3 el resultado de interpolar la función de prueba  $\phi(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$  calculada en forma exacta sobre la malla gruesa  $\Omega_1$  sobre los nodos de una



mallas más finas con tamaño de elemento cuatro veces menor. De la figura se observa una correcta reconstrucción de los valores de la función sobre la malla fina y cómo se suaviza la solución al reconstruir dichos valores con el presente interpolador de alto orden.



**Figura 6.3:** Reconstrucción de valores de la función de prueba  $\phi(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$  impuesta en una malla gruesa ( $\Omega_1$ ), sobre una malla más fina ( $\Omega_2$ ).

### 6.3. Reconocimiento automático de frontera de interpolación

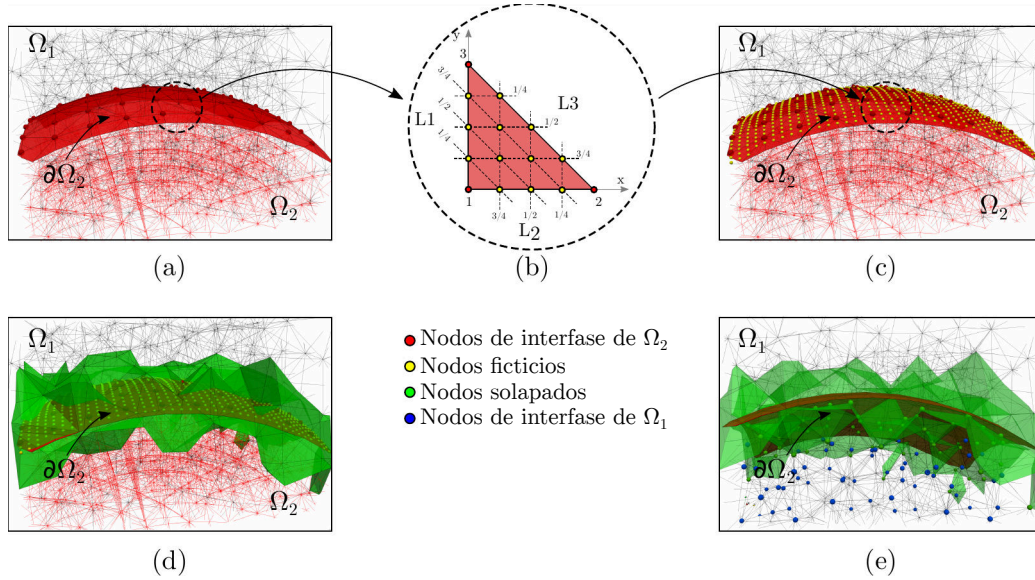
El algoritmo del proceso de búsqueda de nodos que define la interfase de interpolación sobre la malla de fondo en dominios completamente solapados para mallas tridimensionales puede plantearse en forma similar al utilizado en mallas de dos dimensiones, aunque algunas cuestiones, como ser la generación de los nodos ficticios sobre mallas de superficies tridimensionales, no son directas, por lo que se trata a continuación.

Se consideran dos dominios solapados  $\Omega_1$  y  $\Omega_2$ . En la Fig. 6.4 se puede observar una fracción de ambos dominios. El proceso de detección de los nodos de interpolación se puede resumir en los siguientes pasos:

1. Teniendo la malla de superficie de  $\partial\Omega_2$ , se recorren todos los elementos triangulares de la frontera generando los nodos ficticios (en amarillo) para lograr una mejor definición de  $\partial\Omega_2$ .
2. Con la malla de  $\Omega_1$  (conectividades y coordenadas de los nodos) y las coordenadas de los nodos de interfase (en rojo) y de los nodos ficticios, se realiza una búsqueda de los elementos de  $\Omega_1$  en los que se alojan dichos nodos. Estos elementos se representan en verde en la Fig. 6.4d y pertenecen a un grupo  $S$ .
3. Se recorren todos los nodos de los elementos del grupo  $S$  y solo aquellos que se encuentran contenidos dentro del dominio  $\Omega_2$  se agrupan en el conjunto  $T$ . Estos nodos, denominados

como *nodos solapados*, se encuentran representados en verde en la Fig. 6.4e y se encontrarían en la primera capa.

- Se realiza una búsqueda de los vecinos topológicos de los nodos de  $T$ , y sólo aquellos que se encuentren contenidos dentro de  $\Omega_2$  y no pertenezcan a  $T$ , conforman el grupo de *nodos de interfaz* de  $\Omega_1$  representados en azul en la Fig. 6.4e y se encontrarían dispuestos en la segunda capa.



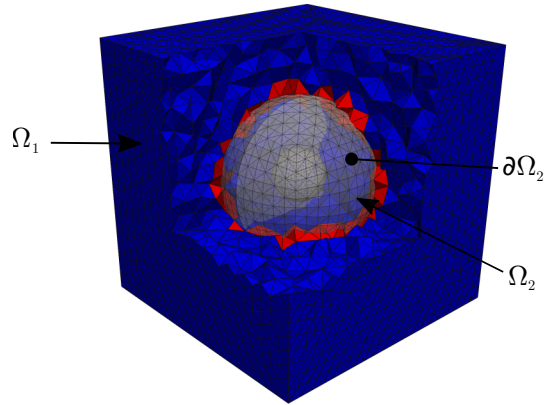
**Figura 6.4:** Proceso generación de nodos ficticios sobre  $\partial\Omega_2$  y detección de los nodos de interpolación sobre la malla de fondo  $\Omega_1$ . Se consideran dos mallas completamente solapadas  $\Omega_1$  y  $\Omega_2$

En la Fig. 6.5 se observa una correcta detección de los elementos pertenecientes a  $\Omega_1$  que son intersectados por la frontera  $\partial\Omega_2$ . Una correcta detección de dichos elementos hace referencia a que una capa continua de elementos deben ser detectados, como puede observarse en la Fig. 6.6b, desvinculando de esta forma los nodos de  $\Omega_1$  internos a  $\Omega_2$  del resto de nodos externos a  $\Omega_2$ . Para esto, es fundamental el agregado de nodos ficticios en el borde de  $\Omega_2$ , ya que si no se utilizan, puede suceder que nodos huecos del dominio  $\Omega_1$  en los cuales la solución no es relevante, queden vinculados a nodos independientes los cuales si pertenecen a la solución global, pudiendo introducir errores en ellos (ver Fig. 6.6a).

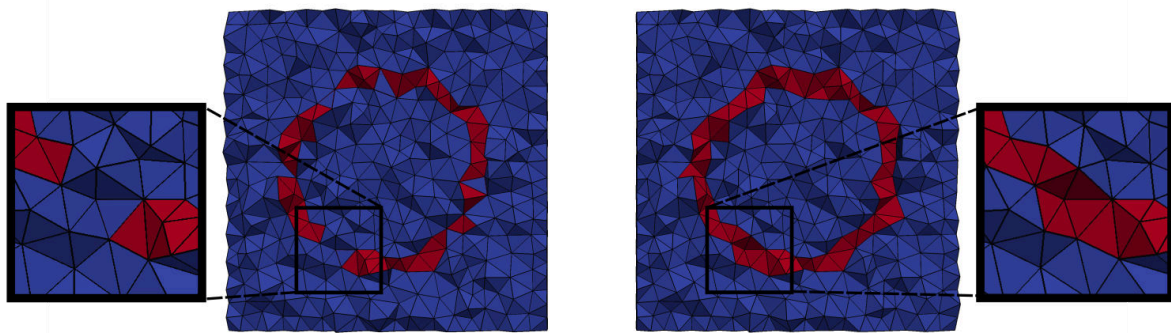
## 6.4. Convergencia y error del método quimera en 3D

En el ejemplo numérico tridimensional del Anexo A se comparan la convergencia del solver iterativo y el error en la norma  $L_2$  de cuatro mallas que poseen similar número de grados de libertad:

- Una malla conforme con tamaño de elemento prácticamente constante en todo el dominio.



**Figura 6.5:** Correcta detección de elementos en  $\Omega_1$  (tetraedros rojos) intersectados por  $\partial\Omega_2$ .



(a) Sin utilización de nodos ficticios

(b) Con utilización de nodos ficticios

**Figura 6.6:** Elementos de  $\Omega_1$  que son intersectados por  $\partial\Omega_2$  en una sección de la Fig. 6.5. Se observa que al no utilizar nodos ficticios, nodos internos de  $\Omega_1$  pueden quedar vinculados a nodos *huecos*, pudiendo introducir errores en la solución

- Una malla conforme con tamaño de elemento no homogéneo la cual se refina hacia el centro del dominio donde se encuentran los mayores gradientes.
- Una malla quimera compuesta por dos dominios  $\Omega_1$  y  $\Omega_2$ , la cual es generada a partir de la malla conforme no homogénea mediante la rotación y traslación de elementos internos en el dominio.
- Una malla quimera compuesta por dos dominios  $\Omega_1$  y  $\Omega_2$  generadas en forma independiente, es decir,  $\Omega_1$  es una malla con tamaño de elemento homogéneo que cubre enteramente el dominio computacional, mientras que  $\Omega_2$  es también generada en forma independiente, de forma estructurada, con un refinamiento progresivo hacia la zona de altos gradientes.

De los resultados obtenidos en un estudio del error y convergencia en malla para el problema numérico presentado en el Anexo A, similares a los obtenidos para mallas de dos dimensiones, se resaltan los siguientes aspectos:

- El acople de los dominios quimera requiere de menor cantidad de iteraciones utilizando BiCGS-

tab, comparado con un problema de malla conforme utilizando CG.

- La convergencia en malla de FEM se mantiene durante el acople, es decir  $\epsilon \approx \mathcal{O}(N^{-2/3})$  de la solución, y  $\epsilon \approx \mathcal{O}(N^{-1/3})$  de su derivada, cuando se utiliza el algoritmo de interpolación de alto orden. Al utilizar un interpolador de bajo orden, el error en la norma  $L_2$  aumenta y la convergencia se pierde.
- Menor error en la norma  $L_2$  es conseguido por la malla quimera regular, ya que al poder generar las mallas en forma independiente, uno puede controlar fácilmente la regularidad de los elementos y, consecuentemente, la calidad de la malla.

## 6.5. Test de optimización térmica en un dominio de tres dimensiones

Con el fin de evaluar el desempeño del método quimera acoplado con los optimizadores implementados en pyOpt, a continuación se resuelve un problema térmico con geometrías relativamente sencillas, en el cual se requiere optimizar la posición de cuatro esferas las cuales tienen impuestas una temperatura relativamente elevada en su superficie, con el fin de maximizar la temperatura media del dominio, mientras que este último se encuentra disipando calor hacia el medio mediante una condición de convección. A continuación se describen las ecuaciones de gobierno del problema térmico y de optimización.

### 6.5.1. Configuración del problema

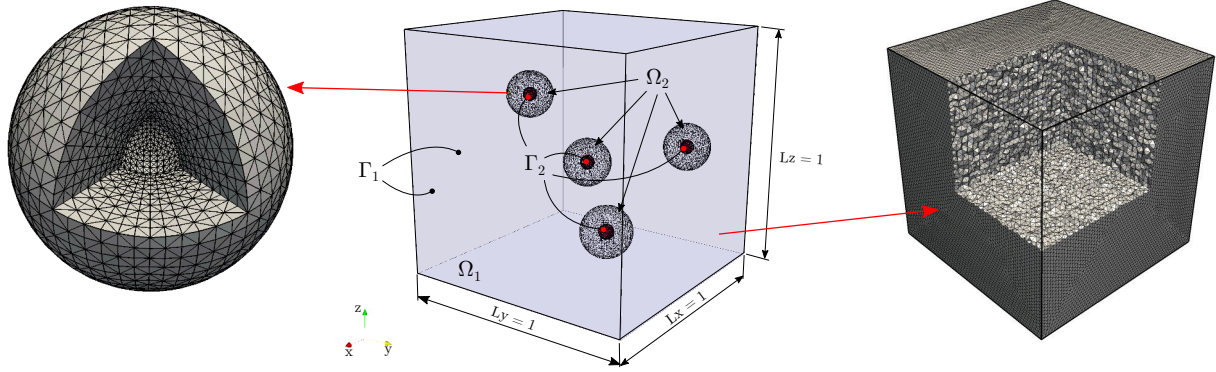
#### 6.5.1.1. Ecuaciones de gobierno del problema térmico

Es considerada la ecuación de transmisión de calor en régimen estacionario en dominios tridimensionales dada por la Ec. (6.9). La conductividad térmica  $\kappa$  en ambos dominios es adoptada igual a 1 [ $\text{W m}^{-1} \text{°C}^{-1}$ ]. En cuanto a las condiciones de borde, las superficies internas de las esferas que conforman la malla *patch*  $\Omega_2$  (ver Fig. 6.7) tienen una temperatura impuesta de 100 [ $\text{°C}$ ], mientras que la malla de fondo  $\Omega_1$  tiene impuesta una condición de borde convectiva sobre toda su superficie. Los valores del coeficiente de convección  $h$  y la temperatura del medio  $T_{\text{amb}}$  son  $h = 1$  [ $\text{W m}^{-2} \text{°C}^{-1}$ ], y  $T_{\text{amb}} = 50$  [ $\text{°C}$ ], respectivamente, por lo que el dominio disiparía calor al ambiente. Entonces

$$\begin{aligned}
 \kappa \Delta T &= 0 && \text{en } \Omega_i, \text{ con } i = 1, 2 \\
 \kappa(\nabla T / \nabla \vec{n}) &= h(T - T_{\text{amb}}) && \text{en } \Gamma_1 \subseteq \partial\Omega_1 \\
 T &= \bar{T} && \text{en } \Gamma_2 \subseteq \partial\Omega_2
 \end{aligned} \tag{6.9}$$

donde  $\Omega_1$  es un dominio cúbico de longitud 1[m], mientras que cada cuerpo del dominio  $\Omega_2$  tiene diámetro interior y exterior de 0.1[m] y 0.22[m], respectivamente. La malla de  $\Omega_1$  consiste de 1.7

millones de elementos tetraédricos no estructurados, mientras que las mallas estructuradas de las esferas que componen  $\Omega_2$  están conformadas por un total de 430.000 elementos tetraédricos (ver Fig. 6.7).



**Figura 6.7:** Dominios  $\Omega_1$  y  $\Omega_2$  del problema de optimización, donde  $\Omega_1$  consiste en un cubo de longitud unitaria, mientras que  $\Omega_2$  consiste en 4 esferas huecas, cuyo bordes internos  $\Gamma_2$  tienen impuestas una condición tipo Dirichlet. Se visualizan también las mallas utilizadas para discretizar ambos dominios.

### 6.5.2. Problema de optimización

El problema de optimización se basa en maximizar la temperatura del dominio  $T_{\text{media}}$ , sujeto a la restricción de mantener una distancia mínima  $c_{\text{min}}$  entre pares de centros de esferas  $c_j$  con el fin de que éstas no se superpongan entre si. Entonces, el problema de optimización queda planteado como

$$\begin{aligned}
 & \underset{\bar{\mathbf{x}}}{\text{máx}} && f(\bar{\mathbf{x}}) = T_{\text{media}} \\
 & \text{sujeto a} && g_j(\bar{\mathbf{x}}) = c_j - c_{\text{min}} \leq 0 && j = 1, \dots, m \\
 & && x_{i_L} \leq x_i \leq x_{i_U}, && i = 1, \dots, n
 \end{aligned} \tag{6.10}$$

En la Ec. (6.10),  $x_{i_L}$  y  $x_{i_U}$  son las cotas mínimas y máximas que pueden tomar las variables. En este ejemplo, dichos valores son 0.15 y 0.85, respectivamente, de manera que los objetos de la malla *patch* permanezcan siempre dentro del dominio  $\Omega_1$ . Este ejemplo posee un total de 12 variables de diseño que definen las posiciones de las esferas en el espacio 3D, y se encuentran condensadas en el vector

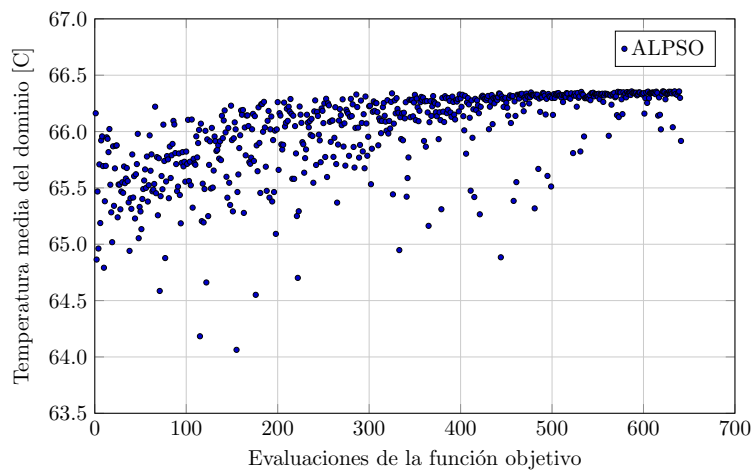
$$\bar{\mathbf{x}} = \left[ \underbrace{x_1, y_1, z_1}_{\text{Esfera 1}}, \underbrace{x_2, y_2, z_2}_{\text{Esfera 2}}, \underbrace{x_3, y_3, z_3}_{\text{Esfera 3}}, \underbrace{x_4, y_4, z_4}_{\text{Esfera 4}} \right] \tag{6.11}$$

Los valores de los elementos de  $\bar{\mathbf{x}}$  son arbitrarios al comienzo del proceso de optimización. En cuanto a los parámetros utilizados en los optimizadores, se utilizaron los mismos valores presentados en

el problema descrito en el Capítulo 4, ya que mostraron buenos resultados de convergencia en problemas similares.

### 6.5.3. Resultados

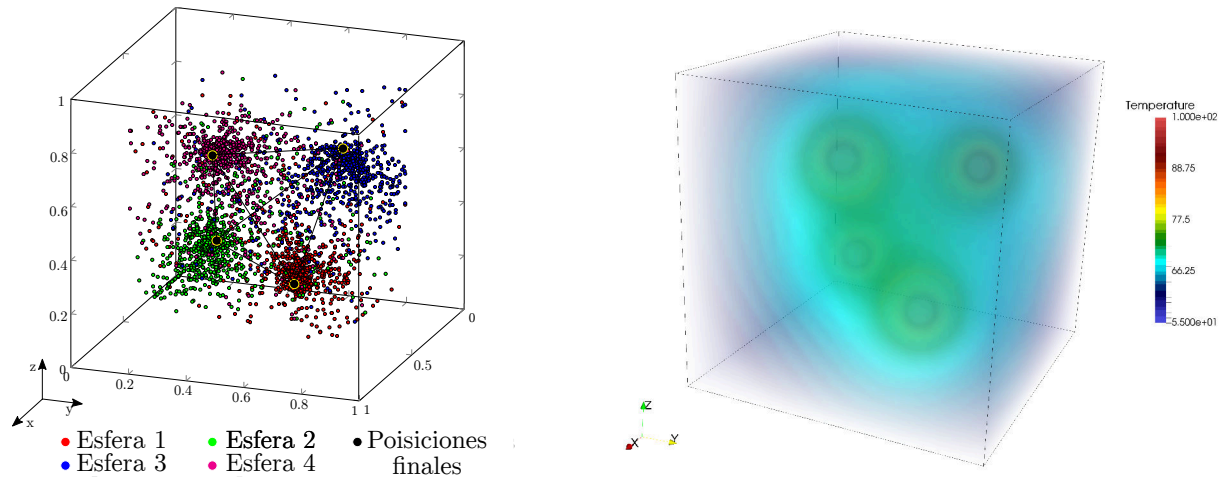
Puede observarse en la Fig. 6.8 la convergencia del problema de optimización. El problema de optimización requirió aproximadamente 1 hora y 12 minutos utilizando 20 procesadores (5 nodos) del *cluster* Seshat. De la Fig. 6.8 puede observarse que la peor condición obtenida por el optimizador obtiene un valor de la temperatura media del dominio de 64.03 [°C], mientras que el arreglo optimizado de los objetos alcanza una temperatura media del dominio de 66.35 [°C]. En la Fig. 6.9 se muestra la configuración optimizada de las esferas. En la Fig. 6.9a se puede observar el valor



**Figura 6.8:** Convergencia de la función objetivo obtenida por el optimizador ALPSO.

tomado por las variables de diseño durante el proceso de optimización. También se visualiza cómo dichas variables exploran en gran medida el dominio computacional. En la Fig. 6.9b se observa el campo de temperatura resultante del proceso de optimización.

Notablemente, de los resultados obtenidos se puede observar que, para maximizar la temperatura media del dominio, los centros de las esferas se posicionan sobre los vértices de un tetraedro prácticamente regular, cuya circunferencia tiene un radio de aproximadamente 0.34 [m], y cuyos vértices se encuentran alineados con las diagonales del dominio cúbico. Esta configuración parece ser la óptima ya que permite maximizar la distancia entre las esferas manteniendo una distancia constante al centro del dominio.



**Figura 6.9:** Disposición optimizada de las esferas dentro del dominio cúbico: (a) espacio de búsqueda analizado por cada esfera y posición final obtenida; (b) temperatura obtenida en el dominio mediante la configuración optimizada.

## 6.6. Optimización de la refrigeración de un molde de inyección

Un campo de aplicación donde el método quimera propuesto en esta tesis resulta muy ventajoso con respecto a métodos más tradicionales en FEM, es una optimización de los canales de refrigeración en herramientas de moldeo por inyección. Dada la velocidad y la facilidad de fabricación, el proceso de moldeo por inyección es la forma más común de producción de piezas conformadas por polímeros (Dimla et al., 2005). El proceso requiere la inyección de un polímero fundido dentro de una cavidad mecanizada en el molde con la geometría de la pieza a fabricar, que luego es enfriado y la pieza es retirada. Las tres etapas principales en un proceso de moldeo por inyección son entonces el llenado, el enfriamiento, y la expulsión de la pieza. De estas tres fases, el enfriamiento es la etapa más importante ya que puede representar hasta un 80 % del tiempo por ciclo de fabricación (Dimla et al., 2005), por lo que determina la velocidad de producción de las piezas. Dicho tiempo de enfriamiento se encuentra influenciado directamente por la forma y configuración de los canales refrigerantes encargados de extraer calor del molde de inyección.

El diseño de los canales de refrigeración convencionalmente es realizado mediante un mecanizado por CNC (Control Numérico Computarizado). Sin embargo, la principal desventaja de este procedimiento es que solo pueden mecanizarse canales rectos, por lo tanto imposibilita el desarrollo de canales en tres dimensiones de mayor complejidad que permitan mayor proximidad de los canales a la pared del molde y así una evacuación más eficiente y uniforme del calor. Desarrollos en las últimas décadas en la tecnología de prototipado rápido aplicada a los procesos de fabricación, como ser el proceso de manufactura por adición AM (*Additive Manufacturing*) dentro de los cuales se encuentran el SLM (*Selective Laser Melting*), o el DMD (*Direct Metal Deposition*), permitieron

diseñar y fabricar canales refrigerantes más eficientes y con geometrías más intrincadas. Gracias a estas nuevas tecnologías de fabricación, surgieron los canales de refrigeración conformes (Dimla et al., 2005; Ring et al., 2002; Shayfull et al., 2013), los cuales se ajustan a la geometría de la pieza en la cavidad del molde. Los mecanizados convencionales simplemente no pueden lograr las formas y geometrías de los canales de refrigeración posibles de construir con el enfriamiento conforme. La tecnología del enfriamiento conforme provee a la industria de un mejor control térmico en los moldes de inyección.

Si bien hubieron grandes avances en las últimas décadas respecto a las metodologías de fabricación que permitieron lograr diseños de moldes de inyección con mejor desempeño, no sucedió lo mismo con los métodos numéricos utilizados para analizar y diseñar dichos moldes. En efecto, el proceso de optimización mediante FEM continúa, en el mayor de los casos, siendo por «prueba y error», es decir, se propone una nueva configuración de los canales refrigerantes, luego esta geometría es diseñada en algún software CAD que luego es discretizada por un mallador, donde por último se analiza el desempeño de la nueva disposición por algún software de elementos finitos.

En esta tesis se propone un enfoque completamente automatizado y eficiente para la optimización del *layout* de los canales de refrigeración utilizando el método quimera y pyOpt con el objetivo de maximizar el flujo térmico a través de los mismos. Las variables de diseño que definen la geometría de los canales son controladas por el optimizador sin intervención del usuario, mientras que las geometrías con sus respectivas mallas tanto del molde, como de los canales y de la cavidad de la pieza son generadas en forma independiente.

Lo más novedoso de la presente metodología es que en cada análisis de la función objetivo realizada por el optimizador, sólo es necesario un remallado de los canales, el cual se realiza en forma automática. Esto logra un gran ahorro en costo computacional, ya que sólo se debe remallar una pequeña porción del dominio y, sumado al cómputo en paralelo de la función objetivo, hacen de la presente metodología una herramienta eficiente y robusta que puede ser utilizada tanto en el entorno científico como en el industrial.

### 6.6.1. Modelo matemático y físico del problema

En el presente caso se analiza el molde de inyección utilizado para fabricar un componente del volante de un automóvil hecho de poliuretano. El molde para fabricar el producto tiene las dimensiones  $70 \times 50 \times 22$  [cm], cuyo material es acero AISI P20, cuya conductividad térmica  $\kappa$  es de aproximadamente  $29$  [ $\text{W m}^{-1} \text{K}^{-1}$ ], suponiendo una temperatura media del molde de  $473$  [K]. El molde tiene dos canales de refrigeración, uno superior y uno inferior (ver Fig. 6.10), cuyos diámetros son de  $1$  [cm]. El fluido utilizado como refrigerante es agua que ingresa en los canales a temperatura



Parámetros del refrigerante			
Caudal ( $\dot{m}$ )	10 [l min <sup>-1</sup> ]	Reynolds (Re)	21200
Velocidad ( $v$ )	2.12 [m s <sup>-1</sup> ]	Prandtl (Pr)	6.97
Calor específico ( $C_p$ )	4.182 [kJ kg <sup>-1</sup> K <sup>-1</sup> ]	Nusselt (Nu)	144.82
Viscosidad ( $\mu$ )	$1 \times 10^{-3}$ [Pa · s]	Coef. pelicular ( $h$ )	8690 [W m <sup>-2</sup> K <sup>-1</sup> ]
Conductividad ( $\kappa_f$ )	0.6 [W m <sup>-1</sup> K <sup>-1</sup> ]		

**Tabla 6.2:** Parámetros físicos del refrigerante que circula por los canales en el molde por inyección.

ambiente. La física del fluido refrigerante no fue simulada y en su lugar se utilizó una condición de borde convectiva (ver Ec. (6.12)) para el análisis térmico. El coeficiente pelicular  $h$  del refrigerante es estimado mediante la correlación de Dittus-Boelter. Todos los parámetros del fluido y del flujo se detallan en la Tabla 6.2.

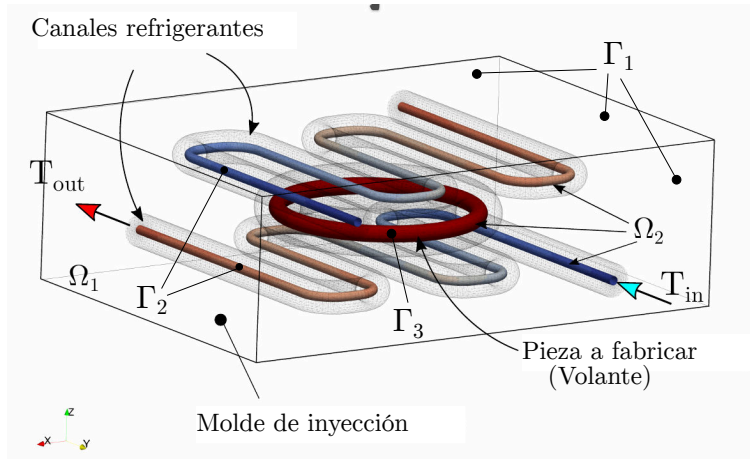
La conducción térmica dentro del molde y en sus fronteras es gobernada por la ecuación de conducción de calor en régimen estacionario:

$$\begin{aligned}
 \kappa \Delta T &= 0 && \text{en } \Omega_i, i=1,2 \\
 \nabla T / \nabla \vec{n} &= 0 && \text{en } \Gamma_1 \subseteq \partial \Omega_1 \\
 \kappa (\nabla T / \nabla \vec{n}) &= h(T - T_{\text{fluido}}) && \text{en } \Gamma_2 \subseteq \partial \Omega_2 \\
 T &= \bar{T} && \text{en } \Gamma_3 \subseteq \partial \Omega_2
 \end{aligned} \tag{6.12}$$

El dominio  $\Omega_1$  se discretiza con una malla de fondo que cubre enteramente el molde de inyección. La malla es estructurada y está compuesta por 10.5 millones de elementos tetraédricos de tamaño uniforme. A la frontera exterior del molde  $\Gamma_1$  se le impuso una condición de borde adiabática debido a que el calor disipado al ambiente es despreciable con respecto al cedido al refrigerante. Mientras que la malla del dominio  $\Omega_2$  se compone por las mallas que rodean a ambas tuberías, como así también la malla que rodea la pieza. La pieza posee una temperatura impuesta en la frontera  $\Gamma_3$  de 473 [K], que es la temperatura a la cual el material fundido entra a la cavidad. En la superficie interna de las mallas de los tubos  $\Gamma_2$  se impone una condición de convección que modela la transferencia de calor del molde hacia el refrigerante. En la siguiente sección se detalla el proceso de creación automática de la malla  $\Omega_2$  para abordar el proceso de optimización.

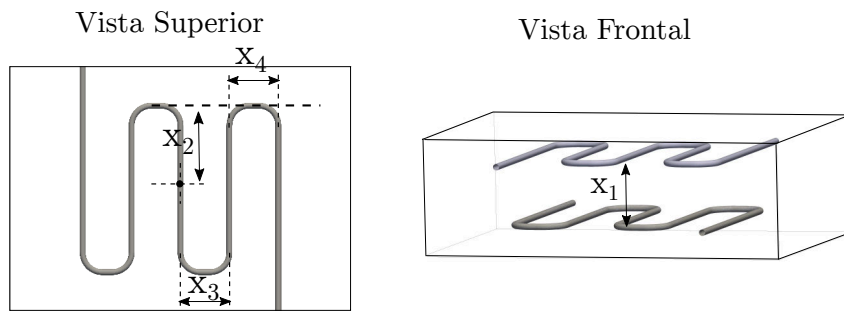
### 6.6.2. Generación de malla de tuberías según las variables de diseño

Dada la naturaleza simétrica del presente problema, en este caso se optó por definir enteramente la geometría y disposición de las tuberías con solo cuatro variables de diseño. En la Fig. 6.11 pueden observarse cómo las variables  $x_2$ ,  $x_3$  y  $x_4$  definen completamente un tramo del canal superior que



**Figura 6.10:** Modelo esquemático del molde de inyección, de la pieza a fabricarse y de los canales refrigerantes.

luego, al realizar una simetría de la malla y una rotación a una distancia  $x_1$  (explicado con mayor detalle a continuación), la geometría de los canales queda completamente definida.



**Figura 6.11:** Vista superior y lateral de la geometría de los canales de refrigeración.

Para la generación de una malla estructurada de forma completamente automática se utilizó el software de código libre SALOME (Ribes y Caremoli, 2007.) utilizando un script en Python, que es ejecutado por consola sin necesidad de acceder a la interfase del programa, y se siguieron los pasos descritos a continuación. Primero, a partir de las variables  $x_2$ ,  $x_3$  y  $x_4$  se genera una malla unidimensional, como se muestra en rojo en la Fig. 6.12a, la cual parte del centro del molde desplazado una distancia  $x_1/2$  en el eje  $z$  y finaliza en el borde del molde. Los radios de empalme son de 3 [cm]. Luego se genera una malla estructurada en dos dimensiones alrededor del tubo (ver Fig. 6.12a) la cual es extrusada por la malla unidimensional, generando la malla tridimensional mostrada en la Fig. 6.12b. Subsecuentemente, se le realiza una simetría a la malla tridimensional respecto al punto de origen de la malla unidimensional, obteniendo así la malla del canal superior completa, como se puede observar en la Fig. 6.12c. Por último, la malla del canal superior es duplicada y rotada sobre el eje  $y$ , para finalmente obtener la malla de ambos canales, como se muestra en la Fig. 6.12d.

