

Refinamiento h-adaptativo de mallas no estructuradas en problemas estacionarios y transitorios de flujos compresibles

Autor: Ing. Gustavo A. Ríos Rodríguez

Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC)
Instituto de Desarrollo Tecnológico para la Industria Química (INTEC)

UNL - CONICET

Director: Dr. Mario Alberto Storti

Co-director: Dr. Norberto Marcelo Nigro



27 de mayo de 2009

Resumen

La presente tesis trata sobre el desarrollo, la implementación y aplicación de una estrategia de refinamiento adaptativo que permite resolver problemas estacionarios y transitorios de flujos compresibles mediante el método de elementos finitos sobre mallas no estructuradas. La estrategia de adaptación se concibe atendiendo a la calidad de la malla. Se llevan a cabo experiencias numéricas que permiten determinar un esquema de particionamiento para tetraedros que resulta conveniente desde el punto de vista de la calidad y del costo de refinamiento. La calidad de los elementos de la malla es evaluada utilizando una métrica de calidad algebraica, mientras que la calidad de la malla es analizada con un enfoque estadístico, asumiendo que la calidad de los elementos responde a una distribución normal. Las mallas que resultan del proceso de adaptación conservan el tipo de elemento y son no conformes. Luego, la presencia de nodos irregulares obliga a imponer restricciones en el refinamiento para que no exista una diferencia arbitraria en el tamaño de elementos vecinos pertenecientes a regiones con distinto grado de refinamiento. Estas restricciones permiten que el algoritmo de refinamiento sea formulado recursivamente. Por otro lado, dado que no se conoce la solución exacta del problema, se debe recurrir a técnicas de estimación del error en la solución aproximada para identificar las regiones del dominio computacional que necesitan ser refinadas. En este sentido, se describen e implementan dos de los métodos más usuales para la estimación *a posteriori* del error cuando el sistema de ecuaciones que gobierna el problema es hiperbólico. El código de adaptación de mallas se ejecuta secuencialmente en un nodo cualquiera (no necesariamente el servidor) de un *cluster* Beowulf, mientras que el programa que resuelve las ecuaciones del flujo se ejecuta en paralelo. La solución de las ecuaciones del flujo está basada en la implementación del método de elementos finitos estabilizado SUPG (*Streamline-Upwind / Petrov-Galerkin*). La formulación utiliza además técnicas de *shock-capturing* para el tratamiento de los choques. Dicha implementación es parte del *solver* multifísica PETSc-FEM desarrollado en el CIMEC (Centro Internacional de Métodos Computacionales en Ingeniería). Por otro lado, el algoritmo de adaptación se implementó en dos códigos. El primero de ellos se desarrolló en lenguaje *scripting* Octave y fue aplicado en la adaptación de mallas bidimensionales y tridimensionales en la etapa de desarrollo conceptual. Dadas las limitaciones de Octave en cuanto al manejo de los recursos computacionales, una implementación más eficiente se implementó en lenguaje C++. Para ello se recurrió al uso de los contenedores y algoritmos facilitados por las librerías de plantillas estándar (STL) y Boost. Ambos códigos se aplican en la resolución de problemas de flujos no viscosos y compresibles bidimensionales, axisimétricos y tridimensionales de gases perfectos sobre mallas no estructuradas de elementos finitos. Se evalúan la precisión en el cálculo de la solución, la calidad de las mallas adaptadas y la performance del código adaptativo. Los

resultados numéricos son comparados con soluciones teóricas procedentes de la dinámica de gases y con resultados de ensayos experimentales. Finalmente se elaboran conclusiones sobre la estrategia de adaptación desarrollada y se proponen una serie de actividades a realizar en el futuro, destinadas a mejorar e incorporar nuevas capacidades a dicha estrategia.

Abstract

An adaptive refinement strategy to solve steady and unsteady compressible non-viscous flows by the finite element method over unstructured meshes is developed. The quality of the adapted mesh is the main driving force to devise the refinement strategy, since a small quality decrease is often wanted. Numerical experiences allow to choose a partitioning scheme for tetrahedral meshes suited to the quality requirements. The quality of the mesh elements is evaluated using well known quality measures. The type of element from the original mesh is kept due to the chosen refinement strategy. The adapted meshes are also non-conforming, ie there exists *hanging* or *green* nodes in the adapted mesh. Refinement constraints are enforced in order to guarantee a smooth size distribution among neighbour elements. The refinement strategy and refinement constraints are implemented by a recursive algorithm. On the other hand, *a posteriori* error estimators and indicators are used to steer the adaptivity process since the exact solution is not known in advance. Common error estimators and indicators for hyperbolic systems of partial differential equations are introduced. The adaptivity code is partially parallelized, ie the mesh adaption task is executed on a single node of a Beowulf cluster of PC while the fluid dynamic solver is executed in parallel. The *Streamline-Upwind / Petrov-Galerkin* stabilized finite element scheme is used to solve the governing equations together with *shock-capturing* techniques. This scheme is implemented in a code that is part of the PETSc-FEM multiphysics solver developed at the International Center of Computational Methods in Engineering (CIMEC). The adaption algorithm is implemented in two codes. The former is developed using the Octave scripting language while the C++ programming language is used for the latter. Both are used to adaptively solve compressible flow problems over two- and three-dimensional meshes, although the Octave code is restricted to smaller size meshes because of memory management issues. Algorithms and containers are provided by the Standard Template Library (STL) and the Boost libraries. Mesh adaption code performance, adapted mesh quality and accuracy improvements in the solution are evaluated whenever possible. Conclusions are drawn based on these results and research activities for the near future are finally proposed.

Agradecimientos

Este trabajo ha sido posible gracias al financiamiento del Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET) por medio de una beca doctoral. El apoyo financiero también ha sido provisto a través de los siguientes proyectos: CONICET PIP 5271/05, Universidad Nacional del Litoral (UNL) CAI+D 2005-10-64, Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) PICT 12-14573/2003, PME 209/2003, PICT-1506/2006. Sin ellos, no hubiera sido posible contar con los recursos necesarios para elaborar esta tesis.

Agradezco a los doctores Mario Storti y Norberto Nigro por los consejos recibidos en estos años. También quisiera mencionar al ingeniero Ezequiel López por haber colaborado conmigo en la etapa inicial de desarrollo del trabajo. Al Dr. José Tamagno le agradezco las sugerencias y la orientación realizadas sobre el ejemplo de Taylor-Sedov esférico. Finalmente, agradezco el apoyo incansable y la comprensión brindada durante todos estos años de mi esposa María Alejandra y de mis padres Rosa y José.

En el desarrollo de esta tesis se ha hecho uso de software libre del proyecto GNU y otros, tales como Fedora y Mandriva GNU/Linux~OS, MPICH2, PETSc-FEM, GCC, Octave, OpenDX, Netgen, Boost, STL, L^AT_EX (un procesador de textos WYSIWYM) y otros.

Índice general

1. Introducción	1
2. Calidad de malla	3
2.1. Métricas de calidad	3
2.1.1. Matrices Jacobianas de elementos simples	4
2.1.2. Métrica de Liu y Joe	7
2.2. Experiencias numéricas	8
2.2.1. Primera prueba de calidad	10
2.2.2. Segunda prueba de calidad	10
2.2.3. Conclusiones	14
3. Estimación a posteriori del error	16
3.1. Introducción	16
3.2. Indicadores y estimadores de error	18
3.2.1. Técnicas de recuperación o reconstrucción del gradiente para problemas de flujo compresible	19
4. Ecuaciones de flujo compresible y el método SUPG	24
4.1. Ecuaciones de gobierno para flujos no viscosos, no estacionarios y compresibles	24
4.2. Método de elementos finitos estabilizado SUPG para el sistema de ecuaciones de Euler	27
4.2.1. Formulación estándar de Galerkin	27
4.2.2. Estabilización SUPG	28
4.2.3. Tratamiento numérico de los choques (<i>shock capturing</i>) en el contexto del método SUPG	29
5. Implementación de la estrategia de adaptación	31
5.1. Consideraciones previas	32
5.2. Implementación de las estructuras de datos para las entidades geométricas de la malla	33
5.3. Implementación de las adyacencias	34
5.3.1. Adyacencias hacia abajo	34

5.3.2.	Adyacencias hacia arriba	38
5.4.	Descripción del código	39
6.	Esquemas de refinamiento y estrategia de solución adaptativa para problemas no estacionarios	45
6.1.	Descripción del refinamiento	45
6.2.	Extensión del criterio de nodo 1-irregular a mallas 3-D	49
6.3.	Adaptación de la malla en problemas dependientes del tiempo	51
6.3.1.	Frecuencia de adaptación	54
6.3.2.	Descripción del procedimiento de adaptación no estacionario	55
6.3.3.	Proyección del estado	58
6.4.	Tratamiento de propiedades sobre entidades geométricas	64
7.	Aplicaciones	71
7.1.	Flujo transónico bidimensional estacionario sobre una cuña	71
7.1.1.	Descripción del problema	72
7.1.2.	Solución numérica	73
7.1.3.	Resultados	76
7.2.	Flujo transónico tridimensional estacionario alrededor de un cuerpo cónico	81
7.2.1.	Descripción del problema	81
7.2.2.	Solución numérica	82
7.2.3.	Resultados	85
7.2.4.	Calidad de la mallas adaptadas	88
7.3.	Arranque del flujo axisimétrico en una tobera acampanada	97
7.3.1.	Descripción del problema	97
7.3.2.	Solución numérica	98
7.3.3.	Resultados	101
7.4.	Onda explosiva esférica (Problema de Taylor-Sedov)	107
7.4.1.	Descripción del problema	107
7.4.2.	Solución numérica	113
7.4.3.	Resultados	115
7.4.4.	Costo relativo de la estrategia de adaptación de mallas	121
7.4.5.	Escalabilidad del algoritmo de refinamiento	122
8.	Conclusiones, Contribuciones y Trabajos a Futuro	124
8.1.	Conclusiones	124
8.2.	Contribuciones	125
8.3.	Trabajos a futuro	125
A.	Propiedades asociadas a la métrica η	127

B. Sistema de ecuaciones diferenciales ordinarias de Taylor-Sedov	131
Bibliografía	133

Índice de cuadros

7.1. Análisis estadístico sobre de calidad de las mallas - Flujo transónico alrededor de un cono.	93
7.2. Comparación del tiempo reloj requerido para resolver el flujo axisimétrico en una tobera.	106
7.3. Posición de la onda de choque para distintos instantes de tiempo - Comparaciones entre las soluciones numéricas y analítica.	115
7.4. Datos estadísticos sobre la calidad de las mallas para el problema de la explosión esférica.	117

Índice de algoritmos

1.	Pseudo-código de la función <code>driver()</code>	40
2.	Líneas del archivo Makefile para ejecutar el solver del flujo y la adaptación de la malla.	41
3.	Pseudo-código de la función <code>adapt()</code>	42
4.	Pseudo-código de la función <code>proc()</code>	44
5.	Pseudo-código de la función <code>refine()</code> . Fragmento correspondiente a la aplicación recursiva de la restricción al refinamiento.	51
6.	Pseudo-código de la función <code>search_fathers()</code>	57

Índice de figuras

2.1.	Relación entre las transformaciones de los plano lógico, de referencia y físico. . .	6
2.2.	Secuencia de refinamiento 1:8 para tetraedros.	9
2.3.	Métricas de Liu vs. iteraciones de refinamiento - Primera prueba de calidad. . .	11
2.4.	Mínimo ángulo diedro vs. iteraciones de refinamiento - Primera prueba de calidad.	12
2.5.	Comportamiento de la métrica de calidad η para ambas la diagonal óptima y la más corta sobre 100,000 tetraedros de muestra generados en forma aleatoria. . .	14
3.1.	Recuperación del gradiente por promediado - Parche de elementos cuando el nodo A no es irregular.	23
3.2.	Reconstrucción del gradiente por promediado - Parche de elementos cuando el nodo A es irregular.	23
5.1.	Numeración local de aristas y caras para tetraedros.	37
5.2.	Numeración local de aristas y caras para hexaedros.	38
6.1.	Patrones de refinamiento regulares 1:4 para elementos bidimensionales.	47
6.2.	Patrones de refinamiento regulares 1:8 para elementos tridimensionales.	47
6.3.	Refinamientos 1:4 y 1:2 para tetraedros.	48
6.4.	Aplicación de la restricción 1-irregular al refinamiento de una malla de cuadrángulos.	52
6.5.	Restricción 1-irregular al refinamiento para una malla de tetraedros - caso particular.	53
6.6.	Estrategia de adaptación para problemas no estacionarios.	59
6.7.	Interpolación lineal del estado para el primer paso de adaptación de la malla base.	60
6.8.	Proyección del estado - Determinación de la posición del vértice V en la malla correspondiente al paso de adaptación anterior.	61
6.9.	Diagrama de flujo para la estrategia de adaptación en problemas no estacionarios.	63
6.10.	Corrección coordenadas de nodo perteneciente a un contorno curvo.	65
6.11.	Refinamiento de la capa de elementos que están en contacto con un cuerpo cónico truncado. Corte de la malla en el plano de simetría.	67
6.12.	Generación de elementos invertidos producido por el ajuste de coordenadas sobre un contorno curvo.	69
6.13.	Situación no permitida - Proyección de nodo irregular sobre una superficie. . .	69

6.14. Gestión de propiedades sobre aristas y caras.	70
7.1. Angulo β de inclinación de la onda de choque vs. ángulo θ de desviación del flujo, parametrizados en el Mach de la corriente libre ($\gamma = 1.4$).	73
7.2. Distancia de separación δ/c entre la onda de choque y el vértice de la cuña como función del Mach reducido del flujo sin perturbar ξ_∞	74
7.3. Discretización inicial del dominio computacional y condiciones de borde.	75
7.4. Malla adaptada - Primera iteración.	76
7.5. Malla adaptada - Segunda iteración.	77
7.6. Malla adaptada - Tercera iteración.	77
7.7. Solución computada para la tercera iteración en el refinamiento.	78
7.8. Soluciones sobre el eje de simetría hasta el punto de impacto de la cuña.	79
7.9. Patrón del flujo para un cuerpo cónico-cilíndrico en una corriente libre supersónica en el infinito obtenido con la técnica de interferograma - Ref [Sol54].	82
7.10. Posición δ de la onda de choque respecto del punto sónico para distintas configuraciones θ_c y Mach de la corriente libre. Resultados experimentales.	83
7.11. Campo de Mach obtenido del análisis posterior de un ensayo interferométrico - Ref [Sol54].	83
7.12. Corte de la malla de partida sobre un plano de simetría.	84
7.13. Posición de la línea sónica para la onda de choque curva pegada al vértice del cono - Ref [Sol54].	86
7.14. Líneas de contorno para el número de Mach - Solución correspondiente al máximo nivel de refinamiento.	86
7.15. Isosuperficies de Mach para la superficie sónica $M = 1.0$ y el frente de onda $M = 1.39$ - Solución correspondiente al máximo nivel de refinamiento.	87
7.16. Detalle de las líneas de contorno de Mach en el vértice del cono.	88
7.17. Corte en un plano de simetría de la malla adaptada - Primer refinamiento.	89
7.18. Corte en un plano de simetría de la malla adaptada - Segundo refinamiento.	89
7.19. Corte en un plano de simetría de la malla adaptada - Tercer refinamiento.	90
7.20. Gradiente de Mach reconstruido sobre la malla base.	90
7.21. Gradiente de Mach reconstruido sobre la malla con un nivel de refinamiento.	91
7.22. Gradiente de Mach reconstruido sobre la malla con dos nivel de refinamiento.	91
7.23. $\ e \ _{L_2(\Omega_K)}$ obtenida por reconstrucción del gradiente sobre la malla base.	92
7.24. $\ e \ _{L_2(\Omega_K)}$ obtenida por reconstrucción del gradiente sobre la malla con un refinamiento.	92
7.25. $\ e \ _{L_2(\Omega_K)}$ obtenida por reconstrucción del gradiente sobre la malla con dos refinamientos.	93
7.26. Comparación de las soluciones para el número de Mach.	94
7.27. Comparación de las soluciones para la presión.	95

7.28. Histogramas con la distribución de la métrica de calidad η para los elementos de las mallas utilizadas en la simulación.	96
7.29. Tobera acampanada de un motor cohete de propulsión líquida.	97
7.30. Disposición esquemática del problema.	99
7.31. Distribución del tamaño de los elementos en la malla base sobre el plano xy . . .	101
7.32. Comparación de la velocidad axial sobre el eje de la tobera.	102
7.33. Comparación del módulo de la velocidad para $t=3.5e-4s$	103
7.34. Comparación del módulo de la velocidad para $t=6.0e-4s$	104
7.35. Malla adaptada para distintos instantes de tiempo.	105
7.36. Evolución del tamaño del problema para el procedimiento adaptativo.	106
7.37. Explosión esférica en la atmósfera.	108
7.38. Perfiles auto-similares para las variables del flujo Ω, U, P en el problema de Taylor-Sedov esférico.	112
7.39. Detalle de la condición inicial para la presión en un corte de la malla base sobre el plano $X - Z$ para el problema de Taylor-Sedov esférico.	114
7.40. Posición de la onda de choque en función del tiempo - Comparación entre la solución teórica dada por la Ec.(7.22) y las soluciones obtenidas con y sin el procedimiento de adaptación de la malla.	116
7.41. Caras expuestas de la malla adaptada para $t = 0.645ms$	116
7.42. Solución en el plano XZ para la densidad y malla adaptada - $t = 0.645ms$	118
7.43. Solución en el plano XZ para el número de Mach y la presión - $t = 0.645ms$. .	119
7.44. Perfiles de densidad y Mach sobre el eje Z para distintos instantes de tiempo. .	120
7.45. Histograma con la distribución de $\eta(T)$ para los elementos de la malla adaptada $step=1240$	121
7.46. Costo relativo para la adaptación de la malla en el problema de la explosión esférica.	122
7.47. Escalabilidad del algoritmo de refinamiento.	123
A.1. Tetraedro regular R	128

Capítulo 1

Introducción

La comunidad científica dedicada a la resolución de problemas de mecánica de fluidos mediante el método de elementos finitos reconoce la importancia de contar con alguna técnica que permita realizar la adaptación de la malla al cálculo de la solución. En el transcurso de los últimos años se han desarrollado un gran número de métodos adaptativos. Entre los más conocidos se encuentran aquellos que producen el enriquecimiento / empobrecimiento de la malla mediante la subdivisión / colapsado de los elementos de la malla (agregando / quitando por lo tanto grados de libertad al problema y modificando por ende la topología de la malla), aquellos que desplazan los nodos de la malla a las zonas de interés, y finalmente los que modifican el orden de las funciones de interpolación que el método de elementos finitos utiliza para aproximar la solución. Estos métodos son conocidos con los nombres de adaptatividad tipo $-h$, $-r$ y $-p$ respectivamente. De los tres, los métodos tipo $-h$ y $-r$ son los que gozan de mayor popularidad y los más desarrollados. El método que se presenta en esta tesis pertenece a los métodos tipo $-h$.

Por otro lado, también es conocida la influencia que tiene en el cálculo de la solución la presencia de elementos de “mala calidad” en la malla de elementos finitos. La mala calidad se ve reflejada en el mal condicionamiento de la matriz del método de elementos finitos, lo que genera complicaciones en el cálculo de la solución. Por lo tanto, resulta indispensable evaluar y controlar la calidad de la malla obtenida durante el procedimiento de adaptación. Se introducen conceptos que permiten definir métricas de calidad para elementos símplices. En función de ello, se presenta un método para evaluar la calidad de una malla. Luego, se realizan pruebas numéricas que permiten determinar una estrategia que resulta conveniente para particionar mallas de tetraedros, considerando tanto la calidad de la malla como el costo de refinamiento. Estas definiciones y pruebas numéricas son presentadas en el capítulo 2.

Un aspecto fundamental a la hora de desarrollar un procedimiento de adaptación de mallas es el método que se utiliza para identificar las regiones del dominio computacional que necesitan ser refinadas. Para ello se debe contar con procedimientos que permitan estimar el error en la solución aproximada obtenida por el método de elementos finitos. En este sentido, no se pretende desarrollar una nueva metodología para estimar el error en la solución, sino reconocer los procedimientos más utilizados para realizar esta tarea cuando el sistema de ecuaciones que

gobierna el flujo es hiperbólico. Una introducción a estos estimadores de error es presentada en el capítulo 3.

Otro elemento que constituye al procedimiento de solución adaptativo es el *solver* de las ecuaciones del problema fluido-dinámico. El código de adaptación de mallas se debe integrar con aquél que resuelve las ecuaciones del flujo. Para ello se utilizan llamadas al sistema y manejo de variables de entorno. Las ecuaciones que gobiernan el flujo no viscoso y compresible de gases perfectos son resueltas con un *solver* que forma parte del código multifísica en paralelo PETSc-FEM, el cual se basa en la implementación del método de elementos finitos estabilizado *Streamline Upwind Petrov-Galerkin* (SUPG) y en técnicas de *shock capturing*. Las ecuaciones de gobierno y el método de solución se presentan en el capítulo 4.

En el capítulo 5 se describen algunos aspectos relacionados con la implementación de las estructuras de datos que se requieren para representar las entidades geométricas de la malla y sus adyacencias. Se utilizan los contenedores y algoritmos que proporcionan las librerías de plantillas estándar STL y Boost, ambas escritas en C++.

El esquema de particionamiento de los elementos de la malla y la extensión de la restricción al refinamiento conocida como criterio de nodo 1-irregular para mallas tridimensionales son presentados en el capítulo 6. Ambos definen el núcleo del procedimiento de adaptación de la malla. Este núcleo es complementado por otras operaciones que permiten gestionar condiciones de borde y otras situaciones que se plantean al intentar resolver el problema en forma numérica. Se describe una estrategia que permite extender el procedimiento de adaptación a problemas no estacionarios. También se analizan diversos aspectos relacionados con la elección de la frecuencia de adaptación de la malla más adecuada.

El capítulo 7 presenta la aplicación del procedimiento de adaptación desarrollado a la resolución de problemas estacionarios y transitorios de flujos no viscosos, compresibles y sin transferencia de calor de gases perfectos. Se muestra la capacidad del código de adaptación para manejar distintos tipos de condiciones de borde y de elementos. Los problemas elegidos son problemas típicos de la dinámica de gases, de los cuales se conocen resultados experimentales, teóricos o numéricos con los cuales se puede verificar la validez de los resultados calculados en esta tesis. Los ejemplos son utilizados para evaluar distintos aspectos del método de adaptación de mallas, teniendo en cuenta la precisión de la solución obtenida como así también el uso de recursos computacionales.

Finalmente, en función de la experiencia obtenida durante el desarrollo y evaluación del código, se extraen conclusiones, se mencionan las contribuciones de la tesis al desarrollo de métodos de adaptación para resolver problemas de flujos compresibles y se proponen actividades a realizar en el futuro con el objeto de mejorar las capacidades del procedimiento desarrollado.

Capítulo 2

Calidad de malla

Las métricas de calidad de malla son utilizadas comúnmente tanto en la generación de mallas como en técnicas de mejorado de mallas. Las métricas de calidad son usadas en el contexto de procedimientos de refinamiento adaptativo ya que estos generalmente producen elementos que son de una calidad inferior a los de la malla de partida. Si no se tiene en cuenta el aspecto de la calidad, puede ocurrir que el procedimiento de refinamiento produzca como resultado mallas que son de baja calidad.

2.1. Métricas de calidad

En esta sección se definen conceptos que permiten evaluar la calidad de un elemento y de la malla. Con ellos, resulta posible analizar en qué medida la estrategia de refinamiento que se desea implementar deteriora la calidad de la malla. En la literatura dedicada al tema se observa que existe un gran número de métodos para evaluar la calidad de un elemento. Los más usuales se basan en criterios geométricos tales como el volumen del elemento, su relación de aspecto, los ángulos sólidos o los ángulos diedros. Estos criterios son clasificados como *métricas geométricas* de calidad. Sin embargo, Knupp[Knu00] es el primero en presentar una teoría matemática sobre las métricas de calidad de mallas basada en elementos del álgebra tales como las matrices Jacobianas del elemento, las normas de matrices, la traza y el determinante. En este ámbito se define el concepto de *métricas de calidad algebraicas* y las propiedades que las mismas deben satisfacer. En ése trabajo se analizan y clasifican varias métricas, a la vez que se identifican aquellas que son redundantes.

Cualquiera sea el tipo de métrica, Knupp las define de la siguiente manera:

Definición: “Una métrica de calidad de un elemento es una función escalar de la posición de sus nodos que mide alguna propiedad geométrica del mismo.”

Es decir que si un elemento tridimensional tiene K nodos cuyas coordenadas son $x_k \in \mathfrak{R}^3, k = 0, 1, \dots, K - 1$, entonces la función $\hat{f} | \mathfrak{R}^{3K} \rightarrow \mathfrak{R}$ indica a una métrica de calidad para el mismo.

Una característica importante sobre el concepto de métricas de calidad es que las mismas necesariamente deben estar referenciadas a un “elemento ideal”. Esto quiere decir que una métrica mide la desviación en la calidad de un elemento cualquiera con respecto al elemento de referencia elegido. Este elemento “ideal” puede variar de una aplicación a otra. Algunas aplicaciones requieren que los elementos de la malla tengan como ideal a un elemento isotrópico (por ejemplo: triángulo o tetraedro regular), mientras que otras aplicaciones pueden requerir elementos con una orientación preferencial, con lo cual resulta conveniente que el elemento ideal no sea isotrópico. En el caso particular de esta tesis se considera que el elemento ideal o de referencia es isotrópico, aunque esta probablemente puede no ser la elección más adecuada en problemas de flujo compresible[Ber98, BJW00].

2.1.1. Matrices Jacobianas de elementos símplices

Sea T un tetraedro en el espacio físico x y L el tetraedro de coordenadas $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ en el espacio lógico ξ (ver figura 2.1). La transformación que vincula al tetraedro L con el tetraedro T , referenciada al nodo de coordenadas $\vec{x}_0=(x_0, y_0, z_0)$, puede escribirse como

$$\vec{x}(\xi) = \sum_{k=0}^3 \xi_k \vec{x}_k \quad (2.1)$$

donde \vec{x}_k son los vectores de coordenadas de los nodos del tetraedro, con $0 \leq \xi_k \leq 1$ y $\xi_0 + \xi_1 + \xi_2 + \xi_3 = 1$. Entonces se tiene que

$$\vec{x}(\xi) = \xi_1(\vec{x}_1 - \vec{x}_0) + \xi_2(\vec{x}_2 - \vec{x}_0) + \xi_3(\vec{x}_3 - \vec{x}_0) + \vec{x}_0 \quad (2.2)$$

Indicando con A_0 a la matriz Jacobiana asociada a dicha transformación, entonces se observa que las coordenadas de los nodos pueden escribirse de la siguiente manera

$$\vec{x} = A_0 \cdot \vec{\xi} + \vec{x}_0 \quad (2.3)$$

donde la matriz A_0 está expresada como

$$A_0 = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{bmatrix} \quad (2.4)$$

siendo $\vec{\xi} = (\xi_1, \xi_2, \xi_3)^T$. A partir de esta matriz Jacobiana, se pueden generar métricas de calidad que son *independientes de la dimensión*. Esto significa que la definición de la métrica de calidad es independiente de la dimensión geométrica del elemento al cual se desea aplicar. Tal como se observa, al nodo k del elemento símplice está asociada la matriz Jacobiana A_k , que contiene información relacionada con el volumen, la forma y la orientación del elemento.

Sean λ_{ij} los elementos del tensor métrico definido como $A^t A$. Denotamos como $\alpha = \det(A)$ y asumimos que $0 < |\alpha| < \infty$ y que $\alpha \geq 0$ (es decir que no se consideran elementos invertidos).

Evidentemente, si quisiéramos construir una métrica de calidad algebraica para elementos simples a partir de la matriz A_k , entonces dicha métrica sería dependiente del nodo k al cual referenciamos la matriz Jacobiana. Sin embargo, es posible obtener una matriz Jacobiana que es nodalmente invariante, la cual se indica como T . Para ello, pensemos que se tiene una transformación que mapea al elemento L que se encuentra en el espacio lógico ξ al elemento R en el espacio de referencia \mathcal{R} (tal como se mencionara en la introducción, el elemento de referencia que se elige en esta tesis es el tetraedro regular). Igual que antes, llamemos W_k a la matriz Jacobiana asociada a esta transformación referenciada al nodo k del elemento R. Entonces la matriz

$$T_k = A_k \cdot W_k^{-1} \quad (2.5)$$

es la matriz asociada a la transformación que mapea al elemento R del plano de referencia al elemento T en el plano físico. Se demuestra fácilmente que dicha matriz es nodalmente invariante, es decir, es independiente del nodo k elegido para computarla. Para ello se procede de la siguiente manera: sea la matriz M dada por

$$M = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (2.6)$$

Entonces se comprueba que $\mathcal{G} = \{I_3, M, M^2, M^3\}$, siendo I_3 la matriz identidad 3×3 , es un grupo cíclico bajo la multiplicación de matrices. Luego se verifica mediante cálculo directo que las cuatro matrices Jacobianas de un tetraedro están relacionadas unas con otras de la siguiente manera

$$A_k = A_0 M^k \quad (2.7)$$

Si aplicamos la relación cíclica entre las matrices W_k se obtiene

$$W_k = W_0 M^k \quad (2.8)$$

con lo cual se observa que

$$A_0 M^k = T_k W_0 M^k \quad (2.9)$$

Posmultiplicando esta última ecuación por $(M^k)^{-1}$ y despejando la matriz T_k se llega a la siguiente

$$T_k = A_0 W_0^{-1} \quad (2.10)$$

De esta forma, se demuestra que T es independiente del nodo de referencia k elegido, es decir que

$$T = A_k \cdot W_k^{-1}, \quad k = 0, 1, 2, 3 \quad (2.11)$$

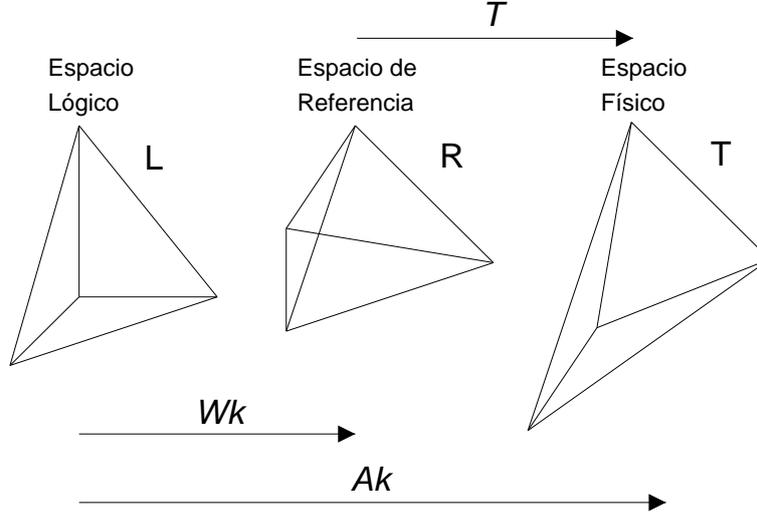


Figura 2.1: Relación entre las transformaciones de los plano lógico, de referencia y físico.

La figura 2.1 ilustra la relación que existe entre las tres transformaciones. Las métricas algebraicas están definidas sobre la matriz T y el tensor métrico $M = T^t T$. Por lo tanto, al utilizar la matriz T para definir métricas de calidad algebraicas, se asegura que la mismas son nodalmente invariantes.

A continuación se transcriben algunas definiciones dadas en[Knu00] importantes para el análisis de las métricas de calidad,

Métricas algebraicas de calidad

Sea M_n el conjunto de las matrices $n \times n$ a coeficientes reales. Sea $f | B_i \in M_n, i = 0, \dots, I \rightarrow \mathfrak{R}$ una función continua desde el conjunto de las matrices reales a los números reales. Entonces f es una métrica de calidad algebraica de mallas si:

1. Las matrices B_i son construidas a partir de A_k, W_k o factorizaciones derivadas a partir de ellas.
2. Las matrices B_i son convertidas a escalares por medio de normas de matrices, el determinante o la traza.
3. f es nodalmente invariante con respecto de la matriz con la que es computada.

Medidas de forma (*shape measure*) para tetraedros

Es una función continua de la posición de los nodos que evalúa la calidad de un tetraedro. Debe ser invariante ante rotación, traslación, reflexión y escalado uniforme del tetraedro. Debe

ser máxima para el tetraedro regular y mínima para un tetraedro *degenerado*. No debe existir otro máximo local que no sea el absoluto ni otro mínimo local que no sea el absoluto. Por conveniencia debe estar escalada al intervalo $[0, 1]$ y debe valer 1 para el tetraedro regular y cero para todo tetraedro degenerado.[DLGC98]

Elemento degenerado

Se define como s mplice degenerado a aqu el tal que todos sus v rtices est n incluidos en un sub-espacio de \mathfrak{R}^d . Es decir, un s mplice en el espacio de dimensi n d es degenerado si sus $d + 1$ v rtices no expanden al espacio \mathfrak{R}^d . Es decir que un tri ngulo es degenerado si todos sus v rtices son colineales o colapsan en un punto, mientras que un tetraedro es degenerado si todos sus v rtices son coplanares, o colineales o colapsan en un punto. En la pr ctica se dice que un tri ngulo es degenerado si su  rea se hace cero, y un tetraedro es degenerado si su volumen se hace cero.

Tal como se menciona en los trabajos de Knupp[Knu00] y Dompierre[DVLG05], en funci n de la definici n dada arriba, los  ngulos diedros m nimos no son una medida de forma v lida para tetraedros, ya que el uso de los  ngulos diedros m nimos como medida de forma no permite detectar algunos tipos de elementos degenerados. Por ejemplo: el menor  ngulo diedro de un tetraedro degenerado tipo *needle* o aguja puede ser tan grande como $\pi/3$. Tampoco lo es la relaci n de longitudes de las aristas del elemento,

$$r = h_{max}/h_{min} \quad (2.12)$$

donde $h_{max} = \max_{ij}(l_{ij})$, $h_{min} = \min_{ij}(l_{ij})$, l_{ij} es la longitud de la arista definida por los nodos $ij = 1, \dots, 4$.

S  lo es la m trica algebraica introducida por Liu[LJ94], la cual se denomina *mean ratio shape measure*. Las medidas de forma son en general m tricas de calidad de malla. Sin embargo, no todas son m tricas de calidad algebraicas.

2.1.2. M trica de Liu y Joe

Liu y Joe[LJ94] introducen una medida de forma para tetraedros que ser  indicada como η , la cual requiere menor esfuerzo de c lculo que ambas la relaci n de aspecto ρ   los  ngulos s lidos m nimos θ_{min} . Esta m trica es definida como

$$\eta(T) = \frac{3\sqrt[3]{\lambda_1\lambda_2\lambda_3}}{(\lambda_1 + \lambda_2 + \lambda_3)} \quad (2.13)$$

donde $\lambda_1, \lambda_2, \lambda_3$ son los autovalores del tensor m trico $M = T^t T$. Se debe satisfacer la condici n de que el tetraedro T en el plano f sico tenga el mismo volumen que el tetraedro regular R en el plano de referencia. Notar que los 3 autovalores son positivos ya que M es definida positiva. Adem s, $0 \leq \eta(T) \leq 1$ con $\eta(T) = 1$ si y s lo si $\lambda_1 = \lambda_2 = \lambda_3$, es decir, si y s lo si el tetraedro T es el regular.

Tal como se demuestra en el apéndice A, esta métrica está relacionada con el volumen v y las longitudes de las aristas l_i , $i = 1, \dots, 6$ del tetraedro T de la siguiente manera

$$\eta(T) = \frac{12(3v)^{2/3}}{\sum_{i=1}^6 l_i^2} \quad (2.14)$$

Dado que $\eta(T)$ está basada en los autovalores del tensor M y que éste es independiente del orden de los vértices de T , entonces se sigue que $\eta(T)$ también posee las mismas propiedades.

Liu y Joe[LJ94] demuestran que la métrica $\eta(T)$ es la relación de la media geométrica a la media aritmética de α^2 , β^2 , γ^2 , es decir

$$\eta(T) = \frac{3\sqrt[3]{\alpha^2\beta^2\gamma^2}}{\alpha^2 + \beta^2 + \gamma^2} \quad (2.15)$$

donde α , β , γ son las semi-longitudes de los 3 ejes principales del elipsoide E inscrito en el tetraedro T . Por lo tanto, se dice que $\eta(T)$ mide la forma del elipsoide E .

2.2. Experiencias numéricas

Se realizan dos experiencias numéricas con el objeto de determinar un procedimiento conveniente para llevar a cabo el particionamiento de elementos tetraédricos, tanto desde el punto de vista de la calidad de los elementos que se obtienen como así también desde el punto de vista del costo computacional. En todos los casos se utiliza, para evaluar la calidad de las mallas, la métrica de Liu (η) descrita en la sección 2.1.2 y el mínimo de los ángulos diedros del tetraedro (Φ_{min}).

Otro aspecto a considerar es cómo evaluar la calidad de la malla a partir de la calidad de sus elementos. En todas las experiencias realizadas en esta tesis se considera que la calidad de la malla η_τ es la de su elemento de más baja calidad, es decir

$$\eta_{min} = \min_{T_k \in \tau} (\eta(T_k)) \quad (2.16)$$

Si bien dicho valor no puede caracterizar por completo a una malla, a medida que el mínimo disminuye más significativo se hace su valor.

Otra forma usual de evaluar la calidad de una malla consiste en realizar un análisis estadístico sobre los valores que el índice de calidad elemental elegido presenta para esa malla en particular. En [DVLG05] se indica que la distribución de los valores que asumen distintos índices o métricas de calidad para elementos símplices responden a una distribución normal. Luego, dicha distribución puede ser caracterizada por cuatro parámetros: el valor mínimo, la media, la desviación estándar y el valor máximo de la métrica elemental elegida. De esta manera, si se utiliza la métrica de Liu como índice de calidad elemental, entonces la desviación estándar de la distribución viene dada por

$$\eta_{Std} = \sqrt{\frac{\sum_k (\eta(T_k) - \bar{\eta})^2}{nele - 1}}, \quad k = 1, \dots, nele \quad (2.17)$$

donde $nele$ es el número de elementos en la malla y $\bar{\eta}$ es la calidad media.

En función de este criterio, una calidad media y mínima elevadas así como también una desviación estándar pequeña son todas indicativas de una buena calidad de malla. Ninguna de las tres medidas por sí solas caracterizan por completo a la distribución de los índices de calidad elemental, por lo tanto todas deben ser tenidas en cuenta al momento de evaluar la calidad de la malla. El valor máximo no resulta significativo ya que siempre es cercano a 1, por lo cual no es considerado.

Secuencia de particionamiento para tetraedros

La secuencia de refinamiento que se propone para elementos tetraédricos es la regular o isotrópica 1:8. La misma consiste en unir entre sí los puntos medios de las aristas del tetraedro, con lo cual se obtienen cuatro tetraedros semejantes al tetraedro de partida (estos elementos poseen la misma calidad que el tetraedro de partida) en los cuatro vértices y un octaedro interior. El octaedro resultante es particionado en cuatro tetraedros más al agregar una arista que une los vértices a_0a_1 , a_2a_3 ó a_4a_5 , de la forma que se observa en la figura 2.2. Existe en la bibliografía una discrepancia en cuanto a la elección de la diagonal más conveniente desde el punto de vista de la calidad de los elementos que se obtienen. Algunos autores [LB92] indican que debe ser la diagonal más corta, mientras que otros [BS97] indican que debe ser la más larga. La experiencia numérica que se propone a continuación pretende arrojar algo de luz sobre esta cuestión.

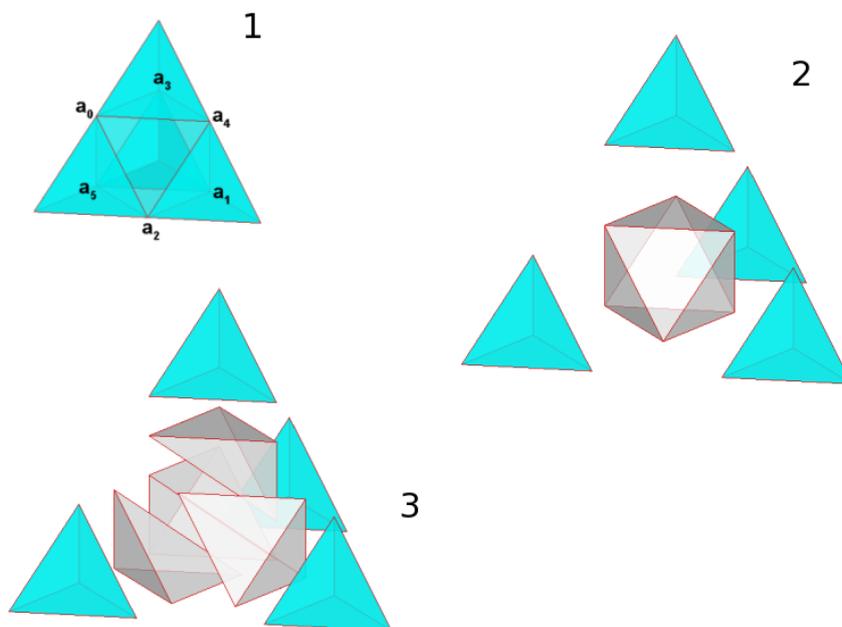


Figura 2.2: Secuencia de refinamiento 1:8 para tetraedros.