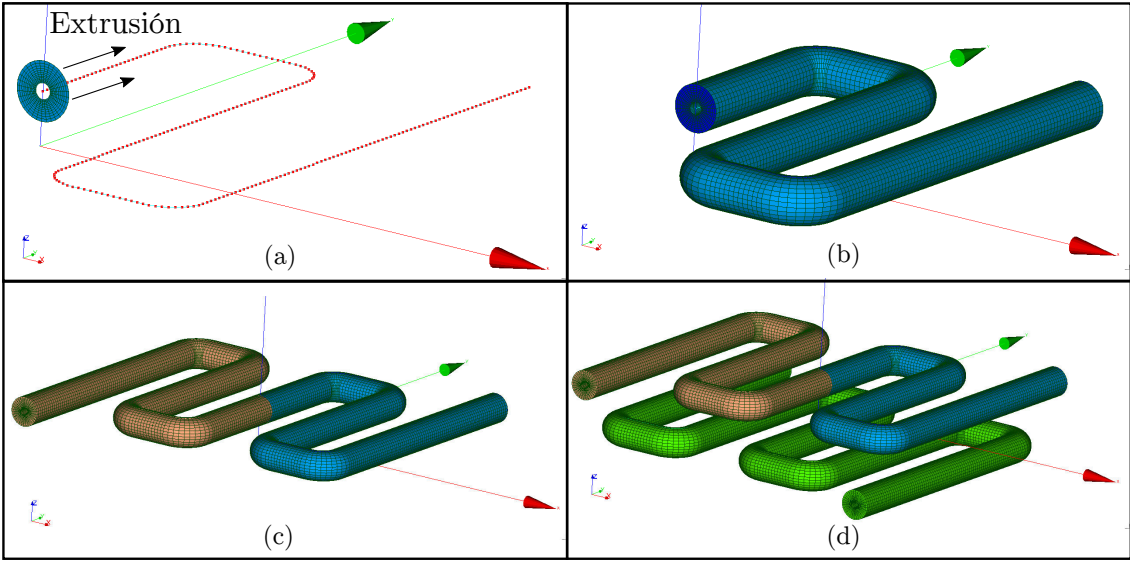
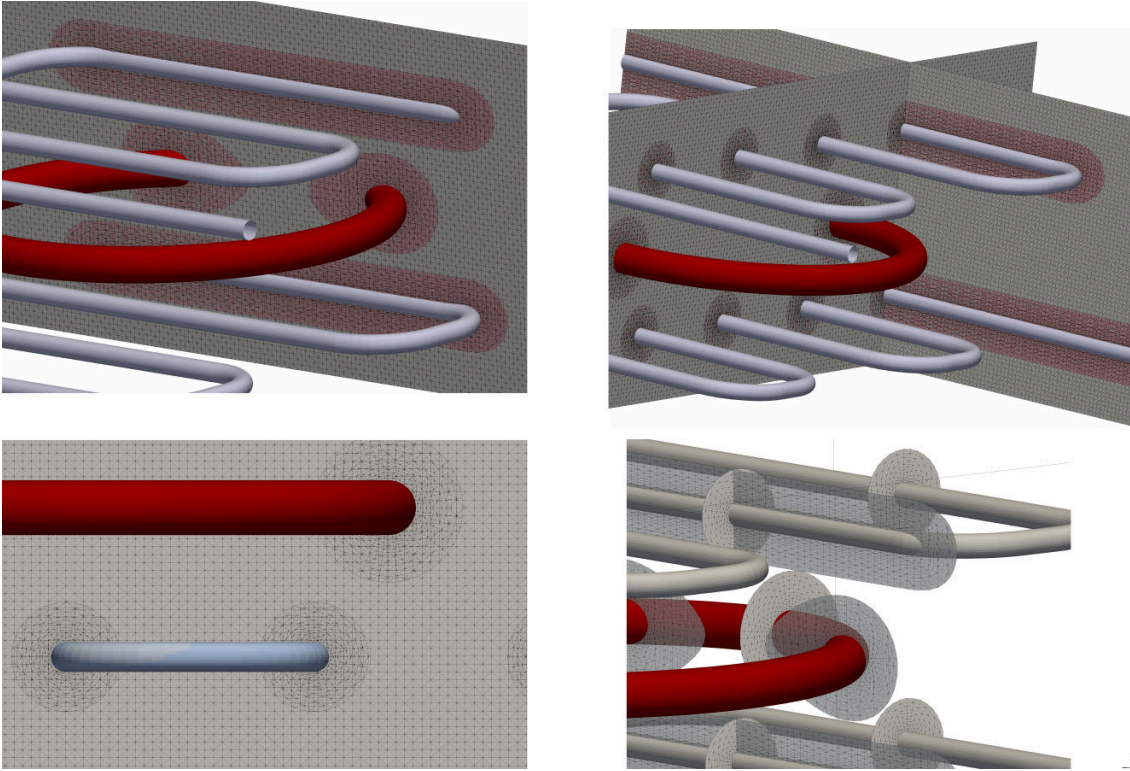


Figura 6.12: Construcción de la malla de las tuberías.



La cantidad de elementos de la malla de los canales puede variar entre 1.05 y 1.65 millones dependiendo de la longitud del canal a ser analizado. Mientras que para la malla del objeto a moldear se utilizaron unas 345.000 celdas.

### 6.6.3. Modelo matemático del problema de optimización

El problema de optimización de los tubos refrigerantes queda planteado de la siguiente forma:

$$\begin{aligned} \max_{\bar{x}} \quad & f(\bar{x}) = q_{\text{pipes}} \\ \text{sujeto a} \quad & x_{i_L} \leq x_i \leq x_{i_U}, \quad i = 1, \dots, n \end{aligned}$$

En este problema se busca maximizar el flujo de calor que pasa del molde hacia las cañerías de refrigeración  $q_{\text{pipes}}$ . El flujo de calor que atraviesa a las cañerías se calcula como:

$$q_{\text{pipes}} = hA(T_{\text{fluido}} - T_{\text{pared}}) \quad (6.13)$$

Las variables de diseño son tratadas como continuas en el optimizador y se condensan en el vector

$$\bar{x} = [x_1, x_2, x_3, x_4] \quad (6.14)$$

Las cotas superiores  $x_U$  e inferiores  $x_L$  se definen de forma tal que los canales permanezcan dentro del molde, no colisionen entre sí o con la pieza. A continuación, en la siguiente tabla se muestran las cotas que puede tomar cada variable:

	$x_1$	$x_2$	$x_3$	$x_4$
$x_L$ [m]	0.06	0.09	0.06	0.06
$x_U$ [m]	0.08	0.18	0.15	0.15

**Tabla 6.3:** Cotas superior e inferior establecidas para las variables del problema de optimización

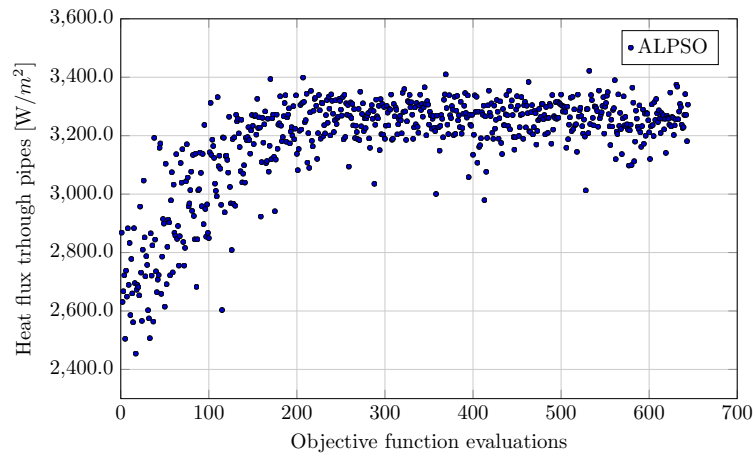
Con respecto a los parámetros utilizados por el optimizador, se utilizaron los mismos que los descritos en el Anexo B utilizados para optimizar la posición de los huecos refrigerantes.

### 6.6.4. Resultados

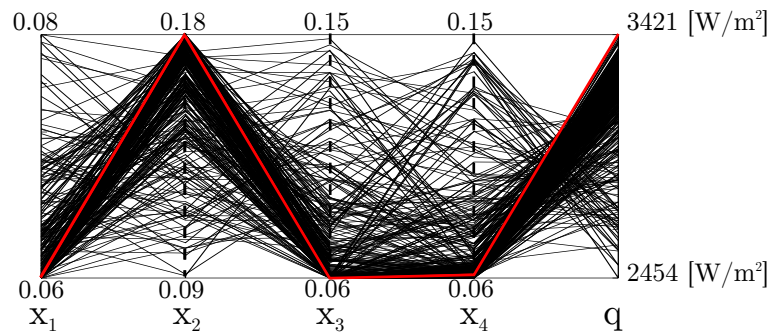
Como se puede observar de la Fig. 6.14, aproximadamente 650 evaluaciones de la función objetivo fueron computadas por el optimizador ALPSO.

El proceso de optimización duró aproximadamente un total de 11 horas utilizando 4 nodos (16 procesadores) del *cluster* Seshat (Seshat, 2018). El valor de flujo máximo alcanzado fue de 3421 [W m<sup>-2</sup>]. La tasa de convergencia visualizada en la Fig. 6.14 no parece ser tan buena como la observada en ejemplos anteriores, esto se puede deber a que los parámetros elegidos para el proceso

de optimización no sean los mejores en el presente problema. Sin embargo, observando la Fig. 6.15 se puede concluir que para lograr el mayor flujo térmico posible hacia los canales, las variables  $x_1$ ,  $x_3$  y  $x_4$  tienden al mínimo valor posible, mientras que la variable  $x_2$  tiende a su máximo. Este resultado es congruente con el problema térmico planteado ya que pone en evidencia que para maximizar el flujo térmico en los canales, estos deben encontrarse dispuestos lo más próximos posible al objeto, lo que explica que  $x_1$ ,  $x_3$  y  $x_4$  tiendan a su mínimo valor, y tengan la mayor prolongación posible en el eje  $y$ , cuyo valor  $x_2$  tiende a su límite máximo.



**Figura 6.14:** Convergencia de la función objetivo obtenida por el optimizador ALPSO.



**Figura 6.15:** Parámetros de entrada del proceso de optimización y flujo térmico en los canales utilizando una representación con coordenadas paralelas.

En las Figs. 6.16 y 6.17 se pueden visualizar el campo de temperatura y gradiente térmico obtenido por las dimensiones óptimas de los canales de refrigeración. Mientras que en la Fig. 6.18 se visualizan secciones transversales de ambos campos donde se puede observar la continuidad de la temperatura y su gradiente entre  $\Omega_1$  y  $\Omega_2$ .

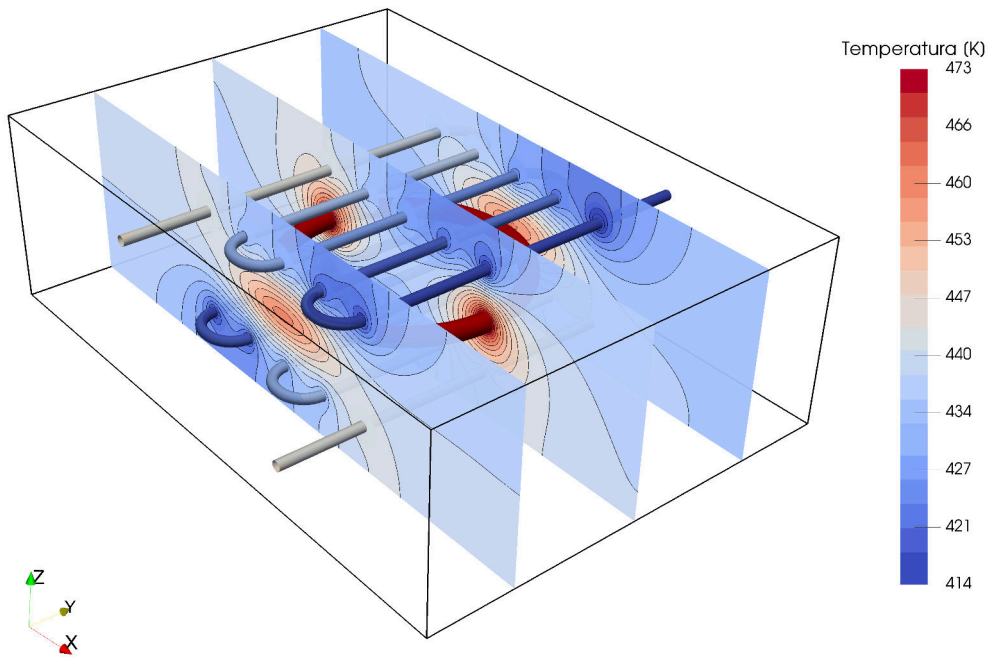


Figura 6.16: Temperatura obtenida por la configuración óptima de los canales.

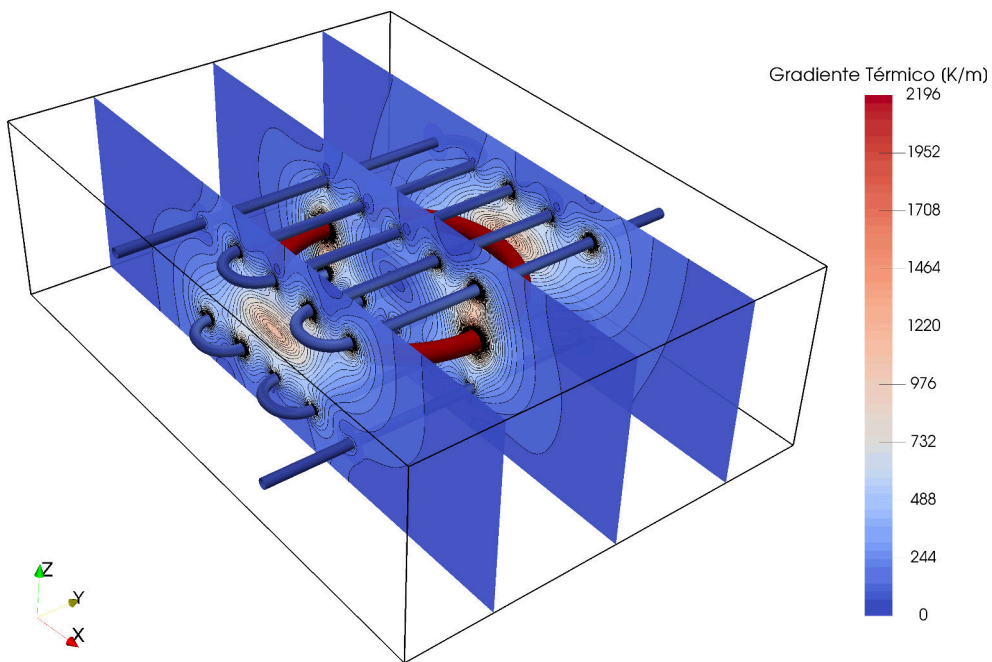
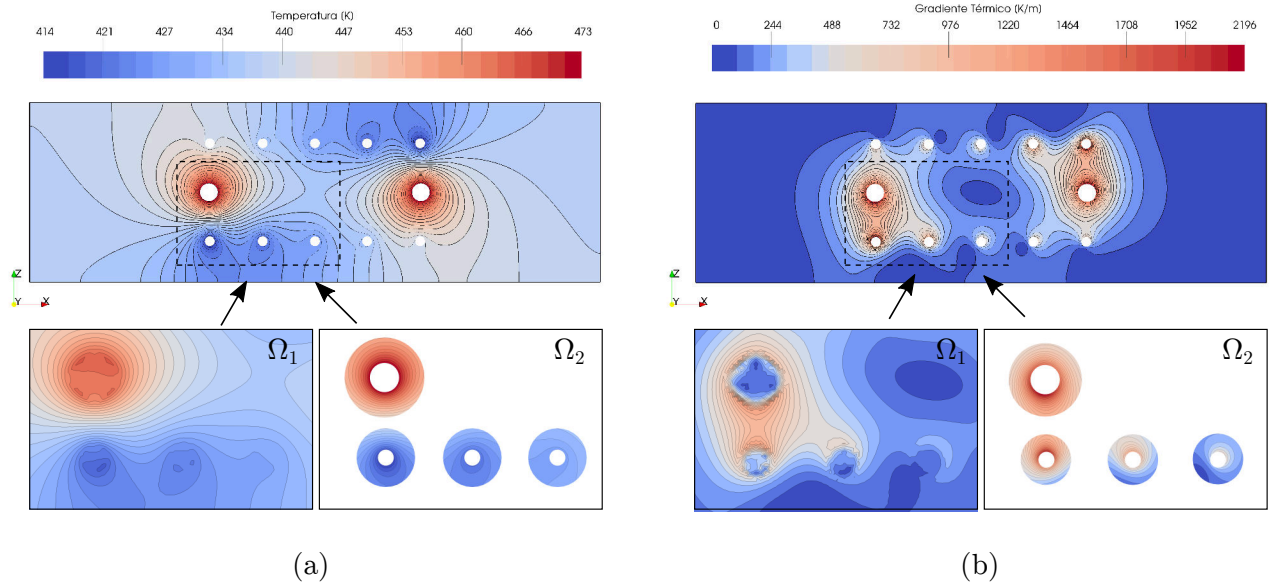


Figura 6.17: Gradiente térmico obtenido por la configuración óptima de los canales.



**Figura 6.18:** Temperatura en el molde y su gradiente térmico obtenidos en  $\Omega_1$  y  $\Omega_2$  presentada en una sección transversal. La continuidad de la temperatura y del gradiente térmico pueden ser observados.

## 6.7. Conclusiones

En este capítulo se abordó la implementación del método quimera con mallas solapadas en tres dimensiones. Si bien el acople *matrix-free* es directo para dominios tridimensionales, algunas consideraciones con respecto a la implementación del algoritmo de interpolación de alto orden y al algoritmo de detección de interfases deben ser tenidas en cuenta, y que fueron tratadas en el presente Capítulo. Con respecto a la convergencia del *solver* iterativo BiCGStab, los resultados observados fueron similares a los obtenidos sobre mallas 2D, es decir, la tasa de convergencia de BiCGStab en los dominios solapados es similar a la tasa de convergencia obtenida en una malla conforme cuyo sistema lineal es resuelto con CG. En cuanto al error en la norma  $L_2$  del método quimera, al igual que lo observado en mallas de dos dimensiones, es comparable al obtenido por una malla conforme. También, el esquema permite construir fácilmente mallas estructuradas alrededor de los objetos lo cual permite lograr mejor calidad y regularidad en los elementos, lo que deriva en la disminución del error en la solución. En cuanto a la aplicación en problemas de optimización, se obtuvieron buenos resultados de convergencia en la función objetivo del acople del método quimera con pyOpt.

Se resolvió un ejemplo numérico a modo de *benchmark* cuyo objetivo fue maximizar la temperatura media de un dominio cúbico que disipa calor al ambiente, optimizando la posición de cuatro superficies esféricas que se encuentran en su interior y que tienen una temperatura impuesta mayor a la del medio. Los centros de las esferas ubicados en los vértices de un tetraedro regular cuyos vértices se encuentran alineados con las diagonales del cubo y cuya circunferencia se encuentra centrada en el cubo, fué la configuración que logró maximizar la temperatura media del dominio. Esto se

debe a que esa disposición logra maximizar la distancia entre las esferas manteniendo a la vez una distancia constante al centro del dominio.

Por último, la metodología se aplicó a un caso ingenieril en el cual se requiere optimizar la configuración de canales de refrigeración de un molde de inyección. La malla de fondo cubría enteramente el molde, mientras que la malla patch se componía por la malla de los canales de refrigeración y de la cavidad de la pieza a moldear. La transferencia térmica hacia el fluido refrigerante fue asimilada con una condición de borde convectiva. Sólo la malla de los canales fue remallada en cada análisis de la función objetivo, la cual es generada en forma automática según las variables de diseño propuestas por el optimizador. El algoritmo probó ser robusto para abordar problemas ingenieriles tridimensionales de formas intrincadas de una manera sencilla, ya que al mallar el molde, la cavidad de la pieza y los canales en forma independiente facilita en gran medida el trabajo de mallado, y en forma eficiente, debido a que el análisis de la función objetivo es paralelizado y se evita el remallado de todo el dominio computacional en cada evaluación.



# Capítulo 7

## Conclusiones

En esta tesis se presentó el desarrollo e implementación de un método de composición de dominios de tipo quimera en el contexto del método de elementos finitos en problemas de transmisión de calor. La principal aplicación fue en problemas de optimización espacial en problemas térmicos utilizando computación distribuida de alto rendimiento mediante *clusters* de computación. En esta tesis la aplicación del esquema desarrollado está abocada principalmente a problemas de optimización en termotransferencia, aunque también puede ser ampliado para abordar otros problemas de la mecánica computacional tales como interacción fluido-estructura o en problemas de multifísica.

Para evaluar el desempeño de la metodología propuesta se presentaron diversos experimentos numéricos sobre mallas tanto estructuradas como no estructuradas, y parcialmente como completamente solapadas, como así también en dos y tres dimensiones espaciales. Las soluciones numéricas obtenidas con el método quimera fueron comparadas con mallas conformes análogas. Los resultados obtenidos muestran que para problemas elípticos, a pesar de que el acople entre subdominios haga perder la simetría de la matriz del sistema de ecuaciones, no obstante, el sistema queda bien condicionado logrando buenas tasas de convergencia iterativa al resolverlo con BiCGStab. Se concluye que el costo computacional requerido por el método quimera es similar al requerido por una malla conforme análoga. A través de los experimentos numéricos se verificó también la continuidad en la solución y su derivada obtenida en la zona de solapamiento.

Como aplicación de la metodología propuesta, primero se resolvió un problema 2D de optimización ingenieril basado en datos experimentales obtenidos sobre álabes de una turbina de gas ensayada exhaustivamente en los laboratorios de la NASA. En este caso se optimizó la posición de los conductos refrigerantes dispuestos en su interior, logrando disminuir su temperatura media, sujeto a restricciones en cuanto a sus gradientes térmicos. Las mallas *patch* rodeando los pasajes refrigerantes podían desplazarse libremente por el álabe dentro de ciertas regiones delimitadas por el optimizador. Tanto la malla *patch* como la malla de fondo son creadas una única vez como parte

del preproceso de la optimización. Como resultado se obtuvo una disposición de los conductos que, a igual caudal y condiciones del refrigerante, mejora la transferencia térmica en el perfil logrando disminuir su temperatura media.

Luego se resolvió un problema 3D en el cuál se requería optimizar la configuración de canales de refrigeración de un molde de inyección. La malla de fondo cubre enteramente el molde, mientras que la malla *patch* se componía por la malla de los canales de refrigeración y de la cavidad de la pieza a moldear. La transferencia térmica hacia el fluido refrigerante fue modelada con una condición de borde convectiva, mientras que sólo la malla de los canales fue remallada en cada evaluación de la función objetivo, la cual es generada en forma automática según las variables de diseño propuestas por el optimizador. Como resultado, el esquema mostró ser robusto en el tratamiento de mallas superiores a los 11 millones de elementos.

El esquema quimera presentó buenas tasas de convergencia en cuanto al proceso de optimización cuando se lo utiliza con optimizadores basados en población, como así también con optimizadores basados en gradiente. Aunque el algoritmo de enjambre de partículas converge rápidamente a la región donde se encuentra la solución óptima, la tasa de convergencia se ve afectada a medida que el algoritmo converge a la solución final. En este caso, continuar la optimización utilizando el *solver* de programación cuadrática basado en gradiente partiendo de la solución obtenida por el optimizador evolutivo demostró ser muy ventajoso, alcanzando la solución final en pocas iteraciones.

Finalmente, la metodología propuesta probó ser una herramienta robusta y eficiente para optimizar el rendimiento térmico de piezas de ingeniería con geometrías intrincadas, reduciendo en gran medida el esfuerzo de mallado y permitiendo grandes desplazamientos de cuerpos dentro del dominio computacional, sin necesidad de un proceso de remallado global.

# Bibliografía

- Abolhasani M., Karami A., y Rahimi M. Numerical modeling and optimization of the enhancement of the cooling rate in concentric tubes under ultrasound field. *Numerical Heat Transfer, Part A: Applications*, 67(11):1282–1309, 2015.
- Andrew M. W., Aaron J. K., William M. C., y Robert L. M. Validation of the Strand Grid Approach. *AIAA Computational Fluid Dynamics*, 2009.
- Bazaraa M.S., Sherali H.D., y Shetty C.M. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.
- Belgacem F.B. y Maday Y. The mortar element method for three dimensional finite elements. *ESAIM: Mathematical Modelling and Numerical Analysis*, 31(2):289–302, 1997.
- Benek J., Buning P., y Steger J. A 3-d chimera grid embedding technique. En *7th Computational Physics Conference*, página 1523. 1985.
- Cai X.C., Dryja M., y Sarkis M. Overlapping nonmatching grid mortar element methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 36(2):581–606, 1999.
- Dalcin L.D., Paz R.R., Kler P.A., y Cosimo A. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- Dimla D., Camilotto M., y Miani F. Design and optimisation of conformal cooling channels in injection moulding tools. *Journal of Materials Processing Technology*, 164:1294–1300, 2005.
- Eaton J., Bateman D., Hauberg S., y Wehbring R. *GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations*. 2015.
- Eberhart R. y Kennedy J. A new optimizer using particle swarm theory. En *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, páginas 39–43. IEEE, 1995.
- Eguzkitza B., Houzeaux G., Aubry R., Owen H., y Vázquez M. A parallel coupling strategy for the chimera and domain decomposition methods in computational mechanics. *Computers & Fluids*, 80:128–141, 2013.
- Eguzkitza B., Houzeaux G., Calmet H., Vázquez M., Soni B., Aliabadi S., Bates A., y Doorly D. A gluing method for non-matching meshes. *Computers & Fluids*, 110:159–168, 2015.

- Eguzkitza Bazar A.B. *HERMESH: a geometrical domain composition method in computational mechanics*. Tesis de Doctorado, Universitat Politècnica de Catalunya, 2014.
- Farhat C. y Roux F.X. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991.
- Guennebaud G., Jacob B., et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- Hill M.D. y Marty M.R. Amdahl's law in the multicore era. *Computer*, 41(7):33–38, 2008.
- Hillier F.S. *Introduction to operations research*. Tata McGraw-Hill Education, 2012.
- Houzeaux G., Cajas J., Discacciati M., Eguzkitza B., Gargallo-Peiró A., Rivero M., y Vázquez M. Domain decomposition methods for domain composition purpose: chimera, overset, gluing and sliding mesh methods. *Archives of Computational Methods in Engineering*, 24(4):1033–1070, 2017.
- Houzeaux G., Cajas J., Eguzkitza B., y Vázquez M. Techniques for parallel, distributed and cloud computing in engineering, volume 36 of computational science, engineering and technology series, chapter chapter 4: Parallel implementation of domain composition methods. *Saxe-Coburg Publications*, páginas 61–92, 2015.
- Hylton L., Mihelc M., Turner E., Nealy D., y York R. Analytical and experimental evaluation of the heat transfer distribution over the surfaces of turbine vanes. *NASA CR 168015*, 1983.
- Jansen P.W. y Perez R.E. Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach. *Computers & Structures*, 89(13-14):1352–1366, 2011.
- Jette M.A., Yoo A.B., y Grondona M. Slurm: Simple linux utility for resource management. En *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*, páginas 44–60. Springer-Verlag, 2002.
- Jiang M., Luo Y.P., y Yang S.Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, 102(1):8–16, 2007.
- Kao K.H., Liou M.S., y Chow C.Y. Grid adaptation using chimera composite overlapping meshes. *AIAA journal*, 32(5):942–949, 1994.
- Kelley C.T. *Iterative methods for linear and nonlinear equations*, volumen 16. SIAM, 1995.
- Kraft D. y Schnepfer K. SLSQP—A Nonlinear Programming Method with Quadratic Programming Subproblems. *DLR, Oberpfaffenhofen*, 1989.
- Lawson C.L. y Hanson R.J. *Solving least squares problems*, volumen 15. SIAM, 1995.
- Liou M.S. y Kao K.H. Progress in grid generation: From chimera to dragon grids. 1994.

- Meakin R. An efficient means of adaptive refinement within systems of overset grids. En *12th Computational Fluid Dynamics Conference*, página 1722. 1995.
- Nielsen E.J., Diskin B., y Yamaleev N.K. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA journal*, 48(6):1195–1206, 2010.
- Parwani A.K., Talukdar P., y Subbarao P. A hybrid approach using CGM and DE algorithm for estimation of boundary heat flux in a parallel plate channel. *Numerical Heat Transfer, Part A: Applications*, 65(5):461–481, 2014.
- Perez R.E., Jansen P.W., y Martins J.R.R.A. pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization. *Structures and Multidisciplinary Optimization*, 45(1):101–118, 2012. doi:10.1007/s00158-011-0666-3.
- Peter G. y Wolfgang A. W. An ALE-CHIMERA method for large deformation fluid structure interaction. En *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics*, páginas 1–14. 2006.
- Quarteroni A. y Valli A. Domain decomposition methods for partial differential equations numerical mathematics and scientific computation. *New York: Oxford University Press*, 1999.
- Ribes A. y Caremoli C. Salome platform component model for numerical simulation. En *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International (http://www.salome-platform.org/)*, volumen 2, páginas 553–564. IEEE, 2007.
- Ring M., Dimla D., y Wimpenny W. An investigation of effectiveness of conformal cooling channels and selective laser sintering material in injection moulding tools. *RPD 2002 Advanced Solutions in Product Development*, 2002.
- Ryoo H.S. y Sahinidis N.V. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers & Chemical Engineering*, 19(5):551–566, 1995.
- Scales L. *Introduction to non-linear optimization*. Macmillan International Higher Education, 1985.
- Schutte J.F., Reinbolt J.A., Fregly B.J., Haftka R.T., y George A.D. Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering*, 61(13):2296–2315, 2004.
- Sedlaczek K. y Eberhard P. Using augmented lagrangian particle swarm optimization for constrained problems in engineering”¿ using augmented lagrangian particle swarm optimization for constrained problems in engineering. *Structural and Multidisciplinary Optimization*, 32(4):277–286, 2006.
- Seshat. <http://www.cimec.org.ar/c3/seshat/>, 2018.
- Shayfull Z., Sharif S., Zain A.M., Saad R.M., y Fairuz M. Milled groove square shape conformal cooling channels in injection molding process. *Materials and Manufacturing Processes*, 28(8):884–

- 891, 2013.
- Shi Y. y Eberhart R.C. Empirical study of particle swarm optimization. En *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volumen 3, páginas 1945–1950. IEEE, 1999.
- Siavashi M., Garusi H., y Derakhshan S. Numerical simulation and optimization of steam-assisted gravity drainage with temperature, rate, and well distance control using an efficient hybrid optimization technique. *Numerical Heat Transfer, Part A: Applications*, 72(9):721–744, 2017.
- Staten M.L., Shepherd J.F., Ledoux F., y Shimada K. Hexahedral mesh matching: Converting non-conforming hexahedral-to-hexahedral interfaces into conforming interfaces. *International Journal for Numerical Methods in Engineering*, 82(12):1475–1509, 2010.
- Stefanica D. y Klawonn A. The FETI method for mortar finite elements. En *Proceedings of 11th International Conference on Domain Decomposition Methods*, páginas 121–129. 1999.
- Steger J. The chimera method of flow simulation. En *Workshop on applied CFD, Univ of Tennessee Space Institute*, volumen 188. 1991.
- Steger J.L. A chimera grid scheme: advances in grid generation. *American Society of Mechanical Engineers Fluids Engineering Division*, 5:55–70, 1983.
- Steger J.L. y Benek J.A. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(1-3):301–320, 1987.
- Storti M., Nigro N., Paz R., Dalcin L., y Lopez E. PETSc-FEM, A General Purpose, Parallel, Multi-Physics FEM Program. *International Center of Computational Method in Engineering (CIMEC), Argentina* <http://www.cimec.org.ar/petscfem>, 2007.
- Van der Vorst H.A. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.
- Wang B., Zhang W., Xie G., Xu Y., y Xiao M. Multiconfiguration shape optimization of internal cooling systems of a turbine guide vane based on thermomechanical and conjugate heat transfer analysis. *Journal of Heat Transfer*, 137(6):061004, 2015.
- Xueyan L. y Zheng X. Swarm size and inertia weight selection of particle swarm optimizer in system identification. En *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, volumen 1, páginas 1554–1556. IEEE, 2015.
- Zheng Y. y Liou M.S. A novel approach of three-dimensional hybrid grid methodology: Part 1. grid generation. *Computer Methods in Applied Mechanics and Engineering*, 192(37-38):4147–4171, 2003.

# Apéndices





# A - A matrix-free Chimera approach based on Dirichlet-Dirichlet coupling for domain composition purposes

El artículo presentado a continuación ha sido presentado en la revista **Computers & Mathematics with Applications**.

B.A. Storti, L. Garelli, M.A. Storti, J. D'Elía, *A matrix-free Chimera approach based on Dirichlet-Dirichlet coupling for domain composition purposes*.

<https://doi.org/10.1016/j.camwa.2020.01.021>.



# A matrix-free Chimera approach based on Dirichlet-Dirichlet coupling for domain composition purposes

B.A. Storti<sup>1</sup>, L. Garelli<sup>1</sup>, M.A. Storti<sup>1</sup>, J. D'Elía<sup>1</sup>.

<sup>1</sup> Centro de Investigaciones de Métodos Computacionales,(CIMEC), (CONICET-UNL)  
Colectora Ruta 168 s/n, Predio Conicet "Dr. Alberto Cassano", 3000 Santa Fe, Argentina

Keywords: Chimera method; Overlapping grids; Non-matching meshes; Domain composition methods; Matrix-free scheme; Finite element method

## Abstract

A novel approach using an algebraic-based Chimera type domain composition method in the context of the finite element method for non-matching overlapping unstructured grids is proposed in this work. The scheme is based on both the transfer of information across each grid interface via Dirichlet boundary conditions and a high-order interpolation algorithm to obtain one global solution of the system. The solution can be obtained iteratively, with a convergence rate that is similar to that obtained with an analogous conformal mesh, and the matrix-vector operator can be computed with completely decoupled operations on both meshes. Furthermore, the scheme can be set as a linear operator that can be fed to a matrix-free efficient iterative solver, such as the BiConjugate Gradient Stabilized method. Several numerical examples using non-matching unstructured grids that are partially and completely overlapped with different element sizes are presented, assessing the precision and convergence rate of the method.

## 1. Introduction

The main purpose of the Chimera method (Brezzi et al., 2001) is to avoid the difficulty represented by the generation of a general unstructured mesh for problems containing complex bodies or time-dependent geometry. The idea of the method is to use a coarse grid on the background of the entire domain while using finer meshes around the bodies. In this way, the meshing work is greatly reduced while maintaining the accuracy of the solution and the computational cost. Several applications can be found in (Chesshire y Henshaw, 1990; Mayer et al., 2010). It is known that the Chimera fixed grid approach in combination with the arbitrary Lagrangian-Eulerian (ALE)-based method is a good choice for solving large deformation fluid structure interaction (FSI) problems (Wall et al., 2008). Additionally, several studies (Andrew M. et al., 2009; Jean-Jacques Chattot, 1998; Houzeaux y Codina, 2003, 2004) have shown the advantages of this approach and its capability for engineering applications, mainly in problems that involve moving bodies. Automation of the meshing process, efficiency, accuracy and low memory overhead are the main reasons for choosing this method.