2.2.1. Primera prueba de calidad

Se desea analizar el comportamiento de la calidad de la malla al aplicar la secuencia de refinamiento mencionada en forma reiterada. La prueba se inicia con el refinamiento de un único elemento. Para iniciar la prueba se eligen tres tetraedros: el regular ($\eta = 1$), el recto ($\eta \simeq 0.84$) y un elemento de baja calidad del tipo *cap*¹ ($\eta \simeq 0.045$). Una vez aplicada la secuencia de refinamiento, se determina cuál es el elemento de más baja calidad entre los 4 sub-elementos obtenidos por particionar al octaedro interior. Este elemento es refinado y se vuelve a repetir el procedimiento.

Para cada tetraedro inicial, el octaedro interior que se obtiene de aplicar el esquema de refinamiento es particionado eligiendo siempre o la diagonal más larga, o la más corta o eligiendo la misma en forma aleatoria. Se considera que la malla resultante es no conforme y que posee nodos irregulares.

Definición: una malla es conforme cuando la intersección de dos tetraedros cualesquiera T_1 y T_2 de la malla es o una cara común, o una arista común, o un vértice común, o el conjunto vacío. En caso contrario la malla es no conforme.

Definición: un nodo irregular es aquél que surge sobre una arista o cara de un elemento pero que no es un vértice del elemento. Si cualquier arista o cara de la malla posee a lo sumo un nodo irregular, entonces se dice que la malla es 1-irregular[CH07].

Las figuras 2.3 y 2.4 presenta el comportamiento de la calidad de la malla en función del número de iteraciones en el refinamiento. Se observa, para los tres tetraedros iniciales elegidos, que la disminución de la calidad mínima debida a la aplicación sucesiva del refinamiento es minimizada si se elige siempre la diagonal más corta del octaedro interior para llevar a cabo el particionamiento. Luego se concluye que la calidad mínima de la malla disminuye únicamente en el primer refinamiento y luego se mantiene constante. Con las otras alternativas la calidad mínima tiende a cero si se la evalúa a través de la métrica de Liu, o a una constante pequeña si se utilizan los ángulos diedros mínimos.

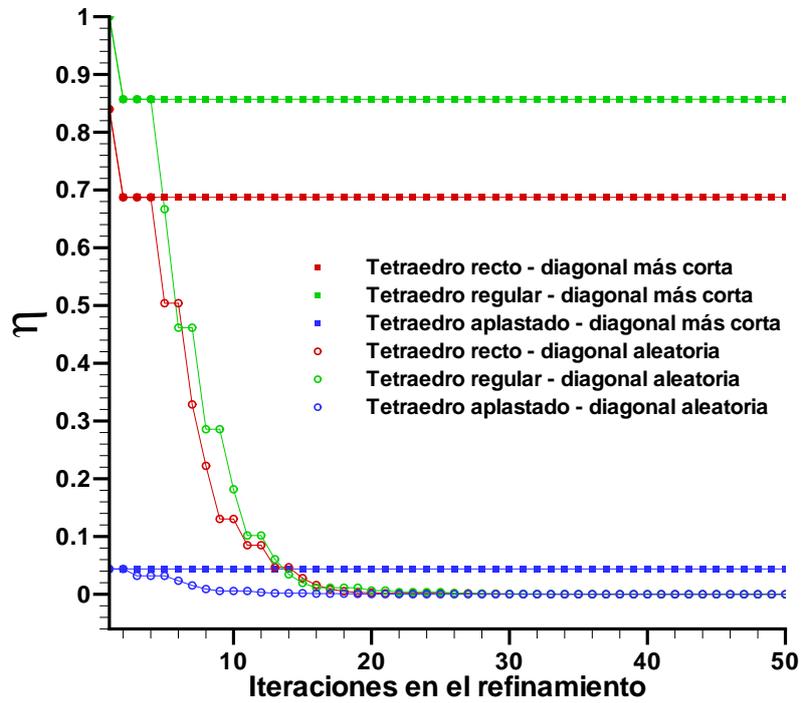
Resultados similares a estos han sido obtenidos por Plaza[PPS05] aplicando otros procedimientos² para particionar tetraedros 1:8.

2.2.2. Segunda prueba de calidad

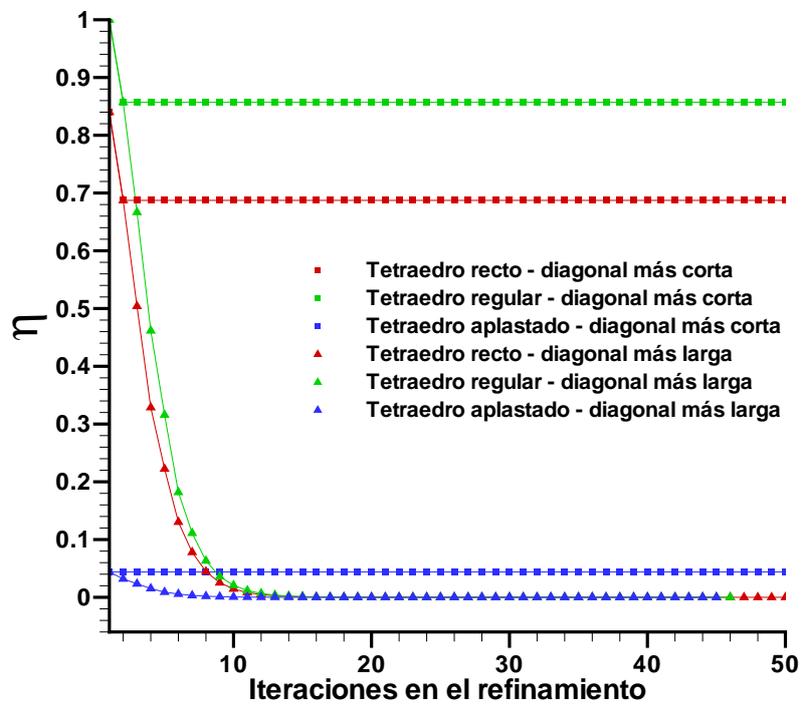
La segunda experiencia numérica tiene por objeto determinar si efectivamente el refinamiento de *cualquier* tetraedro eligiendo la diagonal más corta para particionar al octaedro interior es *siempre* la mejor alternativa. Este objetivo no es demostrado en forma matemáticamente rigurosa, sino aplicando el esquema de particionamiento que se observa en la figura 2.2 a un número

¹Se dice que un tetraedro es del tipo *cap* cuando el mismo posee un ángulo sólido muy grande, casi plano. Para tetraedros de este tipo el radio de la esfera circunscrita es mucho más grande que la longitud de la arista más larga.

²Estos procedimientos son 8T-LE (8 *Tetrahedra Longest Edge*) y aquél propuesto por Liu en [LJ96].

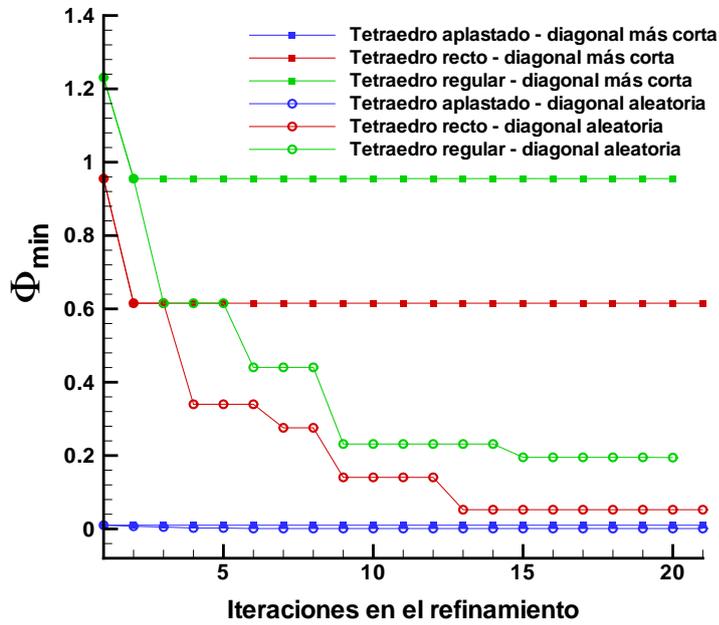


(a) Diagonal más corta vs. diagonal aleatoria

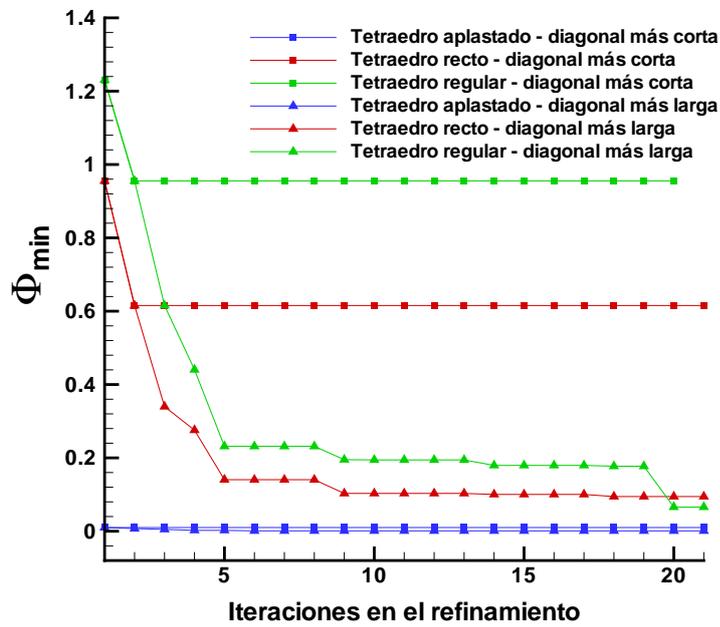


(b) Diagonal más corta vs. diagonal más larga

Figura 2.3: Métricas de Liu vs. iteraciones de refinamiento - Primera prueba de calidad.



(a) Diagonal más corta vs. diagonal aleatoria



(b) Diagonal más corta vs. diagonal más larga

Figura 2.4: Mínimo ángulo diedro vs. iteraciones de refinamiento - Primera prueba de calidad.

suficientemente grande de tetraedros (por ej.:100,000), tales que las coordenadas de sus nodos son generadas en forma aleatoria utilizando una distribución uniforme en el intervalo $[0, 1]$. Para cada tetraedro se elige la diagonal que maximiza el criterio de calidad adoptado (al igual que en la primera experiencia, en uno de los casos este criterio es el valor mínimo de la métrica η y en el otro caso es el mínimo de los ángulos diedros mínimos) y se la indica con el subíndice Opt

$$d_{Opt} \rightarrow d_j \quad \begin{cases} \eta_{Opt} &= \max_{j=1}^3 \min_{k=1}^4 \eta(T_k) \\ \Phi_{Opt} &= \max_{j=1}^3 \min_{k=1}^4 \Phi_{min}^{(k)} \end{cases} \quad (2.18)$$

Por otro lado, también se tienen en cuenta los valores del criterio de calidad obtenidos al refinar el octaedro interior por la diagonal más corta (subíndice $Shortest$)

$$d_j \rightarrow d_{Shortest} \quad \begin{cases} \eta_{Shortest} &= \min_{k=1}^4 \eta(T_k) \\ \Phi_{Shortest} &= \min_{k=1}^4 \Phi_{min}^{(k)} \end{cases} \quad (2.19)$$

Las figuras 2.5 y el análisis posterior de los resultados de esta prueba permiten extraer las siguientes conclusiones:

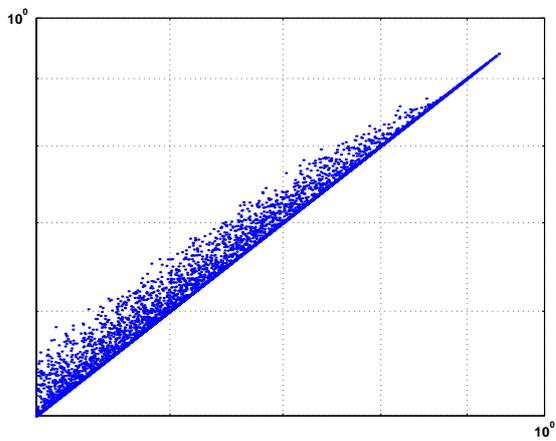
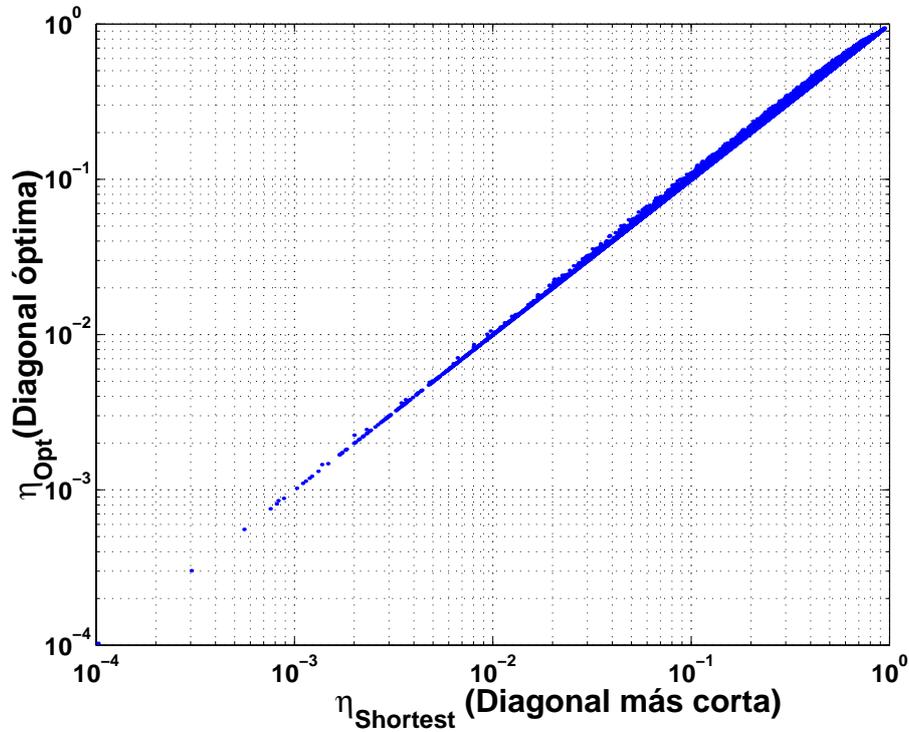
- Ni la métrica algebraica de Liu η ni los ángulos diedros mínimos Φ_{min} son *siempre* maximizados si se elige la diagonal más corta para particionar al octaedro interior.
- Sin embargo, en la mayoría de los casos (aproximadamente el 92 % para la métrica de Liu y el 80 % para los ángulos diedros mínimos), el utilizar la diagonal más corta para refinar al octaedro interior permite maximizar el criterio de calidad.
- Si bien la elección de la diagonal más corta no siempre maximiza el criterio de calidad, la pérdida relativa de calidad más grande que se observa en estos casos es del 13 % para la métrica de Liu y del 30 % para los ángulos diedros mínimos. Esto es

$$\max_{i \in \tau} \left(\frac{\eta_{Opt}(T_i) - \eta_{Shortest}(T_i)}{\eta_{Opt}(T_i)} \right) \simeq 0.13 \quad (2.20)$$

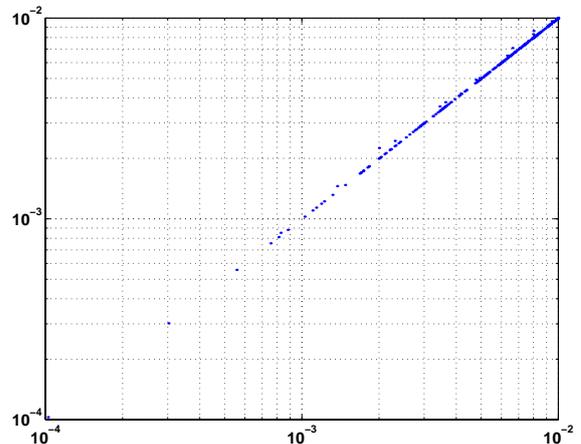
$$\max_{i \in \tau} \left(\frac{\Phi_{Opt}(T_i) - \Phi_{Shortest}(T_i)}{\Phi_{Opt}(T_i)} \right) \simeq 0.13 \quad (2.21)$$

- La pérdida relativa de calidad para tetraedros de baja calidad inicial $\eta(T_i) < 10^{-2}$ y para tetraedros de alta calidad inicial $\eta(T_i) \simeq 1$ es mucho menor que el 13 por ciento mencionado en el ítem anterior (ver detalle en las figuras 2.5(a) y 2.5(b)). Esto resulta importante para los elementos de baja calidad ya que resulta fundamental evitar la disminución de la calidad en estos casos.
- Finalmente, se observa que la estrategia que maximiza el criterio de calidad resulta costosa, desde el punto de vista computacional. Esto es así porque se debería refinar el octaedro

a través de las tres diagonales y computar los mínimos índices de calidad de los cuatro elementos obtenidos para cada una de ellas. Por otro lado, la estrategia de la diagonal más corta sólo requiere calcular las longitudes de las tres diagonales para luego refinar a través de ella.



(a) Detalle para la zona de tetraedros de alta calidad.



(b) Detalle para la zona de tetraedros de baja calidad.

Figura 2.5: Comportamiento de la métrica de calidad η para ambas la diagonal óptima y la más corta sobre 100,000 tetraedros de muestra generados en forma aleatoria.

2.2.3. Conclusiones

De ambas pruebas numéricas se concluye que la estrategia de seleccionar siempre la diagonal más corta para refinar al octaedro interior es la alternativa que presenta la mejor relación

entre el costo de ejecución y la calidad de malla obtenida. Si bien esta es la secuencia elegida para esta tesis, Berzins [BJW00] indica que, en el caso de soluciones que presentan una fuerte anisotropía, puede resultar ventajoso realizar el particionamiento del octaedro de forma tal que la diagonal interior quede alineada con la dirección del flujo.

Por otro lado, se decide no utilizar elementos de transición o plantillas, tales como aquellas introducidas por [KV93], para eliminar la presencia de nodos irregulares y empalmar las regiones de la malla que poseen distinto grado de refinamiento. Esta decisión se respalda en el hecho de que, tal como se menciona en [Sta96], la utilización de elementos plantillas genera un mayor deterioro de la calidad de la malla e implica un costo de gestión importante. En la tesis de Staten[Sta96] se menciona que existen 56 configuraciones de nodos irregulares que deben ser consideradas individualmente.

La presencia de nodos irregulares (*hanging nodes*) hace que la malla resultante sea no conforme. En el caso de utilizar una formulación estándar de Galerkin para resolver el problema de elementos finitos, si las funciones de aproximación que se emplean son lineales, entonces es suficiente con restringir la solución sobre los nodos irregulares al promedio de la solución calculada sobre los nodos que definen a la arista, para asegurar que la solución presente continuidad al pasar de un elemento de la malla que posee un nodo irregular a su vecino. En el caso de utilizar un método que introduzca *upwind* para estabilizar la solución, la estrategia anterior puede generar oscilaciones localizadas. Para corregir este efecto, también debe dársele un tratamiento tipo *upwind* a la contribución del nodo irregular a cada uno de los nodos padres de la arista sobre la cual se encuentra dicho nodo.

Capítulo 3

Estimación a posteriori del error

3.1. Introducción

Un elemento fundamental en todo procedimiento de adaptación de mallas es el método de estimación del error, ya que en general no se cuenta con la solución exacta del problema que se intenta resolver. Dado que el procedimiento de adaptación implica que el refinamiento de la malla se realiza en forma localizada, resulta necesario contar con una estimación del *error local*, más que una estimación del error global, en la solución aproximada obtenida mediante el método de elementos finitos. De esta manera se pueden determinar las regiones del dominio computacional que deben ser refinadas o desrefinadas.

El error en la solución aproximada por el método de elementos finitos u^h depende de parámetros de discretización tales como el orden del espacio de aproximación y el tamaño de malla[Joh95]. Babuska y Rheinboldt[BR78] fueron los primeros en realizar un análisis de la estimación a posteriori del error cuando el problema está gobernado por ecuaciones diferenciales parciales elípticas. Desde entonces, se han desarrollado un gran número de procedimientos para tratar de estimar el error cuando el carácter matemático de las ecuaciones en derivadas parciales no es elíptico. Sin embargo, la teoría relacionada con la estimación del error se encuentra notablemente menos desarrollada para el caso de ecuaciones diferenciales parciales hiperbólicas (por ejemplo, la ecuación de advección-difusión escalar con término advectivo dominante o el sistema de ecuaciones de Euler) y sólo recientemente se han obtenido resultados relevantes para la ecuación de advección-difusión escalar[Joh95, Sül98, Ver98, BL02, MK06]. Resulta interesante el trabajo presentado por Volker[Vol00], en el cual se evalúan y comparan algunos de estos procedimientos para la ecuación de advección-difusión escalar. Allí se analiza la precisión de las soluciones computadas con los distintos estimadores de error.

En este capítulo se presentan algunos de los métodos más usuales para la estimación a posteriori del error local en problemas de advección-difusión o advectivos puros, ya que la clase de problemas a los cuales se desea aplicar el procedimiento de solución adaptativo responde a ecuaciones en derivadas parciales de este tipo.

Las técnicas de estimación del error a posteriori se caracterizan por tratar de estimar en

alguna norma el error en la solución discreta u^h , contando para ello sólo con información proporcionada durante el cálculo de la solución (como la solución discreta en sí misma) y los datos del problema. Por lo tanto, el primer paso en esta tarea consiste en elegir una norma apropiada en la cual evaluar el error estimado. En problemas de advección-difusión es usual utilizar la norma- L_2 $\|e\|_{L_2(\Omega)}$ del error en la solución o en el gradiente de la solución.

Resulta necesario indicar la diferencia que existe entre los usualmente llamados *indicadores de error* y los *estimadores de error*. Los primeros sólo tienen por objeto identificar aquellas regiones del dominio computacional que requieren mayor resolución de la malla, pero no permiten obtener una estimación del error local en la solución aproximada. Por otro lado, los estimadores de error permiten alcanzar ambos objetivos.

Como indicadores de error se utilizan frecuentemente métodos heurísticos, los cuales se basan en calcular sobre los elementos o aristas de la malla la norma del gradiente (o la curvatura) de una variable o una combinación de variables de la solución aproximada u^h , para luego seleccionar aquellos elementos (vía relaciones de compatibilidad si la estimación del error se realiza a nivel de aristas - ver sección 6.1) tales que la magnitud de dicha norma sea mayor a un cierto valor de corte o tolerancia. Estos métodos resultan ser computacionalmente económicos, son fáciles de implementar y son independientes de la naturaleza del problema, pero su éxito depende de la experiencia y habilidad del usuario para elegir las variables indicadoras correctas[Mav95, Vol00].

Otro procedimiento frecuentemente utilizado en la práctica para estimar el error consiste en emplear técnicas de recuperación o reconstrucción del gradiente a partir de la solución aproximada. Dichas técnicas fueron formuladas inicialmente por Zienkiewicz y Zhu[ZZ87] y se basan en obtener una estimación del error en el gradiente a partir de reconstruir una aproximación “mejorada” del gradiente de la solución. Distintas formulaciones para la reconstrucción del gradiente dan origen a distintos métodos. Entre ellos, el procedimiento conocido como *parche de recuperación superconvergente* ó *Superconvergent Patch Recovery* (SPR)[ZZ92a, ZZ92b] es el más representativo y utilizado en la práctica. Sin embargo, esta aproximación permite obtener estimaciones correctas del error cuando el sistema de ecuaciones que se resuelve exhibe un carácter elíptico o parabólico, pero no resulta adecuada cuando el sistema de ecuaciones que gobierna el problema admite soluciones discontinuas. Actualmente se siguen proponiendo métodos para recuperar el gradiente[MK06] específicos para problemas hiperbólicos.

Al igual que en las etapas restantes del procedimiento de adaptatividad, cada una de las técnicas mencionadas debe ser evaluada desde el punto de vista de su efectividad y costo computacional. Mavriplis indica que una forma de evaluar la efectividad de un estimador de error es a través del número de elementos que el estimador selecciona para refinar en relación a la mejora en la precisión obtenida en el cálculo de la solución. En el caso de problemas estacionarios, es importante que se produzca un incremento suficiente en el número de grados de libertad del problema luego de cada refinamiento. Si esto no ocurriera, entonces se incurriría en un esfuerzo para resolver las ecuaciones sobre la malla refinada que no reportaría en una mejora

significativa en la calidad de la solución. Por otro lado, si el estimador o indicador seleccionara demasiadas celdas para refinar, entonces el tamaño del problema podría volverse inmanejable para los recursos computacionales disponibles.

3.2. Indicadores y estimadores de error

Como se mencionara antes, y a pesar de que no permiten obtener una estimación del error en la solución aproximada, el criterio heurístico de seleccionar los elementos a refinar basado en el gradiente elemental de la solución aproximada (o una combinación de variables de la solución) permite detectar adecuadamente las regiones del dominio a refinar. Además, dado que implica un bajo costo extra en el cálculo de la solución, es que el mismo goza de gran popularidad en el ambiente de adaptación de mallas en problemas de flujo compresible [LB92, Mav95, BS97, FLS⁺97, FLÖ⁺98, Wal04, YK05]. Ya que el procedimiento de adaptación que se desarrolla en esta tesis se aplica a la resolución de problemas de flujo compresible, este indicador ha sido el elegido en la mayor parte de los problemas que se resuelven.

En las regiones del campo de movimiento en las que la solución es suave, el error local del método de elementos finitos es proporcional al producto de la m -ésima derivada de la solución multiplicada por una medida representativa del tamaño del elemento (h_e) elevada a la p -ésima potencia.

$$\varepsilon_e \propto h_e^p \frac{\partial^m u}{\partial x^m} \quad (3.1)$$

donde m denota el mayor orden de las derivadas que aparece en la formulación integral o débil y p es el orden de las funciones de interpolación del método de elementos finitos. Por lo tanto, resulta inmediato ver que las regiones del dominio computacional que requieren un mayor grado de refinamiento son aquellas en las cuales se presenta una variación grande del campo de solución.

En la práctica se encuentra que los gradientes de presiones y de densidad permiten detectar características propias del flujo no viscoso tales como ondas de choque, discontinuidades de contacto y abanicos de expansión. Por otro lado, el gradiente del campo de velocidades puede ser utilizado para identificar correctamente estelas y láminas vorticosas.

Para el caso en el que la aproximación por elementos finitos que se utiliza es lineal $p = 1$ y las derivadas que aparecen en la formulación integral débil del problema de elementos finitos son a lo sumo de primer orden, el error elemental se puede aproximar de la siguiente manera

$$\begin{aligned} \varepsilon_e &\simeq |\nabla_e u| \cdot h_e \\ &\simeq \Delta u_e \end{aligned} \quad (3.2)$$

donde $|\nabla_e u|$ es la magnitud del gradiente elemental del campo de solución u y el operador Δ no

es un Laplaciano, sino que representa una diferencia.

El uso del gradiente elemental multiplicado por el tamaño del elemento asegura que el valor de ε_e , el cual debería aproximar el error de la solución, disminuye a medida que se reduce el tamaño de la malla.

3.2.1. Técnicas de recuperación o reconstrucción del gradiente para problemas de flujo compresible

La técnica de estimación del error propuesta por Zienkiewicz y Zhu[ZZ87] basada en la reconstrucción del gradiente, tiene su origen en la siguiente observación: si es posible estimar un gradiente nodal mejorado o recuperado Gu respecto del gradiente ∇u^h obtenido en forma directa a partir de derivar la solución computada mediante el método de elementos finitos (al cual nos referiremos de ahora en adelante como gradiente de bajo orden), luego es posible aproximar el error global en el gradiente de la solución evaluado en la norma- L_2 de la siguiente manera

$$\| e \|_{L_2(\Omega)} = \| \nabla u - \nabla u^h \|_{L_2(\Omega)} \quad (3.3)$$

$$\simeq \| Gu - \nabla u^h \|_{L_2(\Omega)} \quad (3.4)$$

donde $L_2(\Omega)$ es el espacio de funciones que son cuadrado integrable en el dominio Ω . Por definición de la norma- L_2 se tiene que

$$\| e \|_{L_2(\Omega)} = \left(\int_{\Omega} e^T \cdot e d\Omega \right)^{1/2} \quad (3.5)$$

Si se utilizan elementos finitos cuyas funciones de forma son lineales, el gradiente ∇u^h resulta constante por elemento. Luego, el gradiente recuperado debe ser al menos lineal por elemento. Si Ω_K indica al K -ésimo elemento de la malla entonces una cota (inferior) para el error global en la solución viene dada por la contribución de los nel elementos

$$\| e \|_{L_2(\Omega)}^2 = \sum_{K=1}^{nel} \| e \|_{L_2(\Omega_K)}^2 \quad (3.6)$$

$$\simeq \sum_{K=1}^{nel} \| Gu - \nabla u^h \|_{L_2(\Omega_K)}^2 \quad (3.7)$$

Se han diseñado distintos procedimientos para obtener el gradiente recuperado Gu . Estos incluyen el promediado y el procedimiento conocido como parche de recuperación superconvergente (*Superconvergent Patch Recovery*)[ZZ92a, ZZ92b].

- Las técnicas de **reconstrucción del gradiente por promediado** requieren contar con

información acerca de los elementos de la malla que comparten a un nodo (adyacencias `nodo_elementos`). Es decir, dado un nodo A identificar los elementos de la malla que lo comparten. Estos elementos forman un parche de elementos que se indica como ω_A (ver figura 3.1). Luego, el gradiente recuperado en el nodo A de la malla puede ser computado mediante un simple promediado de la siguiente manera

$$Gu(A) = \frac{1}{N(A)} \sum_{K \in \omega(A)} \nabla u^h|_K \quad (3.8)$$

siendo $N(A)$ el número de elementos K que constituyen el parche $\omega(A)$ y $\nabla u^h|_K$ el gradiente de bajo orden en el elemento K . Otra forma usual de reconstruir o *suavizar* el gradiente consiste en utilizar como “peso” la inversa de la distancia $d_{c,A}^K$ desde el baricentro del elemento K al nodo A

$$Gu(A) = \sum_{K \in \omega(A)} \frac{\nabla u^h|_K}{d_{c,A}^K} / \sum_{K \in \omega(A)} \frac{1}{d_{c,A}^K} \quad (3.9)$$

Recordar que en el caso de elementos finitos lineales el único punto de Gauss es coincidente con el baricentro del elemento y que el gradiente de bajo orden es constante en el elemento. El procedimiento de recuperación dado por la Ec.(3.9) también es conocido como reconstrucción por promediado de Gauss[NS04, Usn98]. Otros autores[Vol00] proponen utilizar como “peso” a las áreas o los volúmenes de los elementos que forman el parche en relación al área o volumen total del parche.

Cuando el nodo A es irregular, el procedimiento de reconstrucción es el mismo que se acaba de explicar, pero ahora se agregan a la lista de adyacencias `nodo_elementos` aquellos elementos del parche que comparten al nodo irregular a través de una arista (para elementos 2-D y tetraedros) o una cara (en el caso de hexaedros). La figura (3.2) ilustra esta situación para el caso de tetraedros. En ella se indica cuales son los elementos que forman el parche de recuperación $\omega(A)$. Se observa que si bien el elemento $e_7 = \{2, 3, 4, 5\}$ no posee al nodo 11 en su lista de conectividades, el mismo es nodo irregular de una de sus aristas. Luego, el elemento e_7 también contribuye en el cálculo del gradiente mejorado en la siguiente forma: a) en primer lugar se deben determinar cuáles son los nodos “padres” que definen a la arista sobre la cual se encuentra el nodo irregular A , b) luego se deben calcular los gradientes recuperados en dichos vértices (nodos 3 y 4 de la figura 3.2), c) finalmente el gradiente recuperado en el nodo A se obtiene como el promedio de los gradientes recuperados en los nodos “padres” de la arista

La estimación del error realizada de esta manera es independiente de la clase de problema que se resuelve, es decir, en ella no interviene información sobre la ecuación que gobierna el problema.

Una vez que se obtienen los valores nodales del gradiente recuperado, se procede a inter-

polar los mismos en el interior de cada elemento haciendo uso de las funciones de forma elementales utilizadas en la aproximación por elementos finitos. Finalmente se calcula la contribución al error dada por el miembro derecho de la ecuación (3.7) de la siguiente manera

$$\| Gu - \nabla u^h \|_{L_2(\Omega_K)}^2 = \int_{\Omega_K} (Gu - \nabla u^h)^T \cdot (Gu - \nabla u^h) d\Omega \quad (3.10)$$

$$\begin{aligned} \| Gu - \nabla u^h \|_{L_2(\Omega_K)}^2 = & \underbrace{\int_{\Omega_K} Gu^T \cdot Gu d\Omega}_{I_1} - 2 \underbrace{\int_{\Omega_K} Gu^T \cdot \nabla u^h d\Omega}_{I_2} + \dots \\ & \dots + \underbrace{\int_{\Omega_K} (\nabla u^h)^T \cdot \nabla u^h d\Omega}_{I_3} \end{aligned} \quad (3.11)$$

Si se indica con $N_n(\mathbf{x})$ a la función de forma asociada al nodo n del elemento K y con Gu_n al gradiente recuperado evaluado en el mismo nodo, y teniendo en cuenta que para un punto \mathbf{x} cualquiera de dicho elemento es válida la siguiente expresión

$$Gu(\mathbf{x}) = \sum_{n=1}^{nbe} Gu_n N_n(\mathbf{x}) \quad (3.12)$$

Luego la integral I_1 se puede expresar como

$$\int_{\Omega_K} Gu^T \cdot Gu d\Omega = \int_{\Omega_K} \left[\sum_n Gu_n^T N_n(\mathbf{x}) \right] \cdot \left[\sum_m Gu_m N_m(\mathbf{x}) \right] d\Omega \quad (3.13)$$

$$= \sum_n \sum_m Gu_n^T \cdot Gu_m \int_{\Omega_K} N_n(\mathbf{x}) N_m(\mathbf{x}) d\Omega \quad (3.14)$$

donde las sumatorias en n y m se extienden a los nbe nodos del elemento K . Asumiendo que los elementos finitos son tetraedros lineales ($nbe=4$) y aplicando integración de Gauss-Legendre con un único punto de Gauss al término $\int_{\Omega_K} N_n(\mathbf{x}) N_m(\mathbf{x}) d\Omega$ se tiene que

$$I_1 = \frac{1}{16} V_K \left\{ \sum_{n=1}^4 \sum_{m=1}^4 Gu_n^T \cdot Gu_m \right\} \quad (3.15)$$

en la que V_K indica el volumen del elemento K . Razonando de manera similar para las integrales de I_2 e I_3 se llega a las siguientes expresiones

$$I_2 = \frac{V_K}{2} \left\{ \sum_{n=1}^4 Gu_n^T \right\} \cdot \nabla u^h|_K \quad (3.16)$$

$$I_3 = V_K (\nabla u^h)^T \cdot \nabla u^h|_K \quad (3.17)$$

Un análisis de este estimador de error aplicado a problemas de advección-difusión dominados por el término advectivo es presentado por Volker[Vol00], donde se lo compara con otros métodos. En ése análisis se menciona que el mismo no resulta totalmente satisfactorio para refinar adaptativamente la grilla en problemas advectivos puros. En general se observa que el estimador de Zienkiewicz y Zhu dado por las ecuaciones (3.7) y (3.9) no introduce suficiente refinamiento en la malla. Sin embargo, en la práctica (ver sección 7.2) se observa que la selección de elementos a refinar usando este estimador presenta un mejor comportamiento que el exhibido por el indicador de error basado en el gradiente dado por Ec.(3.2).

- Otra técnica para recuperar el gradiente consiste en aplicar **mínimos cuadrados** para ajustar un polinomio de alto orden, restringido al parche de elementos $\omega(A)$. Los puntos de muestreo que se utilizan en el problema de minimización son puntos particulares en los cuales se demuestra que el gradiente posee una precisión excepcional. Esto quiere decir que en esos puntos se demuestra que el gradiente de bajo orden posee una precisión y un orden de convergencia mayor que en cualquier otro lugar del elemento. Estos puntos son llamados puntos de superconvergencia. La combinación de este ajuste por mínimos cuadrados en los puntos superconvergentes para reconstruir el gradiente da origen a la técnica conocida como **parche de recuperación superconvergente** (SPR). Sin embargo, tal como se menciona en [PE01], no se tiene evidencia acerca de la existencia de esta propiedad de superconvergencia en mallas no estructuradas tridimensionales. Estudios previos[BPR92] muestran que la propiedad de superconvergencia es sensible a la regularidad local de la solución, a la distorsión de los elementos y a la topología de la malla. Todos los resultados obtenidos con este método corresponden a ejemplos 1-D y 2-D simples sobre mallas regulares.

El problema de estimar en forma adecuada el error cometido en la aproximación por elementos finitos para ecuaciones en derivadas parciales hiperbólicas aún se encuentra sin resolver. Sin embargo, dado que resulta necesario contar con una herramienta para seleccionar los elementos a refinar y por las razones expuestas respecto del procedimiento de recuperación superconvergente, sólo se implementan el indicador de error basado en el gradiente y el estimador de error basado en la recuperación del gradiente de Zienkiewicz y Zhu que utiliza la recuperación por promediado.

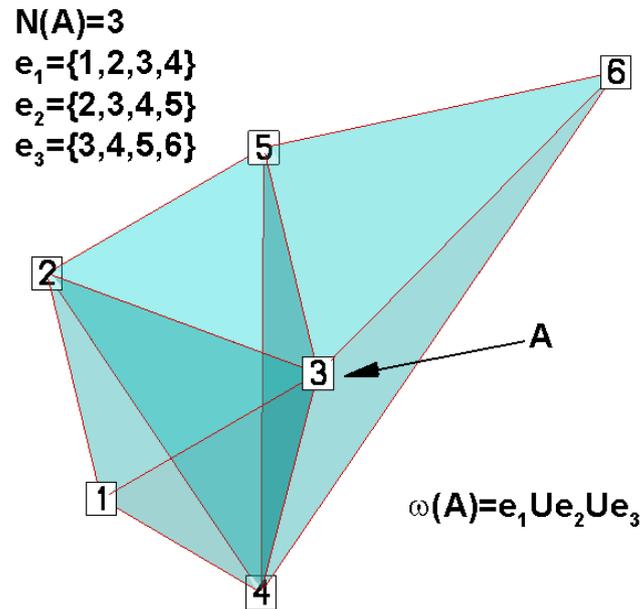


Figura 3.1: Recuperación del gradiente por promediado - Parche de elementos cuando el nodo A no es irregular.

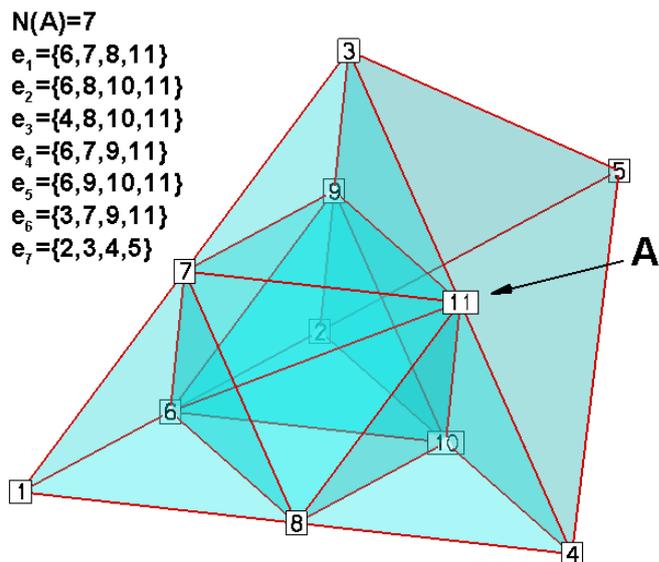


Figura 3.2: Reconstrucción del gradiente por promediado - Parche de elementos cuando el nodo A es irregular.

Capítulo 4

Ecuaciones de flujo compresible y el método SUPG

El presente capítulo tiene por objeto describir las ecuaciones que gobiernan el tipo de problemas resueltos con la estrategia de adaptación de mallas desarrollada en esta tesis. La dependencia de la estrategia de adaptación en el tipo de problema se manifiesta a través del procedimiento de estimación de error utilizado, tal como se menciona en el capítulo 3. Por lo tanto, si bien en principio el *solver* de las ecuaciones del flujo ha sido utilizado sin introducir modificaciones, resulta necesario describir algunos aspectos del mismo. Para mayores detalles acerca de la implementación se remite al lector al manual de PETSc-FEM[SNP⁺08].

Se presenta el sistema de ecuaciones que gobierna el flujo compresible de fluidos no viscosos. Luego se describe la formulación *Streamline Upwind / Petrov Galerkin* desarrollada por Brooks y Hughes[BH82] implementada en el código paralelo multifísica PETSc-FEM para la resolución de sistemas de ecuaciones del tipo advección-difusión utilizando el método de elementos finitos, así como el tratamiento de discontinuidades a través de técnicas de *shock capturing*. La formulación SUPG permite resolver los problemas de estabilización que presenta la formulación estándar de Galerkin cuando es aplicada a problemas donde el término advectivo es predominante sobre el término difusivo.