Most applications of the Chimera method can be categorized into four main groups:

1. Simplified mesh generation: When mesh generators were not capable of addressing complex geometries, the Chimera scheme was proposed as an alternative that created several meshes around the bodies of a computational domain independently (Steger, 1983; Steger y Benek, 1987; Zheng y Liou, 2003).
2. Local refinement: When additional accuracy on the solution is required in specific areas of the computational domain, the Chimera method achieves this by simply locating a finer mesh (patch mesh) over the original (coarse) mesh (Kao et al., 1994; Meakin, 1995; Formaggia et al., 2018).
3. Moving bodies: The Chimera method is capable of treating problems where a large displacement of bodies may exist. In such cases, independent meshes surrounding the bodies move as rigid bodies, and the solution must be coupled on each time step. This approach may be very computationally demanding, although it allows maintaining the boundary layer and the refinement around the bodies in a natural way (Houzeaux y Codina, 2003; Meakin, 1993; Wang y Parthasarathy, 2000).
4. Optimization: A straightforward application of Chimera involves several objects, each one with its respective mesh, that move around over a computational domain searching for the best configuration in terms of an objective function. In this way, the remeshing process between iterations is avoided (Nielsen et al., 2010).

The Chimera method can be classified as a domain composition method (DCM). The main objective of a DCM is to obtain a unique solution of a PDE from the formulation of

the same PDE expressed on two or more subdomains. Such methods work at cross-purposes with domain decomposition methods (DDMs), which are usually intended to decompose the computational domain in order to parallelize the solution process of the PDE (Quarteroni y Valli, 1999; Smith et al., 2004; Toselli y Widlund, 2006) or to combine code bases from different fields for multiphysics applications (Zhang et al., 2014, 2015). Furthermore, there are different DCM strategies, according to how the split solutions on each subdomain are coupled to obtain the global solution. In the literature, such strategies are characterized into three families: mesh-based, equation-based and algebraic-based formulations. Below, we will briefly explain each formulation to put this work in context. We refer the reader to the article Houzeaux et al. (2017) for an extensive review of each strategy.

In mesh-based formulations, the coupling between subdomains is obtained as a pre-process at the mesh level. For instance, the shear-slip mesh update method (SSMUM) (Kalro y Tezduyar, 1993; Behr y Tezduyar, 1999) is used as a sliding-mesh technique to couple fixed and moving domains in a conformal way. The method allows large but regular deformations, such as translations or rotations, by allowing the distortion of a thin layer of elements connecting the fixed and moving domains. If the distortion of such elements is too large, the zone is remeshed by regenerating the elements connectivities. Another mesh-based method is the HERMESH (Eguzkitza Bazar, 2014), which is implemented on non-conformal meshes and based on the generation of new connectivities between subdomains via the creation of new elements named *extension-elements*, whose objective is to extend one mesh to the other by connecting their respective nodes. Regarding equation-based methods, their implementation works at the equation level, both in the strong and the weak formulation. These methods can be categorized into three sub-groups. First, mesh-free methods are based on defining *mesh-free connections* along the interface generated between the meshes by using new elements called *meshless gluing elements* (Kim, 2002; Li y Liu, 2002; Belytschko et al., 1996; Tian y Yagawa, 2007). Second, constrained coupling methods achieve the coupling of the solutions obtained on each disjoint subdomain by restrictions on the governing equations through imposing continuity equations. These restrictions can be imposed via Lagrange multipliers or penalty methods. Such methods include the mortar element method (Belgacem y Maday, 1997; Cai et al., 1999) and the FETI method (Farhat y Roux, 1991; Stefanica y Klawonn, 1999). Additionally, the XFEM method (Březina y Exner, 2017) can be used in combination with non-matching meshes as an alternative to the mortar method to address problems in complex geometries (Fries y Belytschko, 2010). Third, the transmission-condition based

methods achieve the coupling via the transmission of the conditions in the FEM context or by imposing the flux continuity in the FVM context. In the FEM context, the transfer of the conditions allows the continuity of the variables and fluxes through the interfaces. These transmissions can be of the Dirichlet, Neumann or Robin type. The main drawbacks of the Neumann and Robin conditions are the lack of equivalence between the multi-domain and the one-domain solution and the possible degradation of mesh convergence (Quarteroni y Valli, 1999). Finally, in algebraic-based formulations, the composition of the solution is treated directly in the matrix and RHS of each independent algebraic system (Houzeaux et al., 2015). Working the coupling between subdomains at the algebraic level presents two main advantages: The first is that the solution obtained from the formulation expressed on several subdomains is equivalent to the one-domain solution; second, posing the coupling of the systems at the algebraic level allows hiding the transfer of the information between subdomains inside the iterative solver operations, avoiding convergence problems. For instance, in Houzeaux et al. (2018) an algebraic-based coupling methodology is proposed that enables the solution of PDEs for matching and non-matching meshes in a parallel computing environment. The exchange of information between meshes on the interface is made by introducing transmission matrices that transfer the Dirichlet and Neumann data. The methodology allows to address multiphysics problems on several subdomains, such may be the case of FSI problems.

In this work, an algebraic-based Chimera type DCM applied to non-matching overlapped unstructured meshes in the FEM context is introduced. The Chimera mesh is composed of two partially or completely overlapping subdomains with different element sizes, such that the connections between them consist of sharing information across each other's internal boundaries or interfaces. The resulting coupling system is not symmetric positive-definite; nevertheless, the solution can be obtained with the BiConjugate Gradient Stabilized (BiCGStab) method. In this way, the memory requirement is almost the same as the Conjugate Gradient (CG) method, which could be used for an equivalent one-domain mesh. Our strategy allows us to use independent unstructured meshes, generally a coarser mesh (usually referred to as background mesh) placed where smooth gradients on the solution are encountered, and finer meshes (referred to as patch meshes) placed over areas where the gradients of the solution are expected to be significant and higher precision is required. Generally, this requirement occurs around the bodies or locations where highly concentrated source terms are defined.

The paper is organized as follows. In section 2, the formulation of the split algebraic

systems for two overlapped non-matching grids is introduced. The formulation of a high-order interpolation algorithm used for transferring information from one mesh to another is also explained in detail. Error estimation aspects are discussed as well. In section 3, some 2D-3D numerical tests are presented that assess the accuracy and convergence of the method applied to partially and completely overlapped domains.

## 2. Numerical formulation

### 2.1. Algebraic system splitting

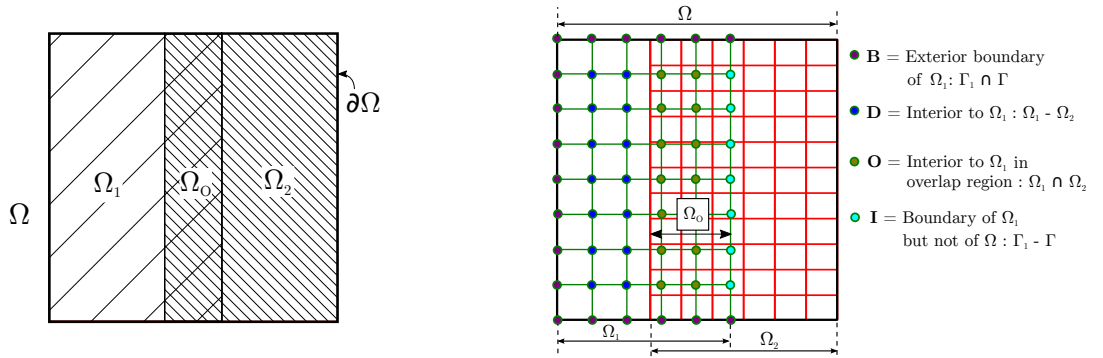
Let us consider a two dimensional domain  $\Omega$ , where we want to solve the Poisson equation with a Dirichlet boundary condition on  $\partial\Omega$ , shown as follows:

$$\begin{aligned} -\Delta\phi &= f & \text{in } \Omega, \\ \phi &= \bar{\phi} & \text{in } \Gamma = \partial\Omega \end{aligned} \quad (\text{A.1})$$

We consider the splitting in two domains  $\Omega_{1,2}$  (see Fig. A.1a). The domains have an overlapping region  $\Omega_o$ . The two meshes do not coincide. Now, consider the solution on the  $\Omega_1$  region. A standard discretization with FEM yields a matrix equation of the form

$$\mathbf{Ax} = \mathbf{b} \quad (\text{A.2})$$

The nodes in the mesh can be subdivided into four main groups according to their location



(a) Splitting of  $\Omega$  domain in two domains:  $\Omega_1$  and  $\Omega_2$ .

(b) Splitting of the nodes of  $\Omega_1$  in groups.

**Figure A.1:** Example of a square domain  $\Omega$  split into two subdomains with an overlapped region  $\Omega_o$  (left) and subsequently, the splitting of  $\Omega_1$  nodes into groups.

(see Fig. A.1b): (i) nodes placed on the boundary of  $\Omega_1$  and on the boundary of  $\Omega$  (group  $B$ ), (ii) nodes located inside  $\Omega_1$  and not belonging to the overlap region (group  $D$ ), (iii) nodes interior to  $\Omega_1$  belonging to the overlapped zone (group  $O$ ), and (iv) nodes on the boundary

of  $\Omega_1$  but not belonging to the boundary of  $\Omega$  (group  $I$ ). It is mandatory for the nodes of  $I$  to be placed inside  $\Omega_2$  in order to be able to perform an appropriate interpolation (fully explained in the next paragraphs). We consider an additional group  $Z$  conformed by all the interior nodes of  $\Omega_1$ , i.e., the union of nodes in  $D$  and  $O$ . This splitting of the nodes induces a splitting of the linear system as follows:

$$\begin{bmatrix} \mathbf{A}_{BB} & \mathbf{A}_{BZ} & \mathbf{A}_{BI} \\ \mathbf{A}_{ZB} & \mathbf{A}_{ZZ} & \mathbf{A}_{ZI} \\ \mathbf{A}_{IB} & \mathbf{A}_{IZ} & \mathbf{A}_{II} \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_Z \\ \mathbf{x}_I \end{bmatrix} = \begin{bmatrix} \mathbf{b}_B \\ \mathbf{b}_Z \\ \mathbf{b}_I \end{bmatrix} \quad (\text{A.3})$$

The values of  $\mathbf{x}_B$  are already known due to the boundary condition. If we knew the values in  $\mathbf{x}_I$ , then we would have a problem with Dirichlet conditions in the whole boundary, and we could certainly solve the problem

$$\mathbf{A}_{ZZ}\mathbf{x}_Z = \mathbf{b}_Z - \mathbf{A}_{ZB}\mathbf{x}_B + \mathbf{A}_{ZI}\mathbf{x}_I \quad (\text{A.4})$$

Although we do not know  $\mathbf{x}_I$ , if we consider interpolating these values from the inside nodes of the  $\Omega_2$  domain, we could obtain a full linear system in the interior values of both domains. Using supra-indices for both domains, we obtain the following algebraic equations:

$$\begin{aligned} \mathbf{A}_{ZZ}^1 \mathbf{x}_Z^1 &= \mathbf{b}_Z^1 - \mathbf{A}_{ZB}^1 \mathbf{x}_B + \mathbf{A}_{ZI}^1 \Pi_{I_{1,2}} \mathbf{x}^2, \\ \mathbf{A}_{ZZ}^2 \mathbf{x}_Z^2 &= \mathbf{b}_Z^2 - \mathbf{A}_{ZB}^2 \mathbf{x}_B + \mathbf{A}_{ZI}^2 \Pi_{I_{2,1}} \mathbf{x}^1, \end{aligned} \quad (\text{A.5})$$

The  $\Pi_{I_{1,2}}$  is a projection operator that interpolates the values of an  $x^2$  vector in the nodes on  $I^1$ . It is assumed that on each domain, there is at least one full element layer in the overlap region so that the interpolation involves only nodes in the  $B^2$  and  $Z^2$  nodes, not on  $I^2$  nodes. Therefore, the linear system (see Eqn. (A.5)) can be solved for the interior values  $\mathbf{x}_Z^1$  and  $\mathbf{x}_Z^2$ .

## 2.2. Iterative solution of the system

The scheme can be set as a linear operator that can be fed to a matrix-free iterative solver such as the Generalized Minimal RESidual (GMRES) or BiCGStab. Matrix-free methods do not need to store the coefficients of the matrix explicitly, but access the matrix by a routine that evaluates matrix-vector products (Kelley, 1995). In this way, when a large matrix is treated, the memory requirement and computational time can be reduced. Below, aspects of the matrix-free implementation and computational costs of the iterative solvers are treated.



### 2.2.1. Matrix-free implementation

The system obtained by the coupling of both subdomains is neither symmetric nor positive definite; however, as will be shown in the numerical experiments presented in section 3, the system is well conditioned such that it can be solved with BiCGstab and achieve convergence rates similar to those obtained with the CG method on the one-domain problem. The last point is of great importance because it means that the linear system has similar characteristics to the one-domain problem. Therefore, to solve the system generated by means of the Chimera approach, Eqn. (A.5) can be rewritten as residuals in the following form:

$$\begin{aligned} \mathbf{r}_Z^1(\mathbf{x}_Z^1, \mathbf{x}_Z^2) &= \mathbf{A}_{ZZ}^1 \mathbf{x}_Z^1 - \mathbf{b}_Z^1 + \mathbf{A}_{ZB}^1 \mathbf{x}_B^1 - \mathbf{A}_{ZI}^1 \Pi_{I_{1,2}} \mathbf{x}_Z^2, \\ \mathbf{r}_Z^2(\mathbf{x}_Z^1, \mathbf{x}_Z^2) &= \mathbf{A}_{ZZ}^2 \mathbf{x}_Z^2 - \mathbf{b}_Z^2 + \mathbf{A}_{ZB}^2 \mathbf{x}_B^2 - \mathbf{A}_{ZI}^2 \Pi_{I_{2,1}} \mathbf{x}_Z^1. \end{aligned} \quad (\text{A.6})$$

To avoid forming and storing the full matrix of the coupled system, a matrix-free formulation is used in this work. To this end, the standard BiCGStab algorithm implemented in PETSc (KSPBCGS Krylov solver) is employed (Balay et al., 2019). Furthermore, for the matrix-vector products, the *matrix shell* (MatShell) implementation of PETSc is used and Algorithm 1 (explained below) is followed for the matrix-free approach.

---

**Algorithm 1:** Compute residuals

---

**Input** :  $\mathbf{x}$

**Output:**  $\mathbf{r} = f(\mathbf{x})$

- 1: Extract  $\mathbf{x}_Z^1, \mathbf{x}_Z^2$  from  $\mathbf{x} = [\mathbf{x}_Z^1; \mathbf{x}_Z^2]$
  - 2: Interpolate  $\mathbf{x}_Z^2$  on nodes  $I^1$
  - 3: Compute residual  $\mathbf{r}_Z^1$  on  $Z^1$  according to Eqn. (A.6) (line 1).
  - 4: Interpolate  $\mathbf{x}_Z^1$  on nodes  $I^2$
  - 5: Compute residual  $\mathbf{r}_Z^2$  on  $Z^2$  according to Eqn. (A.6) (line 2).
  - 6: Combine  $\mathbf{r} = [\mathbf{r}_Z^1; \mathbf{r}_Z^2]$
- 

Algorithm 1 computes the residual from the interior values of both subdomains in each iteration of BiCGStab. It takes as input all the nodes values of  $\Omega$ , i.e.,  $\mathbf{x}^1 \cup \mathbf{x}^2$ , and then, interior node values are extracted ( $\mathbf{x}_Z^1 \cup \mathbf{x}_Z^2$ ) to perform the interpolation over the interface nodes ( $I^1$  and  $I^2$ ). First, the interpolation of nodes on  $I^1$  is computed by means of the interpolation operator  $\Pi_{I_{1,2}}$  and interior node values of  $\Omega_2$ . The values obtained by the interpolation are used to set the Dirichlet conditions on the nodes of  $I^1$ . Afterwards, the residual for the interior values of  $\Omega_1$  is calculated according to line 1 of Eqn. (A.6). Then, in an analogous manner, the values from the interior nodes of  $\Omega_1$  are interpolated on  $I^2$  and the Dirichlet conditions

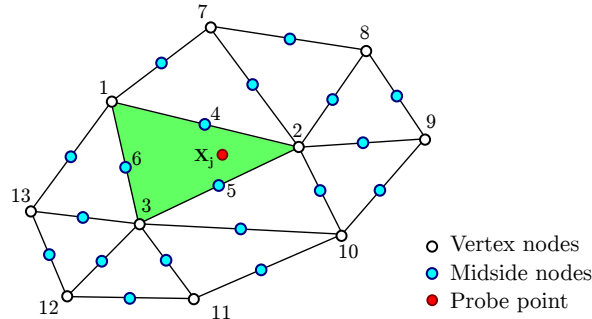
are imposed in the nodes. Finally, the residual for the interior nodes of  $\Omega_2$  is computed, and both  $(\mathbf{r}_Z^1$  and  $\mathbf{r}_Z^2)$  are combined into one single vector  $\mathbf{r}$ .

### 2.2.2. Computational cost of the iterative solvers

The computational cost in storage and in floating-point operations per iteration of the BiCGStab algorithm, it remains constant during the iterative process (in contrast with GMRES for instance), although it is slightly more expensive than the cost of the CG algorithm. This is because BiCGStab requires storing three additional vectors than CG; and, additionally in each iteration, one matrix-vector product and two scalar products must be computed. When large problems are treated, the matrix-vector products operations are dominant. Therefore, the cost per iteration of the BiCGStab algorithm can become twice as much the cost per iteration of the CG algorithm. In contrast, the average cost per iteration of BiCGStab can result much lower than GMRES when large number of iterations are needed to converge. This is because the computational cost of orthogonalization is much higher than that required by a matrix-vector product when the dimension of the Krylov space is large. We refer the reader to (Kelley, 1995) for further details about these aspects.

### 2.3. Interpolation operators

As will be shown in the several numerical examples presented in section 3, the spatial convergence order of the solution is affected by the interpolation scheme. From these results we have concluded that, in order to preserve the second order convergence, a third order interpolation is required. An explanation of the interpolation formulation using Lagrange multipliers implemented in this work is presented below.



**Figure A.2:** Third order interpolation on an unstructured mesh.

Assume that there is a mesh covering  $\Omega_1$  composed of simplices (see figure A.2) and the values of some smooth function  $\phi$  are known at the node vertices (white circles in the figures).

There are some *probe points*  $\mathbf{x}_j$  (which are the nodes of the companion mesh on  $\Omega_2$ ), and a third-order interpolation of the values of  $\phi$  must be obtained at the probe points  $\mathbf{x}_j$ . That means that the interpolation error must be  $O(h^3)$ , where  $h$  is the characteristic mesh size of the  $\Omega_1$  mesh.

The interpolation is based on reconstructing values of  $\phi$  at the midside nodes (in cyan on the figure) and then performing a standard Lagrangian  $P_2$  interpolation. To reconstruct the midside values from the vertex values, a least square fitting is performed from the edge vertices and the topological neighbours. In the case of the figure, to compute the interpolation at probe point  $\mathbf{x}_j$  that belongs to the triangle in green, the values at the midside nodes (4,5,6) are needed in addition to the vertex nodes (1,2,3). To reconstruct the value of  $\phi$  at a midside node such as 5, a least square fitting is performed using the values of the topological neighbours of 2 and 3, namely, the vertices 1-13 (without including midside nodes 4, 5 and 6). Let  $\tilde{\phi}$  be the second-order reconstruction

$$\tilde{\phi}(x, y) = a + bx + cy + dx^2 + e xy + f y^2. \quad (\text{A.7})$$

When applied to the points  $M = \{1, 2, \dots, 13\}$ , the reconstruction yields

$$\tilde{\phi} = \Psi \mathbf{a}. \quad (\text{A.8})$$

where

$$\tilde{\phi} = \begin{bmatrix} \tilde{\phi}_1 \\ \vdots \\ \tilde{\phi}_m \end{bmatrix}, \in \mathbb{R}^m \quad \mathbf{a} = \begin{bmatrix} a \\ b \\ \vdots \end{bmatrix}, \in \mathbb{R}^{n_a} \quad \Psi = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & y_m & x_m^2 & x_m y_m & y_m^2 \end{bmatrix}, \in \mathbb{R}^{m \times n_a} \quad (\text{A.9})$$

where  $m = |M| = 10$  is the numbers of points (vertices of the edge and their neighbours in the example vertices 2 and 3) and  $n_a = 6$  is the number of parameters of the interpolation. The reconstruction cannot be imposed at all on the  $m=10$  points since the system would be overdetermined, considering there would be  $m=10$  equations and  $n_a = 6$  parameters. Therefore, the parameters  $\mathbf{a}$  are chosen so that the reconstruction is the best fit in the least square sense to  $\phi$ . However, the reconstruction at the nodes that are the extremes of the edge (nodes 2 and 3 in the example for midside node 5) is imposed to be exact. Therefore, the vertices are separated in two disjoint sets, with the extremes of the edge  $E = \{2, 3\}$  and  $N = M - E$ .

Thus, the parameters of the reconstruction are determined such that the fit is exact for

the vertices of the edge  $E$  and optimal in the LS sense for the neighbours  $N$ .

$$\mathbf{a} = \operatorname{argmin} \sum_N |\tilde{\phi}_k - \phi_k|^2, \quad (\text{A.10})$$

subject to the restrictions

$$\tilde{\phi}_k = \phi_k, \quad \text{for } k \in E. \quad (\text{A.11})$$

Using Lagrangian multipliers, the Lagrangian functional to be optimized becomes

$$\mathcal{L}(\mathbf{a}, \boldsymbol{\lambda}) = \|\Psi_N \mathbf{a} - \phi_N\|^2 - \boldsymbol{\lambda}^T (\Psi_E \mathbf{a} - \phi_E) \quad (\text{A.12})$$

therefore, the optimal equations are

$$\underbrace{\begin{bmatrix} (\Psi_N^T \Psi_N) & \Psi_E^T \\ \Psi_E & \mathbf{0} \end{bmatrix}}_{\mathbf{L}} \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \Psi_N^T \phi_N \\ \phi_E \end{bmatrix}. \quad (\text{A.13})$$

This is a square system of  $n_a + m_E$ . In general, it is non-singular and well conditioned, except perhaps for points near a boundary. If the system is singular or ill-conditioned, more layers of neighbours can be added (further explained in section 2.3.1). Once the system is solved, the parameters  $\mathbf{a}$  are used to reconstruct the value at the midside node (5 in the example). For a probe point such as  $\mathbf{x}_j$  in the figure, we have to reconstruct the midside values for nodes 4, 5 and 6. The algorithm 2 presented below shows the computation of the interpolation operator.

### 2.3.1. Multiple layers of neighbours

To ensure the second order reconstruction on all the midside nodes, the resulting  $\mathbf{L}$  matrix of the system (see Eqn. (A.13)) must be non-singular and well conditioned to perform its inverse. When that condition is not met, the interpolation algorithm handles this by increasing the number of layers of topological neighbours of the vertex nodes. Below, a case is presented explaining why this might occur.

It is shown in Fig. A.3 an example where one node (probe point in red) of the interpolation boundary  $\partial\Omega_2$  is placed over one element of  $\Omega_1$  (green element) which is located in the boundary of the mesh. From the computations required for the reconstruction of the midside node 4 (see Fig. A.3a), if only the first layer of neighbours is occupied (four nodes in blue), the condition number obtained for the system matrix  $\mathbf{L}$  gives a value of 450.78. Therefore, the system is well-conditioned when only the first layer of neighbours is used. A similar result is obtained when the reconstruction of the value of the midside node 6 (see Fig. A.3c) is

---

**Algorithm 2:** High-order interpolation

---

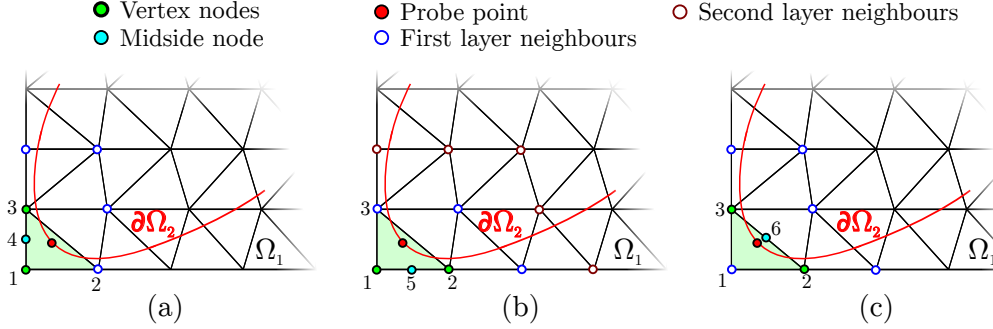
**Input** : Node coordinates, mesh conectivities and probe point coordinates**Output:** Interpolation operator ( $\Pi_{I_{1,2}}$ ) $E \leftarrow$  find elements of  $\Omega_1$  where probe points are overlapped; $Ngbrs \leftarrow$  compute neighbours graph for all nodes of  $E$ ;**foreach** probe point ( $\mathbf{x}_j$ ) on overlap region **do**     $e \leftarrow$  element of  $E$  where  $\mathbf{x}_j$  lives;     $Larea \leftarrow$  compute area coordinates of  $\mathbf{x}_j$  inside  $e$ ;    **foreach** side of  $e$  element **do**         $n_1, n_2 \leftarrow$  vertex nodes of  $side$ ;        **for** nlayers of side **do**             $nnods_1 =$  getneighbours ( $n_1, Ngbrs, nlayer$ );             $nnods_2 =$  getneighbours ( $n_2, Ngbrs, nlayer$ );             $nds =$  union ( $nnods_1, nnods_2$ );            Build  $\Psi_E$  and  $\Psi_N$  according to Eqn. (A.9);            Assemble  $\mathbf{L}$  matrix with Lagrange terms according to Eqn. (A.13);            **if** Condition number of  $\mathbf{L} < tol$  **then**                **break**;        Solve the system of Eqn. (A.13) and extract  $\mathbf{a}$  parameters;

Interpolate values at side centre;

        Add the linear contribution to the interpolation operator  $\Pi_{I_{1,2}}$  //  $Larea$  is needed**return**  $\Pi_{I_{1,2}}$ 

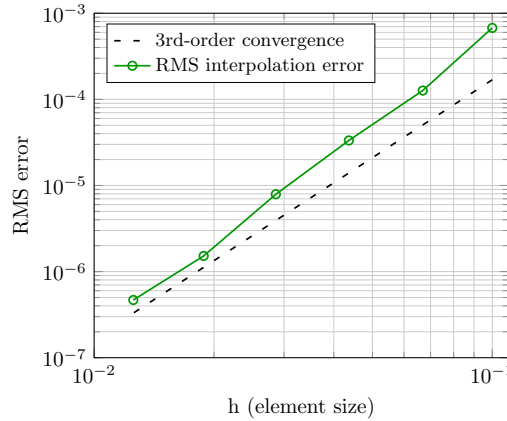
---

performed. In this case, five topological neighbours are employed for the reconstruction and the condition number of the system gives a value of 28.08, proving that the resulting system is well-conditioned as well. However, for the reconstruction of the midside node 5 in Fig. A.3b, there are only three topological neighbours on the first layer, resulting in an ill-conditioned system matrix with a condition number of  $9.5015e+17$ . In this case, the topological neighbours placed in the second layer (red circles) are also occupied for the reconstruction, resulting in a well-conditioned system with a condition number of 86.73.



**Figure A.3:** Topological neighbours employed for the reconstruction of  $\phi$  on the midside nodes of a boundary element. The system for the reconstruction is well-conditioned in (a) and (c), and ill-conditioned in (b), when using the neighbours of the first layer. When the neighbours on the second layer are also used in (b), the system becomes well-conditioned.

## 2.4. Interpolation error



**Figure A.4:** RMS error for the proposed interpolation.

To check the interpolator, unstructured triangular meshes of the growing number of nodes on the square mesh are created. Then, a certain number of probe points ( $N = 100$ ) are randomly chosen in the square, and a certain function  $\phi(x, y) = \sin(\pi x) \sin(\pi y)$  is computed at the mesh nodes and then interpolated at the probe nodes. The mean RMS error is computed

as

$$\epsilon_{\text{RMS}}^2 = \frac{1}{N} \sum_{j=1}^N |\tilde{\phi}_j - \phi_j|^2 \quad (\text{A.14})$$

As seen from the figure, the convergence is  $O(h^3)$ .

### 2.5. Chimera error estimation

Due to the decomposition of the domain  $\Omega$  in the two subdomains  $\Omega_1$  and  $\Omega_2$ , each one has its own error with respect to its own element size. We define the error in  $\Omega_1$  as follows:

$$\|u_{ref} - u_{h_1}\|_{(L2,\Omega_1)} = \sqrt{\sum_{j=1}^{N_j^1} \frac{1}{N_i^1} \left\{ \sum_{i=1}^{N_i^1} (u_{ref}(i) - u_{h_1}(i))^2 \right\} w_{h_1}(j)} \quad (\text{A.15})$$

where  $u_{h_1}$  is the Chimera solution on  $\Omega_1$ ,  $u_{ref}$  is the solution for a uniform-structured grid conformed by  $2048^2$  regular elements interpolated on the nodes of  $\Omega_1$ ,  $N_j^1$  holds for the total elements number of  $\Omega_1$ ,  $N_i^1$  holds for the total number of nodes of the  $j$  element (for instance,  $N_i^1 = 3$  if domains are composed of simplices) and  $w_{h_1}$  is the  $j$  element area. In the same way, we define the error in  $\Omega_2$  as:

$$\|u_{ref} - u_{h_2}\|_{(L2,\Omega_2)} = \sqrt{\sum_{j=1}^{N_j^2} \frac{1}{N_i^2} \left\{ \sum_{i=1}^{N_i^2} (u_{ref}(i) - u_{h_2}(i))^2 \right\} w_{h_2}(j)} \quad (\text{A.16})$$

where, given that  $u_{h_2}$  is the Chimera solution on  $\Omega_2$ ,  $u_{ref}$  is the same solution of the  $2048^2$  mesh interpolated on the nodes of  $\Omega_2$ . To compare the results obtained by the scheme, and in contrast to one-domain solutions, we define the total error in  $\Omega$  for the Chimera solution as:

$$\|u_{ref} - u_h\|_{(L2,\Omega)} = \sqrt{\|u_{ref} - u_{h_1}\|_{(L2,\Omega_1)}^2 + \|u_{ref} - u_{h_2}\|_{(L2,\Omega_2)}^2} \quad (\text{A.17})$$

For the overlapping zone, if we take into account the error on both subdomains, we would be adding the error of the overlap region twice. To overcome this problem, we take into account only the error of the finer mesh in the overlap zone.

### 2.6. Degrees of freedom as a metrics to compare meshes

To be able to compare the mesh convergence and error obtained in the numerical experiments of the following section, the latter explanation on the usage of the number of degrees of freedom ( $N$ ) instead of element sizes ( $h$ ) is introduced. It is known that the classical concept of the  $O(h^p)$  convergence rate in the FEM context is used for uniform meshes with decreasing

element size (Johnson, 2012; Abdulle y Huber, 2016). Then, we have the following expression for the error estimation:

$$\|u - u_h\|_2 \leq Ch^{\min\{p,r\}} \|u\|_{r+1} \quad (\text{A.18})$$

where exponent  $\min\{p, r\}$  specifies the expected rate of convergence,  $p$  is the order of approximation,  $r$  is a regularity index of the solution, and the constant  $C$  hides all constants used in the estimation. However, if an adaptive  $h$  refinement is made, i.e., instead of uniformly refining the mesh, we refine only the elements where the error is large, it does not make sense to estimate the convergence rate in terms of  $h$ . Thus, one should switch to convergence in terms of  $N$  (Demkowicz, 2006). As  $N$  is inversely proportional to  $h$ , if we assume a sufficient regularity of the solution, then we have the error estimation in the function of  $N$  as follows:

$$\epsilon = \|u - u_h\|_2 \leq CN^{-p/d} \quad (\text{A.19})$$

where the norm of the exact solution is incorporated into the constant  $C$ , and  $d$  represents the spatial dimension of the mesh. Therefore, a convergence rate for the FEM solution of a two dimensional problem with linear elements would be  $\epsilon \approx O(h^2)$ , which reduces to  $\epsilon \approx O(N^{-1})$ . While for its derivative, the convergence rate of the  $L_2$ -norm would be  $\epsilon \approx O(h^1)$ , which reduces to  $\epsilon \approx O(N^{-1/2})$ .

Another aspect to clarify is that when we analyze the error on overlapping grids, as we take into account only the error of the finer mesh on overlapped regions, we will apply the same criterion regarding  $N$ , i.e., henceforth, when we refer to  $N$  of overlapping meshes, it would be the sum of the  $N$  of  $\Omega_2$  plus the count of degrees of freedom of nodes belonging to  $\Omega_1$  that are not placed in the overlapped area.

### 3. Numerical examples

#### 3.1. Numerical example 1: Two-dimensional Poisson problem on partially overlapped meshes

In the present numerical example, the Poisson problem of Eqn. (A.1) is solved on two domains partially overlapped, as shown in Fig. A.1a, with a singular source term placed in the finer mesh, and the solution is compared with one-domain results. The interest in this case is in the large curvature of the solution due to the highly concentrated source term. Aspects such as mesh convergence, scheme error, iterative convergence and continuity of the solution are assessed.



### 3.1.1. Numerical setup

For the present numerical test, a singular source term is considered as stated below:

$$\begin{aligned} f &= \frac{\delta}{\cosh((r - 0.15)/\delta)^2}, \\ r &= \sqrt{(x - x_c)^2 + (y - y_c)^2}, \\ \delta &= 0.025; \end{aligned} \tag{A.20}$$

This represents a concentrated source term on the circle  $r = 0.15$ , with its centre placed in  $[x_c, y_c]$ . This is a monopolar layer, and the value of  $\phi$  is almost constant in the interior of the circle. In this numerical test, the centre of the source is placed on  $[0.75, 0.5]$ , which is where the finer mesh will be.