4.1. Ecuaciones de gobierno para flujos no viscosos, no estacionarios y compresibles

El sistema de ecuaciones que describe la conservación de la masa, cantidad de movimiento y energía para el escurrimiento de flujos compresibles, no estacionarios y no viscosos es conocido con el nombre de ecuaciones de Euler. Dicho sistema puede ser escrito en forma conservativa de la siguiente manera

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (4.1)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}) = \rho \mathbf{f}_e, \quad (4.2)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho \mathbf{u} H) = \rho \mathbf{f}_e \cdot \mathbf{u} \quad (4.3)$$

donde ρ , $\mathbf{u} = (u, v, w)^T$, p , E , H son respectivamente la densidad, el vector velocidad, la presión, la energía total por unidad de masa y la entalpía total del fluido por unidad de masa. \mathbf{f}_e e \mathbf{I} son las fuerzas exteriores que actúan sobre el fluido y la matriz identidad 3×3 . Este mismo sistema puede escribirse en forma conservativa y utilizando notación vectorial como

$$\frac{\partial U}{\partial t} + \nabla \cdot F = Q \quad (4.4)$$

tal que

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ \varepsilon \end{bmatrix}, \quad F = \begin{bmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I} \\ \rho \mathbf{u} H \end{bmatrix}, \quad Q = \begin{bmatrix} 0 \\ \rho \mathbf{f}_e \\ \rho \mathbf{f}_e \cdot \mathbf{u} \end{bmatrix} \quad (4.5)$$

en la cual U es el vector de “variables conservativas”, F es el vector de “flujos convectivos” y Q el vector de “términos fuentes”. Se indica con ε a la energía total del flujo por unidad de volumen. Expandiendo al vector de flujos convectivos,

$$F = [\mathbf{f} | \mathbf{g} | \mathbf{h}] \quad (4.6)$$

donde los vectores columnas \mathbf{f} , \mathbf{g} y \mathbf{h} están dados por

$$f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix}, \quad g = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{bmatrix}, \quad h = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wH \end{bmatrix} \quad (4.7)$$

Asumiendo que el fluido es un gas perfecto dichos flujos tienen la propiedad de ser funciones homogéneas de grado 1 del vector de variables conservativas U [Hir90]. Además, el estado termodinámico para un gas perfecto está dado por la siguiente ecuación de estado,

$$\frac{p}{\rho} = \mathfrak{R}_{gas} T \quad (4.8)$$

donde T es la temperatura del gas y \mathfrak{R}_{gas} es la constante para el gas considerado. El sistema de ecuaciones (4.4) representa un sistema puramente advectivo al no existir ningún mecanismo de difusión (la viscosidad del fluido es nula). Introduciendo las matrices Jacobianas A_i para los

flujos convectivos

$$A_i = \frac{\partial F_i}{\partial U}, \quad i = 1, 2, 3 \quad (4.9)$$

el sistema de Ecs.(4.4) se puede reescribir en forma cuasilineal

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = Q \quad (4.10)$$

donde el hiper-vector \mathbf{A} esta constituido por las tres matrices Jacobianas A_i [Hir90, Lan98]

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -u^2 + \frac{\gamma-1}{2}\mathbf{u}^2 & (3-\gamma)u & -(\gamma-1)v & -(\gamma-1)w & \gamma-1 \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ -u[\gamma E - (\gamma-1)\mathbf{u}^2] & \gamma E - \frac{\gamma-1}{2}(\mathbf{u}^2 + 2u^2) & -(\gamma-1)uv & -(\gamma-1)uw & \gamma u \end{bmatrix} \quad (4.11)$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ -v^2 + \frac{\gamma-1}{2}\mathbf{u}^2 & -(\gamma-1)u & (3-\gamma)v & -(\gamma-1)w & \gamma-1 \\ -vw & 0 & w & v & 0 \\ -v[\gamma E - (\gamma-1)\mathbf{u}^2] & -(\gamma-1)uv & \gamma E - \frac{\gamma-1}{2}(\mathbf{u}^2 + 2v^2) & -(\gamma-1)vw & \gamma v \end{bmatrix} \quad (4.12)$$

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -uw & w & 0 & u & 0 \\ -vw & 0 & w & v & 0 \\ -w + \frac{\gamma-1}{2}\mathbf{u}^2 & -(\gamma-1)u & -(\gamma-1)v & (3-\gamma)w & \gamma-1 \\ -w[\gamma E - (\gamma-1)\mathbf{u}^2] & -(\gamma-1)uw & -(\gamma-1)vw & \gamma E - \frac{\gamma-1}{2}(\mathbf{u}^2 + 2w^2) & \gamma w \end{bmatrix} \quad (4.13)$$

Se sabe que la formulación de elementos finitos estándar de Galerkin aplicada a la resolución del sistema de ecuaciones (4.10) produce un esquema numérico que presenta oscilaciones espurias en la solución cuando hay presentes discontinuidades tipo saltos. Para evitar dichas oscilaciones, se debe recurrir a la estabilización de las ecuaciones discretas. El esquema de solución para el sistema de ecuaciones de advección-difusión con término advectivo dominante que implementa PETSc-FEM es el método estabilizado conocido como SUPG (*Streamline Upwind Petrov Galerkin*) desarrollado por Brooks y Hughes[BH80, BH82] que introduce difusión a lo largo de las líneas de corriente.

4.2. Método de elementos finitos estabilizado SUPG para el sistema de ecuaciones de Euler

4.2.1. Formulación estándar de Galerkin

A modo de introducción, en esta sección se describe brevemente el método de elementos finitos que se obtiene a partir de aplicar la formulación estándar de Galerkin a la ecuación de advección (4.4). Luego, dicha formulación será utilizada para introducir la formulación estabilizada SUPG. Para una mayor comprensión acerca de la formulación de Galerkin estándar aplicada a la ecuación de advección-difusión se remite al lector al libro de Claes Johnson[Joh95].

Por simplicidad en la presentación se asume que $U(\mathbf{x}) = 0 \forall \mathbf{x} \in \Gamma_D^{in}$ en el contorno Dirichlet donde el flujo es entrante y que no se aplican condiciones de borde del tipo Neumann. Luego la forma débil de la formulación estándar de Galerkin se lee de la siguiente manera: encontrar la función vectorial $U \in L_2[0, T] \times \mathcal{W}$ tal que se satisface la condición inicial $U(\mathbf{x}, 0) = U_0(\mathbf{x})$ y además

$$\left(V, \frac{\partial U}{\partial t} \right) + a(U, V) - l(V) = 0 \quad \forall V \in \mathcal{W} \quad (4.14)$$

donde

- Ω es la región del espacio en la que se define el problema y $[0, T]$ es el intervalo de tiempo para el que se resuelve el problema.
- $L_2(\Omega)$ es el espacio de funciones que son cuadrado integrables en Ω .
- $L_2[0, T] \times \mathcal{W}$ es el espacio de funciones vectoriales U tales que para cada instante $t \in [0, T]$ se tiene que $U \in \mathcal{W}$ y la $\|U\|_{\mathcal{W}} \in L_2[0, T]$.
- $\mathcal{W} = H_0^1(\Omega)$ es el espacio de funciones vectoriales V tales que cada una de sus m funciones componentes V_i y sus correspondientes derivadas primeras pertenecen a $L_2(\Omega)$ y donde todas componentes de V se anulan en los tramos del contorno con condiciones de borde del tipo Dirichlet en los cuales el flujo es entrante (Γ_D^{in}), es decir

$$\mathcal{W} = \{V : V(\mathbf{x}) \in H^1(\Omega) \mid V(\mathbf{x}) = 0 \text{ para } \mathbf{x} \in \Gamma_D^{in}\} \quad (4.15)$$

$$H^1(\Omega) = \{V : V_i(\mathbf{x}) \in L_2(\Omega) \mid \frac{\partial V_i}{\partial x_j} \in L_2(\Omega), i = 1, \dots, m \text{ y } j = 1, \dots, n_{sd}\} \quad (4.16)$$

indicando con n_{sd} al número de dimensiones espaciales. Observar que el hecho de prescribir condiciones de borde tipo Dirichlet homogéneas para la solución U permite utilizar el

mismo espacio de funciones \mathcal{W} tanto para las funciones de peso como para las soluciones de prueba.

Se ha utilizado la siguiente notación compacta en la Ec.(4.14)

$$\left(V, \frac{\partial U}{\partial t} \right) = \int_{\Omega} V^T \frac{\partial U}{\partial t} d\Omega \quad (4.17)$$

$$a(U, V) = \int_{\Omega} V^T \frac{\partial F_i}{\partial x_i} d\Omega \quad (4.18)$$

$$l(V) = \int_{\Omega} V^T Q d\Omega \quad (4.19)$$

Luego, la discretización $\{\Omega^e\}_{e=1}^{Nel}$ del dominio Ω en Nel elementos introduce el siguiente problema de elementos finitos: encontrar $U_h \in L_2[0, T] \times \mathcal{W}_h$ tal que

$$\left(V_h, \frac{\partial U_h}{\partial t} \right) + a(U_h, V_h) - l(V_h) = 0 \quad \forall V_h \in \mathcal{W}_h \quad (4.20)$$

siendo \mathcal{W}_h el espacio de elementos finitos que aproxima a \mathcal{W} ($\mathcal{W}_h \subset \mathcal{W}$)

$$\mathcal{W}_h = \{V : V(\mathbf{x}) \in H^1(\Omega) \mid V|_{\Omega^e} \in \mathcal{P}_k(\Omega^e) \forall e \text{ y } V(\mathbf{x}) = 0 \text{ para } \mathbf{x} \in \Gamma_D^{in}\}$$

donde $\mathcal{P}_m(\Omega^e)$ es el espacio de polinomios de hasta grado k restringidos al elemento Ω^e . Tal como se menciona en la introducción de este capítulo, la ecuación (4.20) correspondiente a la formulación estándar de Galerkin de elementos finitos para sistemas advectivos produce oscilaciones espúrias en la solución cerca de las discontinuidades (ver Cap.2 - Ref.[DH03] ó Cap.9 - Ref.[Joh95] para una explicación detallada).

4.2.2. Estabilización SUPG

Brooks y Hughes[BH82] proponen una formulación estabilizada *consistente* para el problema de advección-difusión. La misma consiste en agregar en el miembro izquierdo de la ecuación (4.20) un término de la forma

$$r(U_h, V_h) = \sum_{e=1}^{Nel} \int_{\Omega^e} \mathcal{P}_{SUPG}^e(V_h)^T \tau_{SUPG}^e \mathcal{R}_{SUPG}^e(U_h) d\Omega \quad (4.21)$$

definiéndose los siguientes términos,

- τ^e es una matriz de tamaño $n \times n$, siendo n el número de incógnitas. Los elementos de dicha matriz tienen dimensión de tiempo, por lo cual se lo suele llamar el *tiempo característico* o *intrínseco* del elemento, definido como

$$\tau_{SUPG}^e = \frac{h^e}{2 \cdot \|\mathbf{A}\|} \quad (4.22)$$

donde h^e es una medida del tamaño del elemento y $\|\mathbf{A}\|$ es alguna norma del hiper-vector de las matrices Jacobianas de los flujos convectivos, la cual puede calcularse como

$$\|\mathbf{A}\| = \max_{i=1,\dots,3} \left(\max_{k=1,\dots,n_{D.O.F.}} |\lambda_k^i| \right) \quad (4.23)$$

siendo λ_k^i el k -ésimo autovalor de la i -ésima componente (matriz Jacobiana A_i de flujos convectivos) del hiper-vector \mathbf{A} . Esta es la máxima velocidad de propagación de la información (perturbaciones) en el flujo, que para el caso del sistema de ecuaciones de Euler se corresponde con la velocidad de propagación de las ondas de presión en la dirección del flujo.

- $\mathcal{P}_{SUPG}^e(V_h)$ es un operador aplicado a la función de prueba V_h que posee la forma

$$\mathcal{P}_{SUPG}^e = A_i \frac{\partial V_h}{\partial x_i} \quad (4.24)$$

- $\mathcal{R}_{SUPG}^e(U_h)$ es el residuo de la ecuación diferencial resuelta

$$\mathcal{R}_{SUPG}^e(U_h) = \frac{\partial U_h}{\partial t} + \frac{\partial}{\partial x_i} (A_i U_h) - Q \quad (4.25)$$

Se observa que el término que se adiciona a la formulación estándar de Galerkin es una contribución elemento a elemento (es decir que se evalúa en el interior de cada elemento) cuya magnitud depende del residuo local de la Ec.(4.4). Este término residual es el que introduce estabilización ya que si se expande la Ec.(4.21)

$$r(U_h, V_h) = \sum_{e=1}^{Nel} \int_{\Omega^e} \left(A_i \frac{\partial V_h}{\partial x_i}, \tau_{SUPG}^e \frac{\partial U_h}{\partial t} \right) + \underbrace{\left(A_i \frac{\partial V_h}{\partial x_i}, \tau_{SUPG}^e A_i \frac{\partial U_h}{\partial x_i} \right)}_{(*)} - \left(A_i \frac{\partial V_h}{\partial x_i}, \tau_{SUPG}^e Q \right) d\Omega \quad (4.26)$$

se puede apreciar que el término indicado con (*) es el que agrega difusión a lo largo de las líneas de corriente, siempre que la matriz de estabilización τ_{SUPG}^e esté diseñada adecuadamente [DH03].

4.2.3. Tratamiento numérico de los choques (*shock capturing*) en el contexto del método SUPG

La estabilización introducida a la formulación estándar de Galerkin dada por el término (4.26) no es suficiente para eliminar las oscilaciones espúrias que aparecen en las proximidades de la solución cuando existen gradientes muy grandes. Esto es así porque el esquema SUPG es un esquema lineal de alto orden de precisión, y por el teorema de Godunov el mismo no puede ser monótono [DH03]. Por lo tanto, aparecerán oscilaciones en la vecindad de las discontinuidades.

Teorema de Godunov: el mismo afirma que los esquemas numéricos lineales para resolver ecuaciones en derivadas parciales que tengan la propiedad de ser monótonos, a lo sumo pueden ser de primer orden de precisión.

Para evitar las oscilaciones, se recurre al agregado de un término de *shock capturing* que introduce una difusión no lineal, *consistente* con la formulación original, además del término de estabilización estándar que introduce la formulación SUPG. El término de *shock capturing* puede ser diseñado para que la difusión que introduzca sea del tipo isotrópica o anisotrópica. En el primer caso, la difusión no lineal es introducida en todas las direcciones mientras que en el caso anisotrópico, la difusión no lineal es introducida solamente en la dirección transversal al flujo. El objetivo de esto último es que la difusión introducida por el término de *shock capturing* no afecte a la difusión en la dirección de las líneas de corriente introducida por el término SUPG. El término de *shock capturing* isotrópico tiene la misma estructura que el término SUPG estándar dado por la ecuación (4.21),

$$\sum_{e=1}^{Nel} \int_{\Omega^e} \mathcal{P}_{sc}^e(V_h)^T \tau_{sc}^e \mathcal{R}_{SUPG}^e(U_h) d\Omega \quad (4.27)$$

Hughes y Mallet[HM86] describen el tratamiento para la matriz de estabilización que controla el término de *shock capturing* τ_{sc}^e , la cual depende de la definición de la matriz de estabilización τ_{SUPG}^e . Por otra parte, Codina[Cod93] presenta el tratamiento del término de *shock capturing* en el caso anisotrópico.

Capítulo 5

Implementación de la estrategia de adaptación

Se discuten algunos aspectos relacionados con el desarrollo del código de adaptación de mallas, el cual fue realizado en dos etapas. De esta manera, se obtiene primero un código “conceptual” en Octave[Eat07] y luego uno de similares características y funcionalidad en C++. Dividir el desarrollo del trabajo de esta manera tiene como objetivo acelerar la implementación y prueba de ideas dejando de lado aspectos de programación complicados que afectan a lenguajes como C++. Por otra parte, la segunda etapa de trabajo tiene como objetivo lograr una herramienta que sea más *eficiente* en cuanto al manejo de la memoria y reducir el tiempo de ejecución. El principal inconveniente que se encuentra para la implementación realizada en Octave, en lo que se refiere al uso de la memoria, es que dicho lenguaje representa internamente a un número entero como un número real (8 bytes). Teniendo en cuenta que para la implementación en Octave todas las estructuras de datos son representadas a través de arreglos y que a excepción del arreglo de coordenadas de los vértices de la malla el resto de los arreglos deberían contener números enteros, se observa que los recursos de memoria disponibles son desaprovechados. Esto hace que el código esté limitado en relación al tamaño de la malla que permite adaptar. Además, ya que Octave es un lenguaje interpretado, también existe una penalización relacionada con la velocidad de ejecución. Por lo tanto, la implementación en Octave permite adaptar mallas de hasta un millón y medio de tetraedros en una PC que cuente con 1Gb de memoria RAM, y ha sido utilizado para resolver los problemas de flujo compresible que se presentan en las secciones 7.1 (estacionario) y 7.3 (no estacionario).

Un aspecto a tener en cuenta en el desarrollo de códigos de adaptación de mallas son las estructuras de datos que se utilizan para representar las entidades geométricas de la malla y sus adyacencias. Determinar cuál es la estructura de datos óptima es una tarea compleja que no fue considerada en el desarrollo de esta tesis. Aquí sólo se presentan las estructuras de datos que fueron utilizadas en el momento de la implementación y se justifica el motivo por el cual se las eligió, si bien es claro que este es un aspecto del trabajo que debe ser revisado a futuro.

5.1. Consideraciones previas

Siguiendo el trabajo de Remacle[RKS], se definen algunas cuestiones básicas relacionadas con la representación de mallas. Una malla consiste en la discretización de un dominio geométrico mediante entidades de malla que poseen una topología simple (triángulo, cuadrángulo, tetraedro, etc.). Las entidades topológicas básicas en el espacio tridimensional son los vértices (objetos 0-D), aristas (objetos 1-D), caras (objetos 2-D) y elementos (objetos 3-D).

Por otro lado, al vínculo entre las entidades topológicas de distinta dimensión se lo denomina *adyacencias*. Las adyacencias describen la topología de la malla y para una entidad de malla se las puede clasificar de dos maneras: adyacencias *hacia arriba* y adyacencias *hacia abajo*. Las adyacencias hacia arriba son conjuntos no ordenados en el siguiente sentido: por ejemplo, no tiene importancia el orden de los elementos que comparten a una arista o de las caras que comparten a un nodo. Una característica importante de este tipo de adyacencias es que su tamaño es variable y a priori desconocido, y que además no tiene sentido acceder a ellas en forma aleatoria. A diferencia de ellas, las adyacencias hacia abajo sí poseen un tamaño conocido (ejemplo: las aristas que constituyen a un elemento o las aristas de una cara) y resulta conveniente acceder a las mismas en forma aleatoria¹.

De esta manera se observa que según cual sea el tipo de adyacencia que se desea representar, resulta más adecuada la utilización de distintas estructuras de datos para representarlas. En esta tesis, se han utilizado los contenedores que facilita la librería de plantillas estándar STL (*Standard Template Libraries*)[Sil06] para representar estas estructuras de datos. Por ejemplo, para representar las adyacencias caras-aristas (hacia *abajo*) se observa que resulta conveniente utilizar contenedores de acceso aleatorio (*random access*), tales como `vector<>`. Por otro lado, en el caso de requerirse las adyacencias nodos-cara (hacia *arriba*) se cree conveniente utilizar otro tipo de contenedores, como los `multimaps`.

La STL es una librería de *clases* de contenedores, algoritmos e *iterators* escrita en C++. Es una librería *genérica* en el sentido de que la mayor parte de sus componentes son *plantillas* de C++ (para una introducción al lenguaje C++, la programación orientada a objetos, funciones y clases plantillas se recomiendan los libros de Bruce Eckel[Eck00, Eck04]). La elección de la librería STL se fundamenta en el hecho de que la misma proporciona muchos de los algoritmos y estructuras de datos requeridos en este tipo de aplicaciones, cuyas implementaciones poseen una mostrada eficiencia.

¹Con el término *acceso aleatorio* se hace referencia a que el acceso a la adyacencia es una operación $O(1)$, es decir que no se requiere recorrer las adyacencias desde la primera, una por una, hasta llegar a la deseada.

5.2. Implementación de las estructuras de datos para las entidades geométricas de la malla

Las estructuras de datos para las entidades geométricas elementos, caras y aristas son implementadas a través de vectores de objetos de C++. Para la implementación de los vectores se recurre al contenedor `vector<>` de las STL y los objetos son aristas (`edges`), caras (`faces`) y elementos (`elemen`) cuyas clases se declaran de la siguiente manera,

```
#ifndef EDGE_CLASS_H
#define EDGE_CLASS_H
#include<cstdlib>
#include<vector>
class edge{
public:
edge();
~edge();
int middle_node; //nodo medio sobre la arista
std::vector<int>nodes_parent; //nodos de la arista
int parent; //número de la arista madre
std::vector<int>child; //nros de las aristas hijas
int flag; //flag de propiedad
int state; //activa ->1 , inactiva ->0
};#endif // EDGE_CLASS_H ///:~
```

```
#ifndef FACE_CLASS_H
#define FACE_CLASS_H
#include<cstdlib>
#include<vector>
class face{
public:
face();
face(int sz);
~face();
int parent; //cara madre
std::vector<int>child; //caras hijas
int flag; //flag de propiedad
int state; //activa ->1 , inactiva ->0
};
#endif // FACE_CLASS_H ///:~
```

```
#ifndef ELE_CLASS_H
#define ELE_CLASS_H
#include<cstdlib>
#include<vector>
class ele{
public:
```

```

ele();
~ele();
std::vector<int>child; //elementos hijos
int flag; //flag de propiedad
};
#endif // ELE_CLASS_H ///:~

```

Tal como se observa en las declaraciones, la clase `edge` posee los siguientes miembros datos: el entero `middle_node` posee la numeración del nodo medio que surge sobre la arista debido a su refinamiento, el vector de enteros `nodes_parent` guarda la numeración de los pares de nodos que definen a la arista, el entero `parent` posee el número que identifica a la arista *madre*, el vector de enteros `child` guarda la numeración correspondiente a las aristas *hijas*, el miembro `flag` guarda un número entero que sirve para identificar alguna propiedad asignada a la arista y finalmente, el miembro `state` indica si la arista está *activa* o *inactiva* en el sentido que se describe luego en la sección 6.2 para la aplicación de la restricción al refinamiento conocida como criterio de nodo 1-irregular. Las clases `face` y `ele` son similares a la clase `edge`. Luego, con estas declaraciones, las aristas que constituyen la malla son guardadas en un vector de aristas `vector<edge>aristas`, mientras que las caras son guardadas en un vector de caras `vector<face>caras`.

5.3. Implementación de las adyacencias

5.3.1. Adyacencias hacia abajo

Las adyacencias hacia abajo son implementadas utilizando arreglos bidimensionales para permitir el acceso en forma aleatoria. Así, el j -ésimo elemento de la i -ésima fila del arreglo 2D contendrá la información correspondiente a la adyacencia entre la i -ésima entidad geométrica de dimensión K y la j -ésima entidad geométrica de dimensión L , donde $L < K$.

Dado que las STL no proporcionan contenedores para arreglos multidimensionales, se encuentra que una forma de implementarlos es utilizando el contenedor o clase plantilla `vector<>` en forma anidada. De esta manera, si se desea declarar un arreglo bidimensional de `doubles`, entonces se debe proceder de la siguiente manera,

```
std::vector< std::vector<double> > Array;
```

Sin embargo se encuentra que esta implementación presenta un *overhead* significativo en cuanto al manejo de la memoria y que la interfaz resultante puede presentar algunas complicaciones. Por otro lado, si se utilizan arreglos nativos de C/C++, la reserva y liberación del espacio en memoria que requiere el arreglo queda a cargo del usuario, induciendo a errores de programación. Finalmente, si se utiliza una representación pseudo-bidimensional mediante un bloque contiguo de elementos, el manejo de índices para acceder a los elementos del arreglo y la aplicación de los algoritmos de las STL puede volverse una tarea engorrosa para el programador.

De las alternativas que se presentan para reemplazar la implementación `vector<vector<>>` por una más eficiente a la vez que sea sencilla de utilizar y compatible totalmente con los algoritmos y contenedores de las STL, se propone analizar aquella que ofrecen las librerías Boost para arreglos multidimensionales. Boost es un conjunto de librerías *open source* escritas en C++ que se incluye como paquete en distribuciones de Linux como Fedora, Debian o Mandriva. La librería Boost.MultiArray facilita clases plantillas y contenedores para arreglos multidimensionales tal que el diseño de su interfaz sigue los lineamientos que presentan los contenedores de las STL y por ende son compatibles con sus contenedores y algoritmos. Los arreglos proporcionados por Boost.MultiArray presentan un manejo dinámico de la memoria, pueden ser accedidos utilizando la sintaxis de los arreglos nativos de C/C++ y permiten modificar el tamaño (`resize`), la forma (`reshape`) o utilizar “secciones” del arreglo (`views`). La clase plantilla está definida en el archivo de cabecera "boost/multi_array.hpp". Para mayores detalles se remite al lector al manual de la librería [Boo07].

Dado que sólo se van a utilizar arreglos bidimensionales en el código de adaptación de la malla, a los fines de facilitar la escritura y comprensión del código se definen en el archivo de cabecera `my_typedefs.hpp` los siguientes tipos

```
#ifndef MY_TYPEDEFS_H
#define MY_TYPEDEFS_H
#include"multi_array.hpp"
//Arreglos 2-D de la libreria boost
typedef boost::multi_array<int,2> Arr2Di;
typedef boost::multi_array<double,2> Arr2Dd;
#endif // MY_TYPEDEFS_H ///:~
```

De esta manera, un arreglo bidimensional de enteros se declara como

```
Arr2Di my_array(boost::extents[extent1][extent2]);
```

Se realizó una prueba numérica para evaluar y comparar los requerimientos del espacio en memoria que se reserva para arreglos bidimensionales de enteros, implementados como `vector<vector<int>>`, como `Arr2Di` de Boost.MultiArray y como un vector lineal `vector<int>`. La prueba indica que un arreglo 2-D de enteros de 20 millones de filas y 5 columnas requiere en el primer caso 734Mbytes, en el segundo 420Mbytes y en el tercero 393Mbytes. Teniendo en cuenta que teóricamente, para enteros de 4 bytes se requerirían $2.0e+7 * 5 * 4 / 2^20 = 381.47$ Mbytes, entonces se observa que la implementación de Boost.MultiArray requiere un 10 % más de memoria que el valor teórico y un 7 % más que la implementación lineal. Observar que la implementación STL de `vector<vector<int>>` demanda un 92 % más de memoria que el valor teórico!!!.

Las adyacencias hacia abajo utilizadas en la representación de la malla son:

- Elementos-vértices: utilizada por ambos el procedimiento de adaptación de la malla y el *solver* de elementos finitos.

- Elementos-aristas: requerida en la función de refinamiento para:
 1. Aplicar la restricción al refinamiento donde:
 - a) Se construye una lista con las aristas a refinar a partir de la lista de elementos a refinar.
 - b) Se construyen las adyacencias arista-elementos mediante un `multimap<int, int>Edges_ele`.
 - c) Se indican las aristas que permanecen activas / inactivas después del refinamiento.
 2. Obtener la diagonal más corta en el particionamiento del octaedro interior.
- Elementos-caras: utilizada en la función de refinamiento para aplicar la restricción al refinamiento donde se requiere:
 1. Armar una lista de caras a refinar dada la lista de elementos a refinar.
 2. Construir las adyacencias caras-elementos utilizando un `multimap<int, int>Faces_ele`.
 3. Indicar las caras que permanecen activas / inactivas luego del refinamiento.
- Caras-aristas: utilizada en la función de refinamiento para la aplicación de la extensión al criterio 1-irregular ya que, en el caso que existan elementos con caras triangulares, se requiere contar con información sobre las adyacencias arista-caras, para lo cual se construye el `multimap<int, int>Edges_face` sólo para las caras triangulares que aparecen en el centro de las caras refinadas.
- Aristas-vértices: esta información de adyacencias es guardada en la misma definición de la entidad arista (ver clase `edge`).
- Caras-vértices: se requiere sólo la primera vez que se adapta la malla base para:
 1. Generar la lista de caras y asignarle una única *identidad* a cada una de ellas.
 2. Obtener la lista de caras que pertenecen al contorno del dominio geométrico para asignar condiciones de borde a un subconjunto de ellas a partir de una lista de vértices sobre el contorno.

Además, se utilizan tablas que describen la relación local o el orden local para las adyacencias hacia abajo de una misma entidad de malla. La primera de estas tablas describe todas las aristas de un elemento en función de sus vértices. Para tetraedros y utilizando base cero para la numeración local de los vértices,

$$ind_local_edges = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 2 \\ 1 & 3 & 3 & 2 & 2 & 3 \end{bmatrix}^T \quad (5.1)$$

La segunda tabla describe todas las caras de un elemento en función de sus vértices,

$$ind_local_faces = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 2 & 3 \\ 0 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix} \quad (5.2)$$

De esta manera, $ind_local_edges[i][0]$ e $ind_local_edges[i][1]$ da la numeración de los vértices de la i -ésima arista del tetraedro, mientras que $ind_local_faces[i][0]$, $ind_local_faces[i][1]$ y $ind_local_faces[i][2]$ proporciona la numeración de los vértices de la i -ésima cara del tetraedro. La figura 5.1 ilustra la numeración local de los vértices, las aristas y caras, utilizando números romanos para las aristas y las caras.

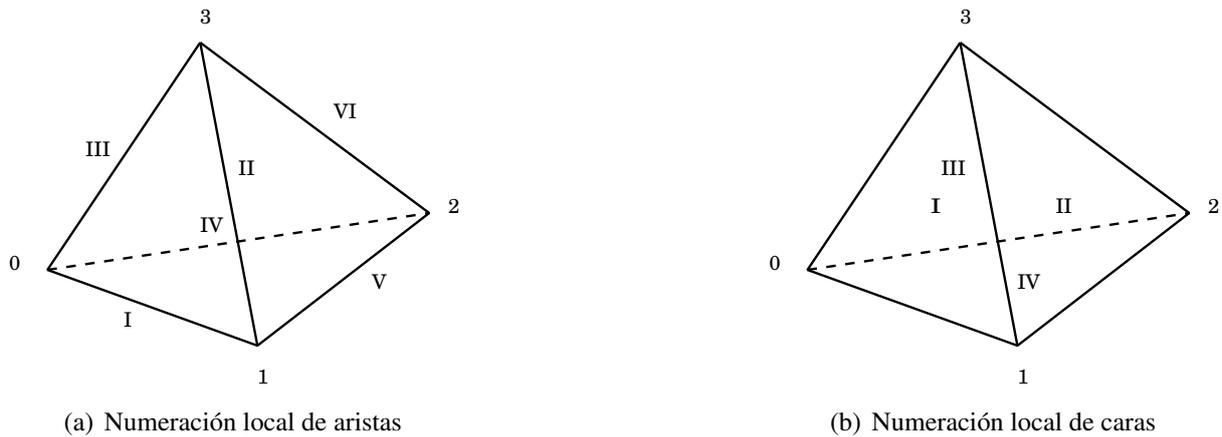


Figura 5.1: Numeración local de aristas y caras para tetraedros.

De la misma manera, la numeración local de las aristas y caras para hexaedros se presentan en la figura 5.2. En este caso, las tablas que describen el orden local de las mismas son las siguientes,

$$ind_local_edges = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 & 5 & 6 & 7 & 4 & 4 & 5 & 6 & 7 \end{bmatrix}^T \quad (5.3)$$

$$ind_local_faces = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 0 & 1 & 5 & 4 \\ 1 & 2 & 6 & 5 \\ 2 & 3 & 7 & 6 \\ 3 & 0 & 4 & 7 \end{bmatrix} \quad (5.4)$$

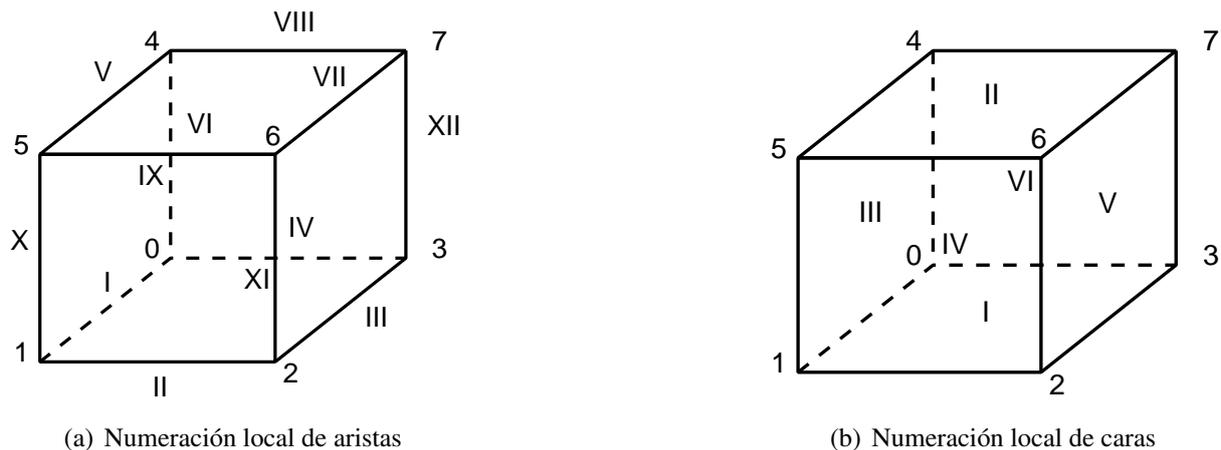


Figura 5.2: Numeración local de aristas y caras para hexaedros.

5.3.2. Adyacencias hacia arriba

La información de las adyacencias hacia arriba es requerida en varias de las etapas que integran al procedimiento de adaptación. Por ejemplo: en la función de refinamiento, a los fines de poder aplicar el criterio de nodo 1-irregular (ver sección 6.2), se requiere contar con la información correspondiente a las adyacencias caras-elementos y aristas-elementos. Para implementarlas se utiliza el contenedor asociativo múltiple ordenado `multimap` de las STL, siendo la clave el *id* de la entidad geométrica de menor dimensión y el valor el *id* de la entidad geométrica de mayor dimensión. De esta manera se tienen las declaraciones de las siguiente adyacencias,

```
multimap<int,int> Edges_ele;
multimap<int,int> Faces_ele;
```

El motivo por el cual se utiliza el contenedor `multimap` de las STL es que precisamente el número de elementos que comparten a una arista o cara es desconocido a priori. Por lo tanto, se requiere asociar un valor a una clave, admitiendo la posibilidad de que la clave se encuentre repetida un número indefinido de veces. Ya que luego se realiza la búsqueda de claves con las funciones miembros `find()`, `lower_bound()`, `upper_bound()` ó `equal_range()` de la clase `multimap`, se garantiza que la complejidad algorítmica de dichas operaciones es en el peor caso logarítmica $O(\log(N))$ [Sil06].

En las funciones que asignan flags de propiedades a aristas y caras a partir de una lista de nodos, también es requerida la información sobre las adyacencias nodos-caras y nodos-aristas. Esto es así porque se considera que una arista o cara posee una determinada propiedad sólo si todos los nodos que la forman pertenecen a la lista de nodos ingresada como argumento a la función de asignación de flags (ver sección 6.4). Por ejemplo: en el caso de asignar un flag asociado a una condición de borde a caras sobre el contorno a partir de una lista de nodos, es necesario contar con información acerca de cuales son las caras pertenecientes al contorno tal que todos sus nodos se encuentran en la lista de nodos que describen ése tramo del contorno. Un razonamiento similar se puede hacer para un grupo de aristas sobre el contorno si una condición

de borde en particular debe ser aplicada “por aristas”, o bien para un grupo de elemento que posean una determinada propiedad. Dado que la asignación de flags de propiedades se realiza sólo en el primer paso de adaptación, esta información de adyacencias es requerida una sola vez en todo el procedimiento de solución adaptativa.

Finalmente, en el caso de utilizar los procedimientos de estimación del error mediante la recuperación del gradiente presentados en la sección 3.2.1, es necesario contar con las adyacencias nodos-elementos, ya que se requiere saber cuáles son los elementos que conforman el parche de elementos que “comparten” a un nodo dado de la malla.

5.4. Descripción del código

En esta sección se describen brevemente los procedimientos más importantes que constituyen al código de adaptación de mallas. A los fines de facilitar la comprensión por parte del lector, se prefiere utilizar pseudo-código antes que presentar explícitamente al código en lenguaje C++. Sin embargo, se asume que el lector posee algún grado de conocimiento sobre dicho lenguaje, sobre la escritura de archivos Make de GNU, de la implementación MPICH/MPICH2 de la interfaz MPI y sobre el sistema operativo GNU/Linux, ya que ocasionalmente resulta necesario recurrir a ellos para explicar detalles de la implementación (ver manuales de Make de GNU[SMS06] y MPICH2[Arg07]).

El código está compuesto por una función principal llamada `driver()`, la cual gestiona el procedimiento de solución adaptativo del problema, esto es, tanto las operaciones que comprenden la adaptación de la malla como la resolución de las ecuaciones del flujo.

Los datos del problema son definidos en un único archivo `data_NAME.tmp` (donde `NAME` es el nombre del problema que se resuelve). Allí se indican tanto las variables involucradas con la física del problema (longitud de referencia, tiempo final de la simulación o número máximo de pasos de tiempo, densidad del gas, calores específicos a presión y volumen constante C_p y C_v , constante del gas \mathfrak{R}_{Gas} , etc.) como las variables requeridas por el procedimiento de adaptación (frecuencia de adaptación de la malla, método de estimación del error, número máximo de niveles de refinamiento, dimensión geométrica del problema, tipo de elemento utilizado en la discretización del dominio, *flags* de propiedades sobre aristas, caras y elementos, etc).

La función `driver()` se describe mediante el pseudo-código 1. Se observa que la misma está integrada por las siguientes funciones:

- `compute_Dt()`: calcula el tamaño del paso de tiempo aplicando la condición de Courant, Friedrich y Lewy (CFL), utilizando información de la malla y del último estado computado (o el estado inicial cuando se inicia el cálculo de la solución).
- `adapt()`: la misma se describe en el pseudo-código 3, e implementa la estrategia no estacionaria que se describe en la sección 6.

Algorithm 1 Pseudo-código de la función `driver()`.

```
//Lectura de archivo con los datos del problema data_NAME.tmp
//Especificacion de parametros que afectan al solver y al adaptador
int main(int argc, char **argv){

    MPI_Init(&argc,&argv);
    string cmd; char* cstr;

    //Estima el tamaño del paso de tiempo (cond. CFL) para iniciar
    //el calculo a partir de la condicion inicial.
    Dt=compute_Dt(.....);

    //acc_time: tiempo de simulacion acumulado
    //final_time: tiempo final para la simulacion
    //nstep: frecuencia de adaptación

    //Lazo sobre el tiempo
    while(acc_time<=final_time){

        //Incrementa el tiempo acumulado
        acc_time=acc_time+Dt*nstep;
        //Llamada via el sistema operativo al solver PETSc-FEM
        cmd="nohup make run >nohup.log";
        cstr = new char [cmd.size()+1];
        strcpy (cstr, cmd.c_str());
        system(cmd);

        //Adapta la malla
        adapt(.....);

        //Estima el tamaño del paso de tiempo (cond. CFL) para reiniciar
        //el calculo a partir del ultimo estado computado.
        Dt=compute_Dt(.....);
        //Incrementa el paso de tiempo
        step+=nstep;
    }
    MPI_Finalize();
}
```

Se observa que se hace uso de las directivas `MPI_Init()` y `MPI_Finalize()`. Esto no significa que la implementación del código de adaptación se realiza en paralelo. Sólo se recurre a estas directivas de MPI para que la adaptación de la malla, que se realiza en forma secuencial, pueda ser ejecutada en un nodo cualquiera del *cluster* de PC con el objeto de no “cargar” el servidor. Por esta razón, el código de adaptación de mallas es compilado con las librerías de MPICH2 y la ejecución del mismo se invoca mediante el *target driver* del archivo makefile incluido en 2.

El *solver* del flujo se lanza desde la función `driver()` utilizando la función `system()` de C/C++, mediante la cual se invocan las instrucciones especificadas en el archivo makefile correspondiente. La función `system()` forma parte de la librería estándar de C y su léxico es el siguiente,

```
#include<cstdlib>
int system(const char *str);
```

La función `system()` pasa la cadena de caracteres apuntada por *str* como un comando al procesador de comandos del sistema operativo, que en este caso es alguna distribución de GNU/Linux. `system()` no retorna el control a la función `driver` hasta que no se termine de ejecutar el comando al cual se ha invocado, en este caso, la ejecución del *solver* durante `nstep` pasos de tiempo. El comando que se invoca al llamar a `make` es el *target run* del archivo makefile y el uso del comando `nohup` que forma parte de las utilidades del sistema GNU permite redireccionar la salida que muestra el *solver* al archivo `nohup.log`. El *target run* dentro del archivo makefile está especificado por las correspondientes líneas en 2.

Algorithm 2 Líneas del archivo Makefile para ejecutar el *solver* del flujo y la adaptación de la malla.

```
##Adaptacion de la malla
driver:

    $(MPI_HOME)/bin/mpirun -machinefile ./machi.dat -n -1 driver.bin

##Solver del flujo PETSc-FEM
NP:=10
run:

    make-depl $(CASE).epl $(CASE).depl
    time $(MPI_HOME)/bin/mpirun -l -machinefile ./machi.dat -n $(NP)
    $(PROG) -case $(CASE).depl
```

donde la variable `CASE` contiene el nombre del caso (por ejemplo `CASE:=nozzle`), `NP` indica el número de procesos que ejecutan en paralelo al *solver* del flujo cuyo nombre es especificado en la variable `PROG`. PETSc-FEM realiza la lectura de un archivo con extensión `.depl` el cual cuenta con información necesaria para lanzar la corrida (ver manual de PETSc-FEM[SNP⁺08]).

Algorithm 3 Pseudo-código de la función adapt ().

```
void adapt(..., step, Nlevelref, ...){
  //Nlevelref: numero maximo de refinamientos permitidos.
  //eles2ref: lista de elementos a refinar
  //step: paso de adaptacion
  if(step==1){//Si se adapta la malla base por 1ra vez.
    //Lazo sobre los niveles de refinamiento.

    for (int level_ref=0; level_ref<Nlevelref; level_ref++){

      //Selecciona elementos a refinar y los guarda en la
      //lista eles2ref.
      select_eles2ref(..., eles2ref, ...);
      if(!eles2ref.empty()) //Si la lista no esta vacia
        //Adapta la malla
        proc(..., eles2ref, .....);
      else
        //Sale del lazo de refinamiento.
        break;
    }
  }else{
    //Selecciona elementos a refinar sobre la ultima
    //malla con mayor refinamiento.
    select_eles2ref(..., eles2ref, ...);
    if(!eles2ref.empty()){

      //Busqueda de padres en forma recursiva. search_fathers()
      //devuelve en eles2ref la lista a refinar en la malla base.
      search_fathers(..., eles2ref, .....);
      //Lazo sobre los niveles de refinamiento.
      for (int level_ref=0; level_ref<Nlevelref; level_ref++){
        if(level_ref>0)

          //Actualiza lista de elementos a refinar escritas en
          //provisoriamente en archivos por search_fathers().
          update(..., eles2ref, .....);

          //Adapta la malla sobre la lista actualizada.
          proc(..., eles2ref, .....);
        }
      }
    }

    //Renombra archivos para relanzar el calculo de la solucion.
    rename_files(...);
  }
}
```

- `proc()`: en la misma se llevan a cabo las siguientes tareas: generación de las estructuras de datos que representan a las entidades geométricas de la malla y sus adyacencias (sólo la primera vez que se adapta la malla base), asignación de flags identificadores de propiedades sobre aristas, caras y elementos (sólo la primera vez que se adapta la malla base), particionamiento de los elementos, caras y aristas. Luego, en la etapa de pos-refinamiento: se actualizan las condiciones de borde, se proyecta el estado sobre la malla adaptada y se actualiza la lista de nodos irregulares. Finalmente se escribe en archivos los datos requeridos por el solver del flujo para reiniciar el cálculo de la solución (ver pseudo-código 4).
- `search_fathers()`: la misma implementa un algoritmo que permite limitar el número máximo de niveles de refinamiento en la estrategia de adaptación para problemas no estacionarios. Una descripción detallada de la misma y el correspondiente pseudo-código se presentan en la sección 6.3.2.
- `update_eles2ref()`: esta función implementa un algoritmo que permite mapear la numeración de los elementos seleccionados para refinar en la estrategia de adaptación para problemas no estacionarios. La misma se describe en la sección 6.3.2.
- `select_eles2ref()`: selecciona los elementos a refinar aplicando el criterio de selección indicado por el usuario (ver sección 3.2).

Algorithm 4 Pseudo-código de la función `proc()`.

```
void proc(eles2ref,...){  
  
    //inicia: indica si la función se aplica a la malla base  
    //o si se aplica a una malla que ya está adaptada.  
    //step: paso de adaptación.  
    if(inicia==1){ //Si se refina la malla base.  
  
        //Carga coordenadas, conectividades, condiciones de borde, etc.  
        if(step==1){ //Si es el primer paso de adaptación.  
  
            //Genera estructura de datos y adyacencias de la malla base.  
            //Asigna flags de propiedades a entidades geométricas de la malla.  
        }else{  
  
            //Cargo estructura de datos malla base desde archivo.  
        }  
    }  
  
    //Apendiza elementos seleccionados por otros criterios a  
    //la lista de elementos a refinar eles2ref (ver refinamiento  
    //selectivo impuesto por el usuario en la seccion 6.4).  
    //Refinamiento de los elementos en eles2ref.  
    refine(eles2ref,...);  
    //Pos-refinamiento  
    //Actualización de condiciones de borde sobre malla refinada.  
    //Interpolación lineal del estado sobre la malla refinada y  
    //actualización lista de nodos irregulares.  
    //Escritura de archivos para el solver del flujo.  
}
```

Capítulo 6

Esquemas de refinamiento y estrategia de solución adaptativa para problemas no estacionarios

En este capítulo se presentan el esquema de refinamiento implementado en el código de adaptación de mallas, se discuten sus ventajas y desventajas frente a otros esquemas y se describe la forma en que se extiende la restricción al refinamiento conocida como criterio de nodo 1-irregular a mallas de elementos tridimensionales. Luego se presenta la estrategia que se propone para aplicar el procedimiento de adaptación de mallas a problemas cuya solución es dependiente del tiempo. Finalmente, se expone el procedimiento que permite gestionar propiedades sobre las entidades geométricas de la malla. Este último permite el tratamiento de condiciones de borde, el refinamiento en zonas selectivas del dominio computacional, el refinamiento sobre contornos curvos y el tratamiento de propiedades elementales.

6.1. Descripción del refinamiento

La estrategia de refinamiento que se implementa en esta tesis corresponde al tipo de estrategias *basadas en elementos*. En este tipo el criterio de selección se aplica a nivel de elementos. Luego los elementos seleccionados son particionados utilizando patrones preestablecidos para cada tipo de elemento. Esto difiere del tipo de estrategia de refinamiento *basada en aristas*, en las que el criterio de selección se aplica sobre las aristas para luego determinar en función de *reglas de compatibilidad para el refinamiento*, cuál es el patrón de subdivisión que se debe aplicar al elemento. Las reglas de compatibilidad en el refinamiento se utilizan a los fines de restringir los posibles casos de refinamiento que se pueden presentar debido al “marcado” de aristas, de manera que dichos casos se ajusten a los patrones de particionamiento preestablecidos para el elemento en cuestión. Esto es, si se aplica el criterio de selección a aristas, entonces puede resultar que la combinación de aristas marcadas para refinar en un elemento no se corresponda con ninguno de los patrones de refinamiento indicados para ese tipo de elemento. De esta

forma, se deben agregar / quitar aristas de la lista de aristas seleccionadas mediante el criterio de selección[LB92, Mav95].

En el caso de elementos bidimensionales, los patrones de refinamiento consisten en la partición regular 1:4 tanto para triángulos como para cuadrángulos. Estos patrones se observan en las figuras 6.1.a) y 6.1.b). Para triángulos, el particionamiento consiste en agregar nodos en las mitades de las aristas del elemento, los que se unen agregando tres aristas en su interior. A los fines de su representación en la estructura de datos, estas nuevas aristas no son *hijas* de ninguna de las aristas del elemento de partida. De esta manera se cuentan **3+6** aristas nuevas por cada elemento refinado. Para elementos cuadriláteros, el particionamiento comprende además la inserción de un nodo en el centro geométrico del elemento y la unión de éste con los nodos que se insertan en las mitades de las aristas agregando 4 aristas nuevas, con lo cual el número de aristas que se generan por cada elemento refinado es igual a **4+8**¹.

Para los elementos tridimensionales tratados en esta tesis (tetraedros, hexaedros y prismas triangulares) el tipo de particionamiento que se aplica también es el regular o isotrópico 1:8,

- Tetraedros: el refinamiento consiste en la inserción de 6 nodos, uno en la mitad de cada una de las aristas del elemento de origen. Luego, en función de lo expuesto en las secciones 2.2.1 y 2.2.2, se selecciona la diagonal más corta para refinar al octaedro que surge en el interior del tetraedro refinado. De esta manera, por cada tetraedro refinado surgen **13+12** aristas nuevas y **8+16** caras nuevas (figura 6.2.a).
- Hexaedros: la partición regular 1:8 comprende la inserción de 19 nodos nuevos. Los primeros 12 nodos se ubican en las mitades de las aristas del elemento de origen. Los 7 que queda se ubican, uno en el centro geométrico del elemento y los restantes en los centros geométricos de las caras. De esta forma, se agregan **30+24** aristas nuevas y **12+24** caras nuevas por cada elemento refinado, tal como se observa en la figura 6.2.b).
- Prismas triangulares: el refinamiento se obtiene insertando 12 nodos nuevos. 9 de ellos se ubican en las mitades de las aristas del elemento de origen y los 3 restantes en los centros geométricos de las caras cuadrangulares. Así se obtiene el patrón de particionamiento que se observa en la figura 6.2.c). Se agregan **21+18** aristas nuevas y **20+10** caras nuevas por cada prisma triangular refinado².

El hecho de utilizar únicamente el esquema de particionamiento regular 1:8 para tetraedros plantea algunas cuestiones que pueden ser consideradas como ventajas o desventajas frente a esquemas que utilizan otro tipo de particionamiento. El refinamiento 1:8 implica que cada vez que se refina, el tamaño de los elementos disminuye más rápido que si se utilizaran refinamientos 1:2 ó 1:4 (ver figura 6.3).

¹Se indica en negrita el número de entidades geométricas que surgen por el refinamiento y que no son hijos/as de ninguna entidad geométrica correspondiente.

²Si bien el esquema de particionamiento que se presenta para prismas triangulares ya está implementado en el código, aún no ha sido aplicado a ningún caso práctico, relacionado principalmente con una capa estructurada de prismas en contacto con un contorno sólido.

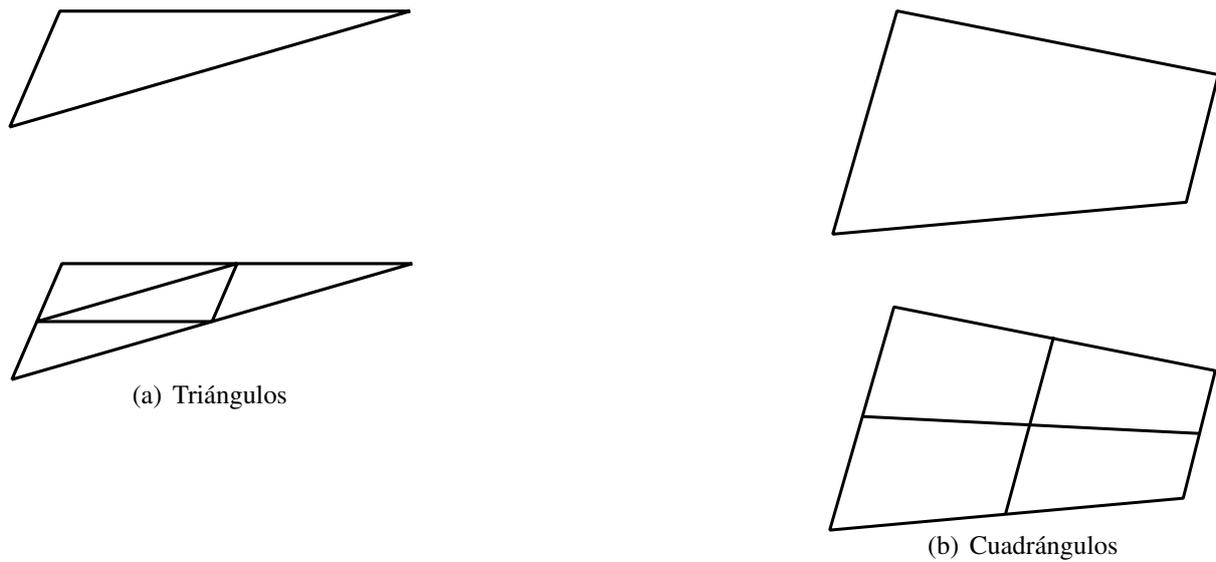


Figura 6.1: Patrones de refinamiento regulares 1:4 para elementos bidimensionales.

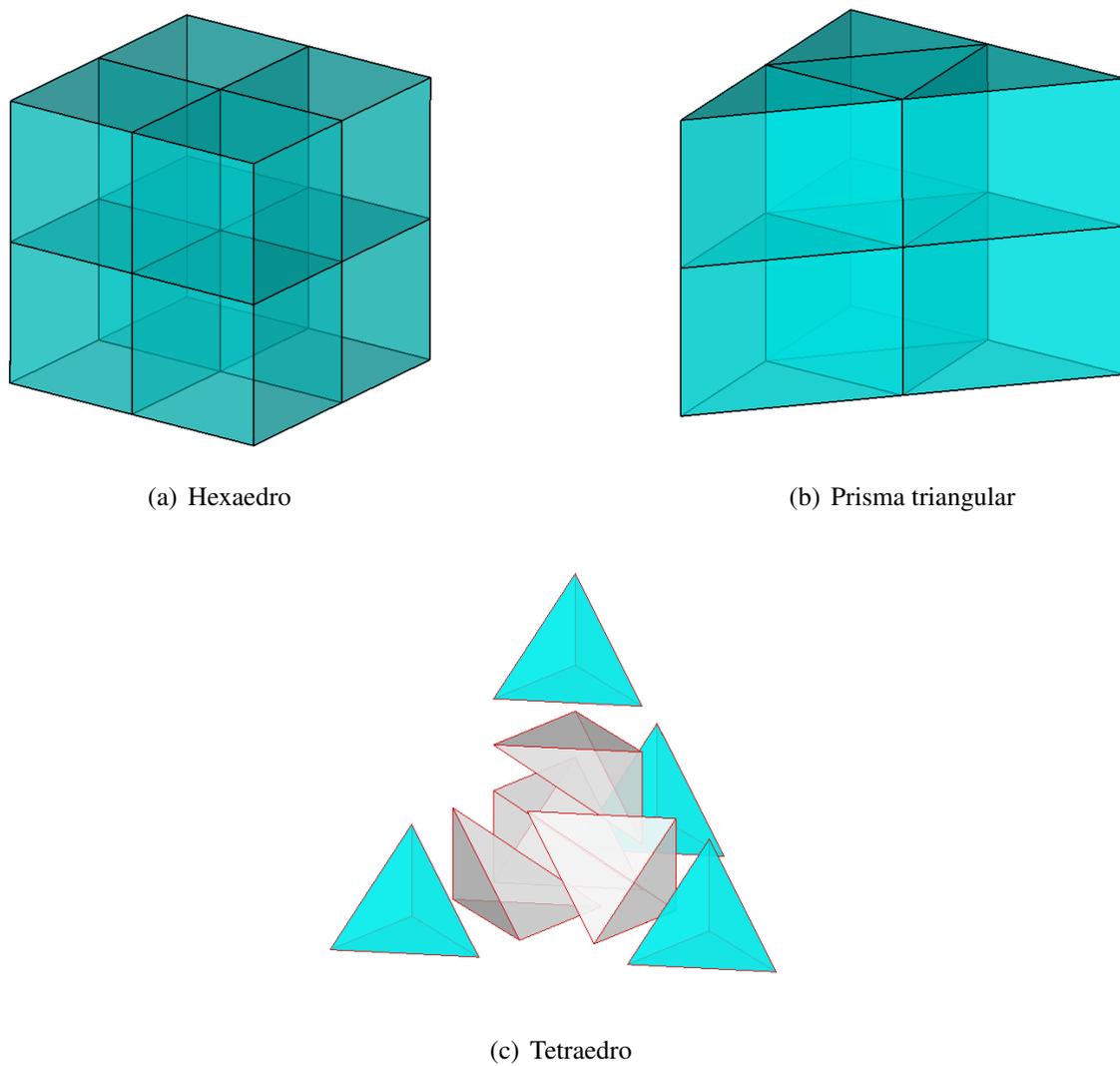


Figura 6.2: Patrones de refinamiento regulares 1:8 para elementos tridimensionales.

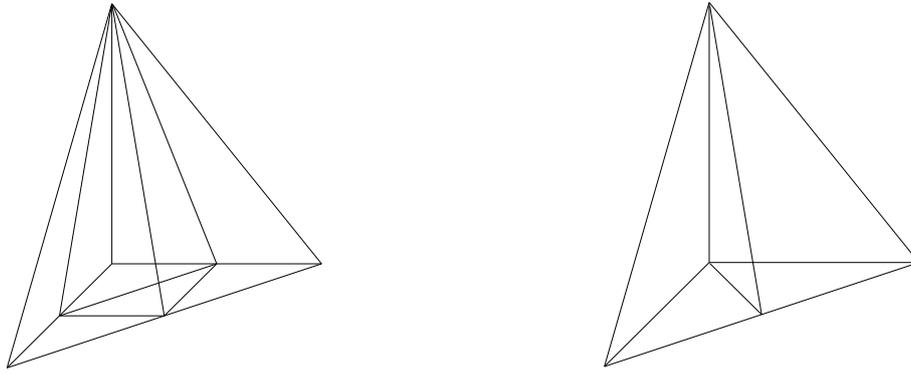


Figura 6.3: Refinamientos 1:4 y 1:2 para tetraedros.

Teniendo en cuenta la ecuación (3.2), esto puede ser considerado ventajoso desde el punto de vista de la rápida reducción del error en la aproximación. Sin embargo, resulta desventajoso si se considera que en el caso de problemas de flujo compresible el paso de tiempo se ve reducido notablemente por la aplicación de la condición de Courant-Friedrich-Lewy que afecta la estabilidad en el cálculo de la solución. Si el programa que resuelve el flujo utiliza un único paso de tiempo para avanzar la solución en todo el dominio computacional, entonces el tamaño del paso de tiempo se ve innecesariamente restringido en aquellos elementos de la malla que poseen un tamaño considerablemente mayor, y el costo de cálculo de la solución para evolucionar hasta un determinado instante de tiempo aumenta. Además, el esquema 1:8 agrega más grados de libertad al problema que los esquemas 1:2 y 1:4, con lo cual el tamaño del problema y el costo de solución también crece más rápido. Por otro lado, los esquemas de refinamiento regulares o isotrópicos no suelen ser adecuados cuando la solución exhibe una fuerte anisotropía. En estos casos resulta necesario agregar más elementos para “capturar adecuadamente” las características del flujo que si se utilizaran refinamientos direccionados o alineados perpendicularmente al gradiente máximo de la solución (refinamiento anisotrópico).

Entre las ventajas del particionamiento elegido se considera que los esquemas regulares aseguran que la calidad de la malla adaptada es aceptable a partir de la calidad de la malla original. Dada la relación que existe entre el número de condición de la matriz global del método de elementos finitos con la calidad de los elementos, el mal condicionamiento debido a elementos de baja calidad puede hacer que la velocidad con la que convergen los métodos de solución iterativos se vea afectada en el sentido de volverse más lenta (por ende costosa) o que incluso no se llegue a dar. Por lo tanto, este aspecto del esquema de particionamiento resulta fundamental. Además, el uso de una única secuencia de refinamiento para cada elemento simplifica la lógica del refinamiento ya que evita el uso de *reglas de compatibilidad* tanto en la etapa de refinamiento como en la de desrefinamiento.

6.2. Extensión del criterio de nodo 1-irregular a mallas 3-D

La restricción al refinamiento conocida como criterio de nodo 1-irregular[BR78] asegura que en el caso de mallas 2-D no exista más que un nivel de refinamiento de diferencia entre elementos *vecinos*. Para ello basta con exigir que “no exista más de un nodo irregular sobre una arista compartida”. En este caso, el criterio de *vecindad* se da a través de las aristas.

El objetivo de aplicar esta restricción al refinamiento es tener un control sobre la variación en el tamaño de los elementos al atravesar zonas con distinto grado de refinamiento. Así se asegura que no habrá diferencias arbitrariamente grandes entre los tamaños de los elementos vecinos en la malla, ya que la máxima relación de longitudes entre aristas compartidas es igual a dos.

Para poder describir el procedimiento de refinamiento que asegura el cumplimiento de la restricción al refinamiento, se deben tener en cuenta las siguientes definiciones:

- Cara / arista *activa*: se dice que una cara o arista está *activa* si *al menos uno* de los elementos que son vecinos a través de la misma no ha sido refinado. En caso contrario, se dice que la cara / arista ha dejado de existir o está *inactiva*.
- La *madre* de una cara / arista es aquella de la cual esta última ha surgido por aplicación del refinamiento. Es de observar, sin embargo, que en tres dimensiones la aplicación de un patrón de refinamiento a un elemento dado genera aristas y/o caras que no son *hijas* de ninguna de las que forman al elemento que se refina. Por ejemplo: la arista que se inserta en el centro del octaedro en el esquema de particionamiento propuesto en la sección 2.2 para tetraedros, no posee madre ya que no proviene de ninguna de las aristas del tetraedro al que se le aplica el refinamiento (ver figura 6.5.b). Lo mismo sucede con las caras 6-11-7, 6-11-8, 6-11-10, 6-11-9, 8-11-10, 6-7-8, 7-9-11 y 6-9-10.
- Todos los elementos que se refinan pasan al estado *inactivo*.
- Se observa, para las secuencias de particionamiento regulares que se proponen, que las aristas, caras y elementos refinados poseen 2, 4 y 8 hijos/as respectivamente. En el caso de mallas 2-D los esquemas de refinamiento que se aplican a los elementos son los mismos que se aplican a las caras en 3-D.
- Una entidad geométrica posee una *jerarquía* menor que la de su madre y una mayor que la de sus hijos/as.

La figura 6.4 presenta un ejemplo de la restricción al refinamiento para una malla de cuadrángulos. En ella se indica en color verde al elemento que se desea refinar y con puntos de color rojo a los nodos irregulares. Tal como lo menciona el criterio, para poder refinar al elemento que se indica en la figura 6.4.c), primero se debe refinar al elemento indicado en color gris en la figura 6.4.e), perteneciente a un nivel mayor en la jerarquía de elementos. Esto es así porque la

arista \overline{rp} es *madre activa* de la arista \overline{rs} y por ende el elemento en color gris debe ser refinado antes de proceder con el elemento en color verde. Una vez refinado el elemento en color gris, la arista \overline{rp} pasa a estar *inactiva* (o deja de existir como tal) y por ende el nodo s deja de ser irregular.

El criterio se puede extender a mallas tridimensionales si se tiene en cuenta que los elementos son vecinos no sólo a través de sus aristas sino también a través de sus caras. Esto es así porque en el caso tridimensional no es suficiente con exigir que *no exista más de un nodo irregular sobre una arista y/o sobre el centro de una cara compartida*. Esto es suficiente para que el criterio se satisfaga sólo cuando los elementos son vecinos a través de aristas y/o caras cuadrangulares. Sin embargo, si los elementos son vecinos a través de caras triangulares, entonces se debe tener en cuenta la situación particular representada en la figura 6.5. El ejemplo consiste en refinar en primer lugar al elemento 1-2-3-4 de la malla original - Fig.6.5.a) - con lo cual se obtiene la malla de la figura 6.5.b). Luego, se refinan los elementos 8-7-1-6 y 10-9-11-6. El refinamiento del elemento 8-7-1-6 no representa ninguna complicación, pero el refinamiento del 10-9-11-6 requiere un cuidado especial. Esto es así porque los elementos 2-5-3-4 y 6-9-10-11 no comparten aristas (las aristas 9-10, 10-11 y 11-9 no tienen madres) y tampoco existe un nodo irregular sobre la cara que comparten (la cara 9-10-11). Si sólo se tiene en cuenta la existencia de nodos irregulares sobre aristas y/o caras compartidas, entonces puede ocurrir la situación que se presenta en la figura 6.5.c). Se observa en este caso que puede haber más de un nivel de refinamiento entre elementos vecinos.

Luego, la función que aplica la secuencia de refinamiento a los elementos de la malla y que además satisface la restricción de nodo 1-irregular es diseñada en forma recursiva y se resume en el pseudo-código del algoritmo (5). La lógica del algoritmo es la siguiente: en una primera instancia se genera una lista `edges2ref` con las aristas que pertenecen a los elementos que se van a refinar (ingresada como argumento a la función). Cuando la función `refine()` es llamada por primera vez, la lista de elementos se corresponde con aquellos indicados por el estimador de error. De la lista de aristas `edges2ref`, sólo se retienen aquellas que aún no han sido refinadas. Luego, sobre este grupo, se realiza la búsqueda de las aristas madres que aún se encuentran activas. Si existen aristas en esta condición, eso significa que antes se debe refinar a los elementos a los cuales pertenecen estas *aristas madres activas* para no violar la restricción al refinamiento. Utilizando la información en las adyacencias aristas-elementos se genera la nueva lista de elementos que se deben refinar `eles2refNEW` en el nivel de recursión actual. Un razonamiento semejante se aplica a las caras. Finalmente, si la lista `eles2refNEW` no está vacía se vuelve a llamar a la función `refine()` con esta lista como argumento. Evidentemente, la recursión se detiene cuando ya no es necesario que se agreguen elementos en la lista a refinar para que se satisfaga el criterio 1-irregular. Las adyacencias aristas-elementos y caras-elementos son implementadas en la forma que se describe en la sección 5.3.2.

Por otro lado, la figura 6.5.d) presenta la solución al caso particular para elementos vecinos a través de una cara triangular. Tal como se describe en el algoritmo (5), si se presenta este caso

entonces se debe agregar a la lista de caras a refinar *faces2ref* a las caras *que se encuentran en el centro de las caras triangulares* (cara 9-10-11) y que tienen al menos una arista en la lista de aristas a refinar *edges2ref*. Luego, la madre de la cara 9-10-11 es la cara 2-3-4. Como la misma se encuentra *activa*, ya que no todos los elementos que la comparten se han refinado, entonces se busca cuales son los elementos de los cuales forma parte. Se encuentra que este elemento es el 2-5-3-4 y por ende se lo agrega a la lista *eles2refNEW*.

Algorithm 5 Pseudo-código de la función `refine()`. Fragmento correspondiente a la aplicación recursiva de la restricción al refinamiento.

```

refine(eles2ref,...){

    //Refinamiento de aristas
    //. Busca las aristas de los elementos eles2ref que NO tienen hijas y
    // las guarda en edges2ref.
    //. Busca las aristas madres activas de las aristas en edges2ref y
    // las guarda en ActParentEdges2ref.
    //Refinamiento de caras
    //. Busca las caras de los elementos eles2ref que NO tienen hijas y las
    // guarda en faces2ref.
    if (caras triangulares){ //Caso particular figura (6.5.d)

        //Agrega a faces2ref las caras que se encuentran en el centro
        //de las caras triangulares refinadas y que tienen al menos una
        //arista en edges2ref.

    }
    //. Busca las caras madres activas de las caras en faces2ref y
    // las guarda en ActParentFaces2ref.

    //Arma la lista de elementos que se agregan por
    //restricción al refinamiento.
    //. Busca elementos a los que pertenecen las ActParentEdges2ref y
    // los agrega a eles2refNEW (requiere adyacencias hacia arriba
    // aristas-elementos).
    //. Busca elementos a los que pertenecen las ActParentFaces2ref y
    // los agrega a eles2refNEW (requiere adyacencias hacia arriba
    // caras-elementos).
    if (!eles2refNEW.empty()) refine(eles2refNEW,...);
    //Aplica refinamiento a los elementos y actualiza estructuras
    //de datos.
    ....
    ....
    ....
}

```

6.3. Adaptación de la malla en problemas dependientes del tiempo

La estrategia de adaptación para problemas no estacionarios se diseña considerando como factor principal al costo de adaptación de la malla. En problemas no estacionarios la malla debe

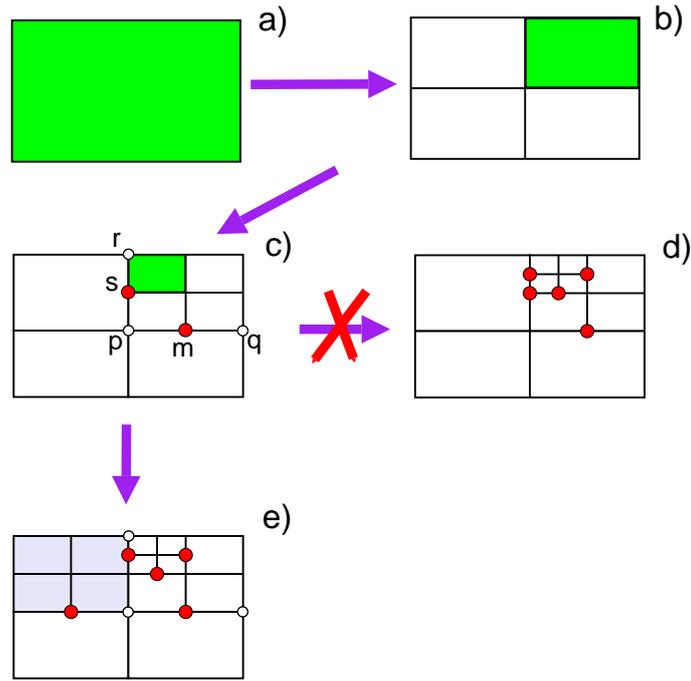


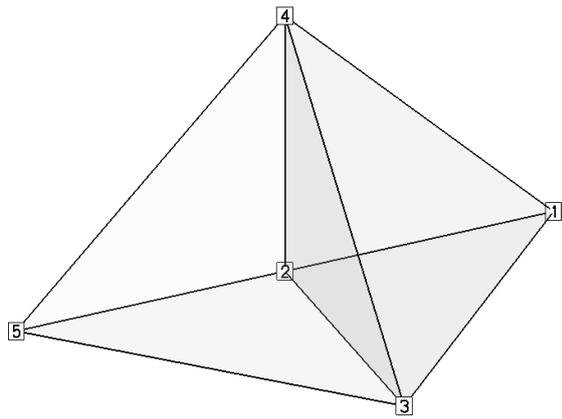
Figura 6.4: Aplicación de la restricción 1-irregular al refinamiento de una malla de cuadrángulos.

ser adaptada frecuentemente a los fines de poder “seguir” con el refinamiento / desrefinamiento el desplazamiento de las estructuras del flujo en el dominio del problema. Por lo tanto, el costo de adaptación de la malla puede representar un porcentaje considerable del costo total de solución del problema. Waltz[Wal04] indica que en procedimientos de solución adaptativos parcialmente paralelizados (lo cual significa que la resolución de las ecuaciones fluido-dinámicas se lleva a cabo en forma distribuida mientras que la adaptación de la malla se realiza en forma secuencial, tal cual es el caso del procedimiento implementado en esta tesis) aplicados a problemas de flujos compresibles, no estacionarios y tridimensionales, la adaptación de la malla demanda aproximadamente el 10 % del tiempo computacional.

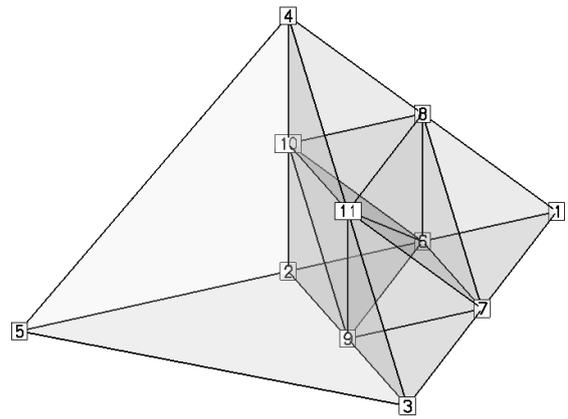
En el esquema de solución que se propone en esta tesis, la solución en todos los elementos de la malla es avanzada utilizando un único paso de tiempo. Es decir que no se realiza una adaptación del problema en la variable temporal. En problemas de flujo compresible, el tamaño del paso de tiempo está limitado por la condición de Courant-Friedrichs-Lewy[Hir90],

$$\Delta t = K \underset{e}{\text{mín}} \left(\frac{h_e}{(\lambda_{\max})_e} \right) \quad (6.1)$$

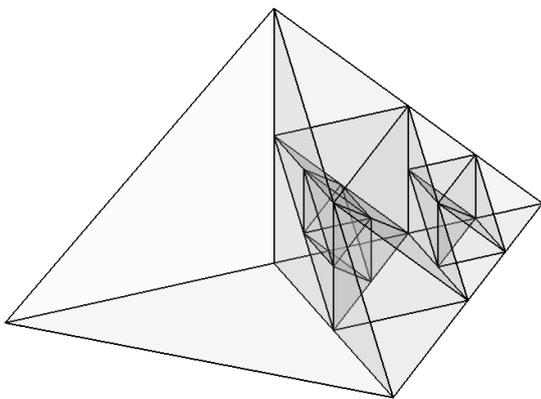
donde $(\lambda_{\max})_e$ es la velocidad de la onda más rápida, calculada en función del estado en alguno de los nodos del elemento e , h_e es una medida del tamaño del elemento y K es el número de Courant.



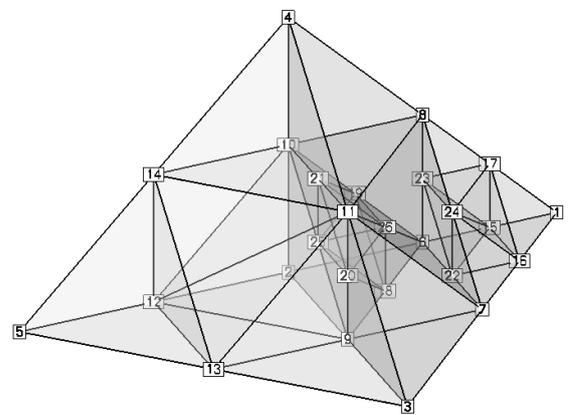
(a) Malla original



(b) Primer refinamiento



(c) Refinamiento no permitido



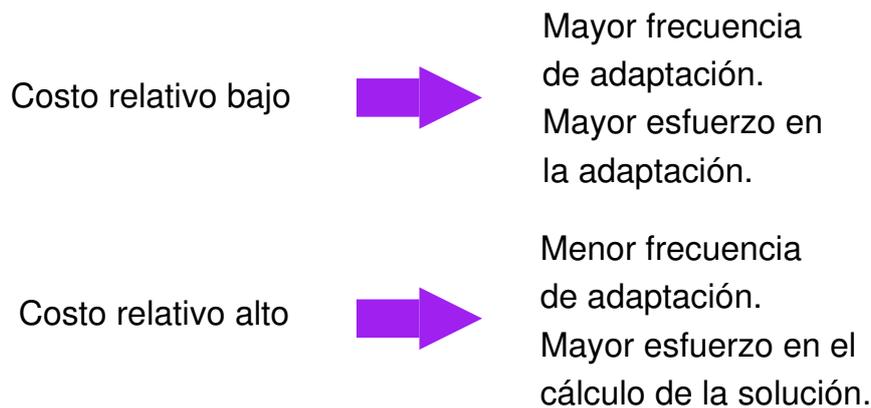
(d) Segundo refinamiento

Figura 6.5: Restricción 1-irregular al refinamiento para una malla de tetraedros - caso particular.

6.3.1. Frecuencia de adaptación

Una cuestión crítica cuando se resuelven problemas no estacionarios es aquella relacionada con la frecuencia de adaptación de la malla. Se observa que algunos autores [BC89, LB92, BBJW94, Wal04, YK05] no actualizan la malla en todos los pasos de tiempo, mientras que otros [Bod97, Qui91, OE04] proponen adaptar la malla cada vez que sea posible. El criterio que define inclinarse a favor de una u otra alternativa depende del costo relativo de adaptar la malla en relación al costo de cálculo de la solución. En teoría, los algoritmos de adaptación de mallas deberían demandar sólo una pequeña fracción del tiempo total requerido para calcular la solución. Si así no fuera, entonces al no modificar la estructura de datos que se utiliza para la representación de la malla cada vez que fuera posible, se pensaría que el algoritmo de solución adaptativa aumenta su eficiencia (es decir, dedica menos tiempo para adaptar la estructura de la malla en comparación al tiempo requerido para calcular la solución sobre la misma). Sin embargo, en este caso las regiones con mayor grado de refinamiento deberían extenderse más allá de la vecindad inmediata de las estructuras del flujo que se desean resolver en forma precisa gracias al refinamiento, con el objeto de asegurar que las mismas quedasen contenidas en las zonas refinadas mientras se desplazan en el lapso de tiempo durante el cual la malla no es modificada. Así se observa que a menor frecuencia de actualización de la malla, mayor es la proporción del dominio computacional que debe ser refinada y por ende mayor es el costo en el cálculo de la solución.

Por el contrario, si el costo de adaptación de la malla es relativamente bajo en comparación al costo de cálculo de la solución, entonces puede resultar conveniente actualizar la malla con una frecuencia mayor y trasladar el esfuerzo computacional a la etapa de adaptación. En resumen



En esta tesis se considera que la frecuencia de actualización de la malla se mantiene constante durante todo el procedimiento de cálculo de la solución. En general se observa [LB92] que frecuencias de actualización de la malla de 5~10 pasos de tiempo resultan adecuadas. Sin embargo, la frecuencia de adaptación de la malla, medida en pasos de tiempo de avance de la solución, debe ser un compromiso entre todos los factores mencionados anteriormente. A estos factores se agrega el hecho de que los esquemas de refinamiento propagan, debido a la aplicación del criterio de nodo 1-irregular, la zona refinada más allá de lo estrictamente indicado por

el estimador / indicador de error utilizado.

6.3.2. Descripción del procedimiento de adaptación no estacionario

La adaptación de la malla en función de la información provista en la etapa de estimación del error se realiza aplicando sucesivos refinamientos a partir de una *malla base*, tal como se observa en el esquema de la figura 6.6. La malla base no debe ser una malla cualquiera, sino que debe presentar una distribución adecuada de sus elementos. Al momento de generar la malla se deben tener en cuenta la geometría del dominio y si se dispone, de alguna información acerca de la solución, de las condiciones iniciales y de condiciones de borde del problema. La importancia que adquiere la generación de una malla base o de partida adecuada es tal por los siguientes motivos:

- El algoritmo de refinamiento no permite desrefinar la malla base. Luego, si esta presenta una distribución inadecuada en el tamaño de los elementos debido a la presencia de elementos pequeños en regiones del dominio que no lo requieren, el costo en el cálculo de la solución implicado por dichos elementos no podrá ser reducido por el procedimiento de adaptación.
- La malla base debe poseer una distribución de sus elementos tal que sea factible resolver sobre ella todas las estructuras del flujo que resultan de interés en los primeros pasos de tiempo, hasta que la misma sea adaptada por primera vez. De otra manera, puede resultar que el procedimiento de indicación/estimación de error no sea capaz de determinar adecuadamente las regiones que deben ser refinadas porque dichas estructuras del flujo ni siquiera forman parte de la solución computada sobre la malla base.
- La predicción adecuada en la distribución del tamaño de los elementos en la malla base contribuye a reducir el costo de adaptación de la malla. En caso contrario se deberá refinar más veces sobre determinadas regiones del dominio para reducir el error hasta las cotas deseadas, con lo cual el costo de adaptación se incrementa. Dado que los elementos de las mallas adaptadas se encuentran anidados en los elementos de la malla base, si esta última posee elementos grandes en regiones del flujo donde se puede predecir que existirán discontinuidades, entonces el procedimiento de adaptación deberá introducir muchos elementos hasta que la solución exhiba la precisión deseada.

A la aplicación de sucesivos refinamientos a partir de la malla base se la indica como un *paso de adaptación*. Para cada paso de adaptación (excepto la primera vez que se adapta la malla base) se aplica el criterio de estimación del error (ver sección 3.2) y se indican los elementos que deben ser refinados. Si se analiza el paso de adaptación n , la estimación del error se realiza utilizando el último estado computado sobre la malla más refinada obtenida en el paso de adaptación previo, indicado como $n - 1$ en la figura 6.6.

Numerosos autores [LB92, BS97, Wal04, YK05] proponen limitar el número máximo de niveles de refinamiento. El objetivo que se persigue es no refinar un número indefinido de veces a los elementos que ocupan una misma región en la malla. Esta situación se presenta cuando existen gradientes muy grandes en la solución. En estos casos el indicador o estimador del error elemental predice que los elementos seleccionados para refinar son siempre aquellos que se encuentran sobre las discontinuidades. Luego estos elementos pueden volverse demasiado pequeños y el tamaño del problema crece en forma indefinida, con lo cual el costo en el cálculo de la solución se ve comprometido. Teniendo en cuenta que la estructura de datos utilizada para representar a los elementos de la malla no cuenta con información relativa al nivel de refinamiento al que pertenece cada elemento, la restricción sobre el número de niveles de refinamiento se realiza mediante las siguientes operaciones:

1. Se obtiene una lista de elementos seleccionados para refinar sobre la malla con el máximo nivel de refinamiento permitido, la cual es guardada en `list2search`.
2. Se recupera la estructura de datos `ele_ref` de la malla correspondiente al nivel de refinamiento actual. `ele_ref` posee, para un elemento dado, información sobre los elementos hermanos, el elemento padre y una propiedad elemental asignada.
3. Dado que al ser refinado un elemento deja de existir como tal, el primer hijo de éste elemento refinado pasa a tener el mismo número de identificación que su padre. Luego si el elemento-*i* posee el máximo nivel de refinamiento permitido por el usuario, entonces posee hermanos en éste nivel. De lo contrario la lista de hermanos para el elemento, en éste nivel de refinamiento, se encuentra vacía. Con este criterio se generan dos listas: una con los elementos que no tienen hermanos (`ind_no_brothers`) y que por lo tanto no pertenecen al nivel de refinamiento actual y otra con los que sí (`ind_brothers`).
4. Se obtienen la numeración de los elementos padres de aquellos elementos que se encuentran tanto en `list2search` como en `ind_brothers` y se construye una lista `fathers_num`. Si estos elementos no tienen padres en este nivel de refinamiento, entonces se los guarda en una lista de elementos que no tienen padres `eles_without_fathers`. El número de identificación del padre de un elemento es directamente el índice en el vector de elementos `ele_ref`.
5. Se construye una lista de elementos `list2search` para generar el nivel de refinamiento actual (con la misma profundidad) en el próximo paso de adaptación. Esta lista resulta de la unión de las listas `fathers_num` y `eles_without_fathers` (recordar que estas operaciones se realizan sobre la estructura de datos pertenecientes al paso de adaptación *n-1*).
6. Se guarda en archivo la lista `list2search`.

7. Se repite el procedimiento desde el ítem 2, subiendo un nivel en la profundidad del refinamiento pero utilizando ahora la lista `list2search` para identificar a los elementos que tienen hermanos de los que no, hasta llegar al nivel de refinamiento cero (malla base).