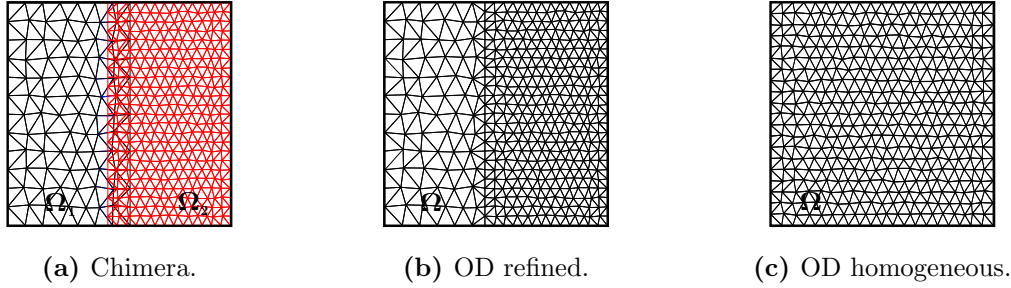
For overlapped meshes, the problem is solved with two domains spanning 0.525 of the  $x$  side, i.e.,

$$\begin{aligned} \Omega_1 &= [0, 0.55] \times [0, 1], \\ \Omega_2 &= [0.45, 1] \times [0, 1], \end{aligned} \tag{A.21}$$

and the overlap region is then

$$\Omega_1 \cap \Omega_2 = [0.45, 0.55] \times [0, 1]$$

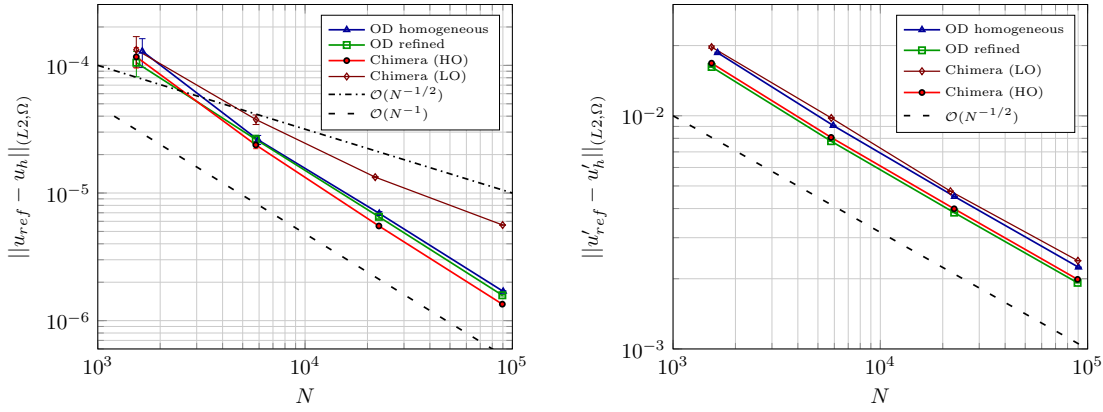
The meshes are Cartesian homogeneously refined in both domains with  $N_{1y} = 11$  elements in the  $y$  direction for  $\Omega_1$  and  $N_{2y} = 23$  for  $\Omega_2$ . Therefore, the meshes are not coincident. The mesh sizes in the  $x$  direction are chosen to keep the aspect ratio (AR) of the elements. The mesh is then refined in steps of  $\sqrt{2}$ , i.e., the mesh in refinement step  $n + 1$  is  $1/\sqrt{2}$  the one in step  $n$ . First, both subdomains are discretized in structured quadrangular elements and then divided into simplices. Subsequently, in each simulation, a certain perturbation is added to the nodes in order to consider the most general case. The node perturbation is achieved by applying a random displacement to the nodes, restricted by the fact that new element sizes cannot exceed a certain factor (greater than one) of the original element size. A schematic image of the Chimera meshes generated on both subdomains can be seen in Fig. A.5a. The results obtained by the scheme proposed for overlapping domains will be compared with two other meshes with similar  $N$ : an analogous conformal one-domain mesh referred to as *OD refined* (see Fig. A.5b), generated from the overlapped meshes, and a one-domain homogeneous mesh referred to as *OD homogeneous* (see Fig. A.5c), with nearly constant element size over the entire domain. The OD refined mesh is generated on each computation from the overlapping grids by removing the elements that are placed on the overlapping area and then performing a triangulation by means of the Delaunay algorithm.



**Figure A.5:** Examples of the meshes used for the current numerical test. The OD refined mesh is generated from Chimera meshes by removing  $\Omega_1$  overlapped elements and then using a Delaunay triangulation in the overlapping area. The OD homogeneous mesh has near constant element size. The three meshes have approximately the same  $N$ .

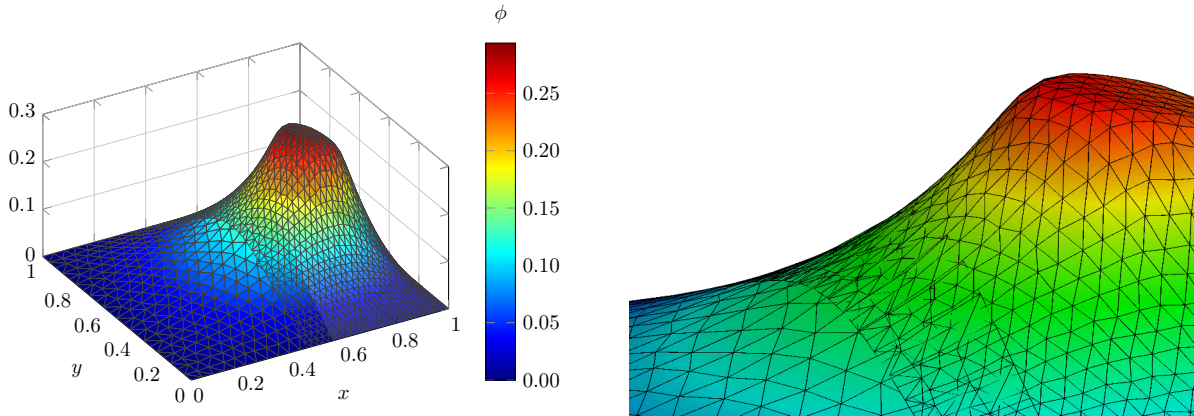
### 3.1.2. Results

For the convergence analysis, ten simulations were carried out for each mesh, each one with approximately 1640, 5930, 22801 and 90300 degrees of freedom. Then, the mean error is computed as well as its standard deviation. To assess how the interpolation order affects the solution of the coupled scheme, a low-order (LO) interpolation algorithm, which is based on simple linear interpolations, and the high-order (HO) interpolation algorithm proposed in section 2.3 were employed. Fig. A.6 shows the mesh convergence for the meshes of the case study. Two important aspects must be highlighted from these results: 1. When the HO interpolation scheme is used, the coupling scheme does not affect the convergence rate of the discretization method, i.e., the second-order convergence for the FEM solution,  $\epsilon \approx O(N^{-1})$ , and the first-order convergence for its derivative,  $\epsilon \approx O(N^{-1/2})$ , is maintained. 2. The  $L_2$ -norm error of the Chimera (HO) solution is very close to the error obtained by an analogous conformal mesh, i.e., the interpolation strategy does not introduce practically any additional error to the solution.



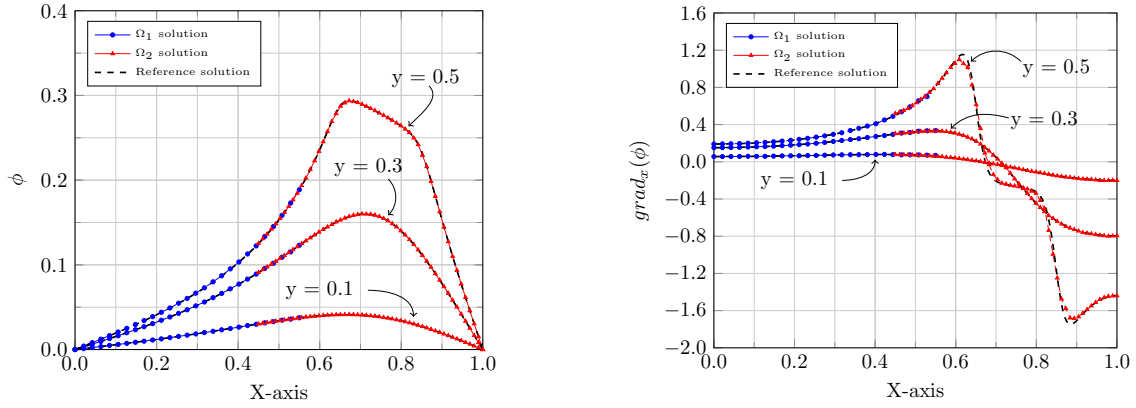
**Figure A.6:** Mesh convergence of the solution (left) and its derivative (right) of the Poisson problem for a delta source term with  $\bar{x}_c = [0.75, 0.5]$ ,  $\delta = 0.025$  and  $r = 0.15$  evaluated on OD homogeneous, OD refined and partially overlapped Chimera meshes.

Fig. A.7 clearly illustrates the continuity of the solution between both subdomains in the overlapped area. Furthermore, in Fig. A.8, a good agreement between the reference solution for the Poisson problem with the delta source term and the solution on  $\Omega_1$  and  $\Omega_2$  projected over several parallel planes can be observed. Additionally, the derivative of the solution on the domains over the planes are in accordance with the reference solution as well. The resulting



**Figure A.7:** Solution of numerical example 1 solved on overlapping meshes showing good agreement on the overlapped area and overlapping meshes. Left: solution of both subdomains ( $\Omega_1$  and  $\Omega_2$  patch together). Different element sizes of meshes and the overlapped region can be observed. Right: A detailed view of the overlapping zone showing the continuity of the solution between both subdomains.

linear systems were solved using CG for the homogeneous and conformal meshes, while for the Chimera mesh, the BiCGStab method was used. Additionally, the conformal OD refined mesh was also solved with BiCGStab for further comparison with the coupled scheme. All cases were solved for a relative residual tolerance of  $10^{-8}$ . As seen from Fig. A.9, the Chimera method shows a very good convergence, reaching the preset tolerance in 594 iterations and

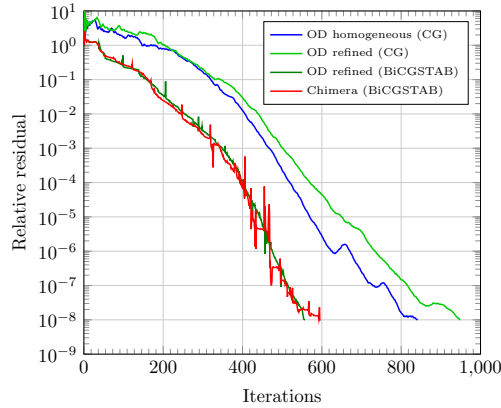


**Figure A.8:** Solution of numerical example 1 on two domains partially overlapped and projected over the planes  $y = 0.5$ ,  $y = 0.35$  and  $y = 0.1$  of the function (left) and its derivative on the x-axis (right), in contrast to the reference solution. The continuity of both the solution and its derivative over the overlapped area is remarkable.

proving that a better convergence can be achieved than that of the OD refined mesh, which takes 947 iterations to converge with CG. Additionally, the number of iterations required for the overlapping grids method to reach convergence is consistently lower than that for the OD homogeneous grid. This brings us to the conclusion that the computational cost and the memory requirement for the resolution of the linear system of the overlapping grids is similar to those required for solving an analogous conformal mesh. Furthermore, it can be observed the similarity in the convergence histories of the OD refined and Chimera meshes solved with BiCGStab, putting in evidence that the linear systems of the one-domain and the coupled problems have similar characteristics. A summary of the results of the iterative solvers are depicted in Table A.1, and its respective histories of the relative residuals are shown in Fig. A.9.

Mesh	N	Elements	Iterations	$L_2$ error ( $\phi$ )	$L_2$ error ( $\nabla\phi$ )
OD homogeneous (CG)	90301	179680	840	1.8296e-06	2.2873e-03
OD refined (CG)	90024	179160	947	1.6387e-06	1.9489e-03
OD refined (BiCGStab)	90024	179160	556	1.6389e-06	1.9489e-03
Chimera HO (BiCGStab)	90012	178840	594	1.2406e-06	1.9499e-03

**Table A.1:** Summary of the results obtained for one case of the mesh convergence study of Fig. A.6



**Figure A.9:** Convergence of the iterative solvers for numerical example 1 on an OD homogeneous mesh, a Chimera mesh composed of two partially overlapped domains, and an OD refined mesh. A relative residual tolerance of  $10^{-8}$  was used in all cases. Mesh sizes are depicted in Table A.1.

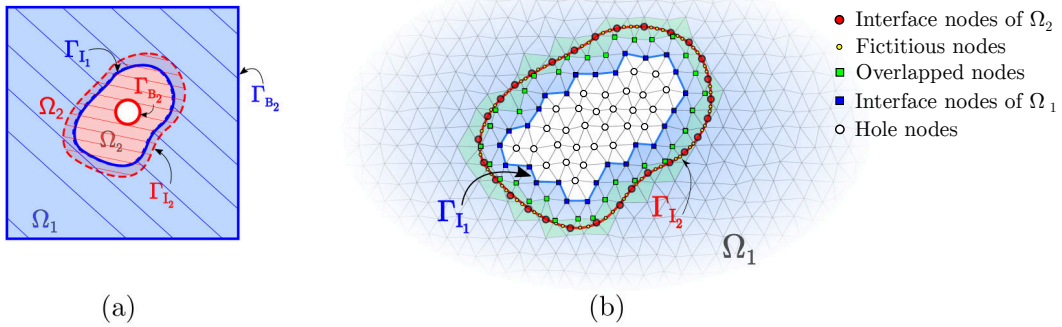
### 3.2. Numerical example 2: Two-dimensional Poisson problem on completely overlapped meshes

In this test case, the finer mesh  $\Omega_2$  is placed entirely inside the coarse mesh  $\Omega_1$  such that it is completely overlapped. This technique allows generation of the background and the patch mesh separately and later places the finer mesh over the region where the gradients are larger. When domains that are partially overlapped are used, the detection of the interpolation interfaces does not represent a difficult task, given that such interfaces would be conformed by boundary nodes that are easy to distinguish. In contrast, for domains that are completely overlapped, the interface nodes of the background mesh, where the values of the fine mesh will be interpolated, are not placed in its boundary (see Fig. A.10a); therefore, a topological detection strategy of the interface nodes must be used. An explanation of the methodology is shown below. Afterwards, the scheme is applied and assessed on a numerical example.

#### 3.2.1. Topological interface detection

To define the interface nodes of  $\Omega_1$ , the procedure is as follows: First, the interface elements of  $\Omega_1$  (elements in green in Fig. A.10b) are recognized, i.e., the elements of  $\Omega_1$  where the boundary of  $\Omega_2$  is. To achieve this, new nodes called fictitious nodes (yellow points in Fig A.10b) between the interface nodes of  $\Omega_2$  (red nodes in Fig. A.10b) are created to have a better frontier resolution. Then, a quadtree search is carried out to trace the elements over the background mesh where the red and yellow nodes are located. Once the interface elements are detected, it is necessary to check which nodes of these elements are inside the overlapped region. These nodes are referred to as overlapped nodes (green nodes in Fig. A.10b), and

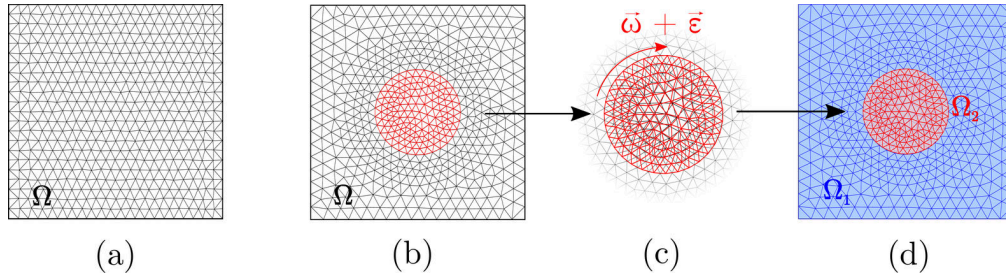
they would be placed in the first layer. Afterwards, a search for the nearest neighbours of the overlapped nodes is made. Only those nodes that are inside the overlapped region are defined as the interface nodes of  $\Omega_1$  (blue nodes in Fig. A.10b), also commonly referred to as *fringe nodes* in the literature (Eguzkitza et al., 2013; Steger y Benek, 1987; Steger, 1983). For the following numerical experiments, the nodes over the third layer are used for the interpolation, which is in accordance with (Vázquez et al., 2000). In this way, two layers of fully overlapped elements are always ensured. Finally, the remaining nodes of  $\Omega_1$  on the overlapped area are referred to as *hole nodes* (see Fig. A.10b) in the literature and are commonly erased from computations, in conjunction with elements delimited by  $\Gamma_{I_1}$  by the so-called *hole cutting process*. In the current approach, such nodes and elements are not erased from the mesh, although a trivial problem is solved on them and its results are not taken into account for the final solution. This strategy allows a constant topology of the meshes to be maintained, which is very beneficial for optimization or transient problems with large displacements, for instance, in a multiprocessor environment.



**Figure A.10:** Automatic topological detection of interpolation boundary on background mesh: (a) domains  $\Omega_1$  and  $\Omega_2$  completely overlapped and interpolation boundaries  $\Gamma_{I_1}$  and  $\Gamma_{I_2}$  where the transmission of information occurs. (b) the different types of nodes of the interface detection procedure.

### 3.2.2. Numerical setup

For the current numerical experiment, the same two dimensional square computational domain is used, and the Poisson problem of Eqn. (A.1) is solved. The Chimera mesh is composed of a square coarse mesh  $\Omega_1 = [0, 0] \times [1, 1]$  covering the entire domain, and a circular finer mesh  $\Omega_2$  of radius 0.2 is placed in the centre of the domain, completely overlapped (see Fig. A.11). The same Eqn. (A.20) is used for the delta source term, although in this case, the parameters are as depicted in Table A.2. In this way, the source term would be even more concentrated in the centre than in the previous case. In the same manner as the previous case,



**Figure A.11:** Examples of the meshes occupied for the current numerical test: a) A one-domain homogeneous mesh with nearly constant element size (OD homogeneous). b) A one-domain mesh with progressive refinement towards the domain centre (OD refined). c) Red elements of the OD refined mesh are rotated and randomly perturbed to generate the  $\Omega_2$  domain. d) The overlapped meshes generated from the OD refined mesh (Chimera mesh).

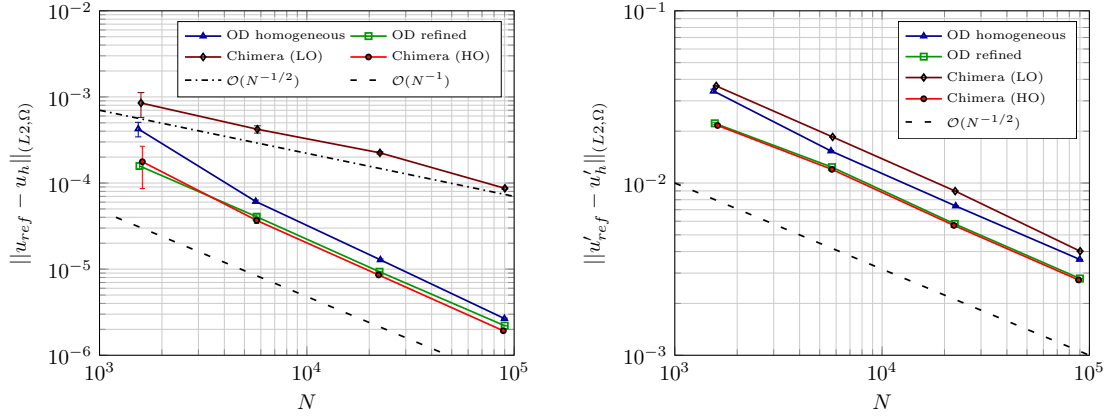
$[\mathbf{x}_c, \mathbf{y}_c]$	$\delta$	$\mathbf{r}$
$[0.5, 0.5]$	0.015	0.1

**Table A.2:** Delta source term parameters for numerical example 2.

a random perturbation is added to the nodes in each computation in order to guarantee that the algorithm works in the most general situation. The meshes implemented for the present study are shown in Fig. A.11. Three meshes are compared: a one-domain homogeneous mesh with nearly constant element size (see Fig. A.11a), a conformal refined mesh whose element sizes are progressively diminished towards the domain centre and whose smaller elements are approximately half the size of the larger elements (see Fig. A.11b), and the Chimera mesh conformed by two non-matching overlapping meshes generated from the refined mesh, as explained below (see Fig. A.11d). As the main purpose of this study case is to assess the accuracy of the coupling methodology using the interface detection scheme and to verify that no additional error is introduced, the same mesh of Fig. A.11b is used as background mesh, while for the patch mesh, the elements of the same mesh contained in a circle of radius 0.2 (in red in Fig. A.11b) but slightly rotated and perturbed, as shown in Fig. A.11c, are used. In this way, both mesh topologies of the refined and overlapping meshes are practically unchanged, and the proper results comparisons can be made.

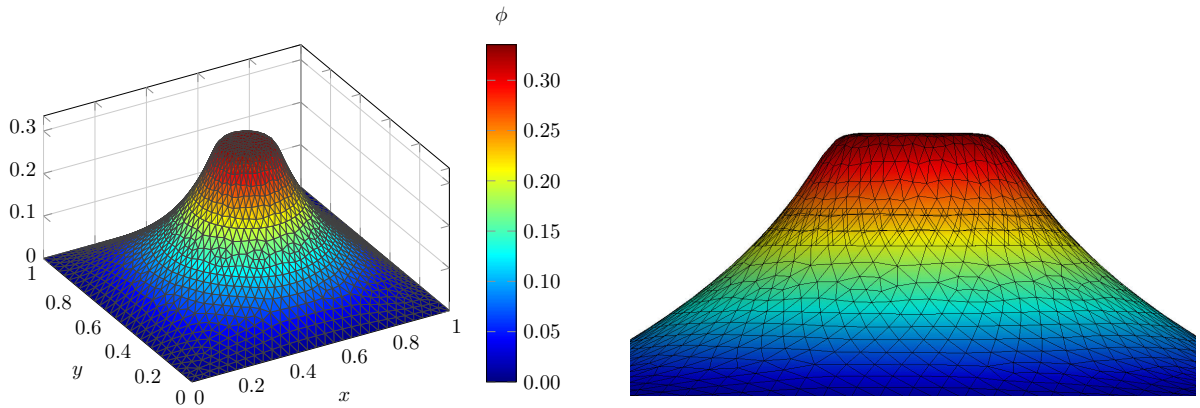
### 3.2.3. Results

Ten runs were carried out for each point of the convergence analysis, and the average value and standard deviation of the  $L_2$ -norm were obtained. Meshes with approximately 1540, 5660, 22650 and 89810 degrees of freedom were used. From Fig. A.12, it can be seen that the second-order spatial convergence of the FEM formulation is maintained using the present Chimera



**Figure A.12:** Mesh convergence of the solution (left) and its derivative (right) of the Poisson problem for a delta source term with  $\bar{x}_c = [0.5, 0.5]$ ,  $\delta = 0.015$  and  $r = 0.1$  evaluated on OD homogeneous, OD refined and fully overlapped Chimera meshes.

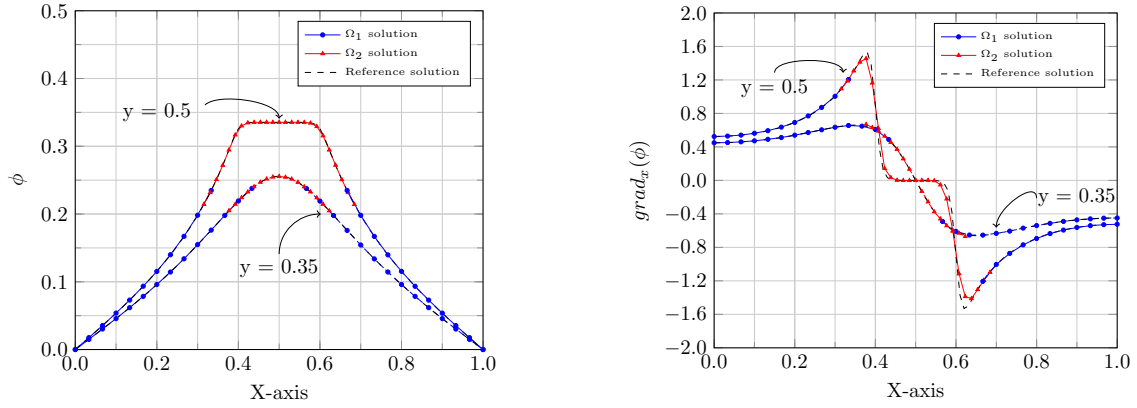
HO approach, as is the first-order convergence of the  $L_2$ -norm of the gradients. Furthermore, it can be seen from Fig. A.12 that the error of the Chimera HO method is similar to the error produced by the one-domain refined mesh, i.e., the interpolation does not introduce any additional error to the FEM solution.



**Figure A.13:** Solution of numerical example 2 for a delta source term (with parameters defined as depicted in Table A.2) solved on two completely overlapping meshes. Left: solution of both subdomains ( $\Omega_1$  and  $\Omega_2$  patch together). Rotation and distortion of  $\Omega_2$  elements can be observed. Right: A detailed view of the overlapping zone. The continuity of the solution between both subdomains can be observed.

The continuity of the solution between the background and patch meshes can be observed in Fig. A.13. Furthermore, the non-matching nodes between both subdomains  $\Omega_1$  and  $\Omega_2$  resulting from the rotation and perturbation of the patch mesh can be seen from the detailed view of Fig. A.13. Additionally, the continuity of the solution and the continuity of its derivative over the planes  $y = 0.5$  and  $y = 0.35$  through the overlapped area, in contrast with the reference solution, are shown in Fig. A.14. In Fig. A.15 and Table A.3, the conver-



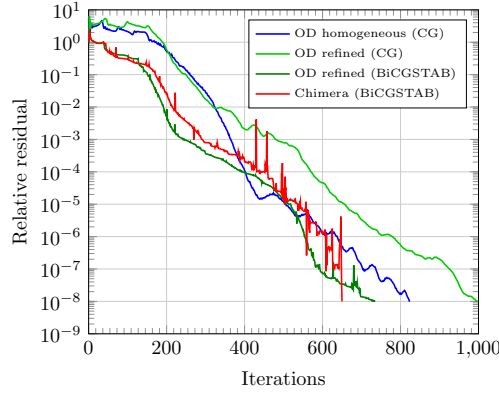


**Figure A.14:** Solution of numerical example 2 on two domains that are partially overlapped and projected over the planes  $y = 0.5$  and  $y = 0.35$  of the function (left) and its derivative on the x-axis (right), in contrast with the reference solution. The continuity of both the solution and its derivative can be observed, in addition to a good agreement with the reference solution.

gence history and the error obtained for one case of the convergence study are depicted. A good convergence of the iterative solver can be seen for the overlapped grids, and the error is similar to the one obtained by the conformal one-domain solution.

Mesh	N	Elements	Iterations	$L_2$ error ( $\phi$ )	$L_2$ error ( $\nabla\phi$ )
OD homogeneous (CG)	89401	177610	823	2.6790e-06	2.8344e-03
OD refined (CG)	89445	177780	997	2.2360e-06	2.7715e-03
OD refined (BiCGStab)	89445	177780	734	2.2361e-06	2.7715e-03
Chimera HO (BiCGStab)	89445	177780	649	1.7436e-06	2.7898e-03

**Table A.3:** Summary of the results obtained for one case of the mesh convergence study of Fig. A.12



**Figure A.15:** Convergence of the iterative solvers for numerical example 2 on an OD homogeneous mesh, a Chimera mesh composed of two completely overlapped domains, and an OD refined mesh. A relative residual tolerance of  $10^{-8}$  was used in all cases. Mesh sizes are depicted in Table A.3.

### 3.3. Numerical example 3: Boundary condition imposed on patch mesh completely overlapped

As a third numerical example, we consider the same square domain, although in this case, a hole of radius 0.1 is placed in the centre of the computational domain (see Fig. A.16). The particular interest in this example is that the largest gradients on the solution will not be given by a source term but will instead be given by a boundary condition imposed over an object, which in this case is a circle.

#### 3.3.1. Numerical setup

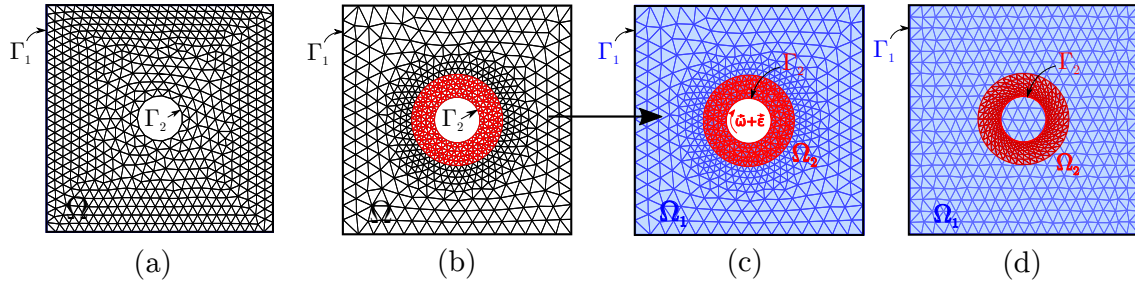
The Laplace equation is solved with Dirichlet conditions imposed in its boundaries. Then, the one-domain problem and the Chimera problem take the form depicted in Table A.4. The

One-domain problem	Chimera problem
$\Delta\phi = 0$ in $\Omega$ ,	$\Delta\phi = 0$ in $\Omega_i$ , $i = 1, 2$ ,
$\phi = 0$ in $\Gamma_1 \subseteq \partial\Omega$ ,	$\phi = 0$ in $\Gamma_1 \subseteq \partial\Omega_1$ ,
$\phi = 1$ in $\Gamma_2 \subseteq \partial\Omega$	$\phi = 1$ in $\Gamma_2 \subseteq \partial\Omega_2$

**Table A.4:** The Laplace problem stated in a one-domain mesh and on a Chimera mesh conformed by two subdomains.

results over four meshes are compared: 1. A conformal homogeneous mesh with nearly constant element size over the entire domain  $\Omega$  (see Fig. A.16a). 2. A one-domain mesh with decreasing element size (refinement) towards the hole (see Fig. A.16b). 3. Two overlapped

meshes generated from the conformal refined mesh, i.e., with almost the same topology that the conformal mesh has (see Fig. A.16c). 4. Two overlapped meshes generated in an independent manner, i.e., the background mesh ( $\Omega_1$ ) conformed by an unstructured homogeneous mesh with nearly constant element size, and a structured mesh ( $\Omega_2$ ) around the hole with progressive refinement towards the boundary  $\Gamma_2$ . We refer to this mesh as *Chimera regular* (see Fig. A.16d).

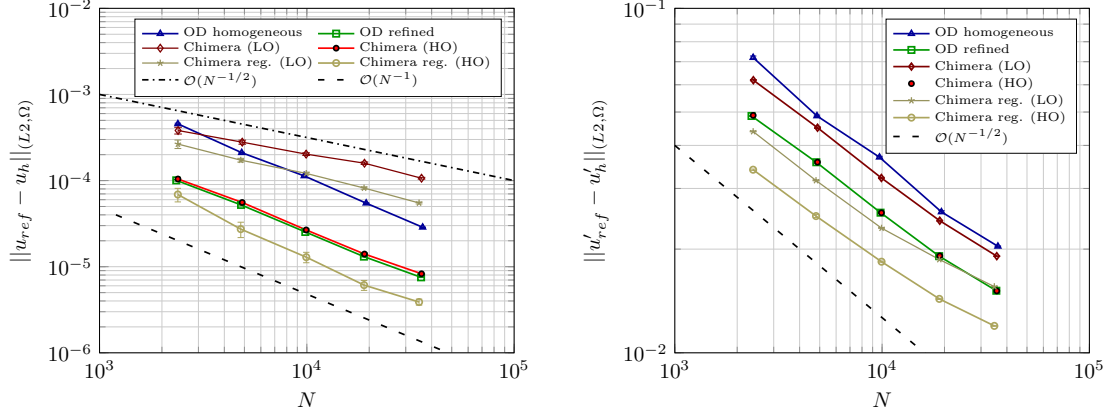


**Figure A.16:** Examples of the meshes occupied for the current numerical test: a) OD homogeneous mesh with nearly constant element size. b) OD refined mesh with its refinement towards  $\Gamma_2$ . c) The elements in red of the OD refined mesh are rotated ( $\bar{\omega}$ ) and randomly perturbed ( $\bar{\epsilon}$ ) to generate the  $\Omega_2$  mesh of the Chimera mesh. d) Chimera regular mesh. A non-structured homogeneous mesh is used for the background, and a structured mesh is progressively refined towards  $\Gamma_2$  for the patch.

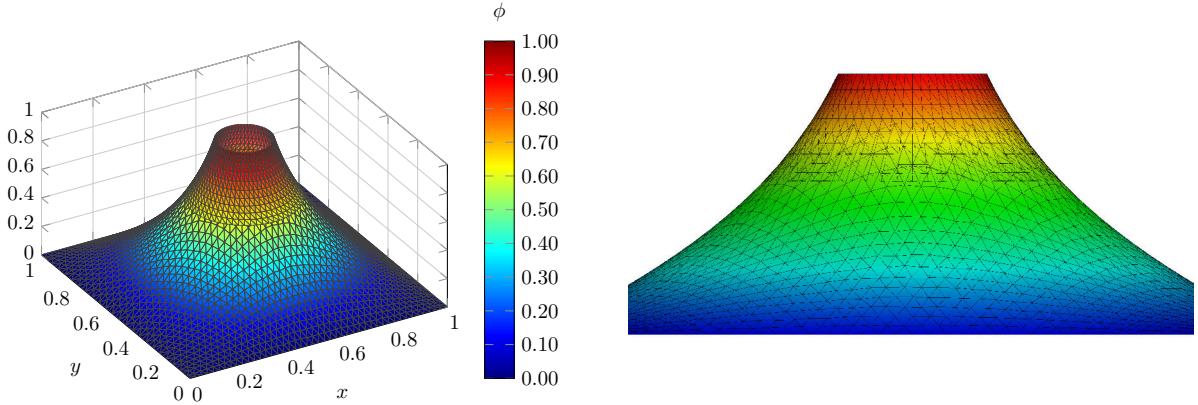
### 3.3.2. Results

Ten runs were carried out for the meshes with approximately 2390, 4850, 9710, 19330 and 36200 degrees of freedom, and the average value and standard deviation of the  $L_2$ -norm were obtained. From the convergence analysis presented in Fig. A.17, several aspects must be emphasized: on the one hand, similar to the two numerical examples presented before, the convergence rate of  $\epsilon \approx O(N^{-1})$  for the solution and  $\epsilon \approx O(N^{-1/2})$  for its derivative is maintained, and the error of the two overlapping domains for the case of Fig. A.16c is similar to the one obtained by the analogous conformal mesh of Fig. A.16b. On the other hand, the overlapping meshes of Fig. A.16d generated independently show an even better approximation of the reference solution. This outcome is due to the regularity of the meshes of  $\Omega_1$  and  $\Omega_2$ . Such regularity is controlled by the worst value of the shape regularity of the elements conforming the mesh which, for triangular elements, is defined as the ratio between the longest side of the element and the radius of the inscribed circle (Johnson, 2012; Hiptmair et al., 2012; Duprez et al., 2019), scaled by a constant  $C_t$  taken in such a way so as to give a unit value for the ideal element ( $C_t = \sqrt{3}/6$  for equilateral triangles). The shape regularity measure indicates the degree to which a triangle is distorted and affects directly in the error

estimation of Eqn. (A.19) given that it is included in the constant C. Is easy to see from Fig. A.16 that this parameter can be better controlled in the generation of the OD Chimera regular mesh. The minimum value obtained during the mesh convergence study in the OD refined meshes was 0.621, while for the Chimera regular meshes was 0.711. It will be shown in section 3.4 that such difference is even greater for three-dimensional meshes.