Como se observa, la lógica que se describe en los ítems 2-7 se repite en forma recursiva. Las listas `list2search` que se guardan en archivo en los pasos 5 y 6 del procedimiento descrito son “preliminares”, en el sentido de que las mismas deberán ser modificadas en la siguiente etapa de la adaptación. El procedimiento descrito es representado por flechas de color azul en el diagrama de la figura 6.6 y es implementado en la función `search_fathers()` cuyo pseudo-código se presenta en 6.

Algorithm 6 Pseudo-código de la función `search_fathers()`.

```
void search_fathers(step2search, list2search, ref_level, nstep){  
    //Variables  
    //step2search: paso de adaptacion en el que se aplica la busqueda.  
    //list2search: lista de elementos cuyos padres se desea encontrar.  
    //ref_level: nivel de refinamiento. La primera vez que se llama a  
    //la funcion es el maximo nivel permitido.  
    //nstep: frecuencia de adaptacion.  
  
    //Recupera la estructura de datos ele_ref para el nivel  
    //de refinamiento ref_level y paso de adaptacion step2search dados.  
  
    //Busca en ele_ref los elementos que no tienen hijos/hermanos en el  
    //ref_level actual y los guarda en la lista ind_no_brothers.  
  
    //Obtiene por complemento la lista ind_brothers con los indices en  
    //ele_ref de los elementos que poseen hermanos.  
  
    //Busca los padres de los elementos que se encuentran en ele_ref,  
    //que poseen hermanos (ind_brothers) y que estan en list2search a la  
    //vez y genera la lista fathers_num.  
  
    //Obtiene por complemento entre list2search y fathers_num los  
    //elementos que no tienen padres en este nivel de refinamiento y  
    //genera eles_without_fathers.  
  
    //Actualiza list2search como la union de fathers_num y  
    //eles_without_fathers.  
  
    if(ref_level==0)  
        //Escribe list2search en archivo.  
    else{  
        //Escribe list2search en archivo.  
        //Sube un nivel de refinamiento en la jerarquia.  
        ref_level=ref_level-1;  
        //LLamada recursiva a la funcion.  
        search_fathers(step2search,list2search,ref_level,nstep);  
    }  
}
```

Una vez construidas las listas de elementos a refinar, se inicia un lazo sobre los niveles de refinamiento el cual comienza con la aplicación de la lista `list_level_0` a la malla base. Luego, para avanzar a los niveles de refinamiento $1, 2, \dots, \text{max_level}$ se debe actualizar la numeración de los elementos a refinar que se guardaron en las listas correspondientes `list_level_1, list_level_2, \dots, list_level_max`, ya que para un dado nivel de refinamiento r la numeración de los elementos de la malla en el paso de adaptación n no se corresponde con la numeración de los elementos en el paso de adaptación $n - 1$. La actualización de la numeración se realiza comparando las estructuras de datos `ele_ref` para las mallas en el nivel de refinamiento r en los pasos de adaptación n y $n - 1$. Se debe determinar cuál es la numeración de los hijos de los elementos previamente refinados para reemplazar la vieja numeración por la nueva. Por ejemplo, en la figura 6.6 se observa para la lista `level_ref_2`, que los hijos del elemento 3 en la malla correspondiente al paso de adaptación $n - 1$ son los elementos $\{3, 4, 5, 6\}$. Pero como en el paso de adaptación n se refinan elementos que no se refinaron en el $n - 1$ (se refinan los elementos $\{1, 2\}$), entonces estos elementos poseen en el paso de adaptación n la numeración $\{3, 10, 11, 12\}$. Luego, si bien el elemento 3 sigue siendo el mismo en ambas mallas, el elemento 4 (que debe ser refinado) ahora es el elemento 10.

Por otro lado, se debe observar que a la lista `level_ref_2` actualizada se agregan todos los hijos de los elementos $\{1, 2\}$, ya que si no se lo hiciera, es posible que la zona finalmente refinada no incluya a las estructuras del flujo que se desean capturar con el mayor grado de refinamiento. De esta forma, al elemento 1 se le agregan los elementos $\{4, 5, 6\}$ y al 2 se le agregan los $\{7, 8, 9\}$. Este procedimiento se aplica siempre que se utiliza más de un nivel de refinamiento y se encuentra implementado en la función `update_els2ref()`.

Actualizada la numeración de los elementos a refinar, se aplica la lista correspondiente al nivel r para luego avanzar al nivel $r + 1$ dentro del paso de adaptación n , hasta que se llega al máximo nivel permitido. La actualización de la numeración de los elementos a refinar es representada por las flechas de color verde en la figura 6.6.

6.3.3. Proyección del estado

Una vez que se alcanza el máximo nivel de refinamiento admitido se debe reiniciar el cálculo de la solución sobre la malla adaptada. En esta instancia se deben proporcionar un estado inicial y las condiciones de borde correspondientes a la nueva malla al *solver* de las ecuaciones del flujo. Aquí se presenta el esquema utilizado para proyectar el estado, mientras que el tratamiento de las condiciones de borde se describe en la sección 6.4. La proyección del último estado computado sobre la nueva malla se realiza de la siguiente manera:

- En el primer paso de adaptación, el estado computado sobre la malla base es interpolado linealmente para obtener el estado sobre la malla más refinada. Dado que en el cálculo de la solución se utilizan elementos finitos cuyas funciones de forma son bilineales (2-D) ó trilineales (3-D) y dado que los nodos introducidos por el refinamiento se posicionan

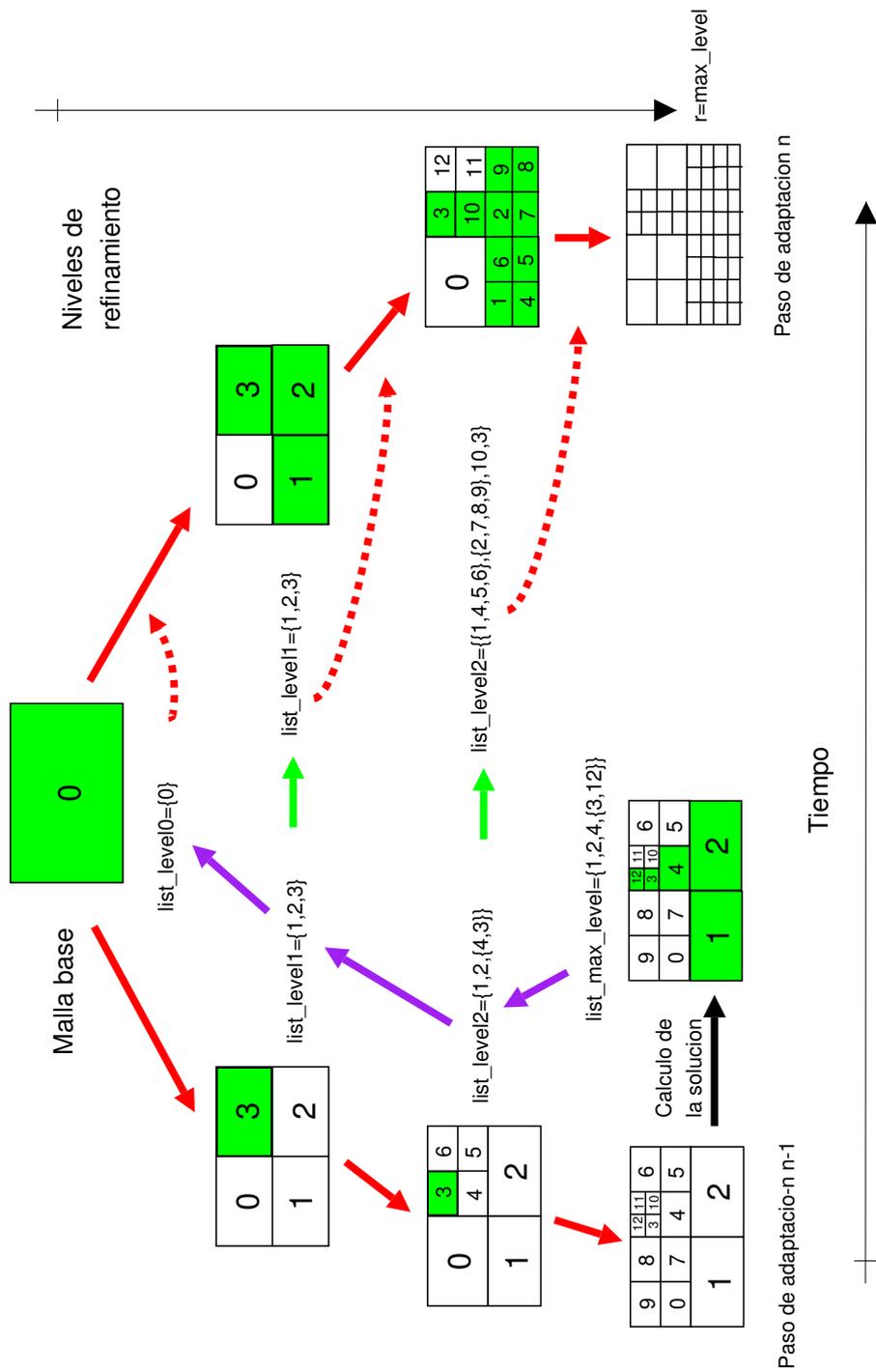


Figura 6.6: Estrategia de adaptación para problemas no estacionarios.

siempre en las mitades de las aristas o en los centros geométricos de las caras o de los elementos refinados (esto ocurre sólo en cuadrángulos y hexaedros), el cálculo del estado en los nuevos nodos como el promedio de los estados en los nodos que definen a la arista, cara o elemento en los que surgen (ver figura 6.7) no viola la conservación de las variables del flujo.

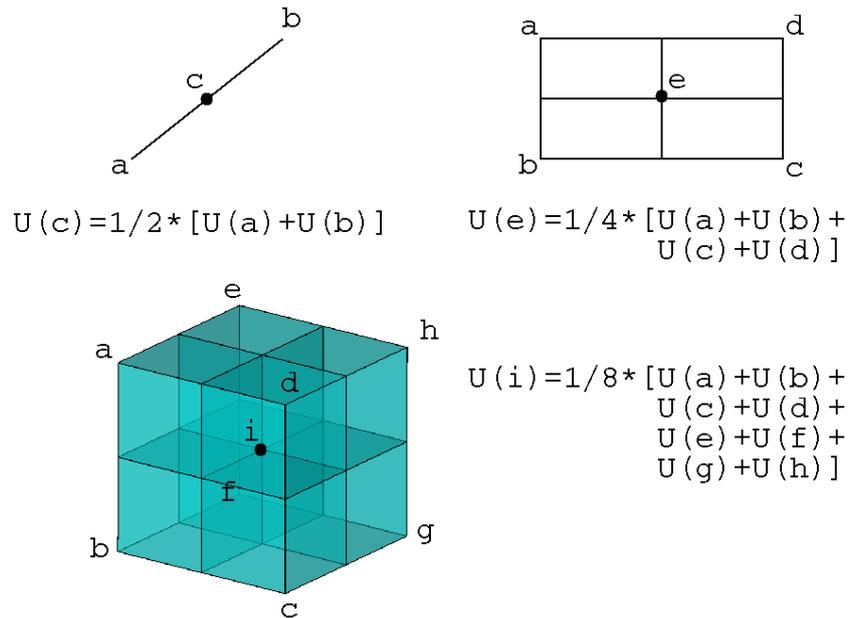
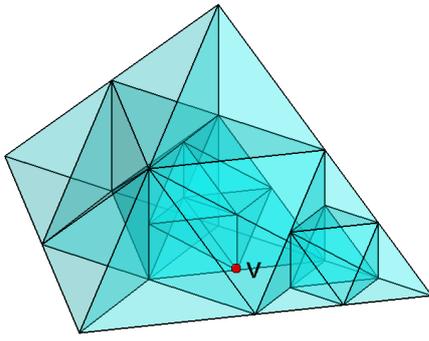


Figura 6.7: Interpolación lineal del estado para el primer paso de adaptación de la malla base.

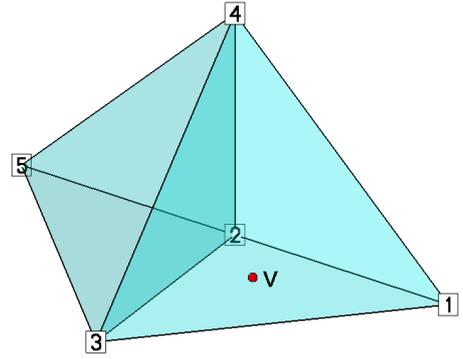
- En los pasos de adaptación siguientes, el estado inicial para reiniciar el cálculo de la solución sobre la nueva malla adaptada se obtiene de la siguiente manera: a) en primera instancia se plantea determinar la posición del vértice V introducido por el refinamiento en el paso de adaptación actual (n) con respecto de los elementos de la malla adaptada perteneciente al paso de adaptación anterior ($n - 1$), b) luego se utilizan los valores que adoptan en el vértice V las funciones de forma del elemento en cuyo interior se encuentra el vértice. Con esta información, el estado en V se puede computar de la siguiente manera

$$U_V^{(n)} = \sum_{j=1}^{nbe} N_j^{(e_i)}(\mathbf{x}_V) \cdot U_j^{(n-1)} \quad (6.2)$$

donde $U_V^{(n)}$ es el estado en el vértice V en el paso de adaptación n , nbe es el número de vértices que definen al elemento e_i de la malla perteneciente al paso de adaptación $n - 1$ en cuyo interior se determina que se encuentra el vértice V (por ejemplo: nbe es igual a 4 para tetraedros y 8 para hexaedros), $N_j^{(e_i)}(\mathbf{x}_V)$ es el valor que la función de forma elemental asociada al nodo local- j adquiere en el vértice V , mientras que $U_j^{(n-1)}$ es el estado computado en el nodo local- j en el paso de adaptación $n - 1$ (observar la figura 6.8).



(a) Posición del vértice V para la malla en el paso de adaptación n



(b) Posición del vértice V para la malla en el paso de adaptación $n-1$

Figura 6.8: Proyección del estado - Determinación de la posición del vértice V en la malla correspondiente al paso de adaptación anterior.

Para resolver el problema de encontrar al elemento de la malla obtenida en el paso de adaptación $n-1$ en cuyo interior se encuentra el vértice V se recurre al siguiente procedimiento

1. Se construye una lista con los centroides de los elementos que forman la malla en el paso de adaptación $n-1$. Para la malla de la figura 6.8.b) se tiene una lista con dos centroides: g_1 asociado al elemento $e_1 = \{1, 2, 3, 4\}$ y g_2 asociado al elemento $e_2 = \{2, 3, 4, 5\}$.
2. Se realiza un búsqueda *aproximada* de los k -elementos vecinos más cercanos al vértice V , determinando para ello los k -centroides más cercanos. Se obtiene así una lista con la numeración de los k -elementos vecinos más cercanos ordenada crecientemente según la distancia de los centros geométricos al vértice V . Para la malla de la figura 6.8.b) la lista encontrada para el vértice V es $l_V = \{e_1, e_2\}$.
3. Se recorre la lista de los elementos vecinos más cercanos encontrados para determinar en el interior de cuál de ellos se encuentra el vértice V . Para ello, es necesario evaluar las coordenadas de área o de volumen del elemento e_i , $i = 1, \dots, k$ en el vértice V . Si se consideran los valores de las coordenadas de volumen de los nodos del elemento e_1 evaluadas en el vértice V se tiene: $N_1(V) = N_3(V) = 0.25$, $N_2(V) = 0.5$ y $N_4(V) = 0$. Luego, el vértice V está en el interior de e_1 .
4. Si se encuentra que el vértice V está en el interior de un elemento, se interrumpe el lazo que recorre la lista de elementos y se utilizan los valores de las coordenadas de área para computar el estado mediante la ecuación 6.2. Si *alguna* de las coordenadas de área es negativa, entonces se sigue recorriendo la lista de elementos más cercanos hasta que se satisface la condición para la cual el vértice está en el interior de alguno de los elementos o bien hasta agotarla.

La búsqueda *aproximada* de los k -elementos vecinos más cercanos al vértice V se lleva a cabo con un algoritmo de búsqueda eficiente implementado en la librería ANN (*Approximate Nearest Neighbour*). ANN es una librería *Open Source* desarrollada por David Mount y Sunil Arya. La misma se encuentra escrita en C++ y permite resolver el problema de la búsqueda aproximada o exacta de los k -vecinos más cercanos a un punto en el espacio realizando una descomposición ortogonal del mismo a través de la implementación de estructuras de datos del tipo *kd-tree* y *box-decomposition trees*. Arya y Mount muestran que la utilización de esta estructura de datos en combinación con la posibilidad de admitir que los puntos encontrados no sean los vecinos más cercanos, pero que tampoco están significativamente lejos de serlo, permite reducir sensiblemente los tiempos de ejecución a $O(\log(n))$. Esta complejidad del algoritmo de búsqueda unida al hecho de que el problema es fácilmente paralelizable (básicamente a través de un paralelismo de datos) permite resolver el problema de la proyección del estado sin comprometer los recursos computacionales ni el tiempo de ejecución. Para mayores detalles sobre la implementación de ANN se remite al lector al manual de usuario de la librería[AM06]. Se observa que al utilizar las funciones de forma elementales para calcular el estado en el vértice V , se asegura que el estado proyectado no viole la conservación de las variables del flujo.

Sin embargo existen inconvenientes vinculados a la conservación debidos al desrefinamiento de la malla. En este caso, no es tenida en cuenta la contribución al estado proyectado de aquél computado sobre un vértice V de la malla en el paso $n - 1$ que no existe como tal en la malla obtenida en el paso de adaptación n . En esta situación es claro que no se respeta la conservación de las variables del flujo. Este problema no ha sido resuelto en la estrategia implementada. A pesar de ello, se piensa que si las regiones refinadas son suficientemente extensas en relación a las dimensiones de las discontinuidades del flujo capturadas con el refinamiento y si el desrefinamiento se presenta en aquellas regiones del flujo alejadas de las discontinuidades, en las cuales la solución es suave, los efectos debidos a la no conservación pueden ser atenuados[LB92]. Si bien no se han realizado evaluaciones en este sentido, los resultados que se obtienen en los ejemplos de flujos no estacionarios que se resuelven en esta tesis parecen confirmar estas suposiciones (ver secciones 7.3 y 7.4).

La figura 6.9 presenta la estrategia de adaptación descrita mediante un diagrama de flujo en el que se incluyen todas las operaciones comentadas. El procedimiento difiere únicamente cuando se realiza la adaptación de la malla por primera vez. En este caso, el problema se resuelve sobre la malla base, obteniéndose un estado sobre el cual se aplica el indicador o estimador de error. Luego se refina hasta que se alcanza la cota del error propuesta para el estimador de error o bien el máximo nivel de refinamiento indicado por el usuario. Luego se reinicia el cálculo de la solución. Por simplicidad, este caso no se incluye en el diagrama de la figura 6.9.

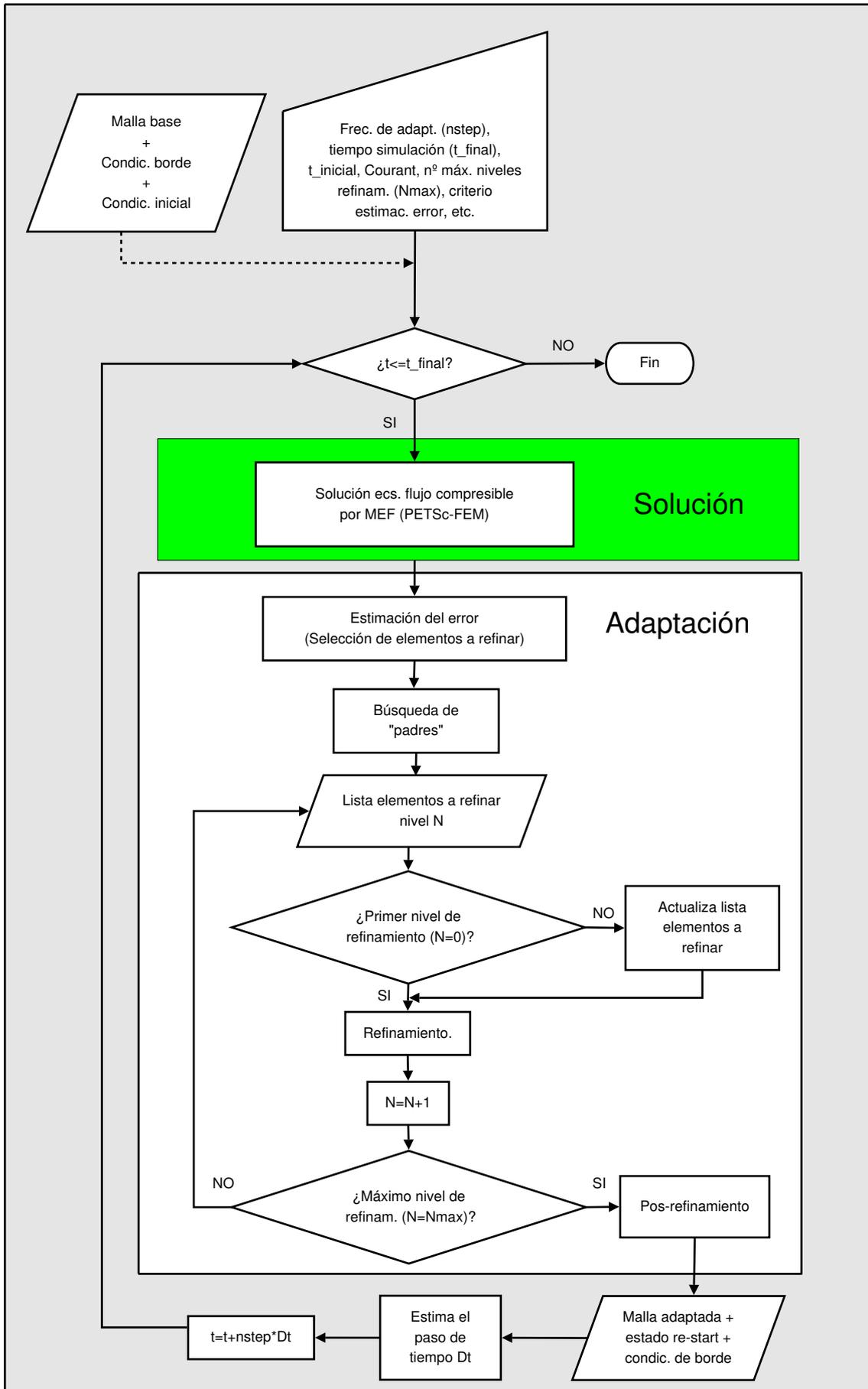


Figura 6.9: Diagrama de flujo para la estrategia de adaptación en problemas no estacionarios.

6.4. Tratamiento de propiedades sobre entidades geométricas

Se describe un procedimiento que permite asociar propiedades a las entidades geométricas de la malla. El procedimiento es implementado en la estrategia de refinamiento por medio de un grupo de funciones `flagsforedges()`, `flagsforfaces()` y `flagsforeles()` que permiten asignar propiedades a aristas, caras y elementos respectivamente. Esta técnica permite gestionar durante la adaptación de la malla las siguientes situaciones:

Condiciones de borde

Las condiciones de borde que usualmente se aplican al resolver problemas de flujos compresibles son:

- Condición de deslizamiento o velocidad normal nula sobre contornos sólidos cuando se resuelven flujos no viscosos.
- Fijaciones de las variables del flujo en algún tramo del contorno.
- Restricciones periódicas y condiciones de simetría.
- Condiciones de borde dinámicas: con ello se hace referencia al procedimiento que utiliza información proveniente de la solución computada el interior del dominio (la cual se propaga a lo largo de las curvas características) hacia los contornos para ajustar en forma dinámica el número y tipo de condiciones de borde que deben ser impuestas en los mismos. Cuando se resuelven problemas de flujos compresibles, es posible que en algún contorno o tramo de contorno el régimen del flujo cambie de subsónico a supersónico o viceversa, o bien es posible que se produzca la inversión del flujo. En estos casos, el número y tipo de condiciones de borde deben ser ajustados dinámicamente para que el problema esté bien planteado. Por ejemplo, en el modelado del arranque del flujo en el interior de una tobera las condiciones de borde en la sección de salida cambian de subsónico a supersónico una vez que la onda de choque alcanza dicha sección (ver sección 7.3).

Por otro lado, se debe garantizar la no reflexión de perturbaciones sobre los contornos “ficticios” utilizados en una simulación para representar una condición en el infinito o en una zona muy alejada de la región de interés. Condiciones de borde que satisfacen este requerimiento son conocidas con el nombre de “absorbentes” o “no reflectivas” y la ventaja que presentan es la de poder posicionar estos contornos ficticios cerca de la región de interés, con lo cual se reduce el costo computacional de la simulación.

En problemas de flujos compresibles y no viscosos (Euler), la determinación del número y tipo de condiciones de borde adecuadas que deben aplicarse en el contorno puede

realizarse por medio de un análisis de los autovalores de la matriz Jacobiada de los flujos convectivos dada por la Ec.(4.9). En esta tesis se utiliza el método presentado en [SNPD08] para manejar condiciones de borde de este tipo.

Contornos curvos

El mismo procedimiento puede ser aplicado para corregir las coordenadas de nodos que por el refinamiento surgen sobre tramos curvos del contorno. Inicialmente, las coordenadas de uno cualquiera de estos vértices son calculadas como el promedio de las coordenadas de los nodos que definen a la arista sobre la que surge el vértice. La implementación desarrollada sólo permite llevar a cabo esta tarea si los contornos pueden ser representados analíticamente por una función (es decir, superficies cónicas, esféricas, cilíndricas, etc.) de la forma $F(x, y, z) = 0$.

Luego, para un vértice o nodo en particular, los pasos que se aplican son los siguientes: a) se calcula la normal en el vértice como el “promedio” de las normales de las caras que lo comparten (en el caso de hexaedros se debe tener en cuenta que si el nodo cuya coordenada se debe corregir surge en el centro de la cara cuadrangular, entonces la normal es directamente la normal a la cara), b) mediante el método de Newton se resuelve el problema de encontrar la intersección de la recta normal que pasa por el vértice con la superficie descrita por la función F , para lo cual se debe contar además con información sobre el gradiente de la función ∇F^3 , c) finalmente se accede al arreglo de coordenadas y se reemplazan las coordenadas del nodo en cuestión.

En el caso 2-D el contorno es descrito por una línea curva $F(x, y) = 0$ y el problema que se resuelve es la intersección de la recta normal en el punto V a la arista \bar{AB} con la curva $F = 0$. La figura 6.10 ilustra esta situación para una malla de triángulos.

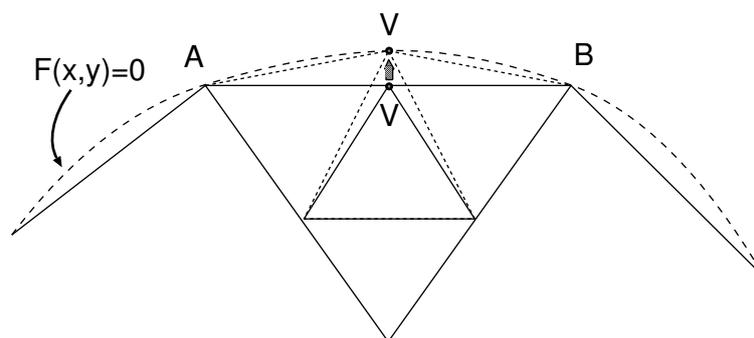


Figura 6.10: Corrección coordenadas de nodo perteneciente a un contorno curvo.

³Si no es posible calcular las componentes del ∇F requeridas por el método de Newton, entonces se puede aplicar algún método que aproxime las derivadas, como el método de la secante.

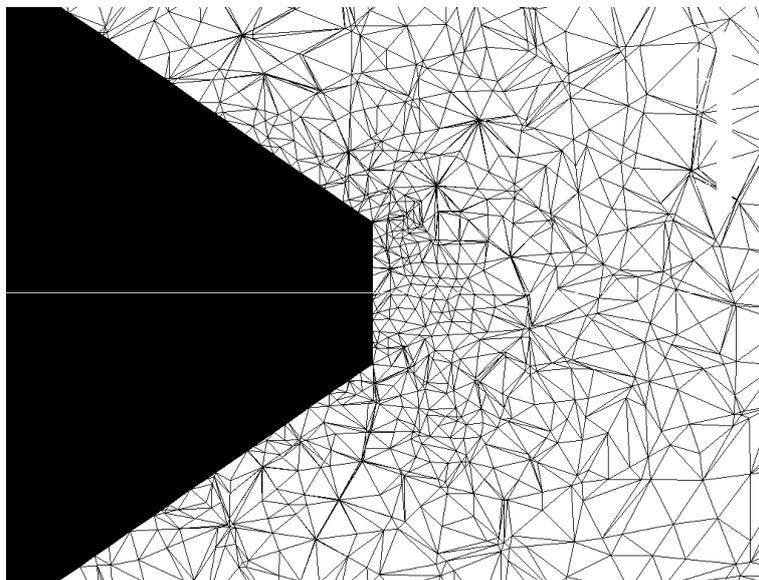
Refinamiento selectivo impuesto por el usuario

Finalmente, otra aplicación de la gestión de propiedades sobre entidades geométricas consiste en refinar una capa formada por todos los elementos que se encuentran en contacto con un contorno o tramo de contorno. La figura 6.11 presenta un ejemplo en el cual se aplican dos niveles de refinamientos a todos los tetraedros que se encuentran en contacto con el contorno sólido de un cuerpo. Esta aplicación requiere asignar un *flag* de propiedad a todos los elementos que “tocan” al cuerpo en la malla de partida con al menos uno de sus vértices.

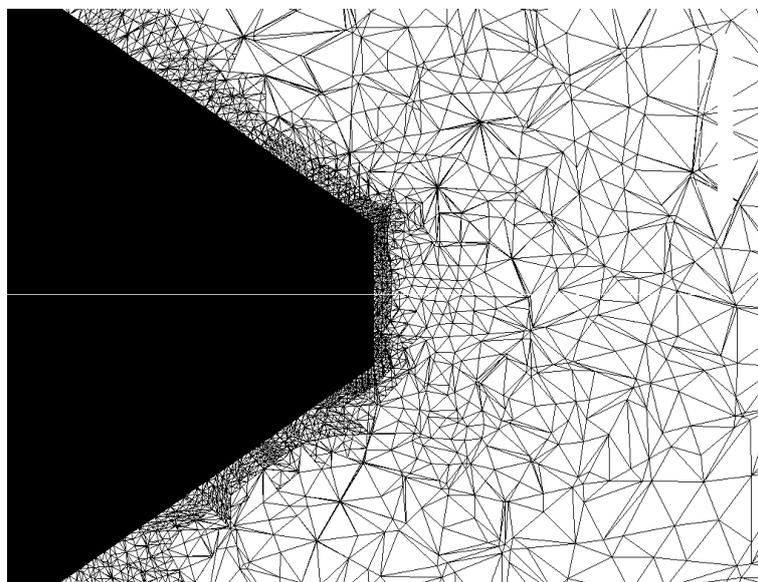
Implementación

La implementación del procedimiento de gestión de propiedades sobre las entidades geométricas de la malla puede describirse en las siguientes etapas:

1. Asignación de un *flag* (un número de indentificación) que permite asociar una propiedad a una entidad geométrica: la asignación del *flag* a la entidad se realiza a partir de determinar que todos los vértices que definen a la entidad geométrica pertenecen a una lista de vértices proporcionados como argumento a la función que realiza la asignación del *flag*. El *flag* es implementado como un miembro dato en las clases que representan a las distintas entidades geométricas (observar las declaraciones de las clases *edge*, *face* y *elem* en la sección 5.2).
2. Heredado del *flag* de propiedad en la etapa de refinamiento: se lleva a cabo asignando a las hijas de una entidad geométrica el mismo *flag* que su madre, tal cual se observa en la figura 6.14. De esta forma, por ejemplo, si el *flag* f_1 está asociado a la fijación del valor de algún grado de libertad del problema sobre la arista $e = \overline{n_1 n_2}$, entonces dicha fijación se hereda a las aristas hijas $e_2 = \overline{m n_2}$ y $e_1 = \overline{n_1 m}$ mediante la asignación del mismo *flag* f_1 . Un tratamiento idéntico se realiza en la etapa de refinamiento de caras y elementos. La figura 6.14 representa esta situación.
3. Identificación en la etapa de pos-refinamiento de todas las entidades geométricas que poseen una misma propiedad asignada. Para ello se realiza una búsqueda por medio de los miembros datos *flags* correspondientes a las entidades geométricas *activas* en el nivel de refinamiento actual (el concepto de entidad *activa* / *inactiva* es el aquél dado en la sección 6.2).
4. Aplicación de una operación o conjunto de operaciones sobre dichas entidades en función del significado del *flag*: el usuario del código es quien asigna un significado al *flag* y por lo tanto sabe como procesar a las entidades correspondientes. En este sentido la implementación de la estrategia de adaptación proporciona funciones que realizan las operaciones usualmente utilizadas, por ejemplo: ajustar las coordenadas de los nodos que surgen sobre las aristas que se encuentran sobre un contorno curvo, generar los archivos con información sobre las condiciones de borde requeridas por el solver del flujo, etc.



(a) Malla sin refinar



(b) Malla con dos niveles de refinamiento

Figura 6.11: Refinamiento de la capa de elementos que están en contacto con un cuerpo cónico truncado. Corte de la malla en el plano de simetría.

Por otro lado, caben explicarse algunas situaciones vinculadas con la corrección de las coordenadas de los vértices que surgen sobre contornos curvos sólo en el caso de mallas tridimensionales. Cuando el refinamiento introduce un nodo sobre una arista (o en el centro de una cara para elementos hexaédricos) perteneciente o asociada a un contorno curvo, si el nodo es irregular entonces sus coordenadas no deben ser ajustadas al contorno. Esto sólo es permitido si el nodo deja de ser irregular, ya que de lo contrario se presentaría la situación que se observa en la figura 6.13. En ella se aprecia que la malla de superficie ha quedado con una “abertura” debido al desplazamiento del nodo irregular I hacia la superficie curva asociada con la arista \bar{ab} . Esta situación es contemplada en la lógica del posrefinamiento. Para determinar si el nodo I es irregular o no, se verifica el estado (miembro `state` de la clase `edge` - sección 5.2) de la arista \bar{ab} . Si el estado de la arista es *activo* entonces el nodo medio que la misma posea es nodo irregular.

Por otro lado, se observa que al modificar las coordenadas de los vértices ajustándolas al contorno, ya no es posible asegurar el comportamiento de la calidad elemental que se menciona en las secciones 2.2.1 y 2.2.2. Además, se debe tener en cuenta que si la discretización inicial que aproxima al contorno curvo no es la adecuada, es factible que el procedimiento de ajuste de coordenadas produzca elementos con volumen cero o que se encuentran “invertidos” (el determinante de la matriz Jacobiana del elemento es menor o igual a cero - ver sección 2.1.1). La figura 6.12 muestra esta situación para el triángulo $t = \{1, 2, 3\}$ que aproxima al contorno curvo C . El refinamiento de este triángulo genera los triángulos $t1$, $t2$, $t3$ y $t4$. Luego, el desplazamiento del vértice V hacia el contorno hace que los triángulos $t1$, $t2$ y $t3$ queden invertidos. Esta situación impide avanzar con el cálculo de la solución y puede ser solucionada aplicando técnicas de *untangling* luego de ajustar las coordenadas del vértice C al contorno. Procedimientos de este tipo aún no han sido incluidos en la estrategia de adaptación, pero se contempla dicha posibilidad a través de las estrategias de *untangling* y *smoothing* simultáneos presentadas en [LNS06], la cual forma parte del código PETSc-FEM utilizado en esta tesis.

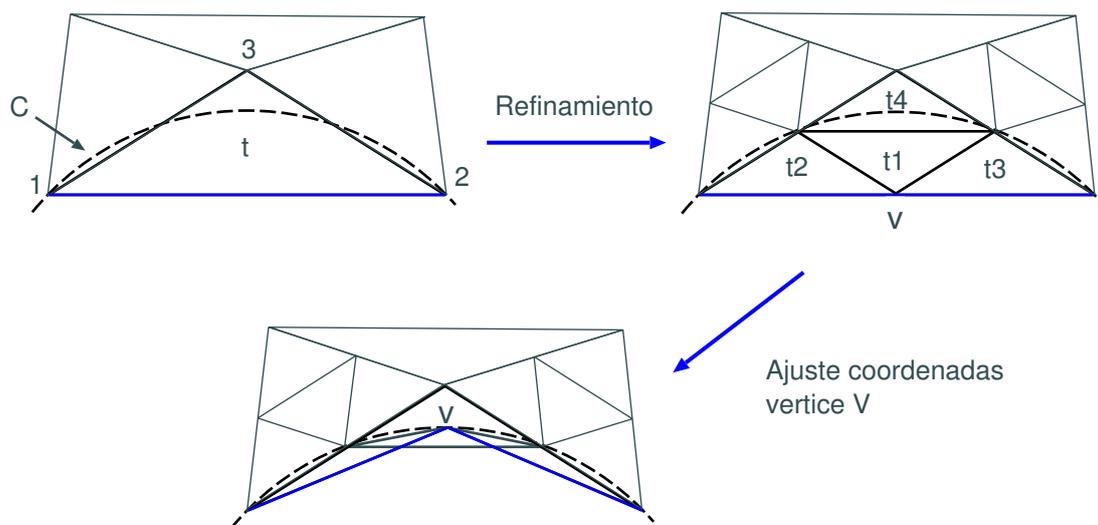


Figura 6.12: Generación de elementos invertidos producido por el ajuste de coordenadas sobre un contorno curvo.

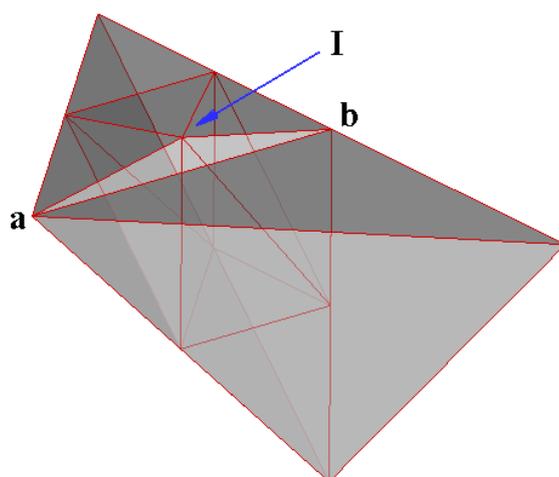
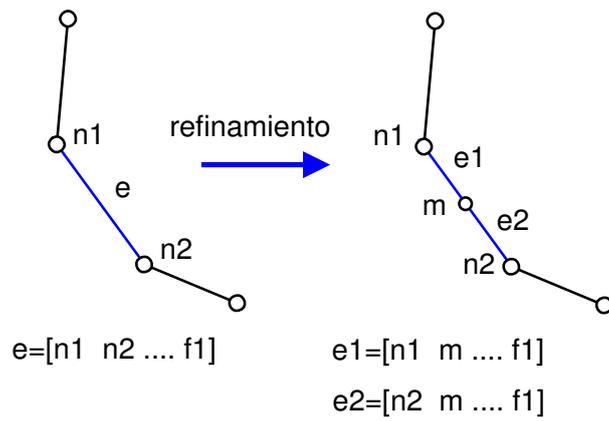
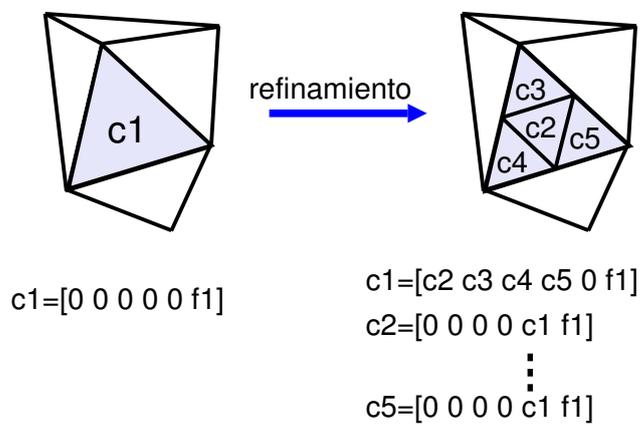


Figura 6.13: Situación no permitida - Proyección de nodo irregular sobre una superficie.



(a) Herencia de propiedades sobre aristas.



(b) Herencia de propiedades sobre caras.

Figura 6.14: Gestión de propiedades sobre aristas y caras.

Capítulo 7

Aplicaciones

Se presentan distintas aplicaciones de la estrategia de adaptación de mallas desarrollada en esta tesis. Las aplicaciones que se resuelven corresponden a problemas típicos de la dinámica de gases. Es decir, son problemas de flujos compresibles, no viscosos, estacionarios o dependientes del tiempo, bidimensionales, axisimétricos o tridimensionales. Las técnicas desarrolladas se utilizan para adaptar mallas bidimensionales y tridimensionales. Para cada ejemplo se describen brevemente la física del problema, las condiciones de borde, la condición inicial (cuando corresponda), las características de la malla, los parámetros utilizados por el código que resuelve las ecuaciones del flujo y el criterio de selección de los elementos a refinar. Los resultados obtenidos en forma numérica se evalúan desde el punto de vista de la precisión en el cálculo. Si el problema se resuelve sobre mallas de tetraedros entonces también se analiza la calidad de las mallas obtenidas por la adaptación. Finalmente, cuando es posible, se analiza la performance del código de adaptación. Cada ejemplo permite evaluar distintas características de la estrategia de adaptación desarrollada y los resultados obtenidos en relación al campo de solución son comparados con aquellos provenientes del área experimental y de la teoría de la dinámica de gases.

7.1. Flujo transónico bidimensional estacionario sobre una cuña

Se propone resolver el flujo transónico que se desarrolla cuando una corriente libre supersónica impacta sobre una cuña. Se asume que el flujo es bidimensional, compresible y no viscoso. También se considera que el fluido se comporta como un gas perfecto. La hipótesis de flujo bidimensional implica que la longitud de la cuña es infinita en la dirección perpendicular a la corriente libre. Las condiciones del problema son tales que se genera una onda de choque curva despegada del vértice de la cuña. Por detrás de dicha onda el Mach del flujo varía desde el régimen subsónico al supersónico. El criterio que controla la adaptación de la malla es el indicador de error basado en el gradiente elemental de la presión. Se comparan, sobre el eje de simetría,

los valores de las variables del flujo inmediatamente detrás de la onda de choque con los valores teóricos correspondientes a un choque recto y se analiza la posición de la onda de choque.

7.1.1. Descripción del problema

La teoría de la dinámica de gases[Sha53] y los ensayos experimentales en túneles de viento determinan que la naturaleza del flujo y el patrón de ondas de choque que se desarrollan alrededor de una cuña bidimensional dependen principalmente de tres parámetros: el número de Mach de la corriente libre M_1 , el ángulo con el cual el flujo incide sobre la cuña α y el ángulo de desviación del flujo o ángulo de la cuña θ . Dados los valores de α y θ , se sabe que a partir de un determinado Mach del flujo y para valores menores a este, no existe una solución que se corresponda con un choque fuerte o débil. Esta situación es representada en el diagrama de la figura 7.1 si se considera que estos parámetros adquieren los siguientes valores: $M_1 = 1.5$, $\alpha = 0^\circ$ y $\theta = 25^\circ$. Tal cual se ve en la figura 7.1, una onda de choque oblicua pegada al vértice de la cuña no es capaz de desviar el flujo en estas condiciones (no existe un ángulo de inclinación de la onda β factible como solución). En estos casos los ensayos en túneles de viento [Bry51] permiten determinar que la única solución posible está dada por la existencia de una onda de choque curva, despegada del vértice de la cuña. Por detrás de la onda se distinguen claramente dos regiones del flujo: una de régimen subsónico y otra de régimen supersónico. La región de flujo subsónico está limitada por la onda de choque, el contorno de la cuña y la línea sónica. Esta última indica la posición, siguiendo una línea de corriente, a partir de la cual el flujo es acelerado de subsónico a supersónico. Ya que la onda de choque es curva, el flujo no es uniforme detrás de ella. Luego la línea sónica no será recta sino curva. La onda de choque es normal a la dirección de la corriente libre por delante del vértice, pero a medida que nos alejamos del cuerpo la onda de choque se va curvando. A una distancia considerable del cuerpo la onda de choque se debilita lo suficiente como para ser considerada una onda de Mach (adquiriendo su inclinación) en vez de una onda de choque. El vértice de la cuña es punto de impacto ya que el flujo es subsónico inmediatamente delante de él.

Desde el punto de vista analítico existen varias dificultades para resolver este problema. Una de ellas es que las ecuaciones diferenciales que gobiernan el flujo transónico son del tipo mixto, es decir, son elípticas en la región de flujo subsónico e hiperbólicas en la región de flujo supersónico. Además, las ecuaciones diferenciales son no lineales, parte del contorno (onda de choque y línea sónica) que determina la región de flujo subsónico es parte de la solución y el flujo por detrás de la onda de choque es rotacional. Las soluciones analíticas que se han encontrado implican asumir una serie de hipótesis y resolver el problema en el plano hodógrafo, lo cual elimina algunas de las complicaciones. Las hipótesis que se asumen conducen a la teoría de pequeñas perturbaciones para flujo transónico. Bryson[Bry51] compara los resultados analíticos obtenidos en [VW52] por medio de dicha teoría con aquellos obtenidos experimentalmente. De dicho trabajo se ha extraído la figura 7.2, la cual presenta el comportamiento de la distancia

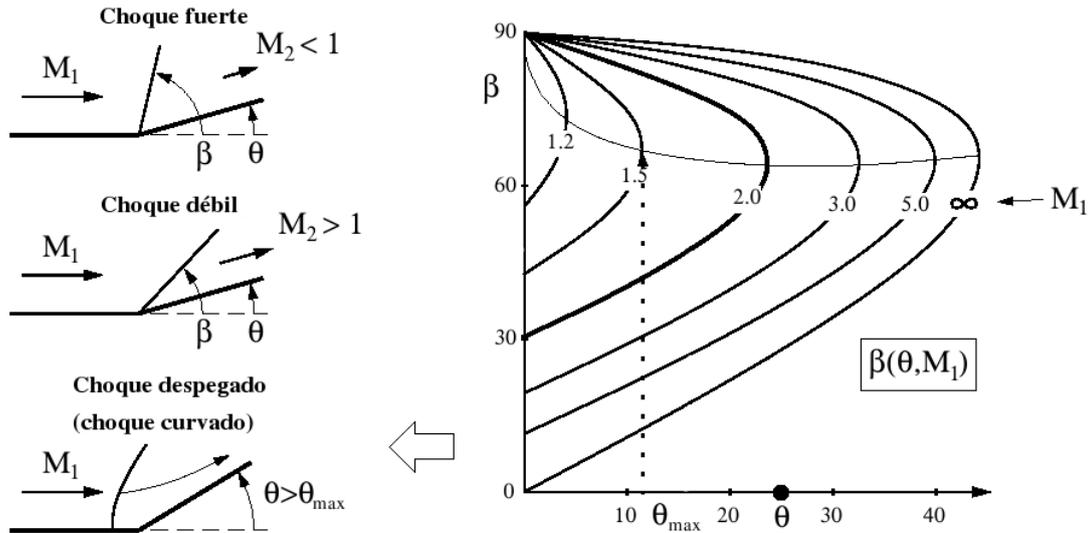


Figura 7.1: Ángulo β de inclinación de la onda de choque vs. ángulo θ de desviación del flujo, parametrizados en el Mach de la corriente libre ($\gamma = 1.4$).

de separación δ de la onda de choque con respecto al vértice de una cuña de cuerda c como una función del número de Mach reducido de la corriente libre ξ_∞ o parámetro de similitud transónica

$$\xi_\infty = \frac{M_\infty^2 - 1}{[(\gamma + 1)t/c]^{2/3}} \quad (7.1)$$

donde $t/c = \tan(\theta)$. Se observa que la distancia de separación de la onda de choque crece rápidamente para valores de $\xi_\infty \lesssim 1.2$. Por otro lado, para los tres semiángulos de cuña que se muestran en la figura 7.2 se observa que no existen diferencias notables en la distancia de separación para el rango $1.0 \lesssim \xi_\infty \lesssim 1.3$. Teniendo en cuenta dicha apreciación y que el valor del Mach reducido para las condiciones del flujo que se resuelven es $\xi_\infty \simeq 1.16$, se puede inferir que la distancia de separación es $(\delta/c)_{exp} \simeq 0.4$.

7.1.2. Solución numérica

Si se tiene en cuenta que el ángulo de ataque α es nulo, el problema puede ser resuelto considerando simetría del flujo. La precisión en el cálculo de la solución obtenida en la vecindad de la onda de choque puede ser evaluada, donde esta es normal a la corriente libre, por medio de las relaciones de Rankine-Hugoniot. Estas relaciones vinculan las variables del flujo por delante y por detrás de una onda de choque recta, normal u oblicua. De esta manera, las variables del flujo inmediatamente detrás de la onda de choque (indicadas con el subíndice 2) sobre el eje de simetría ($y = 0$) poseen los siguientes valores:

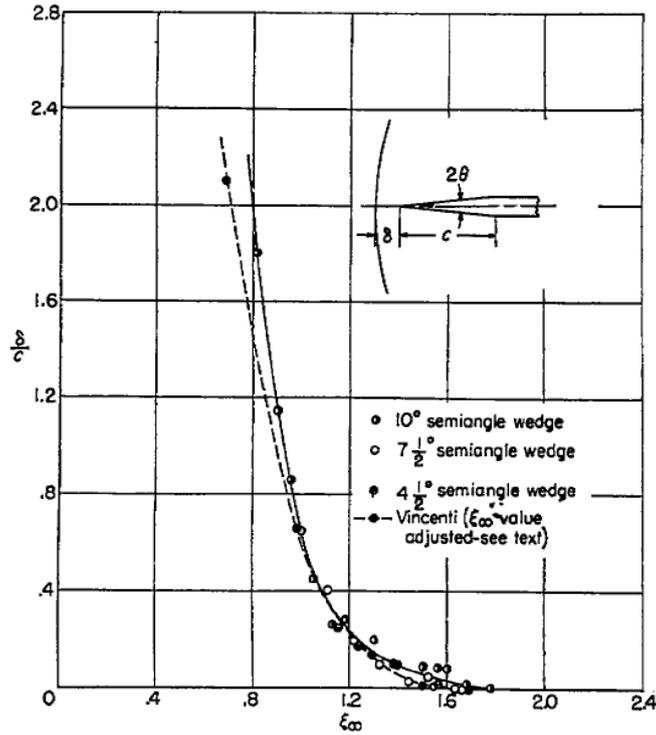


Figura 7.2: Distancia de separación δ/c entre la onda de choque y el vértice de la cuña como función del Mach reducido del flujo sin perturbar ξ_∞ .

$$M_2(y=0) = 0.701088 \quad (7.2)$$

$$\frac{p_2}{p_1}(y=0) = 2.458333 \quad (7.3)$$

$$\frac{\rho_2}{\rho_1}(y=0) = 1.86 \quad (7.4)$$

Para resolver el problema numéricamente, resulta necesario imponer las siguientes condiciones de borde (ver figura 7.3)

- En la sección $\bar{A}\bar{B}$ donde el flujo es entrante se fijan los valores de la presión $p = 0.3176$, la densidad $\rho = 1$ y las componentes de la velocidad $u = 1$ y $v = 0$. Observar que los valores de las variables del flujo se encuentran escalados por conveniencia para realizar la simulación. Los valores de referencia son: $\rho_{ref} = 1.225\text{kg/m}^3$, $u_{ref} = 340\text{m/s}$ y $p_{ref} = 101325\text{Pa}$.
- Los contornos $\bar{B}\bar{C}$ y $\bar{C}\bar{D}$ del dominio computacional donde el flujo es saliente quedan libres, es decir que no se impone condición de borde alguna en ellos.
- Sobre el tramo del contorno $\bar{D}\bar{E}$ que representa al contorno sólido de la cuña se impone la condición de no penetración del flujo. Esto significa que la componente de la velocidad

normal a la superficie de la cuña, evaluada sobre los puntos del contorno Γ_{DE} debe ser cero $(\tilde{\mathbf{V}} \cdot \mathbf{n})|_{\Gamma_{DE}} = 0$.

- La condición de simetría a lo largo del contorno \bar{EA} implica que no puede existir flujo en la dirección normal a la línea de simetría, es decir $(\tilde{\mathbf{V}} \cdot \mathbf{n})|_{\Gamma_{EA}} = 0$.
- Si E es punto de impacto, entonces $\vec{V}(E) = \mathbf{0}$.

Para la discretización inicial del dominio se utiliza una malla de elementos cuadriláteros que consta de 496 nodos y 450 elementos. Dicha discretización se observa en la figura 7.3.

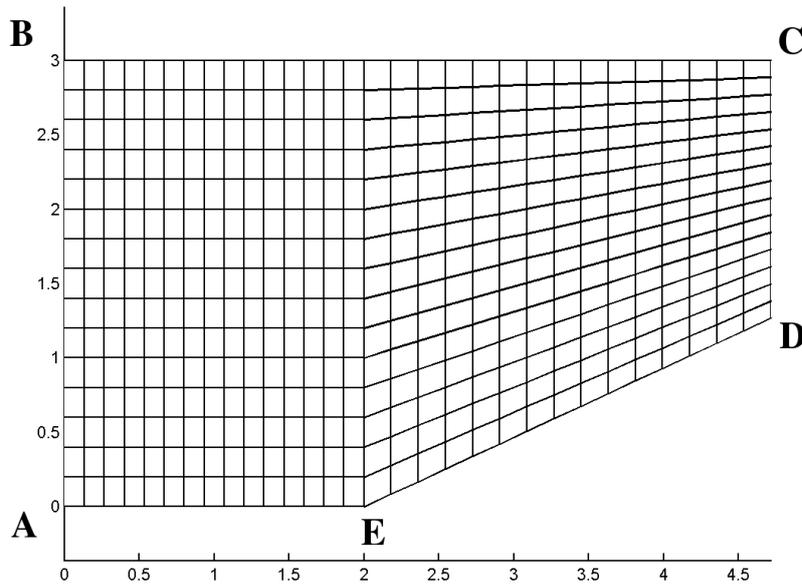


Figura 7.3: Discretización inicial del dominio computacional y condiciones de borde.

Los elementos a refinar se seleccionan utilizando como criterio al indicador de error basado en el gradiente de la solución discreta dado por la Ec.(3.2). Como variable indicadora se elige a la presión. Luego, los elementos seleccionados para refinar son aquellos tales que

$$c_1 \cdot \max_e(\|\nabla_e p\| \cdot h^e) \leq \|\nabla p\| \cdot h \quad (7.5)$$

donde el máximo se aplica a todos los elementos presentes en la discretización. El valor de la constante c_1 es ajustado *ad-hoc* a medida que se adapta la malla.

El criterio de selección anterior es complementado por aquél en el cual se tienen en cuenta los valores nodales de la presión. Dado que las regiones de flujo transónico por detrás de la onda de choque despegada se corresponden a valores de presión (escalados) mayores a 0.7, bajo este criterio se seleccionan todos los elementos tales que sus cuatro vértices poseen un valor de la presión mayor al indicado. El objetivo de esto es tratar de refinar, en las primeras adaptaciones, una región del dominio más extensa y no sólo en la vecindad de la onda de choque. Esta región

incluye a la línea sónica, a través de la cual el flujo vuelve a ser supersónico. Los resultados experimentales [Bry51] indican que la línea sónica debe ubicarse inmediatamente antes del punto de finalización del plano inclinado de la cuña. Por otro lado, en el último refinamiento se seleccionan sólo los elementos que se encuentran sobre la onda de choque seleccionando los elementos con el indicador del gradiente de presiones. Tanto la adaptación de la malla como el flujo son resueltos secuencialmente.

7.1.3. Resultados

El procedimiento de solución adaptativo controlado por los criterios mencionados genera las siguientes mallas¹:

- Primer refinamiento: 1141 nodos y 1044 elementos (Fig.7.4) - $c_1 \simeq 0.1$ y $p_V \geq 0.7$.
- Segundo refinamiento: 2851 nodos y 2670 elementos (Fig.7.5) - $c_1 \simeq 0.14$ y $p_V \geq 0.7$.
- Tercer refinamiento: 3251 nodos y 2991 elementos (Fig.7.6) - $c_1 \simeq 0.18$.

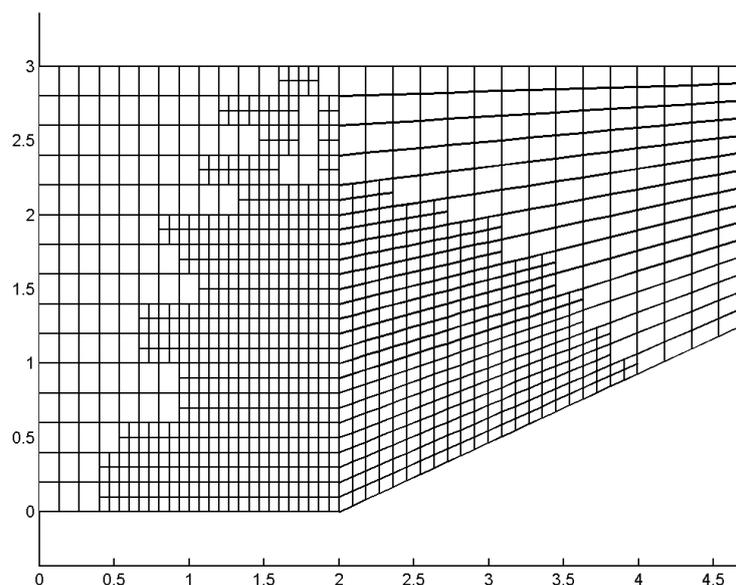


Figura 7.4: Malla adaptada - Primera iteración.

La solución estacionaria para los campos de presión y número de Mach se presentan en las figuras 7.7.a) y 7.7.b). Por otro lado, las soluciones correspondientes sobre el eje de simetría se observan en las figuras 7.8.a) y 7.8.b).

Si se comparan los valores de las variables del flujo que se obtienen numéricamente con aquellos que indican las relaciones de choque recto, se observa una buena concordancia entre

¹Se indica el valor de la constante c_1 utilizado para elegir los elementos en la malla previa.

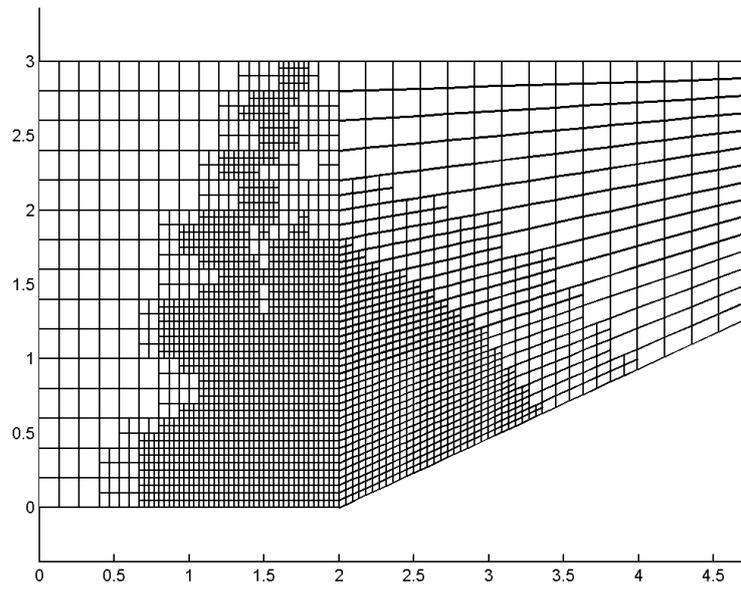


Figura 7.5: Malla adaptada - Segunda iteración.

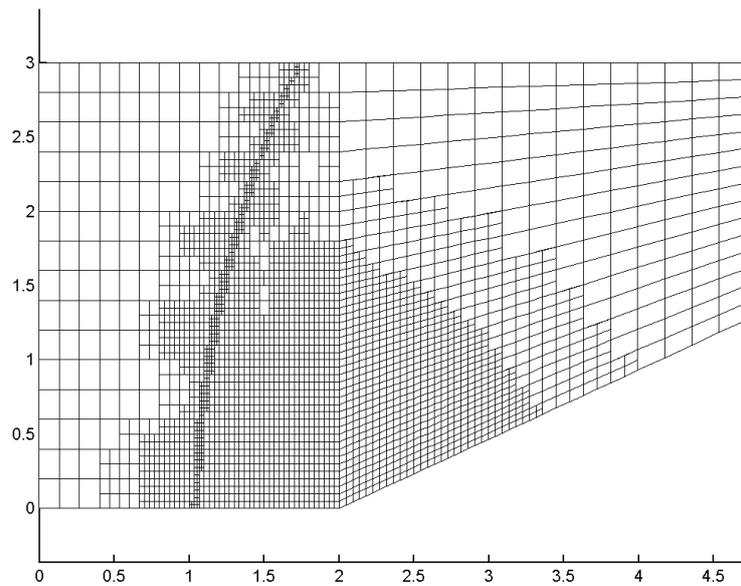
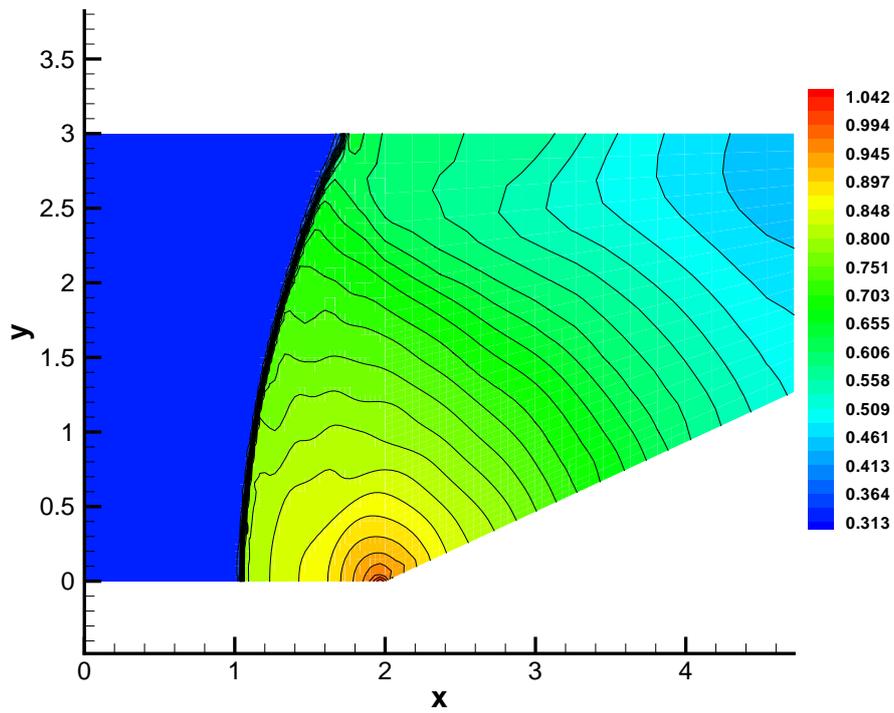
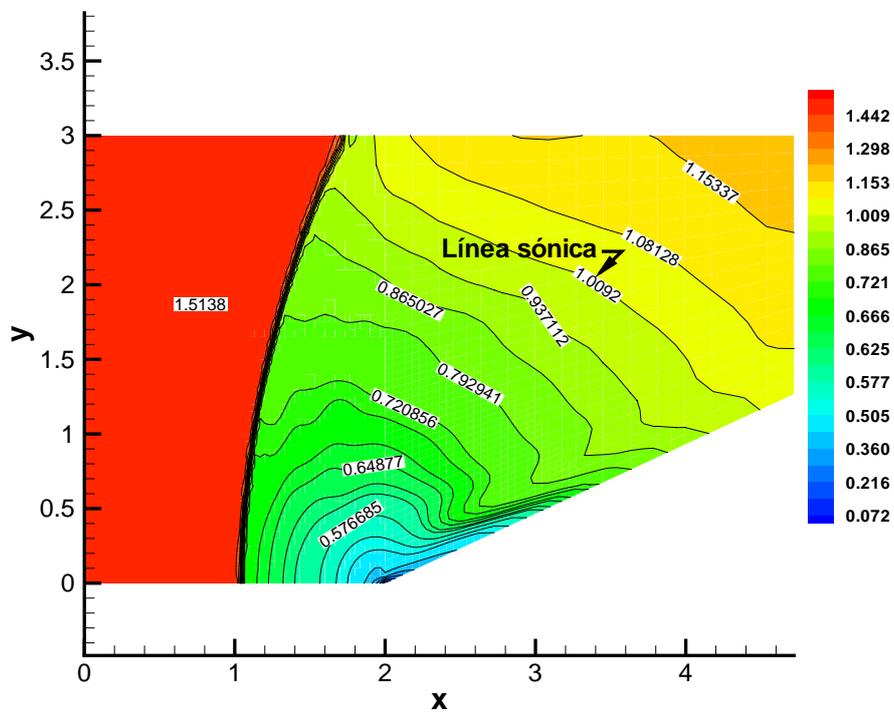


Figura 7.6: Malla adaptada - Tercera iteración.

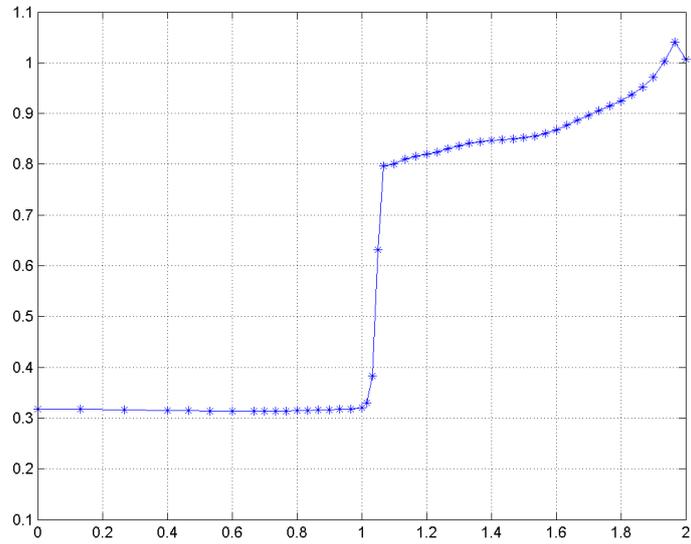


(a) Distribución de presiones.

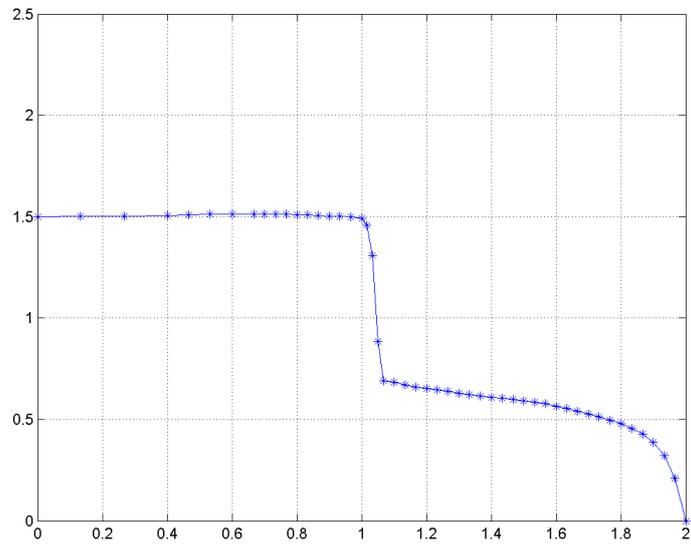


(b) Distribución de Mach.

Figura 7.7: Solución computada para la tercera iteración en el refinamiento.



(a) Distribución de presiones sobre el eje de simetría.



(b) Distribución del número de Mach sobre el eje de simetría.

Figura 7.8: Soluciones sobre el eje de simetría hasta el punto de impacto de la cuña.

ambos resultados. Las relaciones entre los valores de las variables del flujo por delante y por detrás de la onda de choque obtenidos numéricamente son los siguientes

$$\begin{aligned}\frac{p_2}{p_1} &\simeq \frac{0.795}{0.32} = 2.484 \\ \frac{\rho_2}{\rho_1} &\simeq 1.9 \\ M_2 &\simeq 0.685\end{aligned}$$

Los errores porcentuales, calculados en relación a los valores teóricos, son: para la relación de presiones 1 %, para la relación de densidades 2.1 % y para el Mach detrás de la onda 2.3 %. En función de estos errores se puede inferir que al menos en esta región del campo de movimiento el flujo es resuelto con precisión. Por otro lado, en las figuras 7.8.a) y 7.8.b) se aprecia que la onda de choque es resuelta en un espesor menor a 4 elementos de tamaño 1/16 el tamaño de los elementos en la misma región de la malla de partida. Finalmente, si se analiza la posición de la onda de choque sobre el eje de simetría, la distancia de separación obtenida en la simulación es $(\delta/c)_{sim} \simeq 1.0/2.65 = 0.377$. Comparando este valor con aquél obtenido a partir de los datos experimentales de la figura 7.2, se observa que la posición de la onda obtenida en la simulación es correcta.

$$e_\delta = \frac{(\delta/c)_{sim} - (\delta/c)_{exp}}{(\delta/c)_{exp}} \simeq 5.75\% \quad (7.6)$$

7.2. Flujo transónico tridimensional estacionario alrededor de un cuerpo cónico

En esta ocasión, se aplica el esquema de solución adaptativo al cálculo del flujo transónico tridimensional, no viscoso y estacionario de un gas perfecto sobre un cuerpo cónico. Las características del flujo que se desarrolla en la cercanía del cuerpo son tales que se presentan regiones con carácter supersónico y subsónico. El objetivo que se plantea al resolver este ejemplo consiste en evaluar el estimador de error basado en la reconstrucción del gradiente que se describe en la sección 3.2.1 de esta tesis. En principio, se desea determinar si el método implementado permite detectar adecuadamente las regiones de la malla que deben ser refinadas. Para evaluar la precisión en el cálculo de la solución computada se recurre a resultados procedentes de ensayos experimentales.

7.2.1. Descripción del problema

El problema del flujo supersónico alrededor de cuerpos geométricos simples ha sido bien analizado, en sus inicios por experimentos en túneles de viento y luego por métodos teóricos como el método de las características y el plano hodógrafo[Sha53]. Sin embargo, el método teórico no permite resolver todos los regímenes posibles del flujo transónico. En [Sol54] se presentan resultados de una investigación experimental del flujo transónico sobre cuerpos que poseen una punta cónica y que continúan en forma cilíndrica, tal como el ilustrado en la figura 7.9.

Se ha determinado que es posible encontrar una solución al problema del flujo supersónico sobre un cuerpo cónico aplicando el método de las características, pero la misma deja de ser válida cuando el flujo sobre la superficie del cono es subsónico. Es más, para un ángulo dado del cono θ_c no existe una solución teórica si el Mach de la corriente libre M_∞ disminuye por debajo de un determinado valor. Este valor de M_∞ se corresponde con la condición para la cual la onda de choque que se forma en el vértice del cono comienza a despegarse. Varios resultados experimentales sobre la distancia de despegue de la onda de choque han sido recopilados en [Sol54] y son reproducidos en la figura 7.10. En la misma se observa que la posición de la onda de choque despegada depende solamente del diámetro d del cono cuando la distancia que separa a la onda del vértice es grande. Por otro lado, cuando la onda de choque se encuentra cerca del vértice del cono, entonces también existe una dependencia en el ángulo del cono θ_c . Además, los resultados experimentales determinan que existen situaciones en las cuales una onda de choque curva pegada al vértice del cono es capaz de proporcionar los gradientes de presiones necesarios como para desviar el flujo cerca del vértice del cono y que esto ocurre a valores de M_∞ menores a los que indica la teoría para que ocurra el despegue de la onda de choque. En definitiva, los experimentos permiten afirmar únicamente que el despegado de la onda de choque, para un ángulo dado del cono, no puede ocurrir para valores de M_∞ mayores a los que indica la teoría. O bien, para un M_∞ dado, la separación de la onda de choque puede

ocurrir para ángulos del cono algo más grandes a los que indica la teoría, pero no menores.

El régimen del flujo transónico que se pretende resolver en forma numérica es aquél indicado en línea de trazo discontinuo en la gráfica de la figura 7.10. Por lo tanto, es de esperar que el choque se encuentre despegado del vértice del cono, tal como lo muestran las figuras 7.9 ó 7.11. Estas figuras se incluyen sólo a los fines de mostrar cuál es el patrón del flujo que se espera.

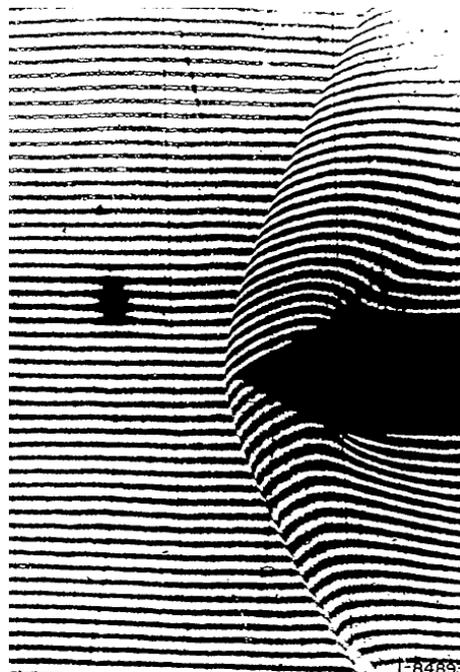


Figura 7.9: Patrón del flujo para un cuerpo cónico-cilíndrico en una corriente libre supersónica en el infinito obtenido con la técnica de interferograma - Ref [Sol54].

7.2.2. Solución numérica

El cálculo de la solución en forma adaptativa se inicia a partir de una malla no estructurada de tetraedros que posee 165688 elementos y 30873 nodos, generada con el software Netgen[SGG04]. Esta malla posee una distribución en el tamaño de sus elementos tal que los mismos se hacen más pequeños en la vecindad del vértice del cono. Se destaca que el cuerpo cónico no continua en un cilindro, tal cual se muestra en la figura 7.11, sino que la base del cono es coincidente con la sección de salida del flujo. De esta manera, la expansión que se observa cuando el flujo pasa del tramo cónico al cilíndrico no estará presente en la simulación numérica.

El dominio computacional es cilíndrico y su eje de simetría es coincidente con el eje del cono. La figura 7.12 presenta el detalle de un corte de la malla de partida en la región del cono. El semiángulo del cono es $\theta_c = 35^\circ$, el Mach del flujo en la corriente libre es $M_\infty = 1.4$ y el cono presenta un ángulo de incidencia $\alpha = 0^\circ$. Para estas condiciones del problema los resultados de los ensayos experimentales que se observan en las figuras 7.9 y 7.10 permiten evaluar ciertas características del flujo tales como el patrón de ondas que se desarrolla alrededor del cuerpo y la posición de la onda de choque despegada.

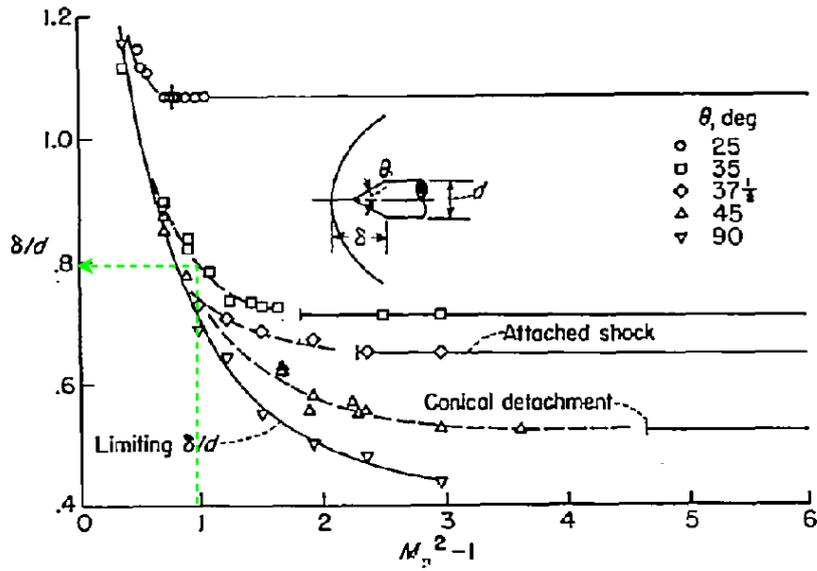


Figura 7.10: Posición δ de la onda de choque respecto del punto sónico para distintas configuraciones θ_c y Mach de la corriente libre. Resultados experimentales.

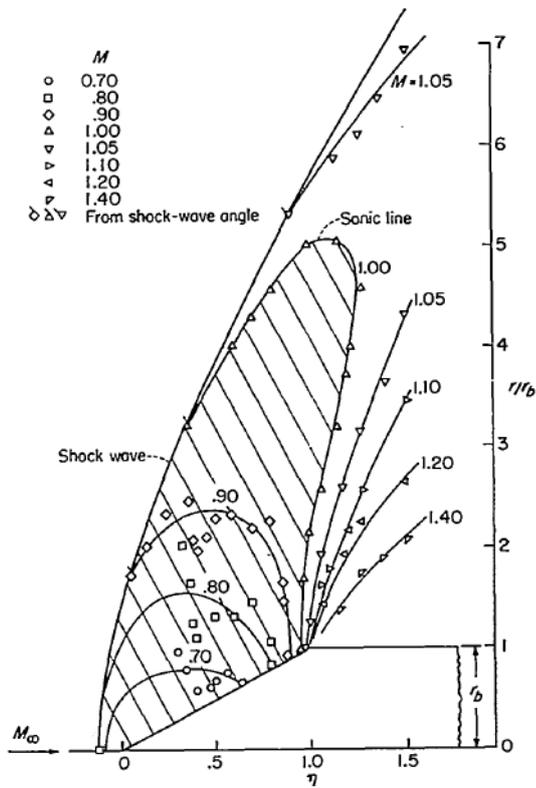


Figura 7.11: Campo de Mach obtenido del análisis posterior de un ensayo interferométrico - Ref [Sol54].

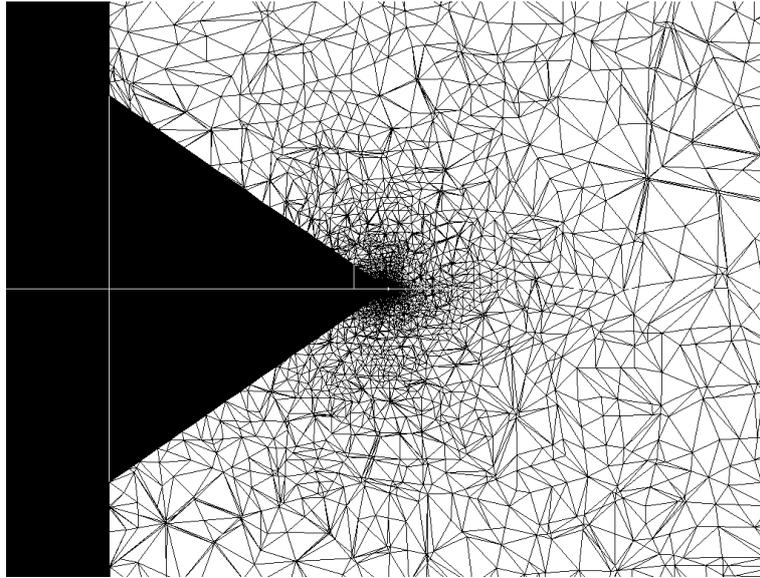


Figura 7.12: Corte de la malla de partida sobre un plano de simetría.

El flujo es resuelto en forma tridimensional aún cuando presenta simetría axial. Las ecuaciones de Euler 3-D se resuelven en paralelo, utilizando 15 procesadores, con el código de advección-difusión implementado en PETSc-FEM. Las condiciones de borde que se imponen para resolver el problema son las siguientes:

- Dado que el flujo en la sección de entrada es supersónico, se fijan los valores de todas las variables en dicha sección. Estos valores, escalados por conveniencia, son los siguientes: densidad $\rho = 1$, componente de la velocidad en la dirección del eje de simetría (eje X) $u = 1$, componentes de la velocidad en las direcciones perpendiculares (ejes Y y Z) $v = 0$, $w = 0$, presión $p = 0.364$. Los valores de referencia son: $\rho_{ref} = 1.225\text{kg/m}^3$, $p_{ref} = 101325\text{Pa}$, y $T_{ref} = 288\text{K}$ y $V_{ref} \simeq 340\text{m/s}$. Por otro lado, dado que los resultados experimentales indican que el flujo subsónico se ve confinado a aquella región limitada por la superficie sónica y el vértice del cono, la sección de salida del flujo debería ser totalmente supersónica. Por este motivo en la sección de salida no se fija el valor de ninguna variable del flujo.
- Ya que se asume que el flujo es no viscoso, se impone la condición de deslizamiento $\vec{V} \cdot \vec{n} = 0$ sobre la superficie del cono. Para gestionar dicha condición de borde, el procedimiento de adaptación requiere la lista de nodos que se encuentra sobre la superficie del cono (ver sección 6.4). Esta lista de nodos también es utilizada para modificar las coordenadas de los nodos que surgen sobre las aristas que pertenecen a la superficie del cono en la etapa de pos-refinamiento.

Las condiciones iniciales para las variables del flujo en todo el dominio son coincidentes con los valores fijados en la sección de entrada. El criterio de selección de los elementos a refinar utiliza la norma-L2 del error en el gradiente, es decir que se refinan todos aquellos elementos tales que

$$0.15 \cdot \max_{K=1}^{Nel} (\| e \|_{L_2(\Omega_K)}) \leq \| e \|_{L_2(\Omega_K)} \quad (7.7)$$

$$\leq \| Gu - \nabla u^h \|_{L_2(\Omega_K)}^2 \quad (7.8)$$

donde la variable del flujo u elegida para evaluar el error en el gradiente es el número de Mach. Se estipula un máximo de 3 niveles de refinamiento.

7.2.3. Resultados

Se analiza si el patrón de ondas obtenido en la simulación numérica se corresponde con los resultados experimentales que se presentan en las figuras 7.9 y 7.11. La figura 7.14 presenta la solución estacionaria para el número de Mach sobre la malla más refinada. En ella se observa que efectivamente se establece una onda de choque curva que se encuentra despegada del vértice del cono. Tal como lo indican los resultados experimentales, para esta condición del flujo el Mach sobre toda la superficie del cono es subsónico. La región de flujo subsónico se ve limitada por la superficie sónica ($M = 1$) y la superficie del cono. Sin embargo, a diferencia de lo que indican los resultados experimentales para las condiciones dadas del flujo (patrón de flujo dado en la figura 7.11), la superficie sónica no se origina en la onda de choque sino que parte desde el vértice del cono y finaliza en la intersección de la superficie cónica con la sección de salida de flujo. Luego existe una región de flujo supersónico entre la onda de choque y la superficie sónica que se asemeja más al cuadro que se presenta en la figura 7.13. Dicho patrón para la línea sónica se corresponde con la situación donde *la onda de choque curva se encuentra pegada al vértice del cono*. Los resultados experimentales que se presentan en Ref.[Sol54] permiten determinar la existencia de este flujo, en el que ocurre una compresión del flujo de supersónico a subsónico entre la onda de choque y la superficie sónica sin la presencia de choques (esta situación se conoce con el nombre de *shock-free compression*) y se corresponde con un número de Mach de la corriente libre mayor al utilizado en la simulación pero a la vez menor al que predicen los resultados experimentales de la figura 7.10, a partir del cual la onda de choque se pega al vértice del cono y comienza a ser válida la solución teórica de flujo cónico ($M_{\infty,attachment} \simeq 1.67$). Además, y coincidiendo con las observaciones experimentales, la compresión shock-free ocurre precisamente cuando la región de flujo subsónico está completamente encerrada por la superficie sónica y la superficie del cuerpo.