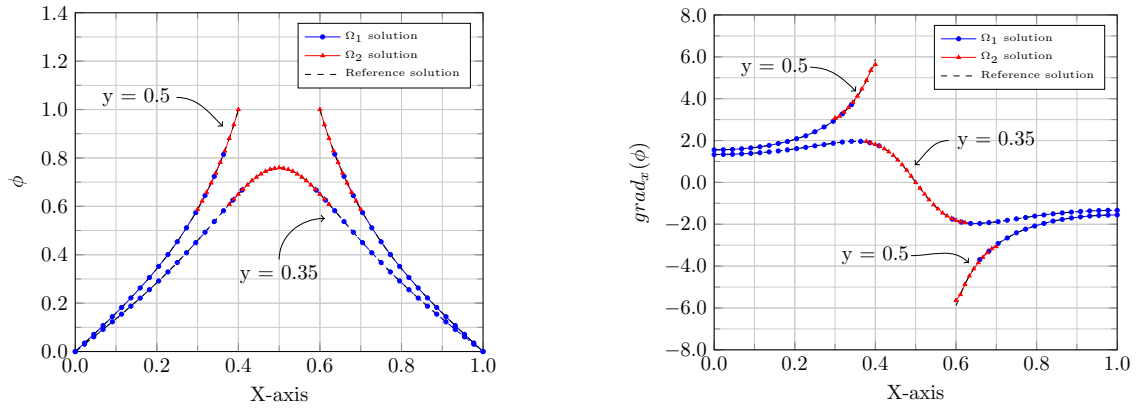


**Figure A.17:** Convergence of the solution (left) and its derivative (right) of the Laplace problem evaluated on OD homogeneous, OD refined, Chimera and Chimera regular meshes.

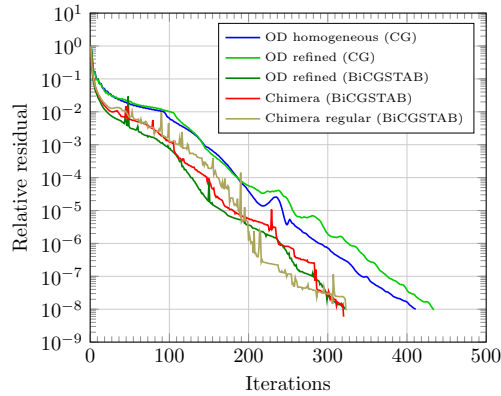


**Figure A.18:** Solution of numerical example 3, stated in Table A.4, solved on a Chimera regular mesh (left), and a detail of the overlapping zone (right). The continuity of the solution in the overlapped area can be observed.

From the numerical results shown in Figs. A.18 and A.19, a smooth transition in the overlapped region between  $\Omega_1$  and  $\Omega_2$  for both  $\phi$  and its derivative is apparent. Additionally, a good agreement with the reference solution can be observed. It can be seen from Fig. A.20 and Table A.5 that given the topology of the Chimera regular mesh, its error on  $\phi$  and  $\phi'$  is even more minor than the error of the refined Chimera mesh and takes approximately the same amount of iterations to converge. This means that the method is convenient when meshes are created independently, such as in optimization problems, where a background mesh covers the entire domain and finer meshes surrounding the bodies move freely, searching



**Figure A.19:** Solution of the Laplace problem for the Chimera regular mesh, conformed by two domains completely overlapped, projected over the planes  $y = 0.5$  and  $y = 0.35$  of the function (left) and its derivative on the x-axis (right), in contrast to the reference solution.



**Figure A.20:** Convergence history of the iterative solvers for the Laplace problem for an OD homogeneous mesh, an OD refined mesh, a Chimera mesh generated from the OD refined mesh and a Chimera regular mesh, where  $\Omega_1$  and  $\Omega_2$  are generated independently. The case corresponds to that presented in Table A.5.

for the best configuration of the system. Therefore, no changes arise in the topology of the meshes between iterations.

Mesh	N	Elements	Iterations	$L_2$ error ( $\phi$ )	$L_2$ error ( $\nabla\phi$ )
OD homogeneous (CG)	36199	71542	410	2.5935e-05	1.9260e-02
OD refined (CG)	35574	70396	433	7.1130e-06	1.5373e-02
OD refined (BiCGStab)	35574	70396	321	7.1388e-06	1.5373e-02
Chimera HO (BiCGStab)	35753	70396	319	8.2912e-06	1.5141e-02
Chimera regular HO (BiCGStab)	34787	68403	322	3.9239e-06	1.1983e-02

**Table A.5:** Summary of the results obtained for one case of the mesh convergence study of Fig. A.20.

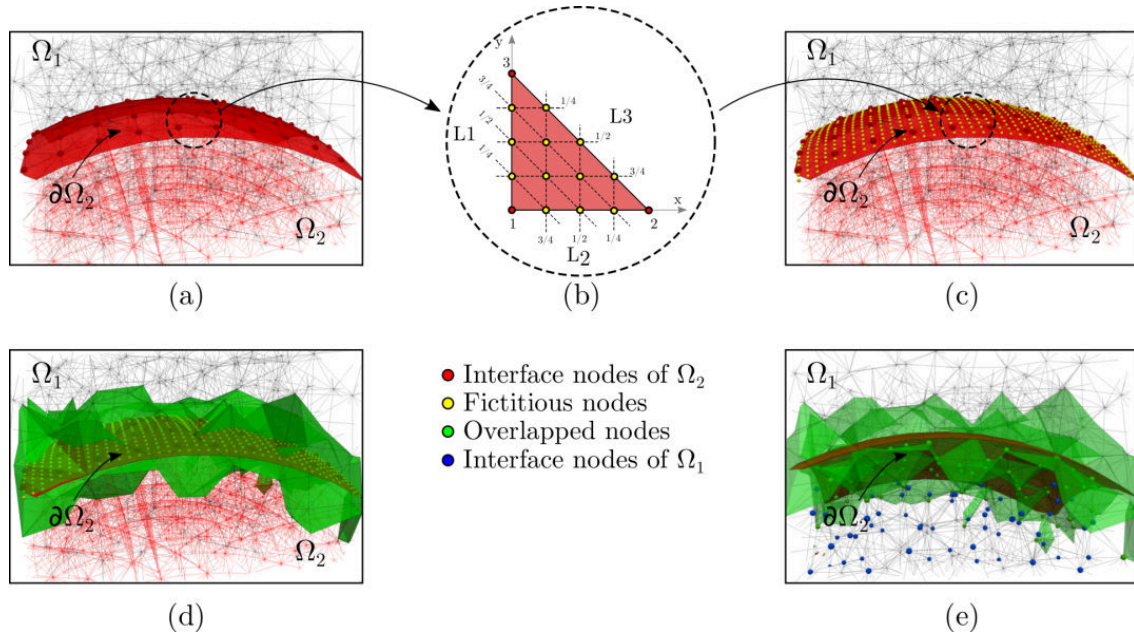
### 3.4. Numerical example 4: Three-dimensional Laplace problem on completely overlapped domains

To further assess the proposed methodology, we consider in this example a three-dimensional cubic domain with a hollow sphere of radius 0.1 in its center. In a similar way to the problem treated in section 3.3, the largest gradients on the solution of this example will be given by a boundary condition imposed over the sphere and no source term will be considered. Although the implementation of the high-order interpolation scheme is straightforward for the three-dimensional meshes, the detection of the interface nodes on the background mesh is not. Hence, some considerations of the strategy used in this work to distinguish such nodes of  $\Omega_1$  in three-dimensional problems are discussed below.

#### 3.4.1. Interpolation interface detection on three-dimensional grids

To define the interface nodes of  $\Omega_1$  in the three-dimensional space, a similar procedure to the one explained previously in section 3.2.1 is followed; however, some aspects regarding the generation of the fictitious nodes are considered and explained below.

Two three-dimensional meshes completely overlapped,  $\Omega_1$  and  $\Omega_2$ , shown in Fig. A.21 are considered. First, the interface elements of  $\Omega_1$  (tetrahedral green elements in Fig. A.21d) must be recognized, i.e., the elements of  $\Omega_1$  that are intersected by the boundary surface  $\partial\Omega_2$ . To this aim, it is necessary to generate the fictitious nodes (yellow points in Fig. A.21b) as evenly spaced as possible throughout  $\partial\Omega_2$ . Therefore, each triangle conforming the surface  $\partial\Omega_2$  (see Fig. A.21a) is mapped into a two-dimensional space where, using area coordinates (Zienkiewicz y Taylor, 2017) (see Fig. A.21), the fictitious nodes are computed and then mapped back to the three-dimensional original surface (see Fig. A.21c). For instance, in Fig. A.21b it is shown a boundary triangular element which is mapped in a two-dimensional space, whose area coordinates  $L_1$ ,  $L_2$  and  $L_3$  are split into four equal fractions, and then the coordinates of the fictitious nodes are computed from its intersections. The choosing



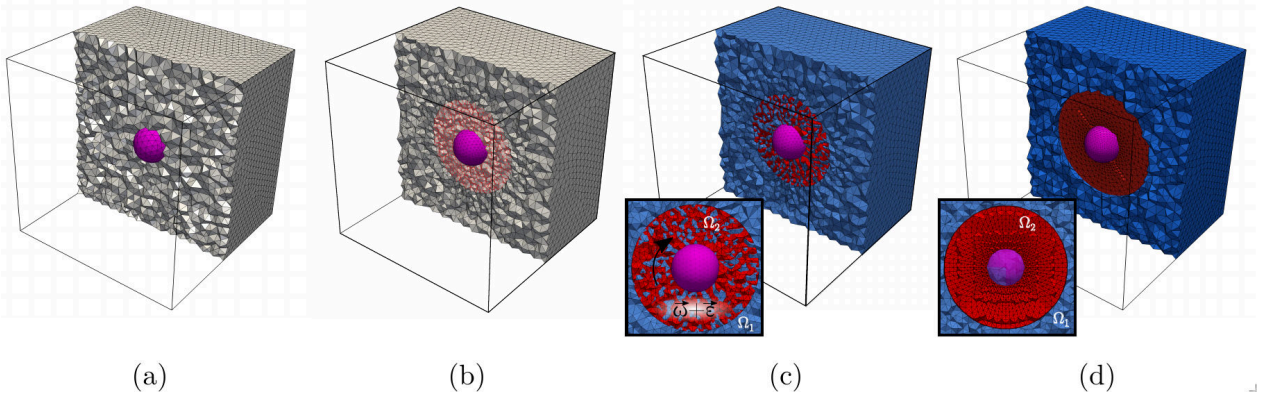
**Figure A.21:** Automatic topological detection of interpolation boundary on background three-dimensional mesh. A detail of the section of two three-dimensional grids completely overlapped,  $\Omega_1$  and  $\Omega_2$ , in conjunction with the interface boundary  $\partial\Omega_2$ , is considered.

of the number for such divisions will depend on the ratio of the element sizes of  $\Omega_1$  and  $\Omega_2$  in the overlapped region, i.e., the larger the ratio between  $h_2$  and  $h_1$ , the larger is the number of fictitious nodes that needs to be added. Afterwards, an octree search is performed to trace the elements of  $\Omega_1$  where the interface and fictitious nodes are located (see Fig. A.21d). Finally, the same procedure explained in section 3.2.1 is followed to distinguish the overlapped nodes (green nodes of Fig. A.21e) and consecutively, by performing a search for the nearest neighbours, the interface nodes on the background mesh (blue nodes in Fig. A.21e) are defined.

### 3.4.2. Numerical setup

The Laplace equation is solved in the three-dimensional domain with Dirichlet conditions imposed on the surface of the cube and in the spherical surface placed in the center (see Fig. A.22). Subsequently, the one-domain problem and the Chimera problem take the same form as that previously depicted in Table A.4. Although in this example,  $\Gamma_1$  and  $\Gamma_2$  are three-dimensional surfaces instead of planar curves. Four meshes are compared: 1. An OD homogeneous mesh conformed by unstructured tetrahedral elements with nearly constant size over the entire domain  $\Omega$  (see Fig. A.22a). 2. An OD refined mesh conformed by unstructured tetrahedral elements with decreasing size towards the spherical surface (see Fig. A.22b).

3. Two overlapped grids generated from the OD refined mesh, i.e., with almost the same topology that the conformal mesh has (see Fig. A.22c). 4. Two overlapped meshes generated in an independent manner, i.e., the background mesh ( $\Omega_1$ ) conformed by an unstructured homogeneous mesh with nearly constant element size, and a structured mesh ( $\Omega_2$ ) around the hole with progressive refinement towards the boundary  $\Gamma_2$ . We refer to this mesh as *Chimera regular* (see Fig. A.22d).



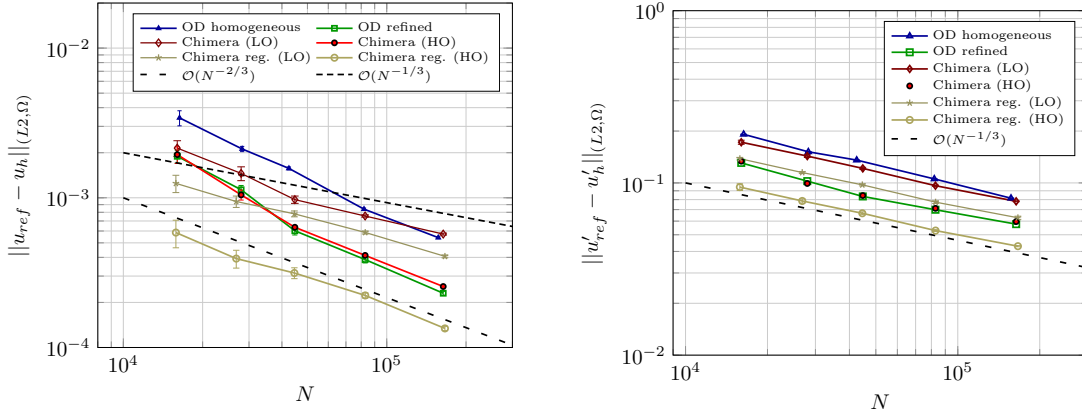
**Figure A.22:** Examples of the meshes occupied for the three-dimensional numerical test: a) OD homogeneous mesh with nearly constant element size. b) OD refined mesh with its refinement towards  $\Gamma_2$ . c) The elements in red of the OD refined mesh are randomly rotated ( $\vec{\omega}$ ) and perturbed ( $\vec{\varepsilon}$ ) to generate the  $\Omega_2$  mesh of the Chimera mesh. d) Chimera regular mesh. A non-structured homogeneous mesh is used for the background, and a structured mesh is progressively refined towards  $\Gamma_2$  for the patch.

### 3.4.3. Results

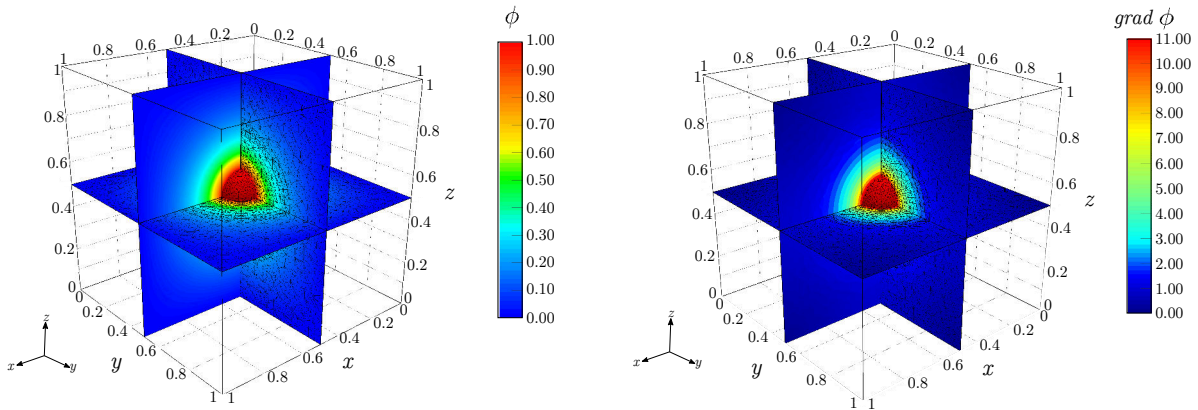
Meshes with approximately 16020, 28000, 44800, 82800 and 166000 degrees of freedom were employed to perform the convergence study. Ten runs were carried for each mesh perturbing randomly the nodes. Then, the average value and standard deviation of the  $L_2$ -norm were computed. According to Eqn.A.19, the expected convergence rates of the FEM solution and its derivative for a three-dimensional problem in terms of  $N$  reduces to  $\epsilon \approx O(N^{-2/3})$  and  $\epsilon \approx O(N^{-1/3})$  respectively. It can be seen from the results shown in Fig. A.23 that the convergence rate of  $\epsilon \approx O(N^{-2/3})$  for the solution obtained by the proposed scheme is maintained when using the high-order interpolation algorithm. Additionally, it can be observed from Fig. A.23 that the error obtained by the overlapping meshes of Fig. A.22c is similar to the one obtained by the OD refined mesh of Fig. A.22b; and, similar to the case presented in section 3.3, a better accuracy is obtained by the Chimera regular mesh shown in Fig. A.22d. The regularity of the meshes have an important role in such solution. In an analogous manner to that presented in the previous numerical example, we define the shape regularity



for a tetrahedron element as the ratio between the radius of the inscribed sphere and the length of its larger side (Parthasarathy et al., 1994), scaled by the constant  $C_t = 1/\sqrt{24}$  to give a unit value for a regular tetrahedron. The minimum value of regularity obtained by the OD refined mesh through the mesh convergence analysis was 0.3102, while for the Chimera regular mesh was 0.4752.



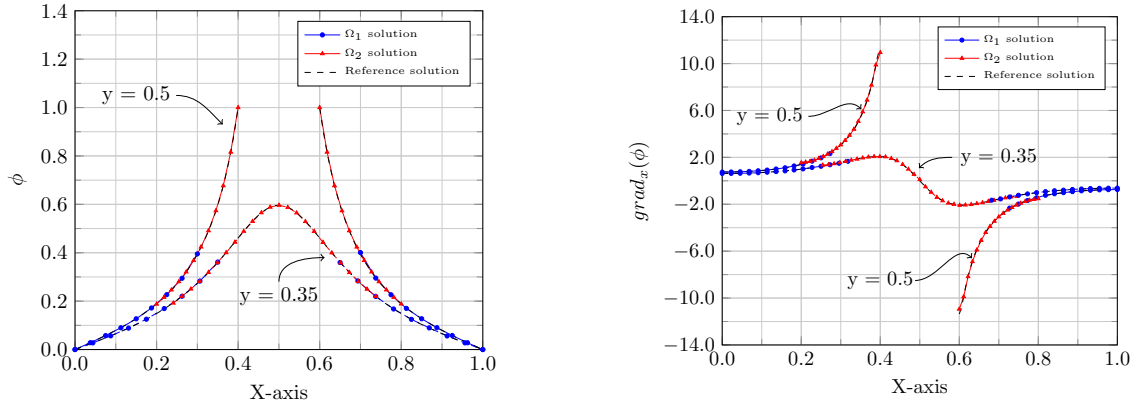
**Figure A.23:** Convergence of the solution (left) and its derivative (right) of the three-dimensional Laplace problem evaluated on OD homogeneous, OD refined, Chimera and Chimera regular meshes.



**Figure A.24:** Solution of numerical example 4 solved on a three-dimensional Chimera regular mesh (left), and its derivative (right). The continuity of the solution and its derivative in the overlapped area can be noted.

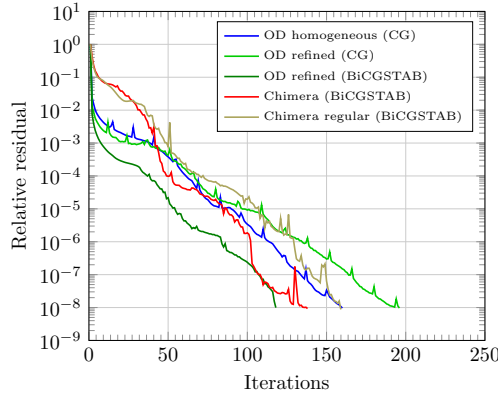
It can be seen from Fig. A.24 the continuity of the solution and its derivative for the three-dimensional Laplace equation over three cross sections of  $\Omega_1$  and  $\Omega_2$  for a Chimera regular mesh. The continuity of the solution and its derivative can also be observed over straight lines resulting from the intersections  $z = 0.50 \cap y = 0.50$  and  $z = 0.50 \cap y = 0.35$  shown in Fig. A.25 and are in accordance with the reference solution.

Good convergence rates can be observed from Fig. A.26 and Table A.6 for the coupling scheme applied to three-dimensional meshes. Furthermore, the error obtained by the Chimera regular mesh is minor than the error obtained by the refined Chimera mesh, taking slightly



**Figure A.25:** Solution of the Laplace problem for the three-dimensional Chimera regular mesh projected on the intersections of the plane  $z = 0.5$  with the planes  $y = 0.5$  and  $y = 0.35$ . Both the solution of  $\phi$  (left) and its derivative on the x-axis (right) are shown in contrast to the reference solution.

more iterations of the iterative solver to converge.



**Figure A.26:** Convergence history of the iterative solvers for the three-dimensional Laplace problem for an OD homogeneous mesh, an OD refined mesh, a Chimera mesh generated from the OD refined mesh and a Chimera regular mesh. The case corresponds to that presented in Table A.6.

## 4. Conclusions and outlook.

A Chimera scheme for non-matching overlapped unstructured grids was presented, as well as the mathematical formulation and algorithm for the iterative solution. The explicit coupling of the independent algebraic systems of each domain and the matrix-free scheme were also introduced. A high-order interpolation algorithm for the transmission of the information between meshes was introduced and explained in detail. The scheme was assessed on four numerical examples for two-dimensional and three-dimensional domains that were partially and completely overlapped, with boundary conditions imposed on both the finer and the coarser mesh. The results showed that the proposed Chimera scheme maintains the

Mesh	N	Elements	Iterations	$L_2$ error ( $\phi$ )	$L_2$ error ( $\nabla\phi$ )
OD homogeneous (CG)	156840	859720	159	5.4192e-04	8.1542e-02
OD refined (CG)	163870	911380	195	2.3043e-04	5.7631e-02
OD refined (BiCGStab)	163870	911380	117	2.3043e-04	5.7631e-02
Chimera HO (BiCGStab)	163870	911380	137	2.5545e-04	5.9631e-02
Chimera regular HO (BiCGStab)	166375	933510	158	1.3408e-04	4.2915e-02

**Table A.6:** Summary of the results obtained for one case of the mesh convergence study of Fig. A.26.

mesh convergence of the finite element method formulation and is proved to have a good convergence rate of the iterative solver using BiCGStab. Additionally, the continuity of the solution and the continuity of its derivative across the overlapped area could be observed. Regarding the computing efficiency, in all tested cases, the iterative solver of the coupled problem required approximately 19-37 % fewer iterations compared to the analogous one-domain problem solved with CG. This means that memory requirements and CPU time for the solution of the Chimera approach are similar to those required by the one-domain problem. Furthermore, although it was shown that similar errors in the  $L_2$ -norm are obtained between the Chimera and the one-domain solution when analogous mesh topologies are implemented, even less error could be achieved when the overlapping meshes are generated independently and thus capable of improving the meshes regularity. This last advantage allows application of the method, for instance, in local refinement, complex geometry treatment and optimization problems, greatly reducing the meshing work in the first two cases and avoiding the remeshing process in the last case.

## Acknowledgment

The authors are grateful for the support provided by the following research projects: Argentinian Council for Scientific Research (grant PIP-588-2015); Argentinian National Agency for Technological and Scientific Promotion (grants PICT-2015-2904 and PICT-2014-2660); Santa Fe Science Technology and Innovation Agency (grant ASACTEI-010-18-2014); National University of the Litoral (grant CAI+D-504-201501-00112-LI). This work was partially performed with resources from *Free Software Foundation/GNU-Project*, GNU/Linux OS, GCC/G++ compilers, GNU/Octave, as well as other *Open Source* tools, such as VTK, ParaView, PETSc, and L<sup>A</sup>T<sub>E</sub>X.



# References appendix A

- Abdulle A. y Huber M.E. Error estimates for finite element approximations of nonlinear monotone elliptic problems with application to numerical homogenization. *Numerical Methods for Partial Differential Equations*, 32(3):955–969, 2016.
- Andrew M. W., Aaron J. K., William M. C., y Robert L. M. Validation of the Strand Grid Approach. *AIAA Computational Fluid Dynamics*, 2009.
- Balay S., Abhyankar S., Adams M., Brown J., Brune P., Buschelman K., Dalcin L., Dener A., Eijkhout V., Gropp W., et al. PETSc users manual. 2019.
- Behr M. y Tezduyar T. The shear-slip mesh update method. *Computer Methods in Applied Mechanics and Engineering*, 174(3-4):261–274, 1999.
- Belgacem F.B. y Maday Y. The mortar element method for three dimensional finite elements. *ESAIM: Mathematical Modelling and Numerical Analysis*, 31(2):289–302, 1997.
- Belytschko T., Krongauz Y., Organ D., Fleming M., y Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47, 1996.
- Březina J. y Exner P. Fast algorithms for intersection of non-matching grids using Plücker coordinates. *Computers & Mathematics with Applications*, 74(1):174–187, 2017.
- Brezzi F., Lions J.L., y Pironneau O. Analysis of a Chimera method. *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, 332(7):655–660, 2001.
- Cai X.C., Dryja M., y Sarkis M. Overlapping nonmatching grid mortar element methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 36(2):581–606, 1999.
- Chesshire G. y Henshaw W.D. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64, 1990.
- Demkowicz L. *Computing with hp-adaptive finite elements: Volume 1 one and two dimensional elliptic and Maxwell problems*. Chapman and Hall/CRC, 2006.
- Duprez M., Lleras V., y Lozinski A. Finite element method with local damage of the mesh. *ESAIM: Mathematical Modelling and Numerical Analysis*, 53(6):1871–1891, 2019.
- Eguzkitza B., Houzeaux G., Aubry R., Owen H., y Vázquez M. A parallel coupling strategy for the chimera and domain decomposition methods in computational mechanics. *Computers & Fluids*, 80:128–141, 2013.
- Eguzkitza Bazar A.B. *HERMESH: a geometrical domain composition method in computational mechanics*. Tesis de Doctorado, Universitat Politècnica de Catalunya, 2014.
- Farhat C. y Roux F.X. A method of finite element tearing and interconnecting and its parallel solution

- algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991.
- Formaggia L., Vergara C., y Zonca S. Unfitted extended finite elements for composite grids. *Computers & Mathematics with Applications*, 76(4):893–904, 2018.
- Fries T.P. y Belytschko T. The extended/generalized finite element method: an overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.
- Hiptmair R., Li J., y Zou J. Convergence analysis of finite element methods for  $H(\text{curl}; \Omega)$ -elliptic interface problems. *Numerische Mathematik*, 122(3):557–578, 2012.
- Houzeaux G., Borrell R., Cajas J.C., y Vázquez M. Extension of the parallel Sparse Matrix Vector Product (SpMV) for the implicit coupling of PDEs on non-matching meshes. *Computers & Fluids*, 173:216–225, 2018.
- Houzeaux G., Cajas J., Discacciati M., Eguzkitza B., Gargallo-Peiró A., Rivero M., y Vázquez M. Domain decomposition methods for domain composition purpose: chimera, overset, gluing and sliding mesh methods. *Archives of Computational Methods in Engineering*, 24(4):1033–1070, 2017.
- Houzeaux G., Cajas J., Eguzkitza B., y Vázquez M. Techniques for parallel, distributed and cloud computing in engineering, volume 36 of computational science, engineering and technology series, chapter chapter 4: Parallel implementation of domain composition methods. *Saxe-Coburg Publications*, páginas 61–92, 2015.
- Houzeaux G. y Codina R. A chimera method based on a dirichlet/neumann (robin) coupling for the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(31):3343–3377, 2003.
- Houzeaux G. y Codina R. A dirichlet/neumann domain decomposition method for incompressible turbulent flows on overlapping subdomains. *Computers & fluids*, 33(5):771–782, 2004.
- Jean-Jaques Chattot Y.W. Improved treatment of intersecting bodies with the chimera method and validation with a simple and fast flow solver. *Elsevier*, 27:721–740, 1998.
- Johnson C. *Numerical solution of partial differential equations by the finite element method*. Courier Corporation, 2012.
- Kalro V. y Tezduyar T. Parallel finite element computation of 3D incompressible flows on MPPs. *Applied Mechanics and Engineering*, 108:99–118, 1993.
- Kao K.H., Liou M.S., y Chow C.Y. Grid adaptation using chimera composite overlapping meshes. *AIAA journal*, 32(5):942–949, 1994.
- Kelley C.T. *Iterative methods for linear and nonlinear equations*, volumen 16. SIAM, 1995.
- Kim H.G. Interface element method (IEM) for a partitioned system with non-matching interfaces. *Computer Methods in Applied Mechanics and Engineering*, 191(29-30):3165–3194, 2002.
- Li S. y Liu W.K. Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1):1–34, 2002.
- Mayer U.M., Popp A., Gerstenberger A., y Wall W.A. 3D fluid–structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 46(1):53–67, 2010.
- Meakin R. Moving body overset grid methods for complete aircraft tiltrotor simulations. En *11th Computational Fluid Dynamics Conference*, página 3350. 1993.
- Meakin R. An efficient means of adaptive refinement within systems of overset grids. En *12th Computational Fluid Dynamics Conference*, página 1722. 1995.

- Nielsen E.J., Diskin B., y Yamaleev N.K. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA journal*, 48(6):1195–1206, 2010.
- Parthasarathy V., Graichen C., y Hathaway A. A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design*, 15(3):255–261, 1994.
- Quarteroni A. y Valli A. Domain decomposition methods for partial differential equations numerical mathematics and scientific computation. *New York: Oxford University Press*, 1999.
- Smith B., Bjorstad P., Gropp W.D., y Gropp W. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge university press, 2004.
- Stefanica D. y Klawonn A. The FETI method for mortar finite elements. En *Proceedings of 11th International Conference on Domain Decomposition Methods*, páginas 121–129. 1999.
- Steger J.L. A chimera grid scheme: advances in grid generation. *American Society of Mechanical Engineers Fluids Engineering Division*, 5:55–70, 1983.
- Steger J.L. y Benek J.A. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(1-3):301–320, 1987.
- Tian R. y Yagawa G. Non-matching mesh gluing by meshless interpolation-an alternative to Lagrange multipliers. *International Journal for Numerical Methods in Engineering*, 71(4):473–503, 2007.
- Toselli A. y Widlund O. *Domain decomposition methods-algorithms and theory*, volumen 34. Springer Science & Business Media, 2006.
- Vázquez M., Houzeaux G., y Codina R. Chimera type domain decomposition methods applied to fractional step finite element schemes for incompressible flows. En *Proceedings of the ECCOMAS 2000 Computational Fluid Dynamics Conference*. 2000.
- Wall W.A., Gamnitzer P., y Gerstenberger A. Fluid-structure interaction approaches on fixed grids based on two different domain decomposition ideas. *International Journal of Computational Fluid Dynamics*, 22(6):411–427, 2008.
- Wang Z.J. y Parthasarathy V. A fully automated Chimera methodology for multiple moving body problems. *International Journal for Numerical Methods in Fluids*, 33(7):919–938, 2000.
- Zhang Y., Pin F.D., y Yim S.C. A heterogeneous flow model based on dd method for free surface fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 74(4):292–312, 2014.
- Zhang Y., Yim S.C., y Del Pin F. A nonoverlapping heterogeneous domain decomposition method for three-dimensional gravity wave impact problems. *Computers & Fluids*, 106:154–170, 2015.
- Zheng Y. y Liou M.S. A novel approach of three-dimensional hybrid grid methodology: Part 1. grid generation. *Computer Methods in Applied Mechanics and Engineering*, 192(37-38):4147–4171, 2003.
- Zienkiewicz O. y Taylor R. *Finite Element Method Fifth Edition Vol. I The Basic*. 2017.





# B - Optimization of an internal blade cooling passage configuration using a Chimera approach and parallel computing

El artículo presentado a continuación ha sido presentado en la revista **Finite Elements in Analysis and Design**.

B.A. Storti, L. Garelli, M.A. Storti, J. D'Elía, *Optimization of an internal blade cooling passage configuration using a Chimera approach and parallel computing*.

<https://doi.org/10.1016/j.finel.2020.103423>.



# Optimization of an internal blade cooling passage configuration using a Chimera approach and parallel computing

B.A. Storti<sup>1</sup>, L. Garelli<sup>1</sup>, M.A. Storti<sup>1</sup>, J. D'Elía<sup>1</sup>.

<sup>1</sup> Centro de Investigaciones de Métodos Computacionales,(CIMEC), (CONICET-UNL) Colectora Ruta 168 s/n, Predio Conicet "Dr. Alberto Cassano", 3000 Santa Fe, Argentina

Keywords: Chimera method; overlapping grids; parallel optimization; turbine vane; particle swarm; sequential quadratic programming

## Abstract

Improvement of gas turbines performance depends almost strictly on maximum temperature of the combustion gases. This temperature is limited by thermo-mechanical strength of the turbine vanes. In this work a Chimera approach for overlapping grids in the finite element method (FEM) context is proposed in order to optimize the disposition of several cooling passages within the vane aiming to minimize its average temperature, thus improving the thermal transfer efficiency. The scheme is based on a fixed background mesh covering the entire domain, while finer meshes surrounding the coolant passages can move around over the airfoil looking for the best configuration of the cooling system. The exchange of information between meshes is made by a high-order interpolation algorithm. The optimization strategy followed is built on the use of the population-based Augmented Lagrangian Particle Swarm Optimizer (ALPSO) to get close to a probable global minimum, and then a refinement of the solution is made using the gradient-based Sequential Least Square Quadratic

Programming (SLSQP) algorithm. Both the gradient, for the gradient-based optimizer, and the function evaluations, for the population-based optimizer, are solved in parallel to speed up the solution.

## 1. Introduction

The thermal efficiency of a gas turbine cycle depends on several factors. Increasing the temperature of combustion gases at the inlet of the expander is one of the main goals of engineers and research studies. Among all the components of a gas turbine, the vanes are the primary component affected by thermal loads. An inappropriate material design or an improper configuration of the cooling system of the vane could reduce its useful life or even lead to its failure (Bunker, 2006). Typically, such a cooling system is achieved by means of compressor bleed air flowing through several passages placed inside the vane. An optimal refrigeration system implies reducing vane thermal loads using the least amount of coolant fluid as possible. Therefore, special care must be taken regarding passage geometry and location. Since laboratory experiments for the optimization of internal cooling passages can be expensive and time consuming, numerical methods are indubitably well suited to address this task.

Several approaches for optimizing hole geometry and arrangement can be found in the literature. Wang et al. (2015), for instance, proposed optimizing the shapes, locations and dimensions of the internal cooling passages to minimize the average temperature of the vane, ensuring its structural strength. Three different passage geometries were studied: circle, super-ellipse and near-surface holes. The authors used a two-dimensional cross-sectional model for the FEM analysis of the thermal conduction within the vane and the gradient-based globally convergent method of moving asymptotes (GCMMA) for the optimization task. A similar approach using GCMMA was followed by Xie et al. (2016) based on previous studies by Nowak y Wróblewski (2011) and Nowak y Nowak (2012), where the sizes, distributions, and shapes of internal cooling channels are optimized considering new passages shaped by Bezier curves on two-dimensional models. Dennis et al. (2003) analyzed two cases considering configurations with a maximum of 13 and 30 small passages near the blade surface. The authors optimized the cooling system by means of genetic algorithms and parallel three-dimensional thermoelasticity finite element analysis. A convective heat boundary condition was defined over the surface to represent the heat flux transferred into the blade. Nowak y Wróblewski (2011) employed a reduced conjugated heat transfer (RCHT) model to make

the optimization problem computationally feasible. The external flow is fixed, and only heat conduction inside the vane is solved.

Optimization of the cooling system appears to be a daunting task involving fields of different disciplines and high computational effort. Hence, simpler and more efficient numerical methods need to be studied and implemented. Regarding the heat conduction FEM analysis of the vane, as passages change their position/shape in each function evaluation of the optimizer, complete remeshing of the vane is required by practically all the aforementioned approaches. This implies that the connectivity of the mesh changes, leading to two numerical issues: (i) the system matrix must be reallocated and recomputed in each simulation, which demands great computational effort and memory requirements; (ii) because meshes are generated automatically, this can result in the generation of highly distorted elements, obtaining poor quality meshes mainly near the boundaries, which could introduce errors in the optimization results. To simplify the mesh generation of the vane and to avoid the remeshing stage in the optimization process, a Chimera overset grid scheme is proposed in this paper. The main idea of the Chimera method (Cheshire y Henshaw, 1990; Franco Brezzi, 1997) is to generate independent meshes for the objects present in a computational domain and to combine them by a coupling strategy to obtain a unique solution of the system. One purpose for using this scheme is to avoid the difficulty in generating a general unstructured mesh for problems containing complex bodies, such as an airplane wing or a wind-turbine blade (Andrew M. et al., 2009). The Chimera scheme can be considered a domain composition method (DCM). The main objective of DCMs is to express the formulation of one problem in two or more subdomains and couple them to obtain a unique solution of the partial differential equation (Houzeaux et al., 2017). DCMs can be divided into three main groups regarding the formulation used: mesh-based, equation-based, and algebraic-based DCMs. In mesh-based formulations, the coupling between subdomains is obtained as a preprocessing step at the mesh level (Eguzkitza Bazar, 2014; Behr y Tezduyar, 1999; Dang y Meschke, 2018). In equation-based formulations, the solution of the problem is obtained by rearranging the equations from the strong and weak formulations of the variational problem (Belgacem y Maday, 1997; Duboeuf y Béchet, 2017), while in the algebraic-based approach, the solution is acquired directly in the algebraic systems of each subdomain (Houzeaux et al., 2015, 2018). The main advantage of algebraic-based formulations is that the solution obtained by expressing one equation into several subdomains is equivalent to the solution obtained by expressing the same equation in a single domain. In this work, an algebraic-based coupling methodology

is proposed. A detailed explanation of some of the previously mentioned formulations can be found in Houzeaux et al. (2017).

The Chimera method has appealing properties that are convenient for applications such as simplified mesh generation (Steger, 1983, 1991; Zheng y Liou, 2003), moving components (Prewitt et al., 2000; Blades y Marcum, 2007), local refinement (Kao et al., 1994; Meakin, 1995) and optimization (Nielsen et al., 2010). The optimization process is a straightforward application in which several objects, each with its respective mesh, can be moved around without the need to remesh the entire computational domain. Then, different optimization techniques can be used to find the optimum configuration of the system in terms of an objective function. In this work, a fixed unstructured coarse mesh (background mesh) is generated to cover the entire turbine vane, while finer structured grids (patch meshes) are generated surrounding the cooling passages and are moved over the blade to search for the optimal configuration that minimizes the average temperature. Furthermore, information passing between the meshes is achieved by a third-order interpolation algorithm to maintain the convergence and accuracy of the FEM method.

Many nonlinear optimization problems involving heat conduction over intricate geometries possess nonconvex design spaces and thus multiple local optimum values. Therefore, finding an optimal solution does not always ensure the presence of a global optimum (Hillier, 2012; Bazaraa et al., 2013; Ryoo y Sahinidis, 1995). On the one hand, although gradient-based algorithms generally converge quickly in the region of the solution, they are highly dependent on the initial trial solutions and are very likely to fall into local minima. On the other hand, population-based algorithms generally utilize some mechanism to improve the initial solutions and perform consistently well in different types of problems; therefore, they are considered to be robust methods that generally avoid local minima, and the final solution is in the vicinity of the global minimum. However, their convergence is very time consuming (Schmid et al., 2005), and for continuous variable optimization, the global minimum of the objective function is not guaranteed. It appears that hybrid approaches that combine the advantages of several optimizers is often the preferred approach (Parwani et al., 2014; Siavashi et al., 2017; Abolhasani et al., 2015). The strategy followed in this work is based on finding the probable global minimum using the constrained population-based ALPSO and then refining the solution by using the SLSQP gradient-based optimizer. Furthermore, to speed up the optimization process, the dynamic process management (DPM) functionality implemented in the pyOpt optimization framework (Perez et al., 2012) is used for ALPSO,

where function evaluations are parallelized using dynamic processor allocation, while for the SLSQP optimizer, the parallel gradient calculation (PGC) implementation is used, where the gradient function evaluations are performed in parallel.

This paper is organized as follows: first, an explanation of the algebraic system generated by the Chimera scheme is presented, and the interpolation operators to transfer the information between meshes are introduced. The algorithm proposed to couple both domains solutions solving with an iterative solver is depicted. Additionally, to maintain second-order convergence of the FEM solution, a high-order interpolator is described. Second, the pyOpt framework is introduced, and the ALPSO and SLSQP optimizers used in this work are briefly described. A simple optimization of a thermal conduction problem is solved by applying Chimera to assess the convergence and speed-up of the parallel implementation. Finally, based on experiments conducted by Hylton et al. (1983), the cooling passage configuration of a gas turbine vane is optimized, where a two-dimensional model of a cross-section of the blade is analyzed, while the thermal loads and coolant conditions are taken from experimental data.

## 2. Numerical modeling and method

An algebraic-based methodology of a Chimera method for the heat conduction equation solved in two overlapping subdomains is described in this section. The scheme consists of a coarse (background) mesh covering the entire computational domain and a finer (patch) mesh surrounding the object. An interface is defined between the two meshes, and information is interchanged by means of an interpolation operator.

### 2.1. The Chimera method for the heat conduction equation

For explaining the Chimera method, a two-dimensional square domain  $\Omega$  with a hole inside is considered (see Fig. B.1a), where it is required to solve the steady-state heat conduction equation:

$$\begin{aligned} \frac{\partial}{\partial x} \left( k \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial \phi}{\partial y} \right) + Q &= 0 & \text{in } \Omega, \\ \phi &= \bar{\phi} & \text{at } \Gamma = \partial\Omega \end{aligned} \tag{B.1}$$

where  $\phi$  denotes temperature,  $k$  is the conductivity,  $Q$  is a heat source, and  $\bar{\phi}$  is the prescribed value of temperature on the boundaries. Next, the splitting of  $\Omega$  into two domains  $\Omega_{1,2}$  is introduced, as shown in Fig. B.1b. The domains have an overlapping region  $\Omega_o$ . The nodes of the meshes  $\Omega_1$  and  $\Omega_2$  are assumed to be not coincident in  $\Omega_o$ . Now, the solution on both

domains is considered as a standard discretization with FEM (Zienkiewicz y Taylor, 2005). Hence, the resulting linear system on each domain yields a matrix equation of the form:

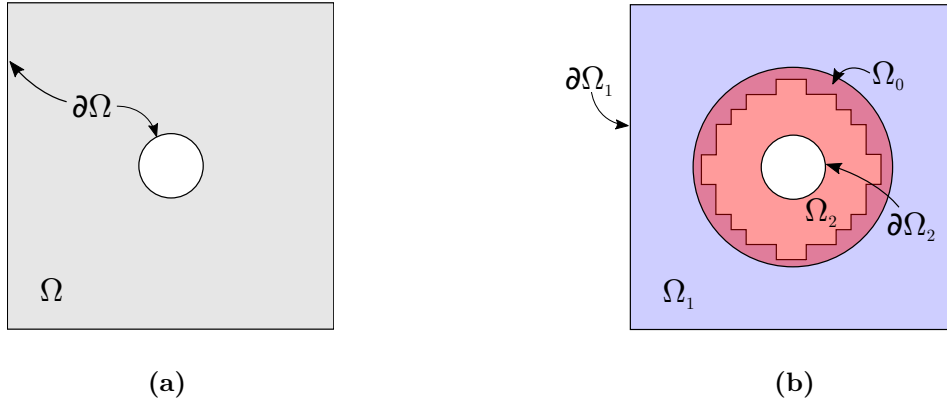
$$\mathbf{K}\mathbf{a} = \mathbf{f} \quad (\text{B.2})$$

where for each element (if the standard Galerkin approach is used), the elemental stiffness matrix and the RHS are stated as:

$$K_{ij} = \int_{\Omega} \left( \frac{\partial N_i}{\partial x} k \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} k \frac{\partial N_j}{\partial y} \right) d\Omega \quad (\text{B.3})$$

$$f_i = - \int_{\Omega} N_i Q d\Omega$$

where  $N$  represents the shape functions of the elements. It can be observed that up to this point, the matrix  $\mathbf{K}$  of each subdomain is symmetric. Now, for simplicity, let us consider



**Figure B.1:** Splitting of domain  $\Omega$  (a) into two subdomains  $\Omega_{1,2}$  with an overlapping region ( $\Omega_0$ ) (b).

that both meshes are formed by structured quadrilateral elements and that  $\Omega_1$  is completely meshed inside the inner boundary  $\partial\Omega_1$ . Then, the nodes of both meshes can be subdivided into several groups according to their location. The nodes of  $\Omega_1$  can be categorized into four main groups (see Fig. B.2a): (i) nodes placed on the boundary of  $\Omega_1$  that also belong to the boundary of  $\Omega$ , i.e.,  $\partial\Omega_1 \cap \partial\Omega$  (group  $B$ ); (ii) nodes that are used for the interpolation, i.e.,  $\partial\Omega_1 \setminus \partial\Omega$  (group  $I$ ), constituting the interpolation boundary (further discussed in section 2.3.1); (iii) nodes that are bounded by the interpolation boundary and that are also in the overlapping region (group  $H$ ); and (iv) the remaining nodes of the domain (group  $Z$ ). Additionally, the nodes of  $\Omega_2$  can be divided into three groups (see Fig. B.2b): (i) nodes placed on the boundary of  $\Omega_2$  that also belong to the boundary of  $\Omega$ , i.e.,  $\partial\Omega_2 \cap \partial\Omega$  (group  $B$ ); (ii) nodes that are used for the interpolation (group  $I$ ); and (iii) the remaining nodes of the domain (group  $Z$ ). It is necessary for the nodes of  $I$  of each domain to be fully overlapped in the corresponding domain to be able to perform an appropriate interpolation from the



other mesh, as fully explained in the next paragraphs. The splitting of the nodes on each subdomain induces a corresponding splitting of the linear system for each as follows:

$$\begin{bmatrix} \mathbf{K}_{BB} & \mathbf{K}_{BZ} & \mathbf{K}_{BI} \\ \mathbf{K}_{ZB} & \mathbf{K}_{ZZ} & \mathbf{K}_{ZI} \\ \mathbf{K}_{IB} & \mathbf{K}_{IZ} & \mathbf{K}_{II} \end{bmatrix} \begin{bmatrix} \mathbf{a}_B \\ \mathbf{a}_Z \\ \mathbf{a}_I \end{bmatrix} = \begin{bmatrix} \mathbf{f}_B \\ \mathbf{f}_Z \\ \mathbf{f}_I \end{bmatrix} \quad (\text{B.4})$$

Temporarily, hole nodes ( $H$  nodes) from  $\Omega_1$  are considered part of the group of the interior nodes ( $Z$  nodes), and their treatment is further discussed in Section 2.4. The values of  $\mathbf{a}_B$  are already known due to the boundary condition. If the values of  $\mathbf{a}_I$  were known, then the problem would be well determined with Dirichlet conditions in the whole boundary, and the following problem for the interior nodes could certainly be solved:

$$\mathbf{K}_{ZZ}\mathbf{a}_Z = \mathbf{f}_Z - \mathbf{K}_{ZB}\mathbf{a}_B + \mathbf{K}_{ZI}\mathbf{a}_I \quad (\text{B.5})$$

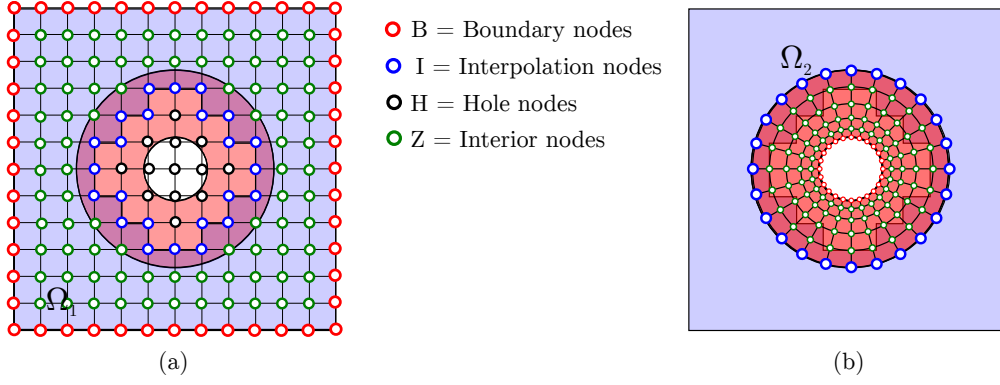
Since the values of  $\mathbf{a}_I$  are not known, the interpolation of these values from the inside nodes of the corresponding overlapped domain can be considered. In this way, a fully linear system in the interior values of both domains can be obtained. Using supra-indices for both domains, we obtain the following algebraic equations:

$$\begin{aligned} \mathbf{K}_{ZZ}^1 \mathbf{a}_Z^1 &= \mathbf{f}_Z^1 - \mathbf{K}_{ZB}^1 \mathbf{a}_B + \mathbf{K}_{ZI}^1 \Pi_{I_{1,2}} \mathbf{a}^2, \\ \mathbf{K}_{ZZ}^2 \mathbf{a}_Z^2 &= \mathbf{f}_Z^2 - \mathbf{K}_{ZB}^1 \mathbf{a}_B + \mathbf{K}_{ZI}^2 \Pi_{I_{2,1}} \mathbf{a}^1, \end{aligned} \quad (\text{B.6})$$

$\Pi_{I_{1,2}}$  is a projection operator that interpolates the values of an  $\mathbf{a}^2$  vector in the nodes on  $\mathbf{a}_I^1$ . In an analogous way, the projection operator  $\Pi_{I_{2,1}}$  interpolates the values from the inside nodes of  $\Omega_1$  over the interpolation boundary of  $\Omega_2$  ( $\mathbf{a}_I^2$ ). It is assumed that in each domain, there is at least one full element layer in the overlapping region such that the interpolation involves only nodes in the  $B^2$  and  $Z^2$  nodes, not on  $I^2$  nodes. Therefore, the linear system (see Eq. (B.6)) can be solved for the interior values  $\mathbf{a}_Z^1$  and  $\mathbf{a}_Z^2$ .

## 2.2. Algorithm of the Chimera scheme

On the one hand, the coupling scheme presented in Section 2.1 that can be set as a linear operator that can be fed to a matrix-free iterative solver. Matrix-free methods do not store the coefficients of the matrix explicitly but access the matrix by evaluating matrix-vector products. In this way, when a large matrix is treated, the memory requirement and computational time can be greatly reduced. On the other hand, the resulting linear system of the coupling is not symmetric or positive definite. However, it is well conditioned, and



**Figure B.2:** Splitting of the nodes of  $\Omega_1$  and  $\Omega_2$  into groups according to their location.

numerical experiments have shown remarkable convergence rates using BiCG-Stab (further explained in Section 2.5). Algorithm 1 computes the residual from the interior values of both subdomains. It takes as input all the node values of  $\Omega$ , i.e.,  $\mathbf{a}^1 \cup \mathbf{a}^2$ , and then interior node values are extracted ( $\mathbf{a}_Z^1 \cup \mathbf{a}_Z^2$ ) to perform the interpolation over the interface nodes ( $I^1$  and  $I^2$ ). First, the interpolation of nodes on  $I^1$  is computed by means of the interpolation operator  $\Pi_{I_1,2}$  and interior node values of  $\Omega_2$ . Subsequently, the residual for the interior values of  $\Omega_1$  is calculated according to line 1 of Eq. (B.7). In an analogous manner, the residual for the interior nodes of  $\Omega_2$  is computed, and both ( $\mathbf{r}_Z^1$  and  $\mathbf{r}_Z^2$ ) are combined into a single vector  $\mathbf{r}$ . This strategy is referred to as iteration-by-subdomain in the literature (Houzeaux y Codina, 2003; Houzeaux et al., 2017).

$$\begin{aligned} \mathbf{r}_Z^1(\mathbf{a}_Z^1, \mathbf{a}_Z^2) &= \mathbf{K}_{ZZ}^1 \mathbf{a}_Z^1 - \mathbf{f}_Z^1 + \mathbf{K}_{ZB}^1 \mathbf{a}_B^1 - \mathbf{K}_{ZI}^1 \Pi_{I_1,2} \mathbf{a}_Z^2, \\ \mathbf{r}_Z^2(\mathbf{a}_Z^1, \mathbf{a}_Z^2) &= \mathbf{K}_{ZZ}^2 \mathbf{a}_Z^2 - \mathbf{f}_Z^2 + \mathbf{K}_{ZB}^2 \mathbf{a}_B^2 - \mathbf{K}_{ZI}^2 \Pi_{I_2,1} \mathbf{a}_Z^1. \end{aligned} \quad (\text{B.7})$$

---

**Algorithm 1:** Compute residuals

---

**Input** :  $\mathbf{a}$

**Output:**  $\mathbf{r} = f(\mathbf{a})$

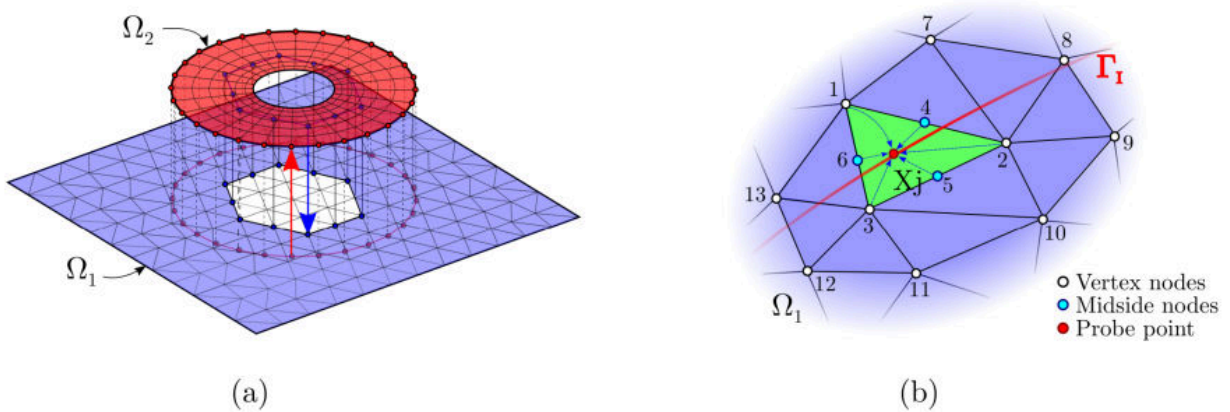
- 1: Extract  $\mathbf{a}_Z^1, \mathbf{a}_Z^2$  from  $\mathbf{a} = [\mathbf{a}_Z^1; \mathbf{a}_Z^2]$
  - 2: Interpolate  $\mathbf{a}_Z^2$  on nodes  $I^1$
  - 3: Compute residual  $\mathbf{r}_Z^1$  on  $Z^1$  according to Eq. (B.7) (line 1).
  - 4: Interpolate  $\mathbf{a}_Z^1$  on nodes  $I^2$
  - 5: Compute residual  $\mathbf{r}_Z^2$  on  $Z^2$  according to Eq. (B.7) (line 2).
  - 6: Combine  $\mathbf{r} = [\mathbf{r}_Z^1; \mathbf{r}_Z^2]$
-

### 2.3. Interpolation scheme

The spatial convergence order and error of the FEM solution are affected by the interpolation scheme. As will be seen in the numerical problem of Section 2.5.1, from multiple simulations performed with decreasing element sizes, we concluded that to preserve the second-order convergence of the FEM method, a third-order interpolation is required. Below, a brief explanation of the interpolation scheme used in the present work is presented.

#### 2.3.1. High-order interpolator

As previously described in Section 2.1, a group of nodes of each mesh ( $I^1$  and  $I^2$ ) are set as interpolation nodes and are used to interpolate the values from one mesh to the other and vice versa (see Fig. B.3a). For the explanation of the interpolation algorithm, an unstructured mesh  $\Omega_1$  composed of simplices (see figure B.3b) is assumed, and the values of a smooth function  $\phi$  are known at the node vertices. Additionally, it is assumed that there are some probe points  $\mathbf{x}_j$  belonging to the interpolation boundary of  $\Omega_2$ , and a third-order interpolation of  $\phi$  is required at the probe points. The interpolation is based, first, on reconstructing the value of  $\phi$  in the midside nodes (in cyan) of the element where  $\mathbf{x}_j$  is placed (green element) and, second, on performing a standard Lagrangian  $P_2$  interpolation. For instance, if the value



**Figure B.3:** High-order interpolation algorithm.

of node 5 needs to be reconstructed, a least squares (LS) fitting is performed using the values of the vertex nodes  $E = \{2,3\}$ , as well as the values of its topological neighbors  $T = \{1,7-13\}$ .  $\tilde{\phi}$  is the second-order reconstruction:

$$\tilde{\phi}(x, y) = a + bx + cy + dx^2 + e xy + f y^2. \quad (\text{B.8})$$

When it is applied to the nodes  $U = N + T$ , the reconstruction yields the following matrix form:

$$\tilde{\phi} = \Psi \mathbf{b}. \quad (\text{B.9})$$

where  $\mathbf{b}$  is a vector of size  $n_b = 6$  that contains all the polynomial coefficients and  $\Psi$  is a matrix of size  $u \times n_b$  containing the variables of the system, where  $u = |U| = 10$  represents the number of nodes used in the reconstruction. Because the reconstruction cannot be imposed at all nodes of  $U$  since the system would be overdetermined ( $u = 10 > n_b = 6$ ), the coefficients of  $\mathbf{b}$  are determined such that the fit is exact for the vertices of the edge (E) and optimal in the LS sense for the topological neighbors (T). Therefore, using Lagrange multipliers, the Lagrangian function to be optimized becomes:

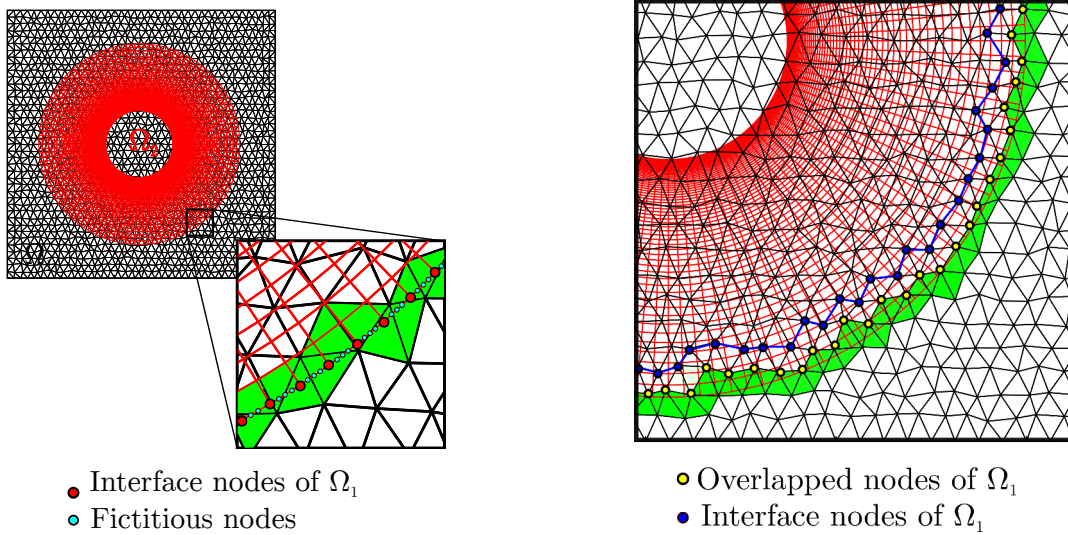
$$\mathcal{L}(\mathbf{b}, \boldsymbol{\lambda}) = \|\Psi_T \mathbf{b} - \phi_N\|^2 - \boldsymbol{\lambda}^T (\Psi_E \mathbf{b} - \phi_E) \quad (\text{B.10})$$

### 2.3.2. Interpolation boundary detection by topological search

When overset unstructured grids are used in a Chimera scheme context, two main scenarios can be encountered: (i) the domains  $\Omega_1$  and  $\Omega_2$  are partially overlapped, or (ii) the domain  $\Omega_2$  is completely overlapped by  $\Omega_1$ . When domains that are partially overlapped are used, the detection of the interpolation boundaries does not represent a difficult task, given that such interfaces would be formed by boundary nodes that are easy to distinguish. In contrast, when domains are completely overlapped, as is the case in this work, the interpolation nodes that are interior to  $\Omega_1$ , where the values of the fine mesh will be interpolated, are not placed in its boundary; hence, a topological detection strategy must be used to identify such nodes. It is important that the algorithm implemented for the recognition meets three main aspects:

1. At least one full layer of elements of the background mesh must be present between both interpolation boundaries;
2. The interpolation border must form a bounded area isolating the interior nodes from the hole nodes of  $\Omega_1$ , i.e.,  $I$  nodes must not be linked to any  $H$  nodes, only to  $B$  and  $Z$  nodes;
3. The resulting geometry of the interpolation boundary of  $\Omega_1$  must be as similar as possible with respect to the geometry of the interpolation boundary of  $\Omega_2$ .

To define the interface nodes of  $\Omega_1$ , the procedure is as follows: First, the interface elements of  $\Omega_1$  (green elements in Fig. B.4a) are recognized, i.e., the elements of  $\Omega_1$  where the boundary



**Figure B.4:** Topological interpolation boundary detection. First, in (a), the interface elements of  $\Omega_1$  (green elements) are recognized from the  $I^2$  nodes and the fictitious nodes. Then, in (b), overlapped and interface nodes of  $\Omega_1$  are distinguished.

nodes of  $\Omega_2$  are found. To achieve this, new nodes called fictitious nodes (cyan points in Fig. B.4a) between the interface nodes of  $\Omega_2$  (red nodes in Fig. B.4b) are created to have a better frontier resolution. The number of fictitious nodes to be added depends on the difference between the element sizes of both meshes,  $h_1$  and  $h_2$ , respectively, in the coupling region. The larger the relation between  $h_2$  and  $h_1$  on the interpolation boundary is, the larger is the number of fictitious nodes that need to be added. Consequently, an efficient quadtree search of the red and yellow nodes is performed, and the interface elements are identified. Then, the nodes of the interface elements that are placed in the overlapping region are referred to as *overlapped nodes* (yellow nodes in Fig. B.4b), and they are placed in the first layer. Subsequently, a search for the nearest neighbors of the overlapped nodes is performed. Only those nodes that are inside the overlapping region are defined as the interface nodes of  $\Omega_1$  (blue nodes in Fig. B.4b), also commonly referred to as *fringe nodes* in the literature (Eguzkitza et al., 2013; Steger y Benek, 1987; Steger, 1983) and would be placed in the second layer. For the following numerical experiments, the nodes over the third layer are used for the interpolation, which is in accordance with (Vázquez et al., 2000). In this way, two layers of fully overlapped elements are always ensured.

#### 2.4. Treatment of the hole nodes on the background mesh

As shown in Sec. 2.1, the hole nodes of the background domain were considered as part of the interior node group during the governing equation assembly. However, those nodes

should not be considered as part of the final global solution. Hence, one could choose to eliminate the hole nodes from the background mesh before the corresponding computations. Of course, this would imply that the nodes and elements are renumbered, thereby changing the connectivity of the mesh. When the Chimera method is applied to problems with moving bodies, either to fluid-structure interaction or optimization problems, for instance, this is the least desirable option since the change in the mesh connectivity on each time-step/objective function computation would lead to high computational cost and memory requirements. Therefore, in the present implementation, the hole elements and nodes are not eliminated to preserve the connectivity of the background mesh and the shape of the system matrix. Because there are no internal nodes linked to a hole node, any value of the solution obtained on a hole node would not be relevant. The main drawback of this approach is that purposeless matrix-vector computations must be performed by the iterative solver. Henceforth, when we refer to the number of degrees of freedom, number of nodes or number of elements of the Chimera mesh, neither hole nodes nor hole elements are considered. Moreover, the solution in these nodes/elements will not be considered on the final global solution.

## 2.5. Mesh convergence and iterative solution

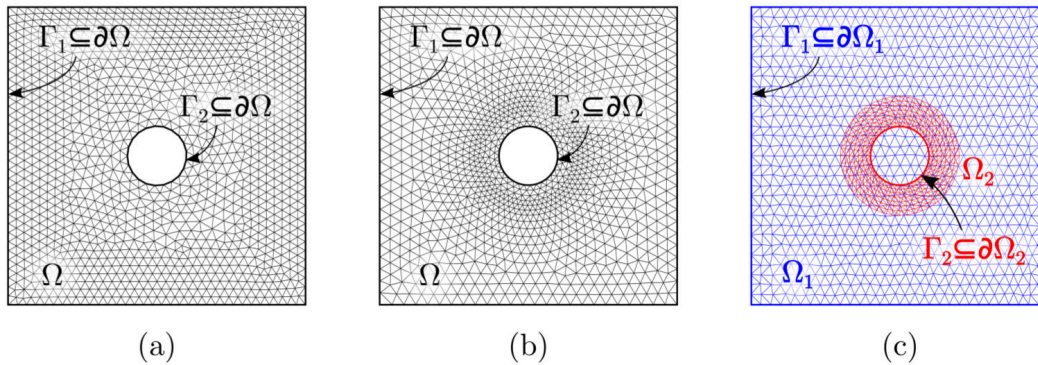
The convergence of iteration-by-subdomain methods, whether they are implemented in the algebraic-based or equation-based context, is often a problem (Houzeaux et al., 2017). There are mainly three approaches to accelerate the convergence: (i) the introduction of a relaxation parameter in the equations, (ii) consideration of different preconditioners and (iii) consideration of efficient solvers such as GMRES or BiCGStab. Although the use of more efficient solvers is not a trivial task and implies some code rewriting, it was the path followed in this work to guarantee convergence. Although the original one-domain (non-Chimera) problem is symmetric positive definite and could certainly be efficiently solved with CG, the multidomain Chimera problem is not; hence, the use of CG would not guarantee the convergence of the system. Therefore, the BiCGStab solver was used in this work. The computational cost per iteration of the BiCGStab algorithm is similar to the cost per iteration of the CG algorithm. The main difference is that BiCGStab has one additional matrix-vector product than CG and requires storing three additional vectors (Kelley, 1995).

One condition expected from the coupling is that the mesh convergence achieved would be  $O(h^2)$ , where  $h$  is the element size, as in the cases of conformal one-domain meshes. Another aspect to consider is the error in the  $L_2$ -norm obtained by the coupled scheme. One would

expect at most an error similar to that of an analogous conformal (one-domain) mesh if the algorithm is implemented correctly. Below, a numerical example is presented treating these aspects.