La figura 7.15 presenta las isosuperficies de Mach para los valores $M = 1.39$ y $M = 1.0$ correspondientes a la solución computada sobre la malla con mayor refinamiento, indicando respectivamente las posiciones de la onda de choque y de la superficie sónica. También se observa la simetría del flujo obtenida en la simulación.

Por otro lado, se evalúa la posición de la onda de choque sobre el eje de simetría. Para ello se cuenta con información obtenida a partir de ensayos experimentales, la cual es resumida en la figura 7.10. En ella se indica que para un ángulo del cono $\theta_c = 35^\circ$ y para un número de

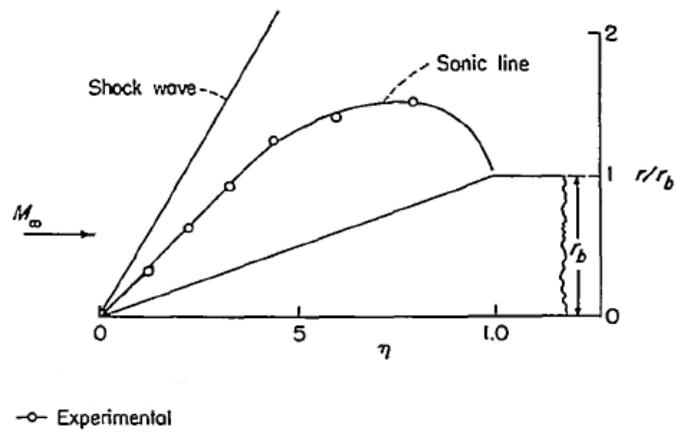


Figura 7.13: Posición de la línea sónica para la onda de choque curva pegada al vértice del cono - Ref [Sol54].

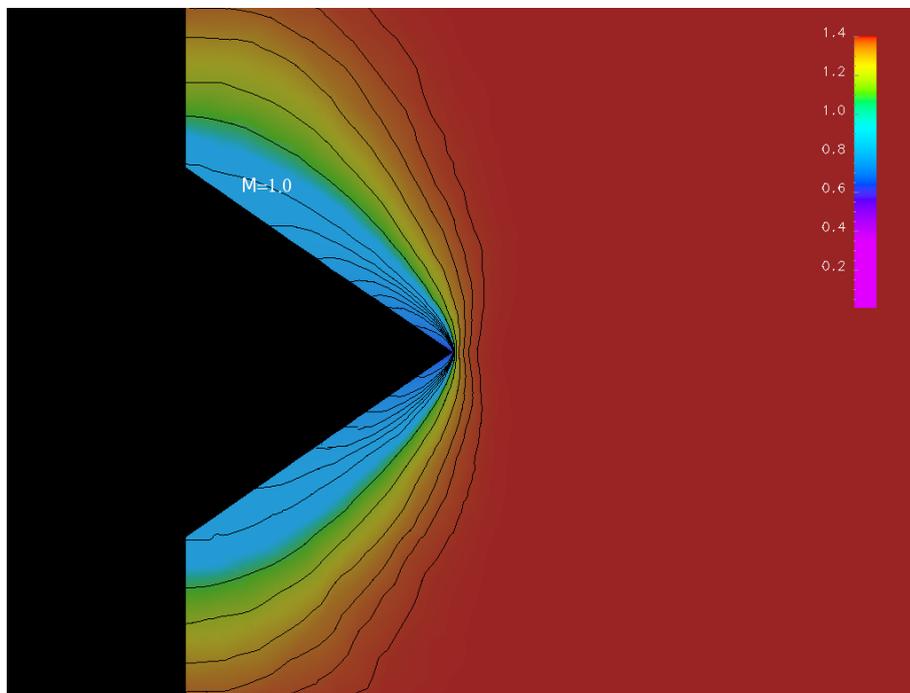


Figura 7.14: Líneas de contorno para el número de Mach - Solución correspondiente al máximo nivel de refinamiento.

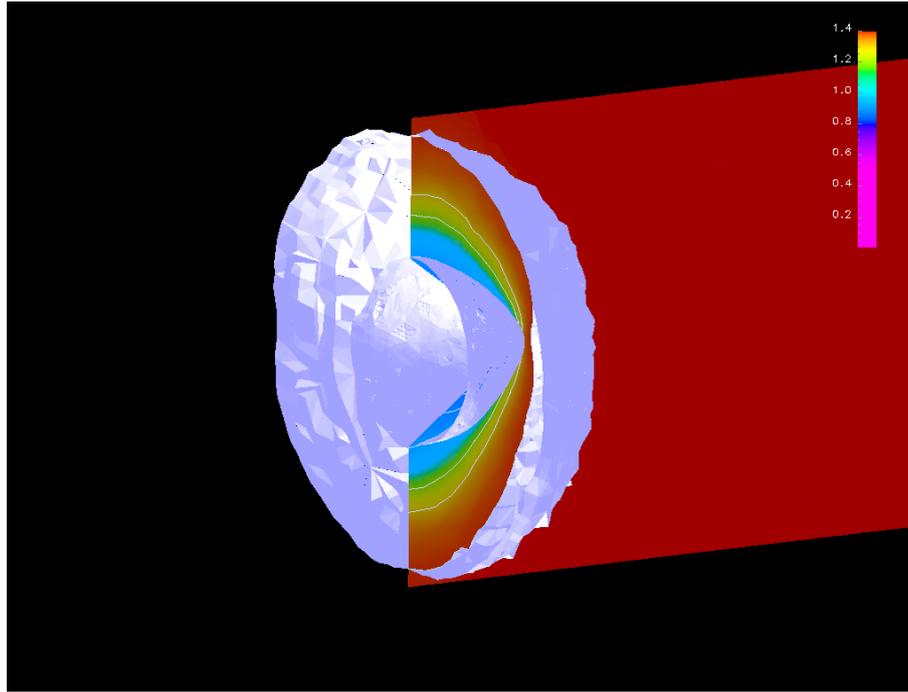


Figura 7.15: Isosuperficies de Mach para la superficie sónica $M = 1.0$ y el frente de onda $M = 1.39$ - Solución correspondiente al máximo nivel de refinamiento.

Mach de la corriente libre igual a $M_\infty = 1.4$, el parámetro $\delta/d \simeq 0.8$, donde δ es la distancia medida desde la base del cono hasta la posición de la onda de choque sobre el eje de simetría y d es el diámetro de la base del cuerpo cónico. Dado que el cono que se utiliza en la simulación numérica posee un diámetro $d = 0.8$, entonces se tiene que $\delta \simeq 0.64$. El detalle del flujo en el vértice del cono que presenta la figura 7.16 permite observar que la posición de la onda obtenida en forma numérica es $\delta_{num} \simeq 0.625$, donde se considera que la posición de la onda está dada por la isolínea de Mach correspondiente al valor 1.39. En este sentido, se observa que existe una buena concordancia entre los resultados numéricos y experimentales.

En las figuras 7.17, 7.18 y 7.19 se pueden observar los cortes en el plano xz de las mallas obtenidas con el procedimiento de adaptación. Si estas se comparan con las figuras 7.20, 7.21 y 7.22, correspondientes al gradiente de Mach reconstruido con el procedimiento que se describe en la sección 3.2.1, se concluye que el refinamiento localizado de la malla no se produce únicamente en las regiones con los gradientes más grandes. Dado que el criterio de selección no es el indicador de error basado en el gradiente sino aquél dado por la Ec.7.7, se concluye que este último resulta más efectivo que el primero. A los fines de poder comparar ambos criterios se incluyen las figuras 7.23, 7.24 y 7.25. En ellas se presentan en mapas de colores la norma-L2 del error local en el gradiente de Mach para las soluciones obtenidas sobre la malla base, con uno y dos niveles de refinamiento respectivamente. Observar en las escalas de colores de dichas figuras que el valor máximo de la $\|e\|_{L_2(\Omega_K)}$ disminuye a medida que aumenta el grado de refinamiento de la malla. Por otro lado, si se hubiera utilizado el indicador basado en el gradiente, entonces los elementos seleccionados para refinar se hubiesen concentrado preferentemente en

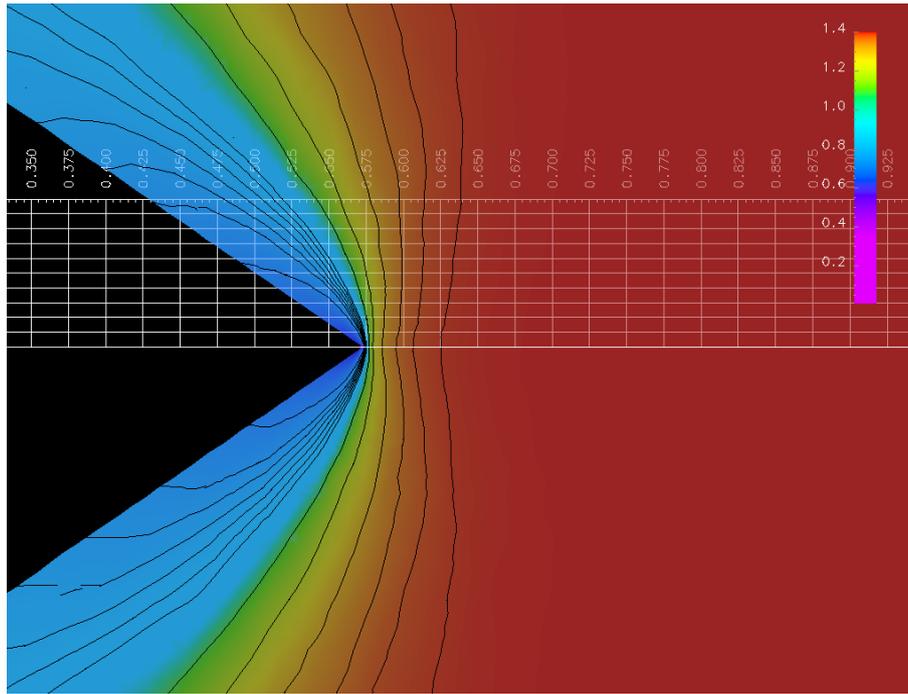


Figura 7.16: Detalle de las líneas de contorno de Mach en el vértice del cono.

la vecindad del vértice del cono, mientras que la región del flujo transónico más alejada de la onda no hubiese sido refinada (tener en cuenta que en la escala de colores del gradiente de Mach se ha utilizado un valor de corte igual a $1.0e+5$ y que la misma ha sido saturada en los valores más bajos de la escala para poner en evidencia las regiones del flujo donde el gradiente de Mach posee valores pequeños).

Finalmente, las figuras 7.26 y 7.27 permiten comparar los cambios que se obtienen en los campos de Mach y presiones cuando se calcula la solución estacionaria sobre una malla idéntica a la malla base utilizada en el procedimiento adaptativo y cuando se aplica dicho esquema de adaptación con tres refinamientos. En ambos casos el flujo es resuelto con mayor precisión si se utiliza el esquema adaptativo, especialmente en la región comprendida entre la onda de choque y la línea sónica.

7.2.4. Calidad de la mallas adaptadas

La calidad de las mallas obtenidas con el procedimiento de adaptación es evaluada siguiendo los lineamientos que se describen en la sección 2.2. De esta manera, se considera que la distribución de la calidad de los elementos evaluada mediante la métrica de calidad $\eta(T)$ dada por la ecuación (2.14) sigue una distribución normal. Luego, la tabla 7.1 presenta un resumen de los valores que caracterizan a estas distribuciones. Se observa que la calidad mínima de las mallas no disminuye con los refinamientos, por lo cual esta medida no es suficiente para caracterizar la calidad de las mallas en este caso. Sin embargo, tanto los valores de la calidad media como los de la desviación estándar mejoran respecto de los de la malla de partida (aproximadamente 2 % de incremento para $\bar{\eta}$ y 9 % de disminución en la η_{Std}). Las figuras 7.28 a), b), c) y d) presentan

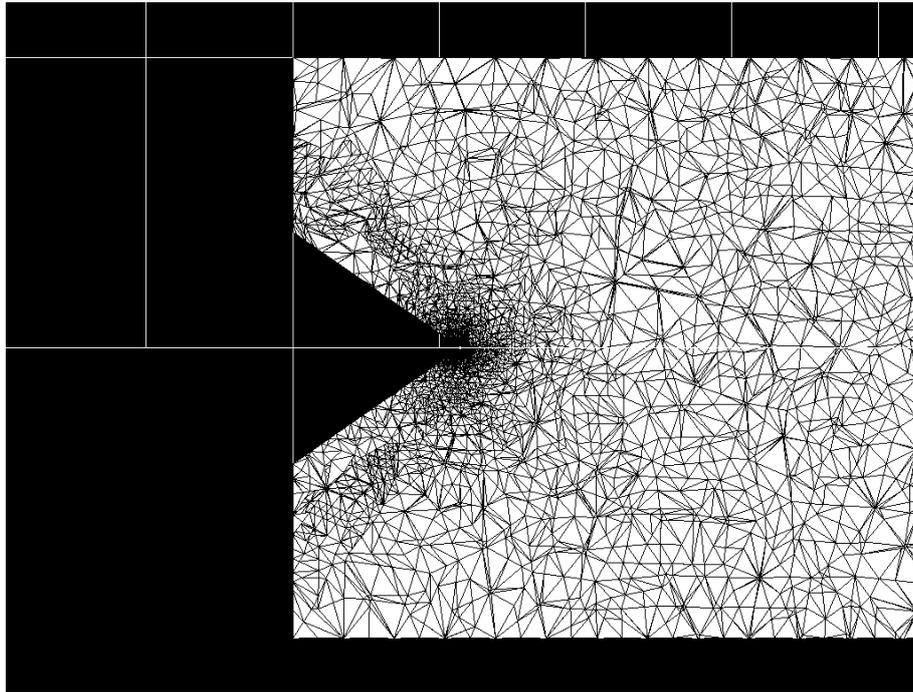


Figura 7.17: Corte en un plano de simetría de la malla adaptada - Primer refinamiento.

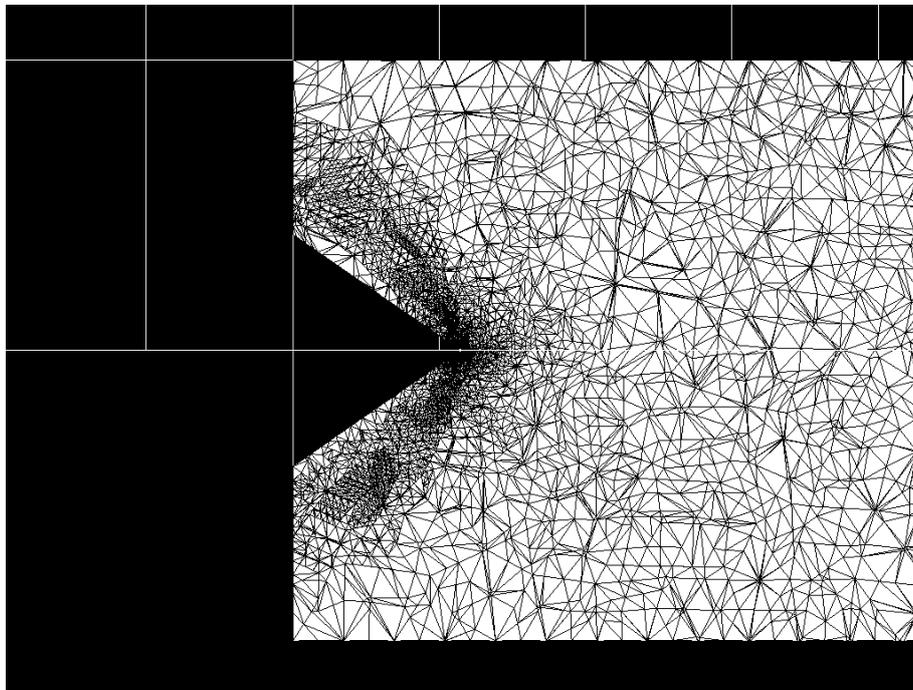


Figura 7.18: Corte en un plano de simetría de la malla adaptada - Segundo refinamiento.

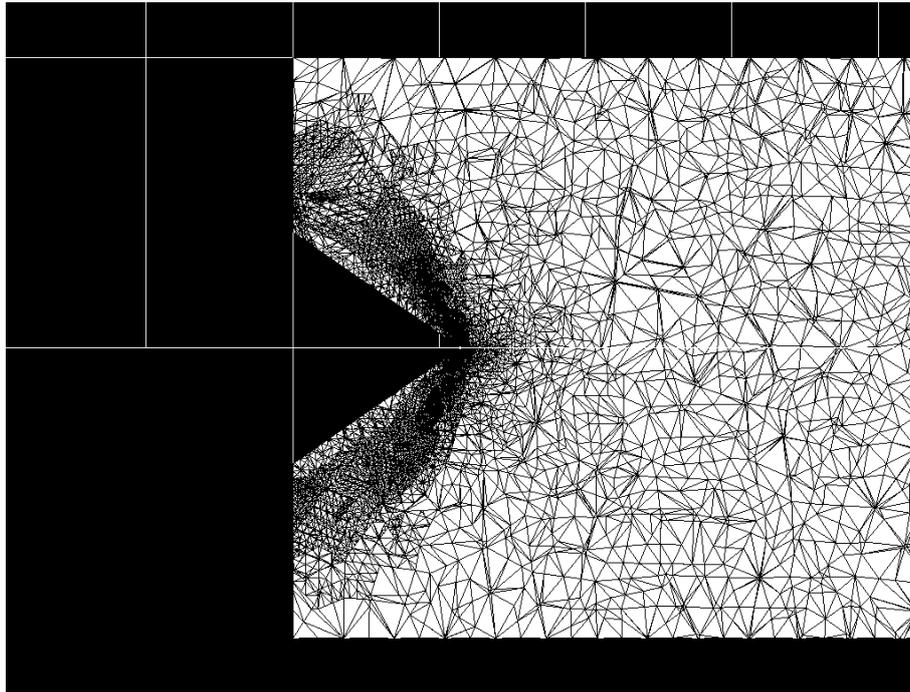


Figura 7.19: Corte en un plano de simetría de la malla adaptada - Tercer refinamiento.

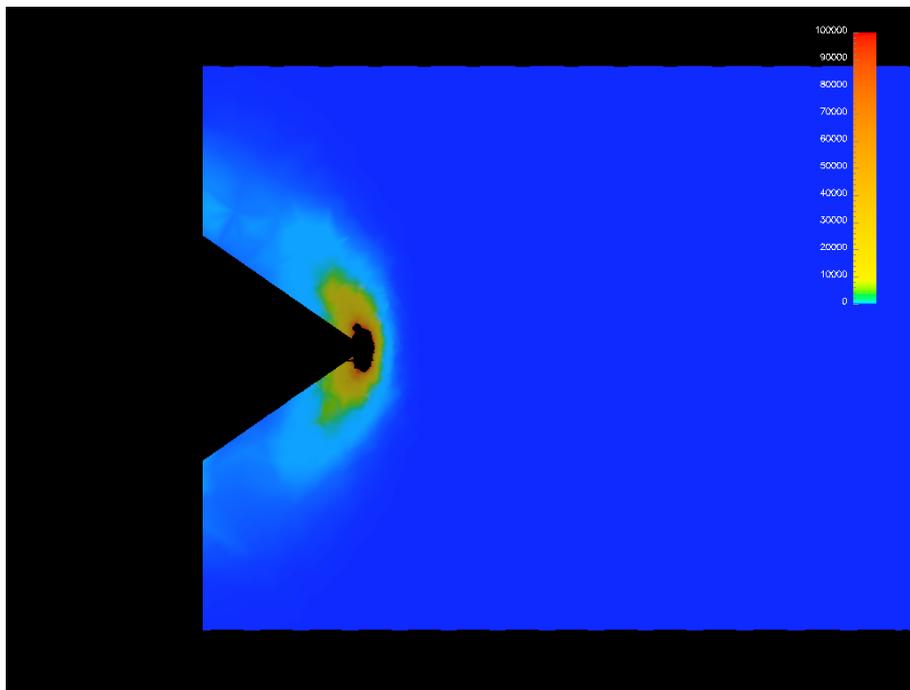


Figura 7.20: Gradiente de Mach reconstruido sobre la malla base.

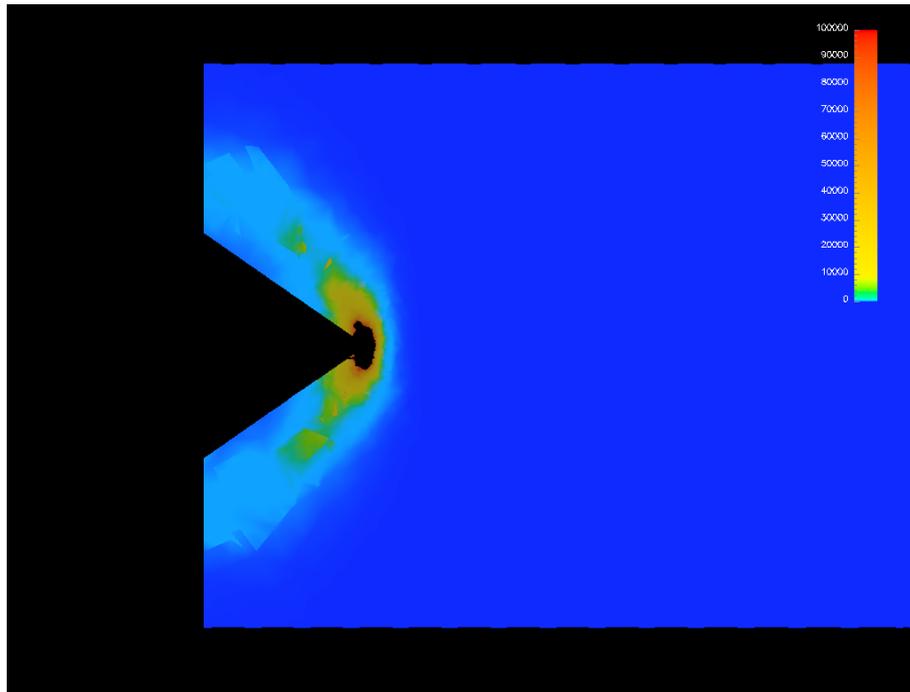


Figura 7.21: Gradiente de Mach reconstruido sobre la malla con un nivel de refinamiento.

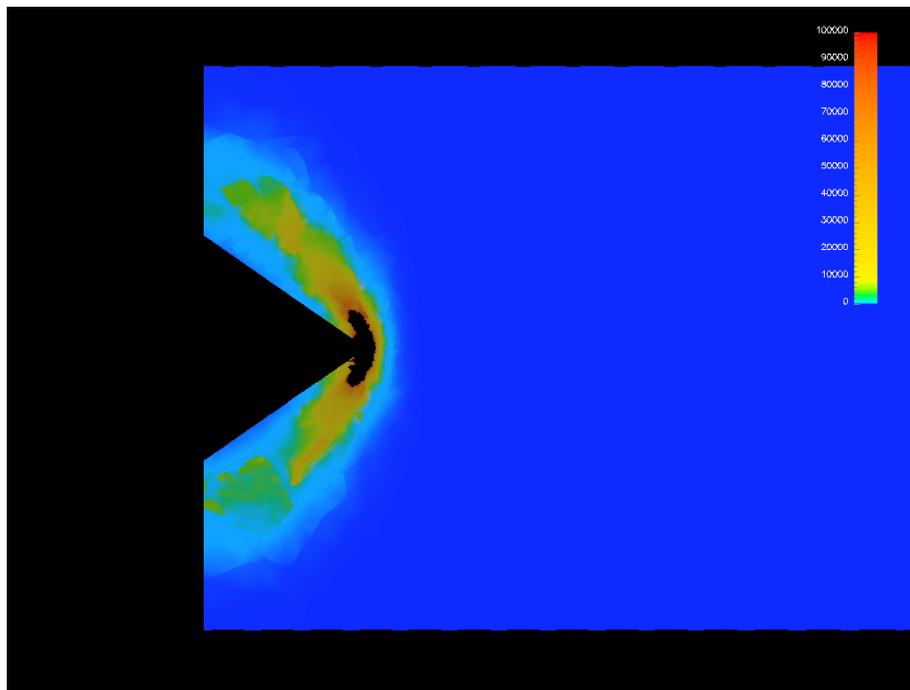


Figura 7.22: Gradiente de Mach reconstruido sobre la malla con dos nivel de refinamiento.

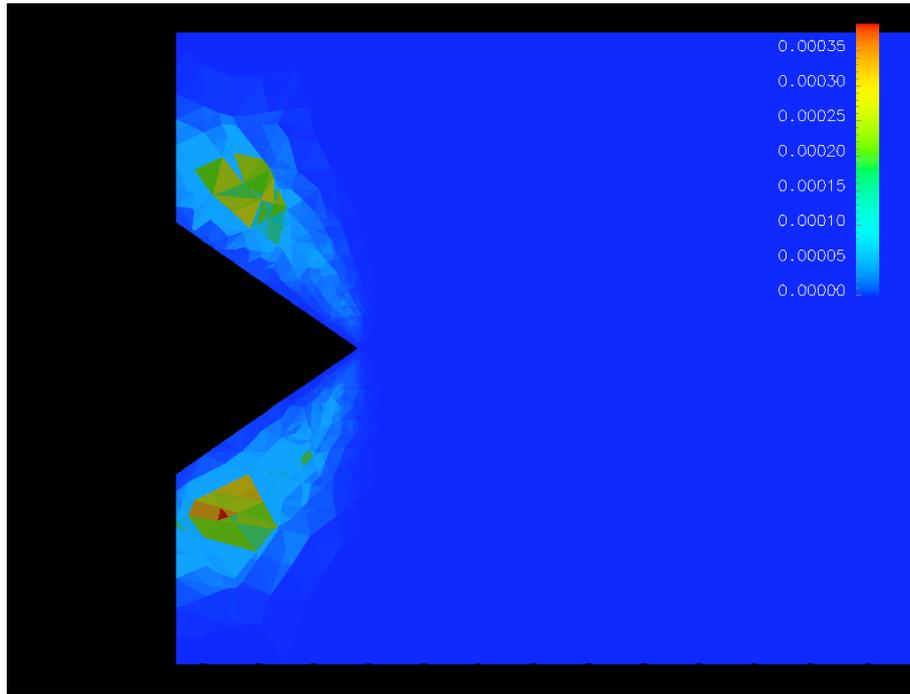


Figura 7.23: $\|e\|_{L_2(\Omega_K)}$ obtenida por reconstrucción del gradiente sobre la malla base.

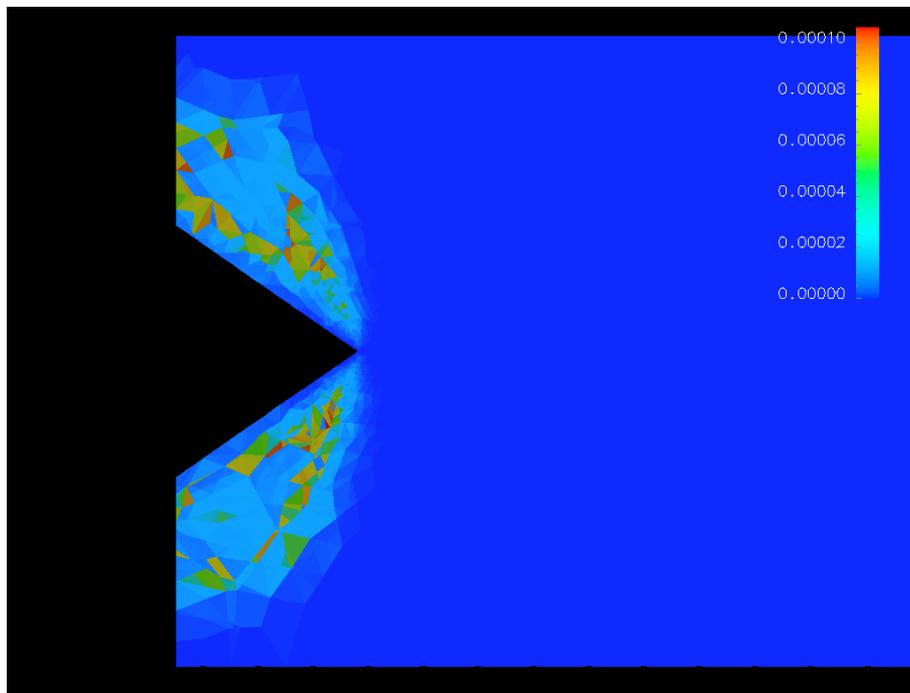


Figura 7.24: $\|e\|_{L_2(\Omega_K)}$ obtenida por reconstrucción del gradiente sobre la malla con un refinamiento.

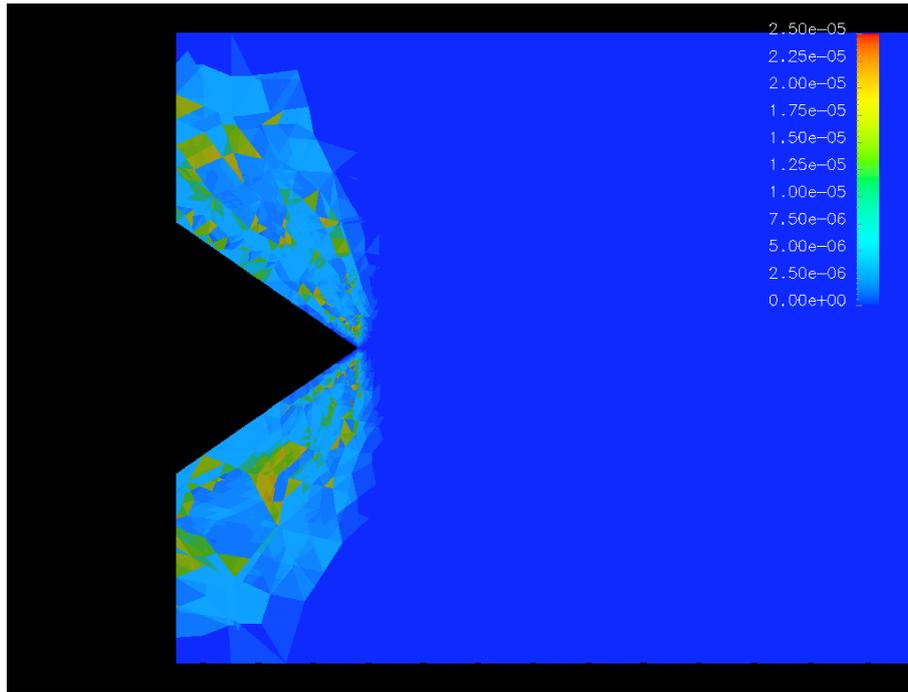
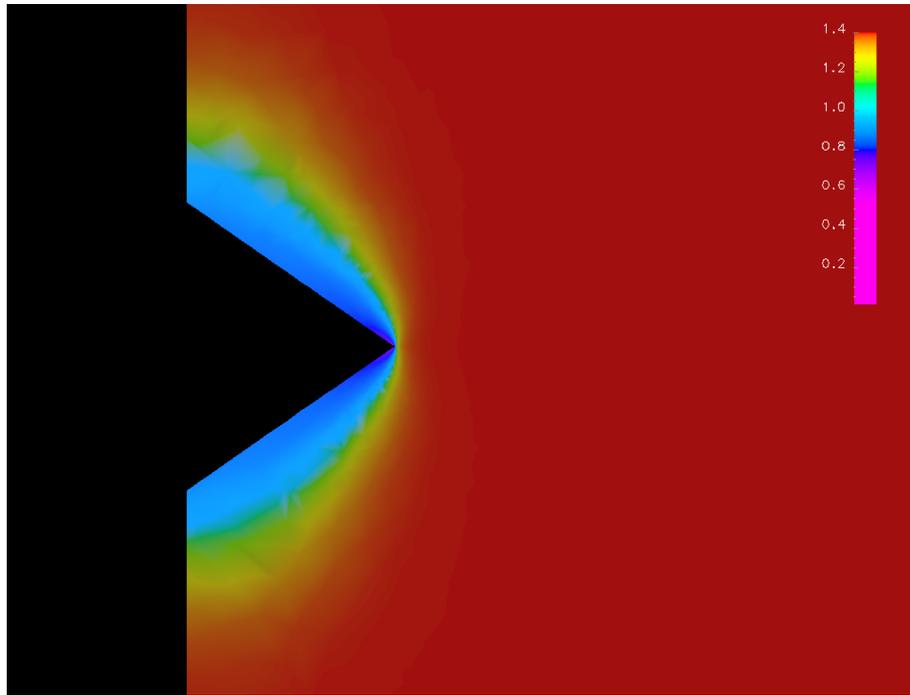


Figura 7.25: $\| e \|_{L_2(\Omega_K)}$ obtenida por reconstrucción del gradiente sobre la malla con dos refinamientos.

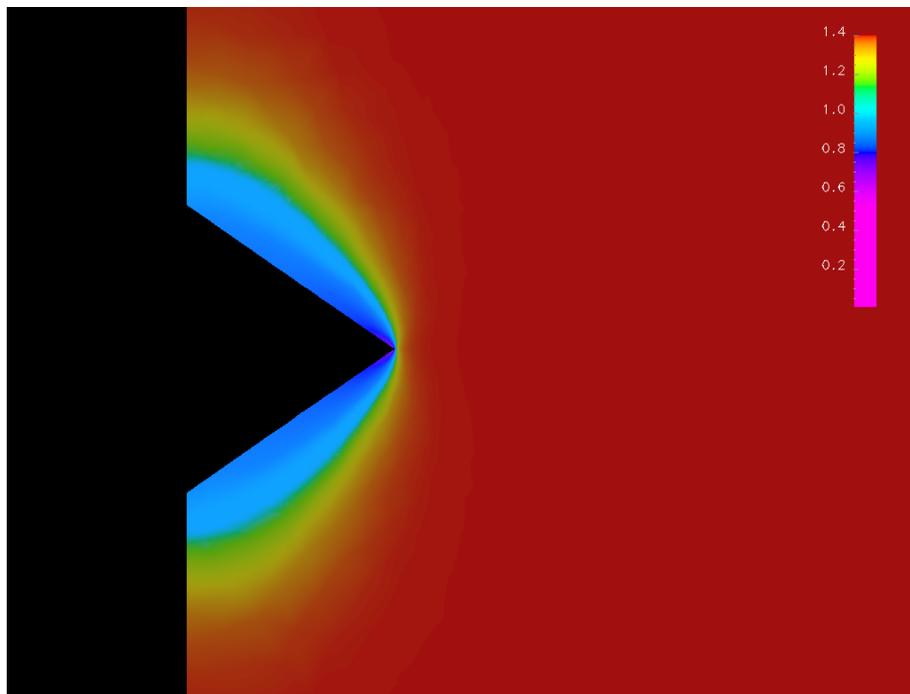
los histogramas con las distribuciones de la métrica de calidad de los elementos para las cuatro mallas utilizadas, a los cuales se superponen las curvas que representan las correspondientes distribuciones normales. Por lo tanto, se concluye que en este ejemplo y respondiendo al criterio de calidad de malla que aquí se utiliza, la calidad de la malla se conserva.

Malla	Nro de nodos	Nro de elementos	$\bar{\eta}$	η_{min}	η_{Std}
partida	30873	165688	0.8109	0.3673	0.1080
primer refinamiento	37024	190566	0.8113	0.3673	0.1071
segundo refinamiento	76090	371782	0.8175	0.3672	0.1026
tercer refinamiento	209264	1035585	0.8261	0.3672	0.0982

Cuadro 7.1: Análisis estadístico sobre de calidad de las mallas - Flujo transónico alrededor de un cono.

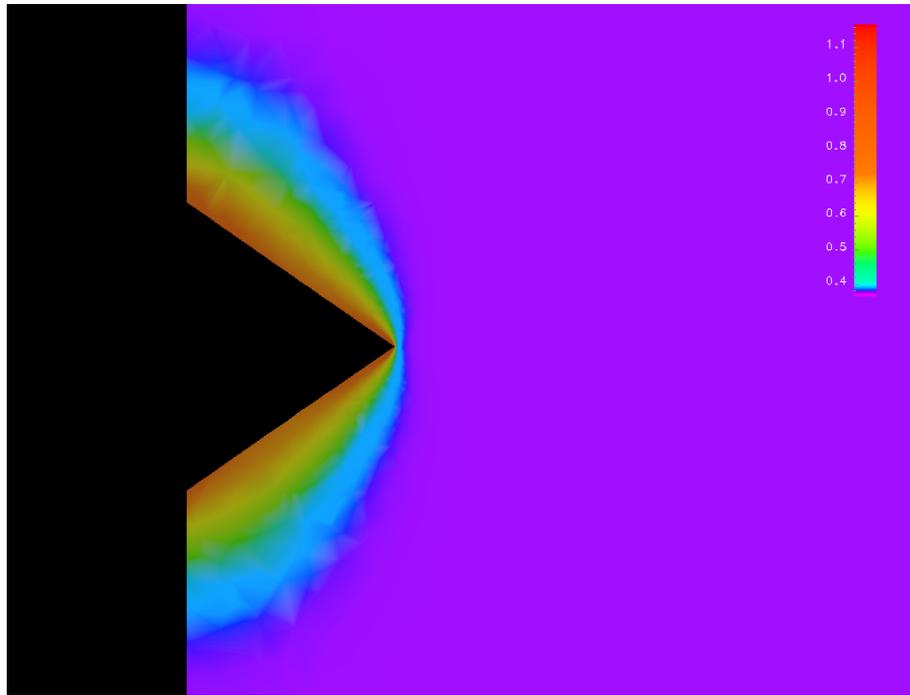


(a) Malla fija.

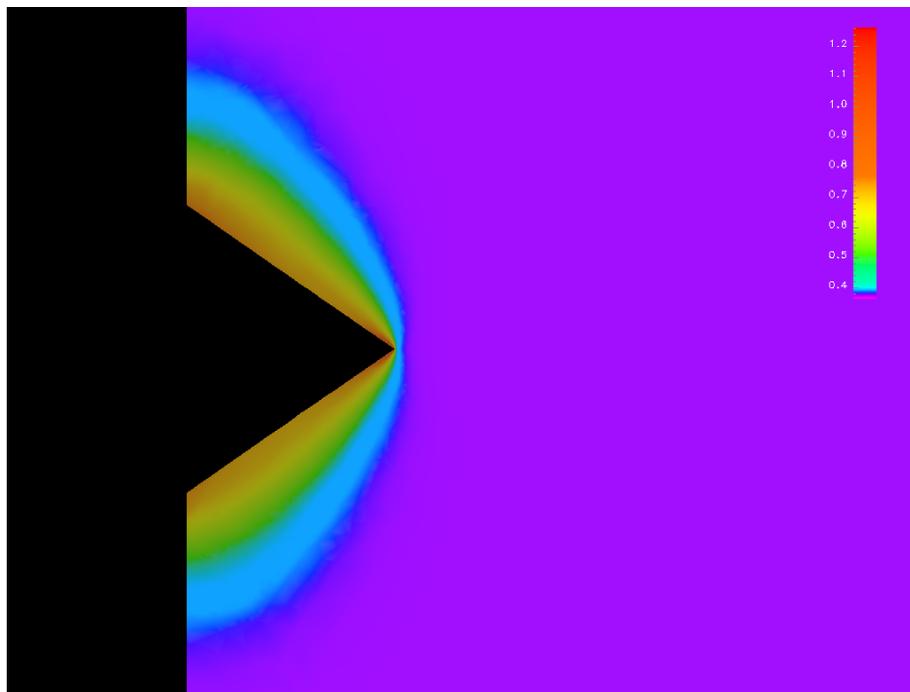


(b) Malla con 3 refinamientos.

Figura 7.26: Comparación de las soluciones para el número de Mach.

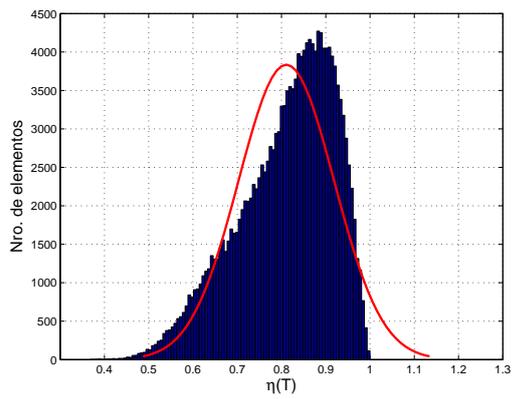


(a) Malla fija.

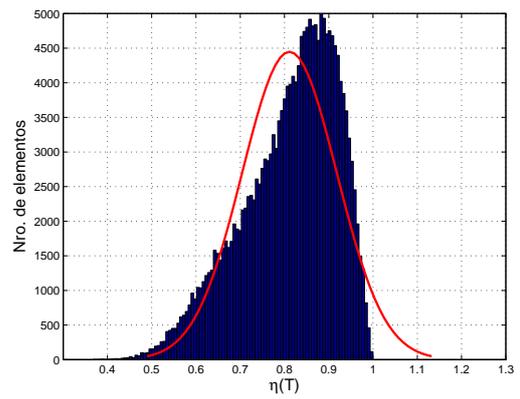


(b) Malla con 3 refinamientos.