### 2.5.1. Error and convergence assessment of the Chimera approach

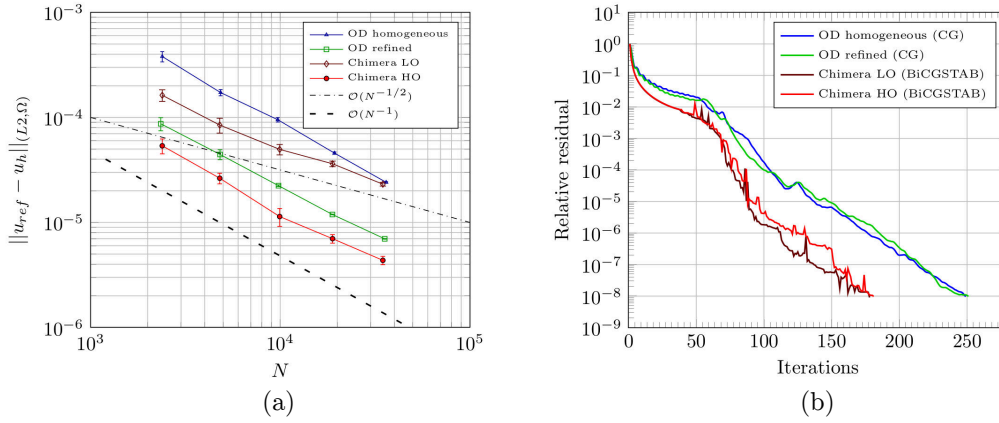
To assess the convergence rate and the error in the  $L_2$ -norm obtained by the Chimera scheme, a heat conduction problem is solved on a two-dimensional squared domain of  $[0, 1] \times [0, 1]$  with a hole in the center with radius 0.1. Two interpolation schemes are tested: (i) a simple Low-Order (LO) linear interpolation, whose convergence order is  $O(h^2)$ , and (ii) the previously detailed High-Order (HO) interpolation algorithm, whose convergence order is  $O(h^3)$ . The results are compared with those obtained by one-domain homogeneous meshes with a practically constant element size throughout the domain (OD homogeneous) and refined meshes with element size progressively decreasing towards the hole (OD refined) (see Fig. B.5). A fixed temperature value of 100 [°C] is imposed in the outer boundary ( $\Gamma_1$ ), and a convective boundary condition is imposed in the inner boundary ( $\Gamma_2$ ). The heat transfer coefficient  $h$  and the temperature inside the hole  $T_{bulk}$  are set to 100 [ $\text{W m}^{-2} \text{°C}^{-1}$ ] and 0 [°C], respectively. A constant value for the conductivity equal to 1 [ $\text{W m}^{-1} \text{°C}^{-1}$ ] is set in the entire computational domain.



**Figure B.5:** Conceptual illustration of the meshes used for analysis.

Ten simulations were computed for the meshes with approximately 2,400, 4,800, 9,700, 19,300 and 36,000 degrees of freedom ( $N$ ). The elements of each mesh were randomly and slightly distorted between each computation to consider the most general case. Then, the average value and standard deviation of the  $L_2$ -norm were obtained. A very fine mesh with 260,680 degrees of freedom (519,060 elements) was used as a reference for the  $L_2$ -norm computation. The results in Fig. B.6a show, on the one hand, that the error convergence rate of

Chimera HO is  $\epsilon \approx O(N^{-1})$ , meaning that the error convergence rate of FEM is maintained (Demkowicz, 2006) when the high-order interpolation scheme is used. On the other hand, when linear interpolation is employed, the error convergence rate is  $\epsilon \approx O(N^{-1/2})$ , putting in evidence the loss of convergence and accuracy of the solution. Although these results suggest that for elements of higher-order one would require even higher-order interpolation schemes to guarantee proper mesh convergence, e.g. fourth-order interpolation when using quadratic elements, this work does not delve into the topic and focuses on the coupling of meshes with linear elements. Fig. B.6a also shows that Chimera HO presents a better approximation to the reference solution than the OD refined mesh. This is due to the mesh quality that one can obtain by generating them independently. Fig. B.6b depicts the convergence histories of



**Figure B.6:** Error and convergence analysis: (a) mean error and standard deviation of the  $L_2$ -norm for ten runs, (b) convergence histories of the iterative solvers for meshes with approximately 9700 degrees of freedom.

the iterative solvers for the three meshes with approximately 9,700 degrees of freedom and a relative residual tolerance set to  $10^{-8}$ . A good convergence rate can be observed with the coupled scheme reaching the preset tolerance in 178 and 181 iterations, for Chimera LO and HO respectively, solved with BiCGStab. While for the OD homogeneous and refined meshes, it required 249 and 251 iterations, respectively, to converge with CG. Summarizing these results, three important outcomes can be highlighted:

- The BiCGStab solver used for the Chimera problem required fewer iterations compared to those required by the CG solver for the one-domain refined problem.
- The proposed coupling scheme maintains the mesh convergence of the FEM formulation when the high-order interpolation is employed.
- The error of the solution in the  $L_2$ -norm obtained by Chimera HO remains lower than that obtained by the one-domain refined mesh. This is because the scheme allows the



generation of the overlapping meshes independently, thereby improving the mesh quality.

### 3. Optimization problems on overlapped grids

Several applications can take advantage of the Chimera method characteristics. Configuration optimization is a straightforward application where a number of objects, conforming to the patch mesh, can move around over a background grid that covers the entire computational domain, searching for the best configuration regarding a specific objective function. In this section, we introduce the pyOpt framework (Perez et al., 2012), an open source code with several optimization solvers for nonlinear and constrained problems. These solvers can be categorized into two main groups: gradient-based optimizers and gradient-free optimizers. We briefly describe in the following sections some of the optimizers used in this work, as well as the parallel and refinement capabilities of pyOpt that were employed.

#### 3.1. Hybrid optimization

Most engineering optimization problems possess multiple local minima and nonconvex design spaces. The use of gradient-based optimizers is very limited for this type of problem given that they are very dependent on the initial trial solutions and can easily fall into a local minimum. The strategy followed in this work is based on finding the probable global minimum using the constrained population-based ALPSO and then refining the solution by using the SLSQP gradient-based optimizer. The object-oriented class structure implemented in pyOpt allowed us to directly pass the ALPSO solution to the SLSQP optimizer and use it as an initial point. The code implementation for the solution refinement is as follows:

```
### Solution refinement with gradient-based optimization ###
alpsso(opt_prob, store_sol=True, disp_opts=True , store_hst=True)
print opt_prob.solution(0)
slsqp(opt_prob.solution(0),sens_type='FD',sens_mode='pgc',sens_step=1e-5)
print opt_prob.solution(0).solution(0)
```

## 3.2. Optimization algorithms

### 3.2.1. ALPSO Optimizer

The ALPSO algorithm (Sedlaczek y Eberhard, 2006) is used to solve nonlinear non-smooth optimization problems subject to equality and inequality constraints. The algorithm uses an augmented Lagrange multiplier approach to transform the constrained optimization problem into an unconstrained optimization problem (Sedlaczek y Eberhard, 2006). Since this approach is based on PSO methods (Eberhart y Kennedy, 1995), it is more likely to find the global minimum of the objective function. Each particle of the swarm is defined by its position and velocity (Eq. B.11), and each constitutes a solution to the problem. First, each particle evaluates the objective function at its random initial position. Then, each particle is updated according to:

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \\ \mathbf{v}_i^{k+1} &= \omega \mathbf{v}_i^k + c_1 r_1 (\mathbf{x}_i^{p_{best},k} - \mathbf{x}_i^k) + c_2 r_2 (\mathbf{x}_i^{g_{best},k} - \mathbf{x}_i^k) \end{aligned} \quad (\text{B.11})$$

where  $p_{best}$  stands for the best value that the particle has achieved thus far;  $g_{best}$  is the best particle among the entire population;  $\omega$  is the inertia weight factor;  $\mathbf{v}_i^k$  and  $\mathbf{x}_i^k$  are the particle velocity and position, respectively, at iteration  $k$ ;  $r_1$  and  $r_2$  are random numbers between (0,1); and  $c_1$  and  $c_2$  are called cognitive and social parameters.

### 3.2.2. SLSQP Optimizer

The SLSQP algorithm implemented by Kraft y Schnepfer (1989) is one that evolved from the LS solver of Lawson y Hanson (1995). The algorithm uses a quasi-Newton Hessian approximation (Bazaraa et al., 2013; Fletcher, 2013; Nocedal y Wright, 2006) with a BFGS update of the B-matrix and an L1-test function in the line search algorithm. The SQP methods are used on mathematical problems for which the objective function and the constraints are twice continuously differentiable. Their algorithms are based on solving a sequence of optimization subproblems, each of which optimizes a quadratic model of the objective subject to a linearization of the constraints. To compute the gradient, two possible options are available in the pyOpt framework: the finite difference (FD) and the complex step (CS). The class *Gradient* has both implementations and can be computed in a serial or parallel manner. In this work, the FD is used for the sensitivity analysis given that the computational cost is lower than that employed by the CS (Martins et al., 2003), and the difference in terms of accuracy would be negligible for the step size used in this work (see section 3.5.2).

### 3.3. Parallelization of the optimization process

In many engineering optimization problems, evolutionary algorithms generally take a large number of function evaluations to converge, which could be computationally very demanding. Furthermore, if a gradient-based algorithm is used, gradient computation can also be expensive in terms of computational effort. Parallel calculation of the objective function, for evolutionary algorithms, or of the gradient, for gradient-based algorithms, can significantly reduce the time required to solve such problems. In this work, the parallel processing capabilities implemented in pyOpt by means of the mpi4py package (Dalcin et al., 2011), in conjunction with the Slurm Workload Manager (Jette et al., 2002), are used. For ALPSO, the dynamic process management (DPM) is used, where functional evaluations are parallelized using dynamic processor allocations. This setting is defined at the optimizer instantiation as follows and is used at run time.

```
# Parallel function evaluations of ALPSO using DPM
alps = ALPSO(pll_type='DPM')
```

For the SLSQP algorithm, the parallel gradient calculation (PGC) was used, and gradient function evaluations were performed in parallel. This setting is defined at run time as follows:

```
# Parallel gradient computation of SLSQP
slsqp(opt_prob, sens_type='FD', sens_mode='pgc')
```

The algorithm is implemented in **Seshat** (Seshat, 2018), a computer cluster located in the Computing Center of the CIMEC Institute. It consists of a total of 69 nodes, where each one has an Intel(R) Xeon(R) CPU E5-1620 v2 @ 3.70 GHz (1 CPU x 4 cores) processor and 16 GB of RAM (DDR3 and 1600 Mhz). The cluster has an InfiniBand network connection with QDR 40 Gbps, which is mandatory for high-performance computing.

### 3.4. Optimization strategy

The optimization procedure consists of two main blocks: the preprocess stage, which is computed only once at the start of the optimization, and the optimization-process stage, which involves the Chimera and optimization algorithms (see Fig. B.7). The modeling and meshing of the background and patch meshes are performed as a preprocess, and the corresponding boundary conditions are set. Additionally, the interpolation nodes of the patch meshes and the number of layers are defined. The optimization stage consists of moving the

patch meshes according to the new set of variable values on each objective function evaluation. Then, proper interpolation interface detection, interpolation operator computing and FEM analysis are performed. The FEM analysis along the Chimera coupling is performed using the open-source software PETScFEM (Storti et al., 2007; Sonzogni et al., 2002), which is a general purpose, parallel, multiphysics FEM code developed by the authors of this work.

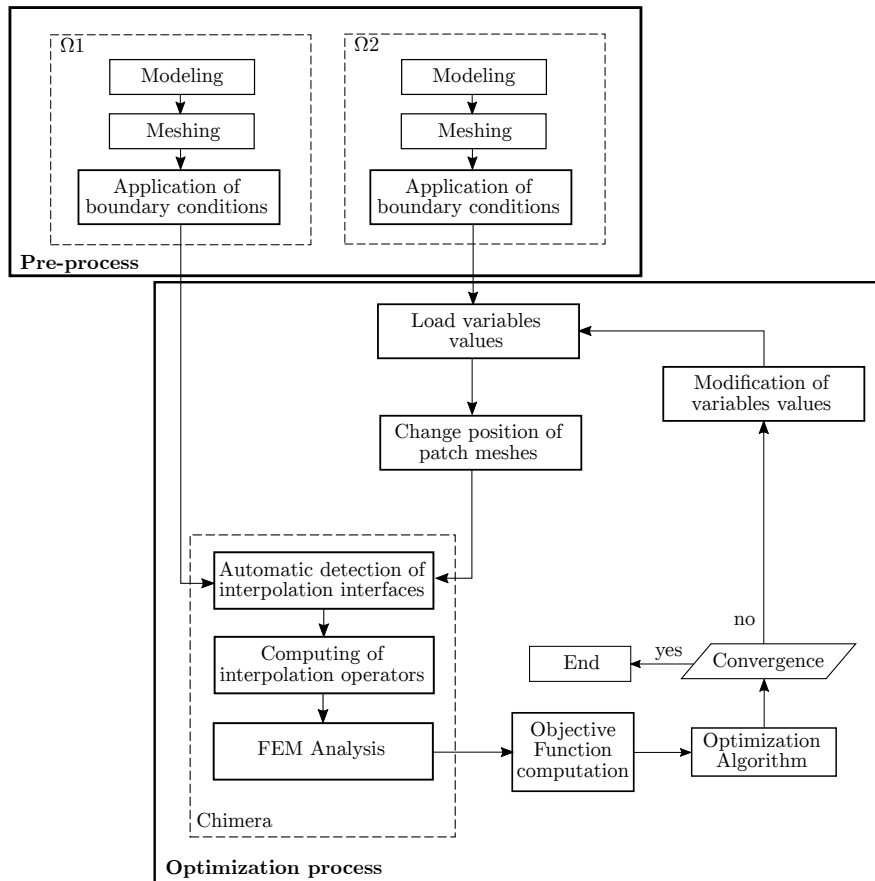


Figure B.7: Optimization strategy using the Chimera method.

### 3.5. Numerical example

Prior to optimizing the cooling passage configuration of the blade, a simpler, although not trivial, numerical test is performed to assess the convergence and efficiency of Chimera coupled with the optimizers, and it will serve as a benchmark case. The main objective of this test is to minimize the average temperature of a two-dimensional quadrangular plate by optimizing the positions of eight cooling passages inside the plate. The refrigeration holes start with completely random positions. The remeshing stage is completely avoided through the entire optimization process.

### 3.5.1. Problem setup

In the following numerical test, the heat conduction equation given by Eq. (B.12) is solved on two-dimensional domains with a constant value for the conductivity equal to 1 [ $\text{W m}^{-1} \text{°C}^{-1}$ ]. The background mesh  $\Omega_1$  has imposed a fixed temperature as a boundary condition of 100 [ $\text{°C}$ ] (see Fig. B.8a), and the patch mesh  $\Omega_2$  is composed of eight small holes of diameter 0.03 with a convective boundary condition imposed over the inner boundary (see Fig. B.8b) acting as refrigeration holes. The  $\Omega_1$  and  $\Omega_2$  domains are formed by 139,400 triangular unstructured elements and 37,120 structured elements, respectively, and 4,640 elements surround each hole. The heat transfer coefficient  $h$  and the temperature of the coolant  $T_{bulk}$  inside each hole are assumed to be constant: 100 [ $\text{W m}^{-2} \text{°C}^{-1}$ ] and 0 [ $\text{°C}$ ], respectively.

The optimization problem is based on minimizing the average domain temperature  $T_{avg}$  (see Eq. (B.13)) subject to the restriction that a minimum distance  $c_{min}$  between each pair of holes  $c_j$  must be guaranteed to avoid superposition between them.

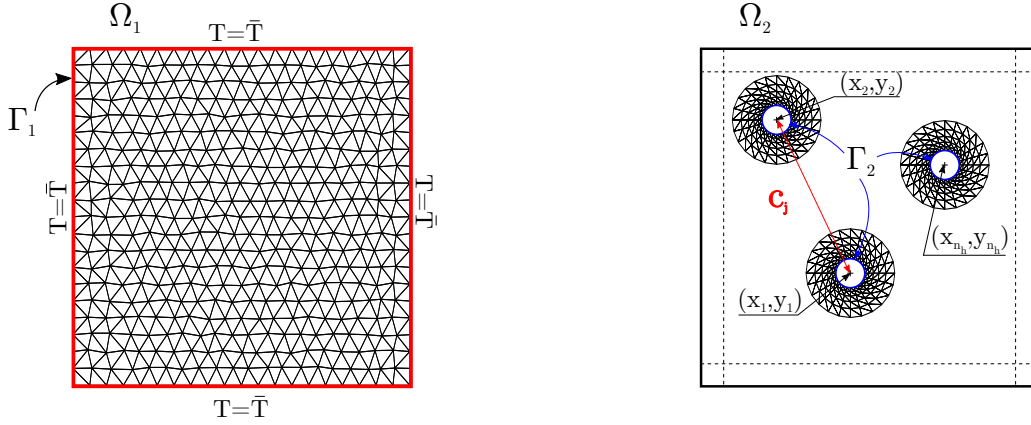
$$\begin{aligned} k\Delta T &= 0 && \text{in } \Omega_i, \quad i = 1, 2, \\ T &= \bar{T} && \text{at } \Gamma_1 \subseteq \partial\Omega_1, \\ k(\nabla T / \nabla \vec{n}) &= h(T - T_{bulk}) && \text{at } \Gamma_2 \subseteq \partial\Omega_2 \end{aligned} \quad (\text{B.12})$$

Therefore, there would be a total of  $m = ((n_h - 1) \times n_h) / 2$  inequality constraints, where  $n_h$  is the total number of holes in the domain.

$$\begin{aligned} \min_{\vec{\mathbf{x}}} \quad & f(\vec{\mathbf{x}}) = T_{avg} \\ \text{subject to} \quad & g_j(\vec{\mathbf{x}}) = c_{min} - c_j \leq 0 && \text{for } j = 1, \dots, m \\ & x_{i_L} \leq x_i \leq x_{i_U}, && \text{for } i = 1, \dots, n \end{aligned} \quad (\text{B.13})$$

From Eq. (B.13),  $x_{i_L}$  and  $x_{i_U}$  represent the value range of the variables. In this example, values of 0.1 and 0.9 are adopted for  $x_{i_L}$  and  $x_{i_U}$ , respectively, i.e., the hole centers can only move inside a square of  $[0.1, 0.9] \times [0.1, 0.9]$ . This boundary limit is set to keep the  $\Omega_2$  domain always inside the background mesh. In this example, a total of 16 design variables are condensed into a single vector, as shown in Eq. (B.14), and are used to represent the eight hole positions. Additionally, a total of 28 inequality constraints are considered, bounding the minimum distance between each pair of holes. The initial condition of the variable values used by the optimizer is a completely random setup.

$$\vec{\mathbf{x}} = \left[ \underbrace{x_1, y_1}_{\text{Hole 1}}, \underbrace{x_2, y_2}_{\text{Hole 2}}, \dots, \underbrace{x_{n_h}, y_{n_h}}_{\text{Hole } n_h} \right] \quad (\text{B.14})$$



**Figure B.8:** Conceptual representation of the background and patch meshes for the optimization procedure of the plate.

### 3.5.2. Optimizer parameters

The PSO algorithm has several parameters (some of them previously depicted in Section 3.2.1) that can determine the performance and efficacy of the optimization process for a given problem. Several works in the literature have studied the influence on the convergence property of the different variables, including the selection of  $\omega$ ,  $c_1$  and  $c_2$ ; velocity clamping  $v_{\max}$ ; and topology of the neighborhood, among others (Jiang et al., 2007; Shi y Eberhart, 1999; Xueyan y Zheng, 2015). For example, Xueyan y Zheng (2015) have analyzed the performance of PSO using different values of swarm size  $N_{\text{sw}}$  and  $\omega$ . After several simulations, they concluded that for values of  $N_{\text{sw}}$  less than 40, the performance of PSO improves with increasing  $N_{\text{sw}}$ ; however, no further improvement is achieved for greater values. Regarding  $\omega$ , values greater than 0.5 appear to be the best option. Furthermore,  $\omega$  can vary as the optimization progresses from an initial inertia weight  $\omega_1$  to a final inertia weight  $\omega_2$  at the end of the search. This approach is used to improve the convergence rate (Sedlaczek y Eberhard, 2007), and it is followed in this work.

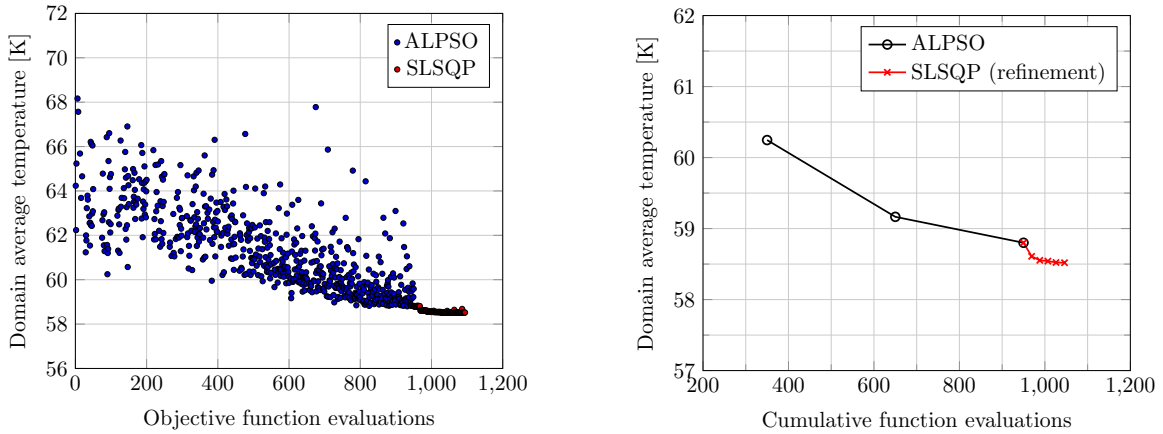
Additionally, Jiang et al. (2007) analyzed the PSO behavior based on different configurations of the tuple  $\omega$ ,  $c_1$ ,  $c_2$  and determined parameter margins to guarantee convergence. Based on the latter and numerical experiments performed by the authors of this work, the ALPSO parameters depicted in Table B.1 proved to be the most suitable for the present communication. Regarding the SLSQP optimizer, where a gradient-based algorithm is used, very few parameters are needed. The appropriate values used in this work are also depicted in Table B.1.

Optimizer	Parameter	Description	Value
ALPSO	$N_{\text{SW}}$	Number of particles	40
	$N_{\text{HD}}$	Number of neighbors of each particle	40
	HoodModel	Neighborhood model	'gbest'
	$v_{\text{init}}$	Initial velocity of particles	1.0
	$v_{\text{max}}$	Maximum velocity of particles	2.0
	$c_1$	Cognitive parameter	2.0
	$c_2$	Social parameter	1.0
	$\omega_1$	Initial inertia weight	0.99
	$\omega_2$	Final inertia weight	0.55
SLSQP	ACC	Convergence accuracy	$10^{-6}$
	MAXIT	Maximum Iterations	50
	sens type	Gradient type computation	'FD'
	sens mode	Flag for parallel gradient calculation	'pgc'
	sens step	Sensitivity step size	$10^{-3}$

**Table B.1:** Optimizer parameters.

### 3.5.3. Results

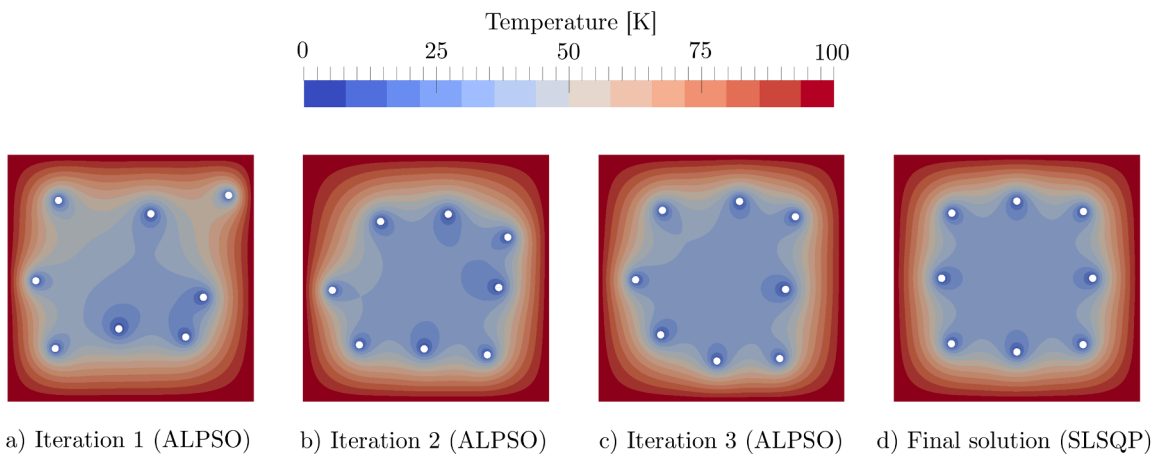
The convergence of the optimization problem is shown in Fig. B.9. The entire optimization process took approximately 50 minutes to converge using 16 cores. Figure B.9a shows the values of the objective function through optimization for both the ALPSO and SLSQP optimizers. The worst arrangement of the refrigeration holes obtained by the ALPSO provides an average value of the domain temperature of 68.16 [K], while for the optimized layout, the average temperature decreases to 58.51 [K]. Figure B.9b shows the values of the objective function in each iteration of both optimizers. Although the domain average temperature decreases quickly between iterations 1 and 2 of ALPSO (from 60.24 [K] to 59.16 [K]), minor improvement is obtained between iterations 2 and 3 (reaching 58.80 [K]). Considering the results shown in both figures, it can be clearly observed that ALPSO converges quickly in the first iterations; however, the convergence speed significantly decreases when reaching the optimum. Therefore, at that stage, it is convenient to switch to a gradient-based optimizer. In Fig. B.9b, when the ALPSO solution is transferred to the SLSQP algorithm, the convergence towards the optimum value accelerates, reaching the optimum value of 58.51 [K]. Although the average temperature of the domain does not vary significantly between ALPSO iteration



**Figure B.9:** Convergence of the objective function obtained by the ALPSO and SLSQP optimization solvers for the numerical example.

3 and the SLSQP final solution, as shown in Fig. B.10c and Fig. B.10d, the hole layout does. As expected, the optimized configuration of the coolant holes that minimizes the domain mean temperature presents three particularities:

1. The distances between a given refrigeration hole and its closest neighbors remain practically constant.
2. The arrangement of the holes is symmetric with respect to the lines  $x = 0.5$  and  $y = 0.5$  and the two diagonals ( $y = 1 - x$  and  $y = x$ ) of the squared domain.
3. The holes are placed near the boundary, where the maximum temperatures occur.



**Figure B.10:** Positions of holes through the optimization procedure. (a), (b) and (c) correspond to the three first iterations of ALPSO, and (d) corresponds to the final solution obtained by the SLSQP optimizer.

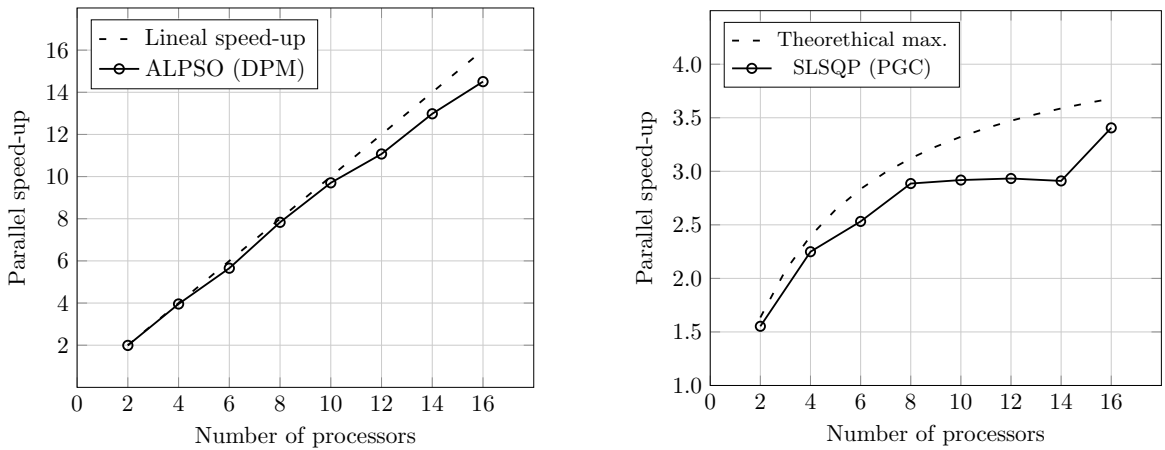


### 3.5.4. Speed-up

Both solvers were run in parallel, and a scalability study was performed. According to Amdahl's law, the ideal speed-up of a parallel computation is defined as follows:

$$S(n_c) = \frac{1}{(1-p) + \frac{p}{n_c}} \quad (\text{B.15})$$

where  $S(n_c)$  is the theoretical speed-up,  $p$  is the fraction of the algorithm that can be parallelized, and  $n_c$  is the total number of cores. In the case of the SLSQP optimizer, the portion of the code that can be parallelized corresponds to the gradient calculation, which is, in this example, approximately 77% of the total computation time. That fraction is dependent on the number of design variables, i.e., the greater the number of design variables that the optimization problem has, the greater the fraction of the code that could be parallelized. Due to the static processor allocation implemented in pyOpt for the parallel gradient computation, it can be observed from Fig. B.11b that when the amount of design variables is an integer multiple of the number of processors, the scalability is close to the theoretical level. When that condition is not met, an increase in the number of processors shows little improvement. On the other hand, from Fig. B.11a, the dynamic process management (DPM) implemented for the ALPSO algorithm shows an almost linear acceleration. These results are in accordance with Schutte et al. (2004).



**Figure B.11:** Speed-up of ALPSO optimizer with dynamic process management (DPM) (left). Speed-up of SLSQP optimizer using parallel gradient calculation (PGC) (right).

## 4. Vane optimization

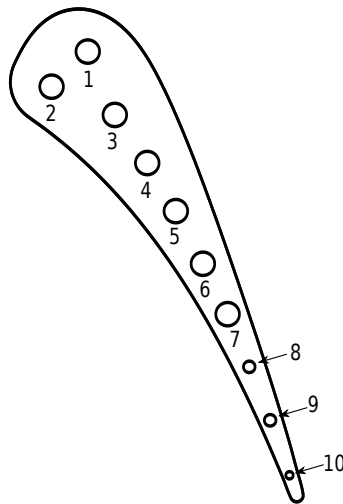
Improvements in modern gas turbine performance are greatly limited by the maximum temperature of the combustion gases. Such limitations arise from the thermomechanical resis-

tance of turbine vanes. To reduce the working temperature of the vane, several coolant tubes are located inside the profile. Therefore, thermomechanical resistance improvement implies that the temperature and stress distribution should be optimized.

In 1983, [Hylton et al. \(1983\)](#) of NASA laboratories worked on an experiment of a cascade of three internal cooled turbine blades subject to several load conditions. Extensive data can be acquired from the results of each test. The airfoil surface was instrumented with approximately eighty thermocouples and thirty static pressure taps located in a near midspan plane. Additionally, each cooling tube was instrumented with thermocouples and static pressure taps at its inlet and exit. The mass flow rate  $\dot{m}$  values for each tube were measured using a calibrated orifice meter.

#### 4.1. Profile geometry and load conditions

Two distinct vane geometries were analyzed during the NASA experiments. In this work, the C3X vane is chosen for the optimization process. The mass flow rate and temperature of the coolant are in accordance with *RUN 157*, as well as the temperature measured by thermocouples placed over the vane surface. The vane geometry and coolant flow conditions of tubes are depicted in [Fig. B.12](#) and [Table B.2](#), respectively. To obtain an accurate tempe-



**Figure B.12:** C3X Vane geometry. Configuration used as a reference.

perature distribution within the vane, a full three-dimensional conjugated heat transfer (CHT) analysis would be desirable to perform a comprehensive study of the involved physics. Of course, such an analysis would be computationally unfeasible since a full CHT simulation is needed for each objective function evaluation. However, works have shown ([von Wolfersdorf](#)

Properties	Passage									
	One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Ten
$\varnothing$ [mm]	6.3	6.3	6.3	6.3	6.3	6.3	6.3	3.1	3.1	1.98
$T_{bulk}$ [K]	342	344	335	336	330	355	336	350	377	387
$\dot{m}$ [kg s <sup>-1</sup> ]	0.0222	0.0221	0.0218	0.0228	0.0225	0.0225	0.0216	0.00744	0.00477	0.00256

**Table B.2:** Coolant and passage characteristics.

et al., 1997; Wang et al., 2015; Xie et al., 2016) that reliable results for the optimization task can be achieved using a simplified model assuming fixed temperatures on the blade surface and for the coolant fluid bulk, and this approach is followed in this work. The coolant flow is assumed to have a steady behavior, i.e., its thermal physical properties do not vary in the FEM simulation. Additionally, the coolant mass flow rate and temperature are considered to be constant and uniform on each passage. The convective heat transfer coefficient between the bulk of the coolant fluid and the vane surface for each passage is computed as follows:

$$h = \frac{k_f \text{Nu}}{d} \quad (\text{B.16})$$

where  $k_f$  is the thermal conductivity of the coolant,  $d$  is the hydraulic diameter of the passage, and Nu stands for the Nusselt number and is defined experimentally according to the Dittus-Boelter correlation for turbulent flows, which states:

$$\text{Nu} = 0.023 \text{Re}^{0.8} \text{Pr}^{0.4} \quad (\text{B.17})$$

where Pr and Re stand for Prandtl and Reynolds numbers, respectively, where Re is defined as:

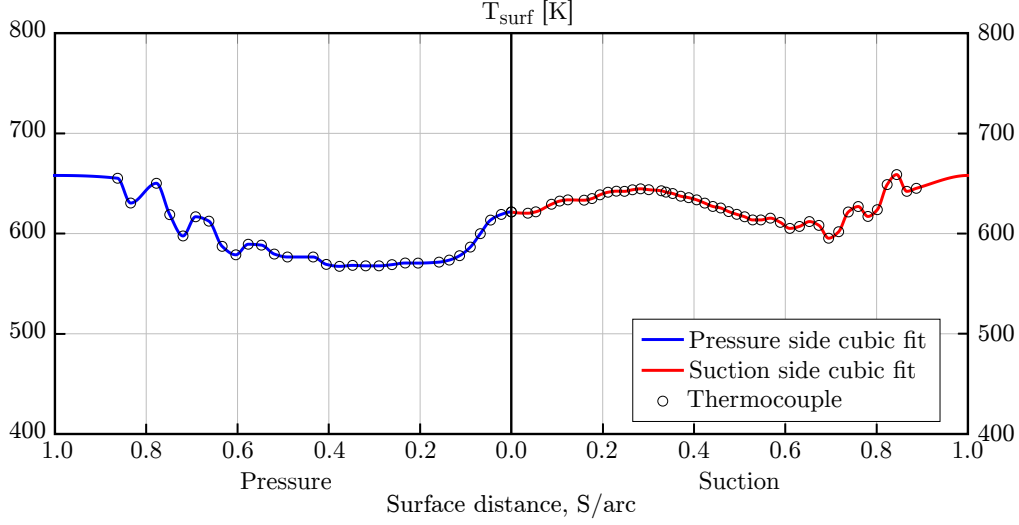
$$\text{Re} = \frac{\rho v d}{\mu} \quad (\text{B.18})$$

where  $\rho$  and  $\mu$  are the density and viscosity of the coolant, respectively, and  $v$  is the speed of the fluid. The Prandtl number is defined as:

$$\text{Pr} = \frac{C_p \mu}{k_f} \quad (\text{B.19})$$

where  $C_p$  is the specific heat capacity of the fluid at constant pressure. The Dittus-Boelter correlation is valid when the following three conditions for the fluid are met: (i)  $0.6 < \text{Pr} < 160$ , (ii)  $\text{Re} > 10000$  and (iii)  $L/D > 10$ , where L and D are the length and diameter of the passages, respectively. All these conditions are fulfilled by the ten holes. From Eqs. B.16 to B.19, one can conclude that for constant values of thermal conductivity, density, viscosity, velocity and specific heat capacity, the convective heat transfer coefficient of each hole is only a function of its diameter.

Furthermore, a fixed temperature is established over the vane surface according to the experimental values measured by the thermocouples (see Fig. B.13). A cubic interpolation is performed to load all the surface nodes. The vane is composed of ASTM 310 stainless steel,



**Figure B.13:** Temperature measured by thermocouples and cubic interpolation over the pressure and suction surface for the FEM analysis.

and its thermal conductivity is approximated using the following linear function (Harvey, 1982):

$$k_s(T) = 0.0115 T + 9.9105 \quad [\text{W m}^{-1} \text{K}^{-1}] \quad (\text{B.20})$$

Viscosity  $\mu$  and thermal conductivity  $k_f$  of the coolant are defined by the Sutherland formula, which states:

$$\mu(T) = \mu_0 \left( \frac{T}{T_0} \right)^{3/2} \frac{T_0 + S}{T + S} \quad [\text{Pa s}] \quad (\text{B.21})$$

$$k_f(T) = k_0 \left( \frac{T}{T_0} \right)^{3/2} \frac{T_0 + S}{T + S} \quad [\text{W m}^{-1} \text{K}^{-1}] \quad (\text{B.22})$$

where for the air, the values of the Sutherland's law coefficients are  $\mu_0 = 1.7894 \times 10^{-5}$  [Pa s],  $T_0 = 273.11$  [K],  $k_0 = 0.0261$  [W m<sup>-1</sup> K<sup>-1</sup>] and  $S = 110.56$ .

## 4.2. Mathematical model of the optimization problem

The mathematical model of the blade cooling optimization problem can be defined as:

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & T_{avg}(\bar{\mathbf{x}}) \\ \text{subject to} \quad & g_j(\bar{\mathbf{x}}) = c_{min} - c_j \leq 0, & \text{for } j = 1, \dots, m_e, \\ & \nabla T(\bar{\mathbf{x}}) \leq \nabla T_{max}, & \text{for } j = m_e + 1, \dots, m, \\ & x_{i_L} \leq x_i \leq x_{i_U}, & \text{for } i = 1, \dots, n \end{aligned} \quad (\text{B.23})$$

The objective function to minimize is the average temperature of the vane, subject to the restriction that maximum thermal gradients cannot be greater than a specified value ( $\nabla T_{max}$ ). As thermomechanical stresses depend directly on thermal gradients, to ensure safe values of thermomechanical stresses on the vane, thermal gradients are constraints on the optimization problem. Additionally, the minimum distance between each pair of hole centers and the vane surface is restricted. The design variables are condensed in the vector  $\bar{\mathbf{x}}$ , defined as:

$$\bar{\mathbf{x}} = \left[ \underbrace{r_1, \alpha_1}_{\text{Passage 1}}, \underbrace{r_2, \alpha_2}_{\text{Passage 2}}, \underbrace{x_3, y_3}_{\text{Passage 3}}, \underbrace{x_4, y_4}_{\text{Passage 4}}, \dots, \underbrace{x_{n_h}, y_{n_h}}_{\text{Passage } n_h} \right], \quad (\text{B.24})$$

where for passages 1 and 2, the variables  $r_i$  and  $\alpha_i$  correspond to their center position in polar coordinates for the transformed domain (see Fig. B.14). For the remaining passages, the variables  $x_i$  and  $y_i$  correspond to the Cartesian coordinates of the hole center  $i$  (for  $i = 3, 4, \dots, 10$ ).  $x_{i_U}$  and  $x_{i_L}$  are the upper and lower bounds of the variables, which will depend on the coordinate transformation, as explained in the following section.

### 4.3. Transformation of design variable domains

To constrain hole movement during the optimization process, a similar coordinate transformation used in (Wang et al., 2015; Xie et al., 2016) is applied in this work for each hole. Each passage with its respective mesh is allowed to move inside a certain region (see Fig. B.14) to remain between the background mesh boundaries and do not overlay above other coolant holes. For holes one and two, a polar transformation is performed as stated in Eq. (B.25).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \end{bmatrix} + \begin{bmatrix} r \cos \alpha \\ r \sin \alpha \end{bmatrix} \quad (\text{B.25})$$

For holes from three to ten, the following transformation is performed:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (\text{B.26})$$

The vector  $\{x_o, y_o\}$  represents the coordinates for the displaced domain, and  $\theta$  represents the angle of its rotation.  $\{x', y'\}$  and  $\{r, \alpha\}$  stand for the variables in the local coordinate system. The coordinate transformation for each hole center is depicted in Fig. B.14. Additionally, Fig. B.14 shows that the regions are defined such that the outer boundaries of the hole meshes keep a minimum distance from the vane boundary of 1 [mm].

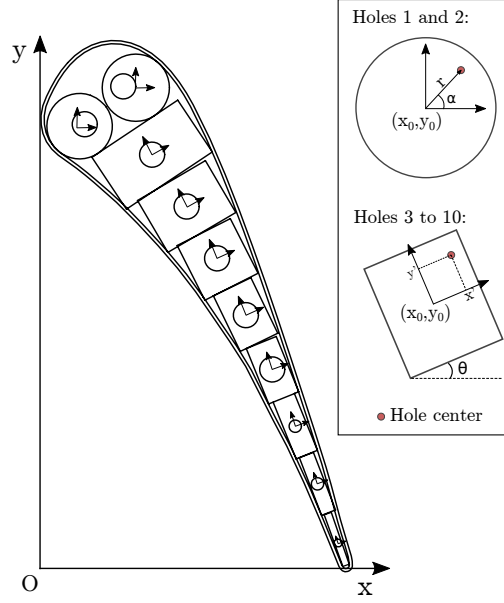


Figure B.14: Transformation of hole coordinate system.

#### 4.4. Grid independence analysis of background and patch meshes

Once the thermal FEM model was set, a mesh convergence study was conducted. A constant temperature of 600 [K] has settled over the vane surface. Several two-dimensional triangular meshes with an increasing number of cells have been tested, varying from approximately 5600 to 77000 elements. The background mesh of the vane is generated using the open-source platform *SALOME* (Ribes y Caremoli, 2007.), while for the patch mesh, an ad hoc script was used. The total number of cells of each overset grid is obtained by summing background and patch mesh elements. In each case, the patch grid is formed by approximately one-fifth of the number of cells of the background mesh. To compare the mesh performance, the parameter  $T_{DIFF}$  was defined to measure the percentage difference of the average temperature between two consecutive meshes. Then, for a given mesh  $i$ , the temperature difference states:

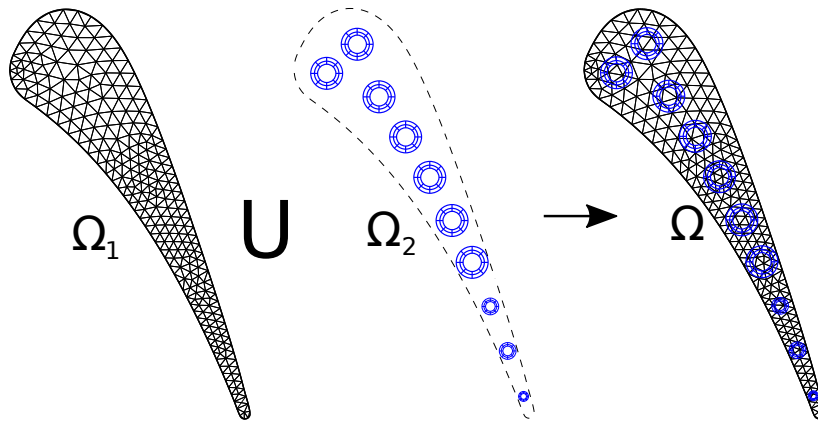
$$T_{DIFF}^{(i)} = \left( \frac{|T_{avg}^{(i)} - T_{avg}^{(i-1)}|}{\Delta T_{mesh}} \right) \times 100 \quad (\text{B.27})$$

where  $T_{avg}^{(i)}$  and  $T_{avg}^{(i-1)}$  are the values of the average temperature obtained by the meshes  $i$  and the previous mesh ( $i - 1$ ), respectively. The parameter  $\Delta T_{mesh}$  represents the average temperature difference between the coarser and finer meshes, i.e.,  $\Delta T_{mesh} = T_{avg}^{(1)} - T_{avg}^{(6)} = 13.72$  [K] (see Table B.3), and is used to properly weight the discrepancy between the results of two meshes. Large differences in the results can be observed in Table B.3 for the first four grids. Smaller differences were obtained between meshes 4 and 5 (2.62%); however, mesh 5 was finally chosen prioritizing the smoothness of the objective function. The error between

Mesh	Elements	$T_{\text{avg}}$ [K]	$T_{\text{DIFF}}$ [%]
1	5632	564.70	-
2	9719	557.14	55.10
3	19109	553.52	26.38
4	26387	551.42	15.30
5	61758	551.06	2.62
6	76620	550.98	0.58

**Table B.3:** Grid independence study.

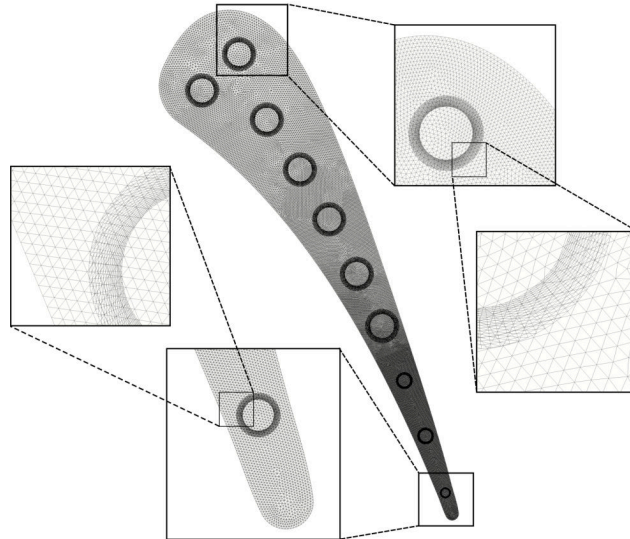
meshes 5 and 6 is less than 1% (0.58%), showing that finer meshes will unnecessarily increase the computational cost of each simulation. Hence, the grid composed of 61,758 cells has been used. The background mesh is formed by 48,960 linear unstructured triangular elements, while the patch mesh is composed of a total of 12,800 linear structured triangular elements (1,280 elements around each passage). A conceptual representation of the meshes used by the Chimera scheme is shown in Fig. B.15, and in Fig. B.16, details of the actual mesh used for the optimization are presented.



**Figure B.15:** Conceptual representation of the background and patch meshes used by the Chimera scheme.

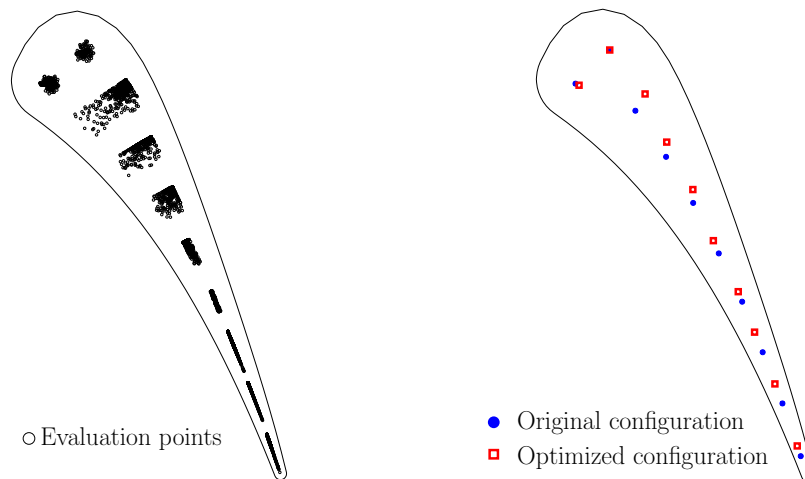
#### 4.5. Results

Almost a thousand function evaluations were computed by the optimizers, covering almost the entire computational domain (see Fig. B.17a) with a total computing time of approximately two hours using 20 cores. The convergence of the blade optimization problem is shown in Figs. B.17 and B.18. For the first evaluations of ALPSO, the blade average temperature takes high values reaching approximately 566 [K], while as it converges to a solution, it



**Figure B.16:** Actual overlapping meshes used for the optimization procedure. The background mesh is formed by 48,960 linear unstructured triangular elements, while the patch mesh is composed of a total of 12,800 linear structured triangular elements (1,280 elements around each passage).

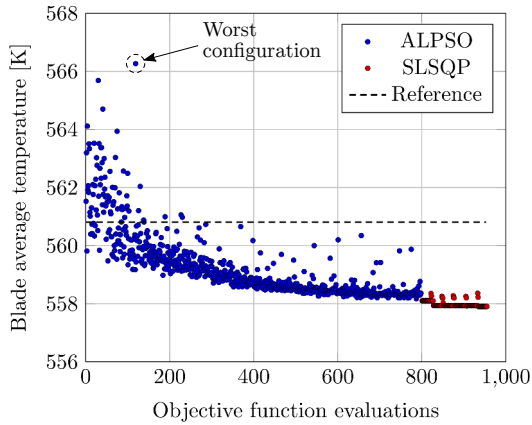
reaches a minimum value of 558.17 [K], outperforming the reference solution (560.81 [K]). Then, the ALPSO solution is passed to the SLSQP optimizer, and a final value of 557.90 [K] is achieved after a few iterations. As shown in Fig. B.19, for the optimized cooling of the



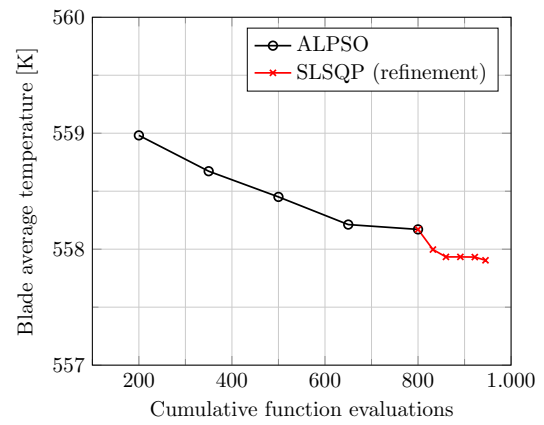
**Figure B.17:** Positions of the passage centers through the optimization process.

vane, passages 6 to 10 are placed slightly toward the leading edge through the camber line with respect to its original locations. The optimized positions of passages 1 and 2 practically coincide with the original positions, meaning that their positions were already optimized and no further improvement on the blade refrigeration can be achieved by changing their locations. For passages 3 to 5, their best locations appear to be displaced slightly towards the suction surface, which is where the maximum temperatures occur. The worst scenario





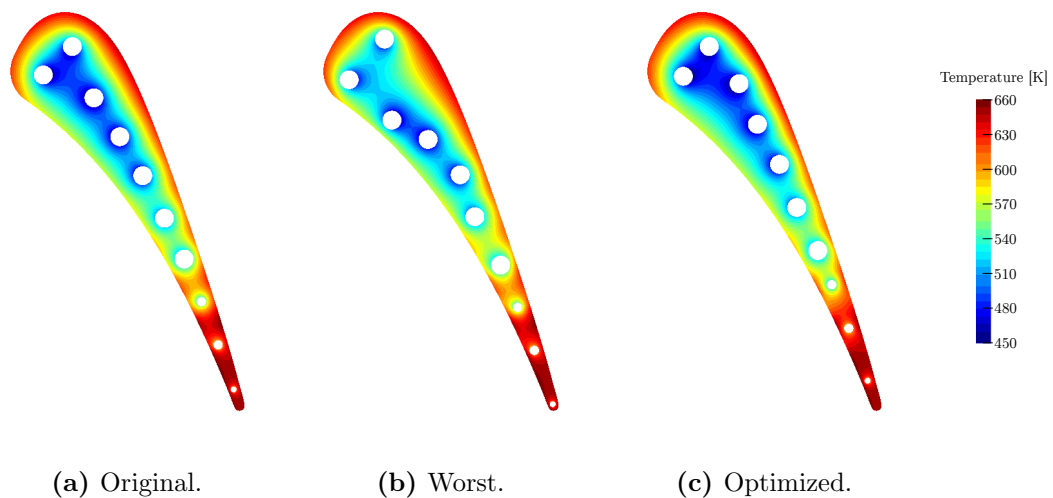
(a) Objective function values.



(b) Iterations of the optimizers.

**Figure B.18:** Convergence of the objective function obtained by the ALPSO and SLSQP optimization solvers for the numerical example.

is presented to obtain a better understanding of the thermal phenomena. As shown in Fig. B.19b, a poor refrigeration condition coincides with a large separation between passages 1 and 2, heading towards the suction side and pressure side, respectively, and passages 3 and 4 moved towards the pressure side where the minimum temperatures occur. Furthermore, in Fig. B.20, the temperature gradients for the original, worst and optimized configurations are shown. The maximum values of the thermal gradients of the optimized configuration (Fig. B.20c) remain lower than those obtained by the original arrangement of the passages; thus, the thermomechanical strength of the blade would not be compromised by the new optimized configuration.

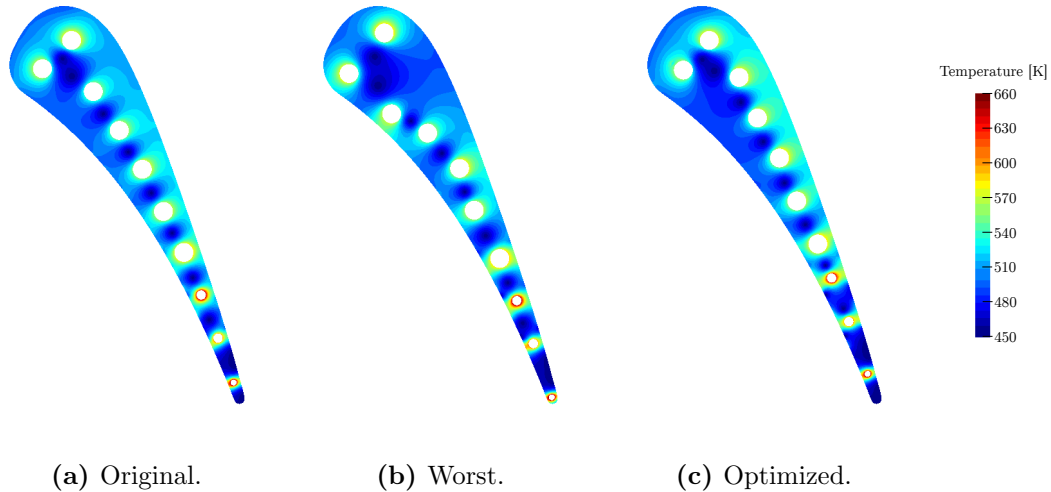


(a) Original.

(b) Worst.

(c) Optimized.

**Figure B.19:** Comparison of the temperature field obtained by the FEM results.



**Figure B.20:** Comparison of the temperature gradient field obtained by the FEM results.

## 5. Conclusions

In this work, a novel approach using a Chimera scheme in the finite element context for overlapped grids was presented to address the optimization of the internal cooling configuration of a gas turbine vane. It was shown that the scheme allowed the generation of the vane mesh and the passage meshes independently, thereby reducing meshing efforts while maintaining the quality of the meshes in the entire computational domain.

To assess the coupled scheme, the configuration of several coolant holes inside a two-dimensional heated plate was optimized. The domain average temperature was minimized, and the final optimized passage locations were as expected: (i) evenly spaced; (ii) symmetric with respect to the vertical, horizontal and diagonal axes; and (iii) directed towards the plate boundary, where the maximum temperatures occur.

Subsequently, the locations of the ten cooling passages of the NASA C3X vane were optimized, minimizing the average temperature of the blade subject to thermal gradient restrictions. Each patch mesh surrounding the holes could freely move within a region, searching for the best configuration. Consequently, the optimized solution outperformed the reference arrangement of the holes.

The scheme showed a good convergence rate when coupled with population-based and gradient-based optimization solvers. Although ALPSO converges quickly to the optimized solution region, the convergence drastically reduces when reaching the optimized value. At this point, the solution refinement with the SLSQP was convenient, reaching an optimized solution in few iterations and the proper passage distribution.

Regarding the scalability of the parallel computation, the dynamic process management implemented for the ALPSO algorithm showed good performance, resulting in an almost linear acceleration. For the SLSQP optimizer, the static processor allocation for the parallel gradient calculation showed a close-to-theoretical scalability only when the number of design variables is an integer multiple of the number of processors. When that condition was not met, an increase in processors shows little or negligible improvement.

The proposed methodology proved to be a robust and efficient tool to optimize and enhance the thermal performance within complex engineering parts with intricate geometries without much meshing effort. As future work, the algorithm will be extended to address three-dimensional optimization problems with meshes of a great quantity of elements, taking full advantage of the matrix-free and parallel capabilities of the proposed methodology. Although the algebraic coupling would be straightforward, considerable effort must be made to correctly detect the interpolation boundary on the background mesh in an efficient way.

## Acknowledgment

The authors are grateful for the support provided by the following research projects: Argentinian Council for Scientific Research (grant PIP-588-2015); Argentinian National Agency for Technological and Scientific Promotion (grants PICT-2015-2904 and PICT-2014-2660); Santa Fe Science Technology and Innovation Agency (grant ASACTEI-010-18-2014); National University of the Litoral (grant CAI+D-504-201501-00112-LI). This work was partially performed with resources from *Free Software Foundation/GNU-Project*, GNU/Linux OS, GCC/G++ compilers, GNU/Octave, as well as other *Open Source* tools, such as VTK, ParaView, PETSc, and L<sup>A</sup>T<sub>E</sub>X.



# References appendix B

- Abolhasani M., Karami A., y Rahimi M. Numerical modeling and optimization of the enhancement of the cooling rate in concentric tubes under ultrasound field. *Numerical Heat Transfer, Part A: Applications*, 67(11):1282–1309, 2015.
- Andrew M. W., Aaron J. K., William M. C., y Robert L. M. Validation of the Strand Grid Approach. *AIAA Computational Fluid Dynamics*, 2009.
- Bazaraa M.S., Sherali H.D., y Shetty C.M. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.
- Behr M. y Tezduyar T. The shear-slip mesh update method. *Computer Methods in Applied Mechanics and Engineering*, 174(3-4):261–274, 1999.
- Belgacem F.B. y Maday Y. The mortar element method for three dimensional finite elements. *ESAIM: Mathematical Modelling and Numerical Analysis*, 31(2):289–302, 1997.
- Blades E.L. y Marcum D.L. A sliding interface method for unsteady unstructured flow simulations. *International Journal for Numerical Methods in Fluids*, 53(3):507–529, 2007.
- Bunker R.S. Axial turbine blade tips: function, design, and durability. *Journal of Propulsion and Power*, 22(2):271–285, 2006.
- Chesshire G. y Henshaw W.D. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64, 1990.
- Dalcin L.D., Paz R.R., Kler P.A., y Cosimo A. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- Dang T.S. y Meschke G. A Shear-Slip Mesh Update–Immersed Boundary Finite Element model for computational simulations of material transport in EPB tunnel boring machines. *Finite Elements in Analysis and Design*, 142:1–16, 2018.
- Demkowicz L. *Computing with hp-adaptive finite elements: Volume 1 one and two dimensional elliptic and Maxwell problems*. Chapman and Hall/CRC, 2006.
- Dennis B.H., Egorov I.N., Dulikravich G.S., y Yoshimura S. Optimization of a large number of coolant passages located close to the surface of a turbine blade. En *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference*, páginas 13–19. American Society of Mechanical Engineers, 2003.
- Duboeuf F. y Béchet E. Embedded solids of any dimension in the X-FEM. Part II–Imposing Dirichlet boundary conditions. *Finite Elements in Analysis and Design*, 128:32–50, 2017.

- Eberhart R. y Kennedy J. A new optimizer using particle swarm theory. En *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, páginas 39–43. IEEE, 1995.
- Eguzkitza B., Houzeaux G., Aubry R., Owen H., y Vázquez M. A parallel coupling strategy for the chimera and domain decomposition methods in computational mechanics. *Computers & Fluids*, 80:128–141, 2013.
- Eguzkitza Bazar A.B. *HERMESH: a geometrical domain composition method in computational mechanics*. Tesis de Doctorado, Universitat Politècnica de Catalunya, 2014.
- Fletcher R. *Practical methods of optimization*. John Wiley & Sons, 2013.
- Franco Brezzi Jacques-Louis Lions O.P. Analysis of a Chimera method. *Numerical Analysis*, 1:655–660, 1997.
- Harvey P.D. *Engineering properties of steel*. American Society for Metals Metals Park, OH, 1982.
- Hillier F.S. *Introduction to operations research*. Tata McGraw-Hill Education, 2012.
- Houzeaux G., Borrell R., Cajas J.C., y Vázquez M. Extension of the parallel Sparse Matrix Vector Product (SpMV) for the implicit coupling of PDEs on non-matching meshes. *Computers & Fluids*, 173:216–225, 2018.
- Houzeaux G., Cajas J., Discacciati M., Eguzkitza B., Gargallo-Peiró A., Rivero M., y Vázquez M. Domain decomposition methods for domain composition purpose: chimera, overset, gluing and sliding mesh methods. *Archives of Computational Methods in Engineering*, 24(4):1033–1070, 2017.
- Houzeaux G., Cajas J., Eguzkitza B., y Vázquez M. Techniques for parallel, distributed and cloud computing in engineering, volume 36 of computational science, engineering and technology series, chapter chapter 4: Parallel implementation of domain composition methods. *Saxe-Coburg Publications*, páginas 61–92, 2015.
- Houzeaux G. y Codina R. A Dirichlet/Robin Iteration-by-Subdomain Domain Decomposition Method Applied to Advection-Diffusion Problems for Overlapping Subdomains. *Domain Decomposition Methods in Science and Engineering*, páginas 435–442, 2003.
- Hylton L., Mihelc M., Turner E., Nealy D., y York R. Analytical and experimental evaluation of the heat transfer distribution over the surfaces of turbine vanes. *NASA CR 168015*, 1983.
- Jette M.A., Yoo A.B., y Grondona M. Slurm: Simple linux utility for resource management. En *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*, páginas 44–60. Springer-Verlag, 2002.
- Jiang M., Luo Y.P., y Yang S.Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, 102(1):8–16, 2007.
- Kao K.H., Liou M.S., y Chow C.Y. Grid adaptation using chimera composite overlapping meshes. *AIAA journal*, 32(5):942–949, 1994.
- Kelley C.T. *Iterative methods for linear and nonlinear equations*, volumen 16. SIAM, 1995.
- Kraft D. y Schnepfer K. SLSQP—A Nonlinear Programming Method with Quadratic Programming Subproblems. *DLR, Oberpfaffenhofen*, 1989.
- Lawson C.L. y Hanson R.J. *Solving least squares problems*, volumen 15. SIAM, 1995.
- Martins J.R., Sturdza P., y Alonso J.J. The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)*, 29(3):245–262, 2003.
- Meakin R. An efficient means of adaptive refinement within systems of overset grids. En *12th Computational Fluid Dynamics Conference*, página 1722. 1995.

- Nielsen E.J., Diskin B., y Yamaleev N.K. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA journal*, 48(6):1195–1206, 2010.
- Nocedal J. y Wright S. *Numerical optimization*. Springer Science & Business Media, 2006.
- Nowak G. y Nowak I. Shape design of internal cooling passages within a turbine blade. *Engineering Optimization*, 44(4):449–466, 2012.
- Nowak G. y Wróblewski W. Optimization of blade cooling system with use of conjugate heat transfer approach. *International Journal of Thermal Sciences*, 50(9):1770–1781, 2011.
- Parwani A.K., Talukdar P., y Subbarao P. A hybrid approach using CGM and DE algorithm for estimation of boundary heat flux in a parallel plate channel. *Numerical Heat Transfer, Part A: Applications*, 65(5):461–481, 2014.
- Perez R.E., Jansen P.W., y Martins J.R. pyopt: a python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, 45(1):101–118, 2012.
- Prewitt N.C., Belk D.M., y Shyy W. Parallel computing of overset grids for aerodynamic problems with moving objects. *Progress in Aerospace Sciences*, 36(2):117–172, 2000.
- Ribes A. y Caremoli C. Salome platform component model for numerical simulation. En *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International (http://www.salome-platform.org/)*, volumen 2, páginas 553–564. IEEE, 2007.
- Ryoo H.S. y Sahinidis N.V. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers & Chemical Engineering*, 19(5):551–566, 1995.
- Schmid F., Hirschen K., Meynen S., y Schäfer M. An enhanced approach for shape optimization using an adaptive algorithm. *Finite elements in Analysis and Design*, 41(5):521–543, 2005.
- Schutte J.F., Reinbolt J.A., Fregly B.J., Haftka R.T., y George A.D. Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering*, 61(13):2296–2315, 2004.
- Sedlaczek K. y Eberhard P. Using augmented lagrangian particle swarm optimization for constrained problems in engineering” *¿ using augmented lagrangian particle swarm optimization for constrained problems in engineering*. *Structural and Multidisciplinary Optimization*, 32(4):277–286, 2006.
- Sedlaczek K. y Eberhard P. Augmented lagrangian particle swarm optimization in mechanism design. *Journal of System Design and Dynamics*, 1(3):410–421, 2007.
- Seshat. <http://www.cimec.org.ar/c3/seshat/>, 2018.
- Shi Y. y Eberhart R.C. Empirical study of particle swarm optimization. En *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volumen 3, páginas 1945–1950. IEEE, 1999.
- Siavashi M., Garusi H., y Derakhshan S. Numerical simulation and optimization of steam-assisted gravity drainage with temperature, rate, and well distance control using an efficient hybrid optimization technique. *Numerical Heat Transfer, Part A: Applications*, 72(9):721–744, 2017.
- Sonzogni V.E., Yommi A.M., Nigro N.M., y Storti M.A. A parallel finite element program on a Beowulf cluster. *Advances in Engineering Software*, 33(7-10):427–443, 2002.
- Steger J. The chimera method of flow simulation. En *Workshop on applied CFD, Univ of Tennessee Space Institute*, volumen 188. 1991.

- Steger J.L. A chimera grid scheme: advances in grid generation. *American Society of Mechanical Engineers Fluids Engineering Division*, 5:55–70, 1983.
- Steger J.L. y Benek J.A. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(1-3):301–320, 1987.
- Storti M., Nigro N., Paz R., Dalcin L., y Lopez E. PETSc-FEM, A General Purpose, Parallel, Multi-Physics FEM Program. *International Center of Computational Method in Engineering (CIMEC), Argentina* <http://www.cimec.org.ar/petscfem>, 2007.
- Vázquez M., Houzeaux G., y Codina R. Chimera type domain decomposition methods applied to fractional step finite element schemes for incompressible flows. En *Proceedings of the ECCOMAS 2000 Computational Fluid Dynamics Conference*. 2000.
- von Wolfersdorf J., Achermann E., y Weigand B. Shape optimization of cooling channels using genetic algorithms. *Journal of Heat Transfer*, 119(2):380–388, 1997.
- Wang B., Zhang W., Xie G., Xu Y., y Xiao M. Multiconfiguration shape optimization of internal cooling systems of a turbine guide vane based on thermomechanical and conjugate heat transfer analysis. *Journal of Heat Transfer*, 137(6):061004, 2015.
- Xie G., Song Y., y Sundén B. Computational optimization of the internal cooling passages of a guide vane by a gradient-based algorithm. *Numerical Heat Transfer, Part A: Applications*, 69(12):1311–1331, 2016.
- Xueyan L. y Zheng X. Swarm size and inertia weight selection of particle swarm optimizer in system identification. En *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, volumen 1, páginas 1554–1556. IEEE, 2015.
- Zheng Y. y Liou M.S. A novel approach of three-dimensional hybrid grid methodology: Part 1. grid generation. *Computer Methods in Applied Mechanics and Engineering*, 192(37-38):4147–4171, 2003.
- Zienkiewicz O. y Taylor R.L. *The finite element method: the basis*. World Books Publishing Corporation, 2005.