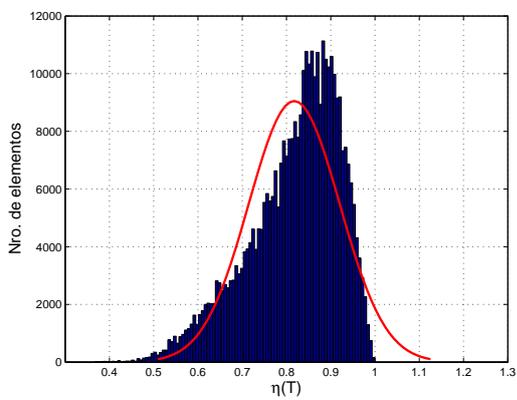
Figura 7.27: Comparación de las soluciones para la presión.



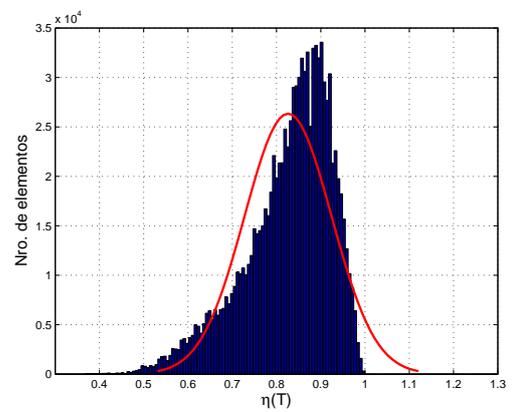
(a) Malla de partida



(b) Primer refinamiento



(c) Segundo refinamiento



(d) Tercer refinamiento

Figura 7.28: Histogramas con la distribución de la métrica de calidad η para los elementos de las mallas utilizadas en la simulación.

7.3. Arranque del flujo axisimétrico en una tobera acampanada

En este ejemplo se aplica la estrategia de adaptatividad desarrollada para problemas no estacionarios en la simulación del flujo transitorio que se desarrolla en el interior de una tobera cuyo contorno posee una forma campanada, tal como la que se presenta en la figura 7.29. El objetivo que tiene este ejemplo es comprobar si la mencionada estrategia es capaz de “seguir” con el refinamiento / desrefinamiento el desplazamiento de las onda de choque que se forman en el divergente de la tobera. Luego se comparan los resultados obtenidos con resultados correspondientes a otra simulación en la cual se utiliza una malla fija “equivalente”. Se analiza la estabilidad en la solución y la precisión con la que se definen las discontinuidades. También se comparan ambos procedimientos de solución desde el punto de vista del esfuerzo computacional requerido.

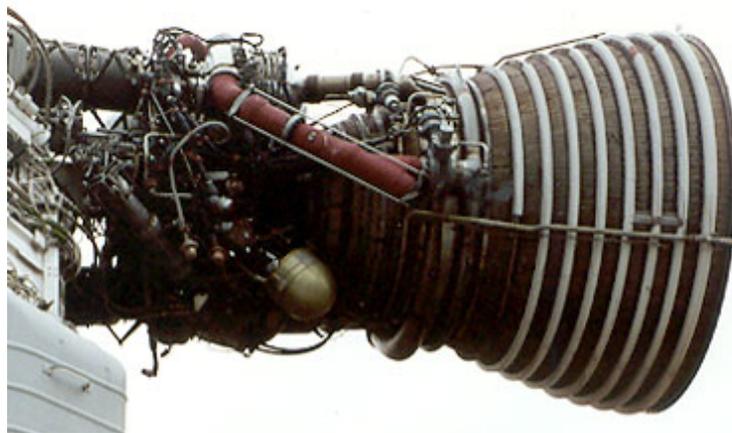


Figura 7.29: Tobera acampanada de un motor cohete de propulsión líquida.

7.3.1. Descripción del problema

La teoría de la dinámica de gases establece que para ciertas condiciones de los valores de estancamiento en la cámara de combustión y en la atmósfera de descarga, se alcanza la condición sónica en la sección de garganta de la tobera. Si además la tobera se encuentra funcionando en la condición de diseño o se halla sub-expandida, entonces el flujo que se desarrolla en todo el divergente de la tobera es supersónico y no se presentan ondas de choque estacionarias. En este caso, las variables del flujo responden únicamente a la relación que existe entre el área de la sección transversal al flujo considerada y el área de la sección de garganta (sección crítica).

En el transitorio, durante el arranque del propulsor, se genera un patrón de ondas de choque no estacionarias que se desplazan hacia la sección de salida de la tobera. Estas ondas son

las que inician el flujo en el interior de la tobera. Resulta de interés en estudios aeroelásticos (no incluidos en esta tesis) estudiar la evolución del flujo en el transitorio, ya que a los fines de reducir el daño estructural los diseñadores de toberas para este tipo de propulsores desean reducir al mínimo posible el tiempo de arranque del flujo en la tobera. Asumiendo que el arranque del motor cohete se produce casi instantáneamente, entonces los fenómenos que ocurren se asemejan a los que aparecen en las toberas utilizadas en tubos de choque, es decir, se genera una onda de choque primaria que se propaga por el interior de la tobera con una velocidad que va en disminución como resultado del aumento de la sección transversal en el divergente. A su vez, se genera una expansión aguas abajo de la sección de entrada del flujo. Luego se forma un choque secundario entre el choque primario y la sección crítica de la tobera, que se mueve aguas arriba en relación al flujo. Dicha onda aparece para empalmar dos regiones del flujo, una con alto número de Mach y baja presión generada en la zona donde se produce la rápida expansión del flujo, y otra con un flujo a bajo número de Mach y alta presión que se genera detrás del choque primario. Debido a que el choque primario se desacelera, las características en el flujo pos-choque primario se vuelven convergentes resultando en la formación del choque secundario[CS04]. El arranque exitoso de la tobera y el establecimiento del flujo cuasi-estacionario ocurre cuando este choque secundario es arrastrado aguas abajo, afuera de la tobera. Si esto no ocurre por no ser suficientemente elevada la velocidad del flujo entrante, el arranque de la tobera falla y el choque secundario puede retroceder hasta la sección de garganta de la tobera. En este caso no se tiene más la condición de flujo sónico en la garganta y el flujo es completamente subsónico en la tobera.

7.3.2. Solución numérica

Se propone resolver el problema aplicando ambas la estrategia de adaptación de mallas para problemas no estacionarios y una malla fija. A los fines de comparar las soluciones y el costo computacional involucrado en la resolución del problema, la malla fija se genera de forma tal que su tamaño sea aproximadamente igual al de la malla adaptada que posee mayor número de elementos. El costo computacional se analiza desde el punto de vista del tiempo reloj que demanda el cálculo de la solución.

El problema es modelado bajo la hipótesis de flujo axisimétrico. Dado que el *solver* resuelve las ecuaciones de Euler en 3-D, se deben imponer restricciones que garantizan la simetría axial del flujo. La malla que discretiza al dominio computacional se obtiene a partir de la extrusión en la dirección circunferencial en un ángulo ϕ , de una malla de cuadriláteros en el plano xy . Luego la malla sobre la que se resuelve el flujo está formada por hexaedros, con un único hexaedro en la dirección circunferencial. Los elementos que se encuentran sobre el eje de simetría poseen forma de cuñas debido al colapsado de las aristas que coinciden con el eje de simetría. La figura 7.30 presenta un esquema del problema.

Además, se asume que el flujo es compresible y no viscoso, aún cuando se sabe que la viscosidad juega un papel importante en la evolución de la capa límite que se desarrolla sobre

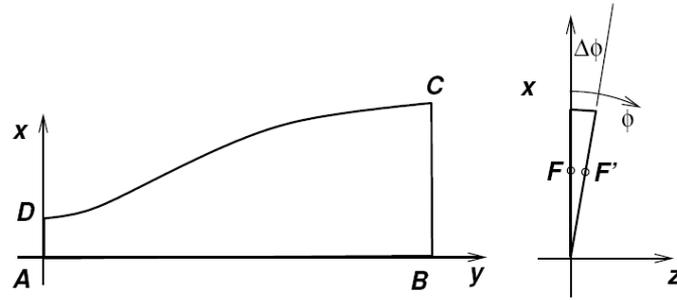


Figura 7.30: Disposición esquemática del problema.

el contorno de la tobera. Pero ya que el propósito principal de este ejemplo es determinar si la estrategia de adaptación es capaz de seguir las ondas de choque en su desplazamiento por el divergente de la tobera, se admite que el flujo sea no viscoso.

Por otro lado, las condiciones iniciales para las variables del flujo en todo el dominio computacional son ($\rho_{ini} = 0.0018\text{kg/m}^3$, $u_{ini} = 0\text{m/s}$, $p_{ini} = 143\text{Pa}$, $T_{ini} = 262\text{K}$), mientras que el coeficiente isoentrópico del gas es $\gamma = 1.17$. La simetría axial del campo de movimiento se garantiza imponiendo las siguientes condiciones de borde,

- Condición de no penetración $\vec{V} \cdot \vec{n} = 0$ sobre el contorno sólido (pared de la tobera) \overline{DC} y sobre el eje de simetría \overline{AB} .
- Condiciones de borde dinámicas [SNPD08] en la sección \overline{BC} de salida de la tobera.
- Flujo axial en la sección de la garganta (la componente $V_x = 0$ en la sección \overline{AD}).
- Presión y densidad constantes sobre la sección de garganta. Ambas se incrementan linealmente en el tiempo, desde los valores ambientes hasta los valores $\rho_0 = 0.47541\text{kg/m}^3$, $p_0 = 0.6\text{MPa}$ y $T_0 = 4170\text{K}$. Se asume que el transitorio hasta alcanzar dichos valores tiene una duración igual a $t = 5\mu\text{s}$.
- Se considera que no existe flujo en la dirección circunferencial ($V_\phi = 0$).
- Existe periodicidad circunferencial de todas las variables del flujo en todos los puntos del dominio. Esto significa que las variables escalares tales como la presión y densidad deben ser iguales para los puntos F y F' de la figura 7.30, mientras que las cantidades vectoriales como la velocidad se encuentran afectadas por la matriz de rotación que se utiliza para rotar la malla de cuadriláteros en el plano xy por el ángulo ϕ .

Las condiciones de la simulación se corresponden con el funcionamiento de la tobera en el régimen sub-expandido, por lo cual se espera que ambas ondas de choque se desplacen rápidamente hacia la sección de salida. Durante el transitorio para los valores de las variables del flujo en la sección de garganta, el cálculo del flujo se realiza con un número de Courant relativamente bajo ($Cou = 0.2$). Una vez que se ha establecido el flujo en la garganta de la tobera y que la onda de

choque ha viajado una distancia considerable aguas abajo en el divergente, el valor del Courant se incrementa a $Cou = 1$.

Los elementos que se seleccionan para refinar se eligen utilizando del indicador de error basado en la magnitud del gradiente elemental de la velocidad. Para ello se calcula el módulo de la velocidad $U = \sqrt{u^2 + v^2 + y^2}$ y se computa la magnitud del gradiente elemental de dicho campo escalar $\|\nabla_e U\|$. Luego, los elementos que se seleccionan para refinar son aquellos tales que

$$c_1 \leq \frac{\|\nabla_e U\|}{\max_{e=1\dots N_{el}} \|\nabla_e U\|} \quad (7.9)$$

donde N_{el} es el número de elementos que discretizan el dominio del problema y el valor de $c_1 \simeq 0.1$ permanece constante durante todo el procedimiento de solución. Se utiliza una frecuencia de adaptación de la malla igual a $n_{step}=5$ pasos de tiempo, es decir que cada 5 pasos de tiempo la malla es adaptada sobre último estado obtenido por el *solver*.

Finalmente, teniendo en cuenta las condiciones iniciales del problema y los valores de las variables del flujo en la sección de garganta, se puede estimar el tiempo que demorará la onda de choque primaria en salir del divergente. De esta manera, si se plantea el problema de una onda de choque no estacionaria con los valores de las variables del flujo por delante de la onda $\rho_1 = 0.0018\text{Kg/m}^3$, $u_1 = 0\text{m/s}$, $p_1 = 143\text{Pa}$, $T_1 = 262\text{K}$ y aguas abajo de la misma $\rho_2 = 0.475\text{Kg/m}^3$, $p_2 = 0.6\text{MPa}$, $T_2 = 4170\text{K}$, $M_2 = 1$, se obtiene una velocidad de desplazamiento de la onda $U_S \simeq 2430\text{m/s}$. Luego, teniendo en cuenta que la longitud de la tobera es $L \simeq 1.775\text{m}$ se determina que la onda demorará aproximadamente 0.73ms en alcanzar la sección de salida. Evidentemente, por las explicaciones dadas en la descripción del problema, la onda de choque se desacelera a medida que viaja por el divergente y tampoco se tiene en cuenta en dicha estimación la presencia de la onda de choque secundaria. Sin embargo, la estimación resulta útil a los fines de tener una idea acerca del tiempo de simulación.

Las ecuaciones de Euler son resueltas en PETSc-FEM utilizando 10 procesadores del *cluster* Aquiles y se prescribe un único nivel de refinamiento para toda la simulación.

La malla base que se utiliza en el procedimiento de solución adaptativo está constituida por 9330 elementos y 9672 nodos. La misma es generada prescribiendo elementos más pequeños en la región de la garganta de la tobera, siendo que el tamaño de los elementos se estima como el radio del círculo circunscrito para el cuadrilátero en el plano xy . La figura 7.31 presenta la distribución del tamaño de los elementos para la malla base.

Por otro lado, la malla fija sobre la que se calcula la solución sin utilizar el procedimiento adaptativo presenta una distribución en el tamaño de sus elementos similar a la de la malla base, pero posee 11430 elementos y 11842 nodos. El tamaño de la malla fija es igual al tamaño de la malla más grande obtenida durante el cálculo adaptativo de la solución. Los criterios por los cuales se utiliza una malla fija de este tamaño son costo y precisión en el cálculo de la solución. Es decir, se desea utilizar una malla fija tal que la precisión con la que se resuelve el flujo y el costo computacional sean comparables a los que se obtienen con el procedimiento de solución

adaptativo. Luego, se tienen en cuenta los siguientes aspectos: a) si bien el mayor tamaño de la malla implicaría un mayor costo de solución de las ecuaciones del flujo, los costos asociados a la adaptación de la malla y a reiniciar el cálculo de la solución cada n_{step} pasos de tiempo no están presentes cuando se utiliza la malla fija, b) si bien el número de elementos en la simulación adaptativa es siempre menor o igual que el de la malla fija, en la primera los elementos que se agregan se localizan en las regiones de interés (sobre las ondas de choque) mientras que en la segunda se encuentran distribuidos en toda la malla, c) finalmente, si se utilizara una malla fija que con el mismo grado de refinamiento que el que se obtiene adaptando la malla base, entonces la malla fija debería poseer 37320 elementos, lo que penalizaría en un costo mayor respecto del cálculo adaptativo.

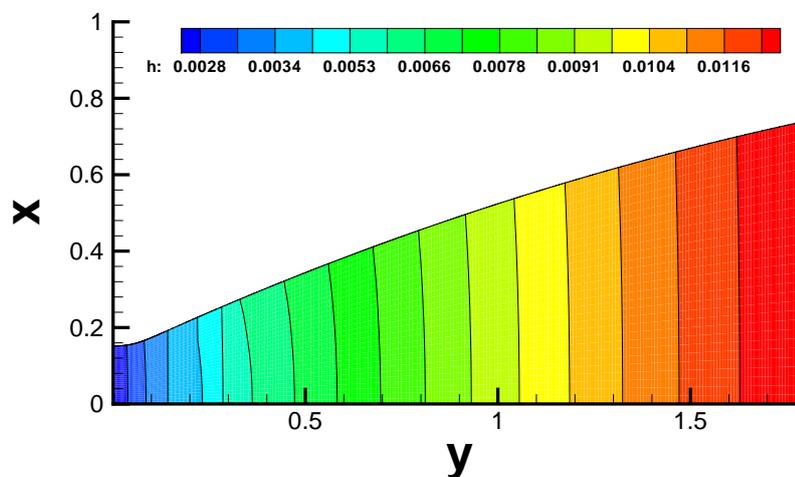


Figura 7.31: Distribución del tamaño de los elementos en la malla base sobre el plano xy .

7.3.3. Resultados

Se comparan los perfiles de la velocidad axial sobre el eje de simetría de la tobera correspondientes a distintos instantes de tiempo, obtenidos con ambos procedimientos de solución. En primera instancia, se observa en la figura 7.32 que se obtiene una mejor definición de ambos choques cuando se utiliza el procedimiento adaptativo. También se ve que el *overshoot* en la velocidad axial que ocurre inmediatamente detrás de la onda de choque secundaria para los instantes $t=0.5\text{ms}$ y $t=0.6\text{ms}$ en la solución obtenida con la malla fija no se observa en la solución obtenida con la malla adaptada. Se menciona que debido al escalado de las variables del flujo utilizado para resolver el problema numéricamente, la magnitud de la velocidad en la figura 7.32 debe ser multiplicada por el valor de referencia $V_{ref} = 1215.16\text{m/s}$ para obtener los valores reales (esta velocidad de referencia es igual a la velocidad del sonido para las condiciones del flujo en la sección de garganta).

Si se analiza el campo de movimiento en su totalidad, se puede ver que también existen diferencias en el flujo. Las figuras 7.33 y 7.34 presentan el campo para el módulo de la velocidad

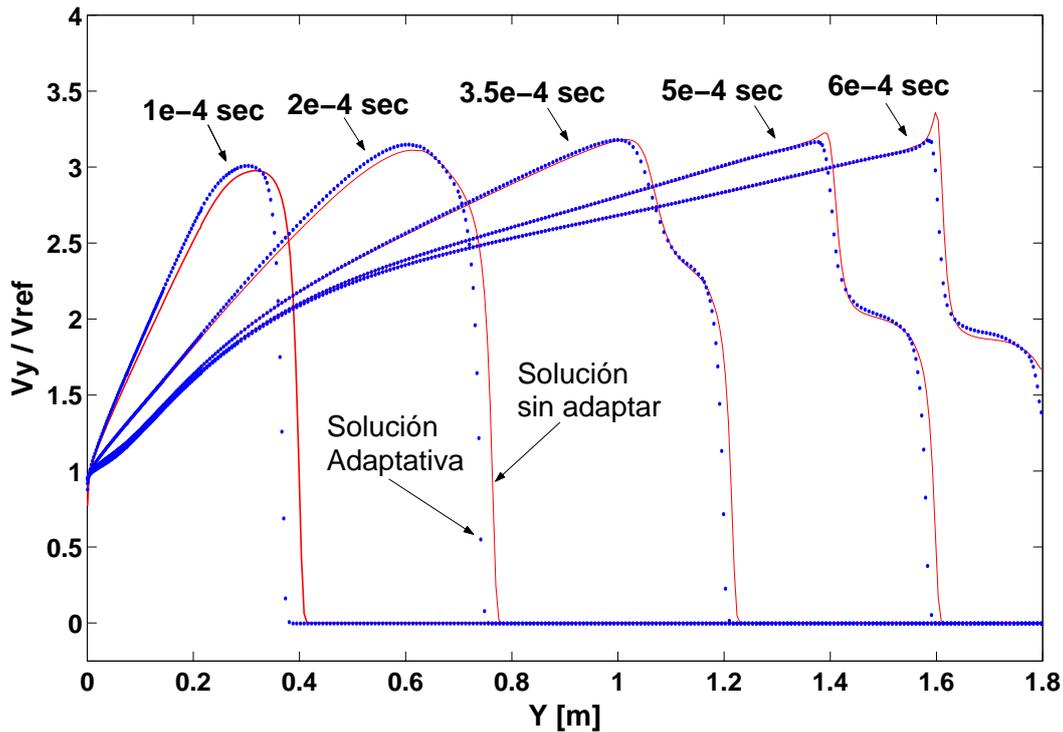
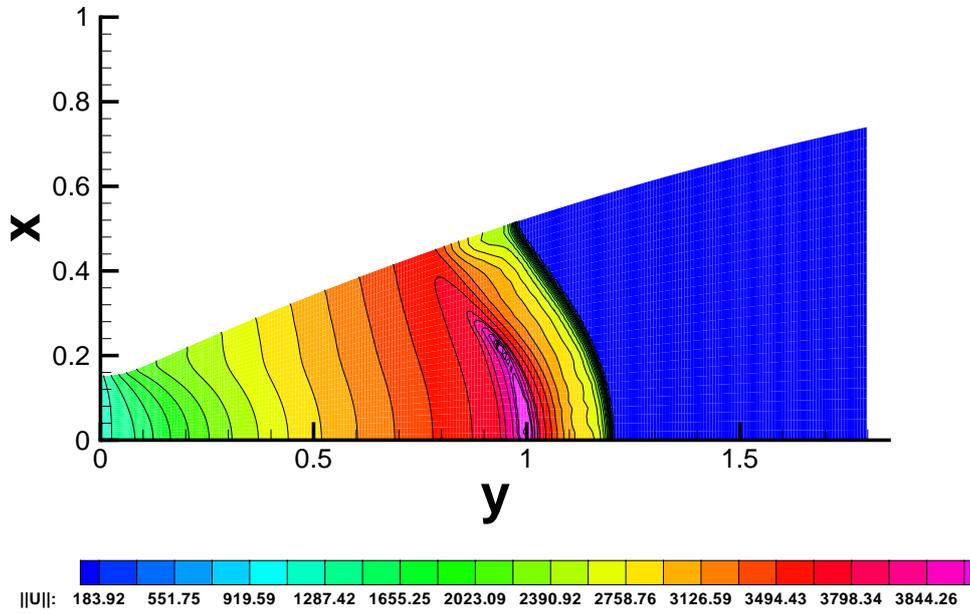


Figura 7.32: Comparación de la velocidad axial sobre el eje de la tobera.

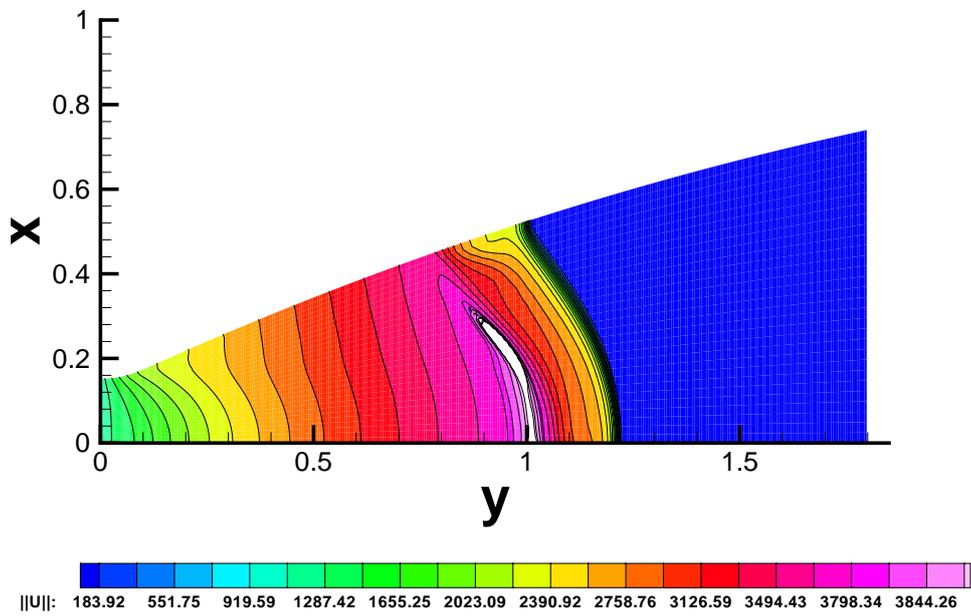
en los instantes $t = 0.35\text{ms}$ y $t = 0.6\text{ms}$. De ambas figuras se puede concluir que los choques son resueltos con mayor precisión cuando se utiliza el procedimiento adaptativo. La región indicada en color blanco sobre el choque secundario, en las figuras correspondientes a las soluciones obtenida con la malla fija, indica la zona donde ocurre el *overshoot* en el módulo de la velocidad. En este sentido, se podría pensar que el procedimiento de solución adaptativo agrega difusión al esquema numérico. Sin embargo, las discontinuidades son resueltas con mayor precisión.

Las figuras 7.35.a) y 7.35.b) presentan el detalle de la malla adaptada para ambos instantes de tiempo. En la figura correspondiente al instante de tiempo $t=0.35\text{ms}$ se observa que el gradiente del módulo de la velocidad sobre el choque secundario no es lo suficientemente grande, en relación a la cota impuesta por la constante c_1 utilizada en la ecuación (7.9) a los fines de disparar el refinamiento sobre dicha discontinuidad en el flujo.

La simulación adaptativa utiliza 2280 pasos de tiempos hasta que la solución alcanza el estado cuasi-estacionario (para que la onda de choque secundaria salga del divergente de la tobera), mientras que la solución obtenida con la malla fija requiere 2219 pasos de tiempo. El hecho de que la solución obtenida con el procedimiento adaptativo requiera una mayor cantidad de pasos de tiempo se debe a la restricción en el tamaño del paso de tiempo por el uso de la condición CFL y al refinamiento en las regiones del flujo donde la velocidad de propagación de las perturbaciones es más grande (ondas de choque). Luego, para ambas simulaciones se encuentra que la onda de choque primaria alcanza la sección de salida para $t \simeq 0.6\text{ms}$ mientras que la onda de choque secundaria lo hace para $t \simeq 0.7\text{ms}$. Si se tiene en cuenta el tiempo estimado de $t = 0.73\text{ms}$ se puede concluir que existe una buena concordancia entre estos valores,

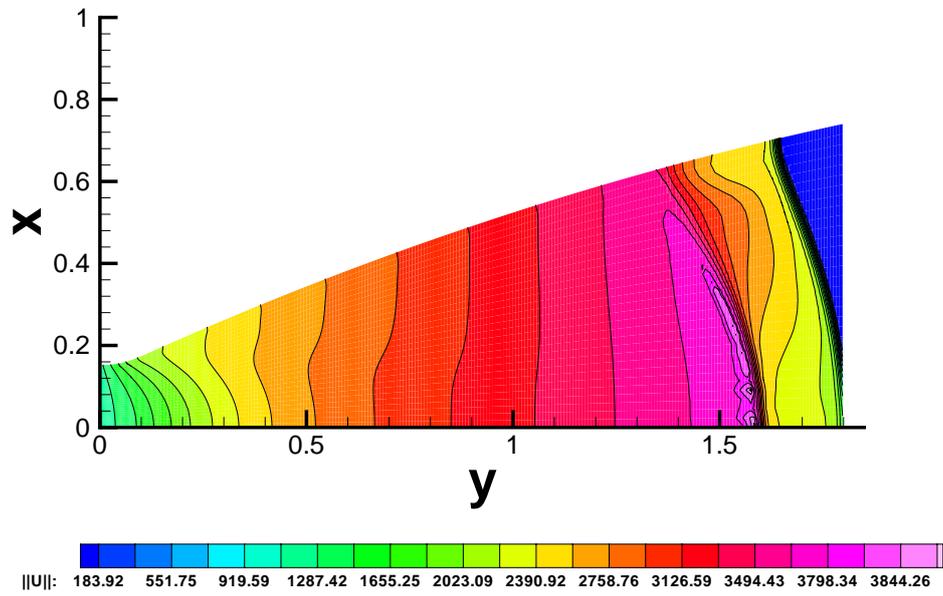


(a) Solución adaptativa.

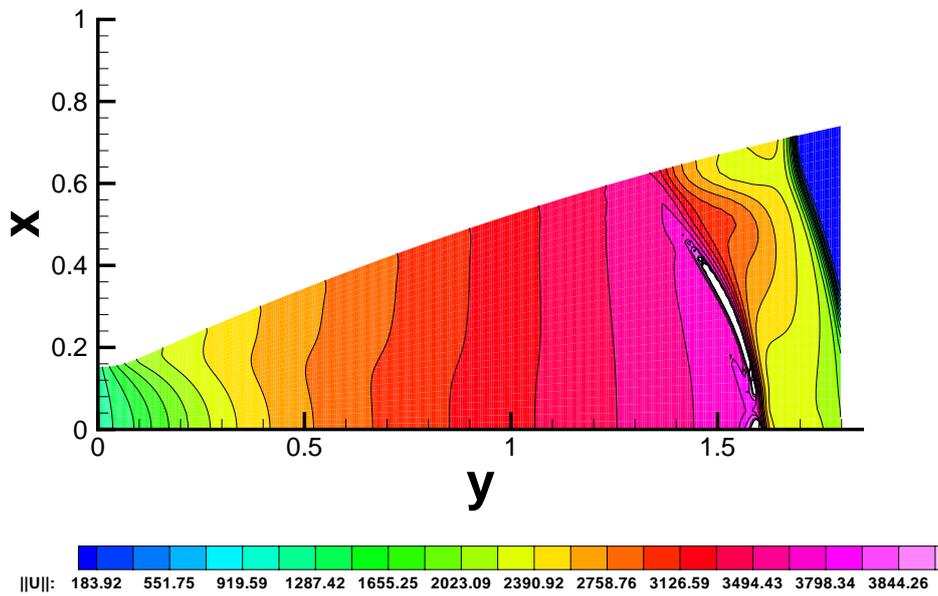


(b) Solución malla fija.

Figura 7.33: Comparación del módulo de la velocidad para $t=3.5e-4s$.

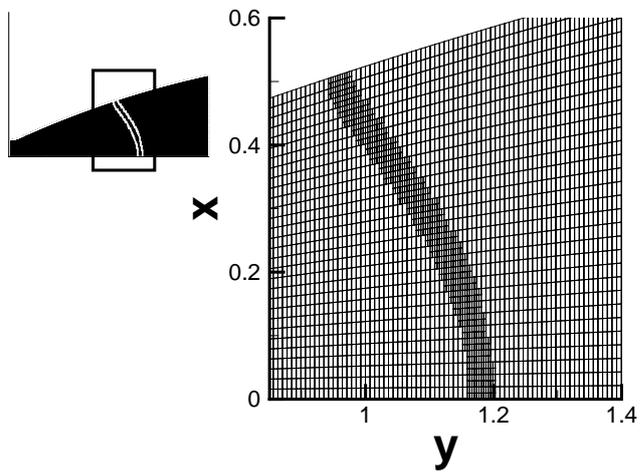


(a) Solución adaptativa.

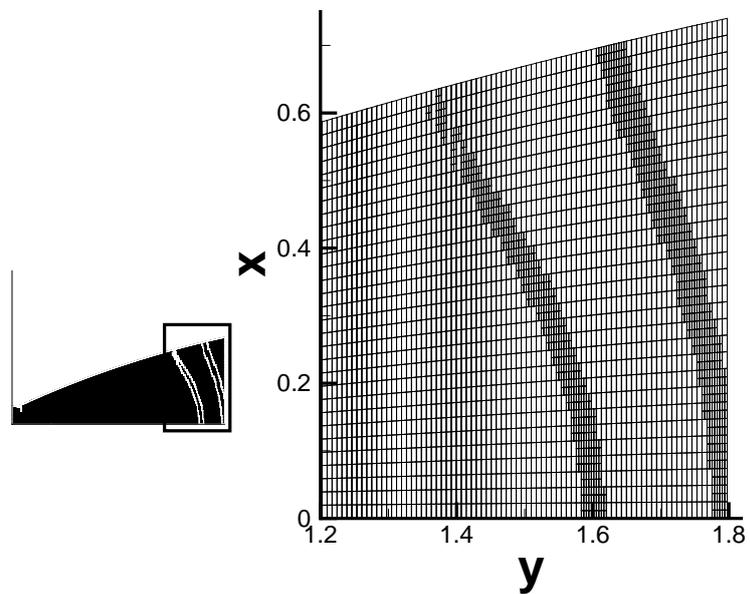


(b) Solución malla fija.

Figura 7.34: Comparación del módulo de la velocidad para $t=6.0e-4s$.



(a) Malla adaptada en $t=3.5e-4s$.



(b) Malla adaptada en $t=6.0e-4s$.

Figura 7.35: Malla adaptada para distintos instantes de tiempo.

aunque la onda de choque primaria en la simulación ha viajado con una velocidad media mayor a la estimada en los cálculos previos.

Por otro lado, a los fines de evaluar si presenta alguna ventaja utilizar el procedimiento de cálculo adaptativo, se miden los tiempos reloj requeridos por ambos procedimientos de solución, los cuales se resumen en la tabla 7.2.

tiempo hh:mm	Sol. Adaptativa	Sol. Malla Fija
flujo	7:49	9:06
adaptación	0:48	—
total	8:37	9:06

Cuadro 7.2: Comparación del tiempo reloj requerido para resolver el flujo axisimétrico en una tobera.

Se observa que el tiempo reloj demandado por el procedimiento de solución adaptativo es ~30 minutos inferior al requerido por el procedimiento que utiliza la malla fija, mientras que el tiempo para adaptar la malla representa aproximadamente el 10 % del tiempo total de solución. Luego, el costo para resolver el flujo, realizar la adaptación de la malla, más el *overhead* para reiniciar el cálculo de la solución cada n_{step} pasos de tiempo (incluido en el tiempo para resolver el flujo) en el procedimiento adaptativo es menor al costo para resolver el flujo con una malla fija “comparable”.

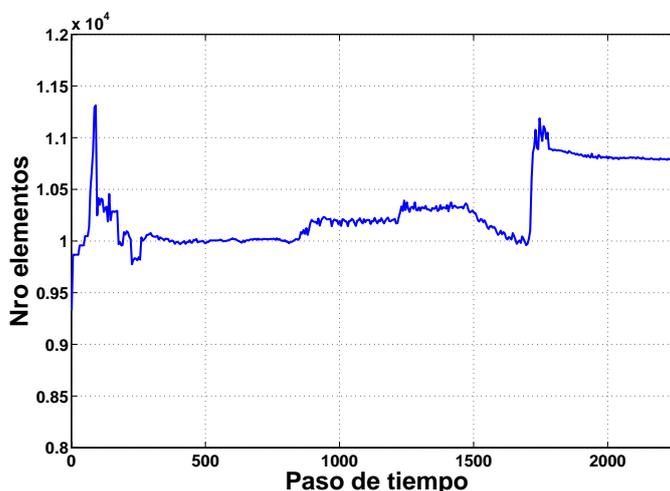


Figura 7.36: Evolución del tamaño del problema para el procedimiento adaptativo.

El comportamiento en el número de elementos de la malla se adjudica a la geometría del problema, a la distribución no uniforme en el tamaño de los elementos y a la evolución misma de las estructuras del flujo que modifican la cantidad de elementos seleccionados para refinar en cada paso de adaptación.

7.4. Onda explosiva esférica (Problema de Taylor-Sedov)

La aplicación del esquema de solución adaptativo al siguiente problema persigue dos objetivos. El primero consiste en evaluar la performance del código implementado en C++ en referencia al tiempo de procesamiento para un problema no estacionario 3D. Para ello, se analiza en qué medida el algoritmo de refinamiento de mallas (ver figura 6.9) escala respecto del tamaño del problema y cuál es el esfuerzo computacional requerido por dicho procedimiento.

El segundo objetivo consiste en analizar si existen beneficios en cuanto a la precisión en el cálculo de la solución con esta estrategia de adaptación. De forma similar al problema del arranque del flujo en la tobera, se utilizan: a) el procedimiento de adaptación propuesto y b) una malla fija. Se analiza si la estrategia de adaptación de mallas introduce inestabilidades en la solución y en qué medida permite resolver las discontinuidades del flujo con precisión.

Se elige el problema de la explosión esférica por haber sido extensamente estudiado y resuelto en forma analítica y mediante métodos aproximados [Fre74, VLB96, Hut00, Rem03, Wal04, KAKY06]. Además, el mismo no presenta complejidad en cuanto a la geometría involucrada.

7.4.1. Descripción del problema

Se describe el problema que plantea el cálculo del flujo generado por la propagación de una onda explosiva esférica originada por una fuente explosiva puntual, en un medio con condiciones de presión y densidad uniformes. Al producirse una explosión, inicialmente se genera y se propaga una onda de choque debido al salto abrupto en las condiciones de presión y densidad entre el foco donde se origina la explosión y el medio circundante. A esto le sigue la generación de una expansión en el foco de la explosión. El campo de movimiento está caracterizado por una onda de choque esférica que viaja a velocidad supersónica en el medio y por un flujo subsónico detrás de la misma, seguida de una expansión. De esta manera, se define como *onda explosiva* a la estructura del flujo que se forma cuando la expansión que se genera en el centro de la explosión evoluciona de tal forma que alcanza y supera a la onda de choque, haciendo que esta se debilite y finalmente desaparezca a medida que se propaga en el medio. Existen varias causas por las cuales se puede producir este fenómeno. En el caso particular de análisis, la causa se corresponde con el aporte de energía en forma limitada. Fenómenos de este tipo ocurren frecuentemente en la naturaleza. Ejemplos de ello son:

- Supernova *remnants*: si bien inicialmente producen frentes de choques y no ondas explosivas, cuando la presión interior se reduce y el material interestelar acumulado o barrido por el frente de choque excede a la masa de la estrella misma, los restos de la supernova desarrollan una estructura del tipo onda explosiva esférica.
- La descarga de energía producida por un rayo durante un lapso de tiempo muy breve en el trayecto recorrido por el mismo. En este caso se puede considerar que la onda explosiva posee simetría cilíndrica en vez de esférica.

- Explosiones en la atmósfera (figura 7.37) o de cargas de profundidad.



Figura 7.37: Explosión esférica en la atmósfera.

Las ondas explosivas esféricas también se conocen con el nombre de ondas explosivas de Taylor-Sedov ya que ambos, los físicos G.I. Taylor y L.I. Sedov, fueron los primeros en analizar el comportamiento de las mismas[Tay46, Tay50a, Tay50b, Sed59]. A partir de dichos análisis, proponen que la solución del problema sea del tipo auto-similar. Esto quiere decir que los perfiles de la solución son independientes del tiempo y que sólo se ven afectados por una escala espacial que es función del tiempo. En este caso la escala espacial elegida es la posición radial $R(t)$ del frente de la onda de choque. Luego, la variable de similaridad que se utiliza es $\xi = r/R$ donde $0 \leq \xi \leq 1$.

Las ecuaciones que describen este movimiento son las de continuidad, cantidad de movimiento y energía (ecuaciones de Euler) para un flujo no viscoso y compresible expresadas en coordenadas radiales. Si además se considera que el gas se comporta como un gas perfecto, entonces se tiene para el caso esférico que las ecuaciones se reducen a la siguiente forma

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \rho u) = 0 \quad (7.10)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + \frac{1}{\rho} \frac{\partial p}{\partial r} = 0 \quad (7.11)$$

$$\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial r} - c_s^2 \left(\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial r} \right) = 0 \quad (7.12)$$

donde $c_s = \sqrt{\gamma \mathfrak{R} T}$ es la velocidad del sonido en el medio, γ es el coeficiente isoentrópico del gas, $\mathfrak{R} = 287 \text{ J/Kg K}$ es la constante del gas, u la velocidad en la dirección radial, ρ la densidad y p la presión.

Interesa encontrar soluciones en las cuales un sólo parámetro ξ describe la forma o el “perfil” de la solución para todo instante y en todo el espacio. Para lograr esto, el sistema de ecuaciones en derivadas parciales 7.10, 7.11 y 7.12 debe transformarse en un sistema de ecuaciones diferenciales ordinarias que involucre funciones adimensionales que dependan exclusivamente del parámetro ξ . A tal fin, Taylor y Sedov proponen expresar las variables del flujo u , ρ y p de la siguiente manera

$$u = \dot{R}U(\xi), \quad \rho = \rho_0(r,t)\Omega(\xi), \quad p = \rho_0(r,t)\dot{R}^2P(\xi) \quad (7.13)$$

en las que el punto indica derivada respecto del tiempo y se observa que la densidad inicial en el medio no perturbado, ρ_0 , es por ahora una función general de la posición y del tiempo. Observar que la solución para la densidad no depende ni de la posición de la onda R ni de su velocidad \dot{R} . Esto quiere decir si la distribución $\rho_0(r,t) = \text{cte}$ es constante, entonces los valores de la densidad se obtienen escalando el perfil $\Omega(\xi)$ por la constante ρ_0 . En el caso de la velocidad radial y de la presión, sí hay una dependencia respecto de la velocidad del frente de choque esférico. Considerando que la variable h es una función de la variable ξ , entonces aplicando la regla de la cadena se pueden expresar las derivadas respecto de la variable temporal y espacial como

$$\frac{\partial h}{\partial t} = -\xi \frac{\dot{R}}{R} h'(\xi) \quad \text{y} \quad \frac{\partial h}{\partial r} = \frac{1}{R} h'(\xi) \quad (7.14)$$

siendo $h'(\xi) = \partial h / \partial \xi$. Utilizando las ecuaciones (7.14) cuando correspondan, es posible obtener el siguiente sistema de ecuaciones diferenciales ordinarias (ver apéndice B), las cuales dependen exclusivamente del tiempo y del espacio en algunos términos

$$\frac{\dot{\rho}_0}{\rho_0} \frac{\dot{R}}{R} \Omega(\xi) + \frac{\rho'_0 R}{\rho_0} U(\xi) \Omega(\xi) + [U(\xi) - \xi] \Omega'(\xi) + U'(\xi) \Omega(\xi) + \frac{2U(\xi) \Omega(\xi)}{\xi} = 0 \quad (7.15)$$

$$\frac{\rho'_0 R}{\rho_0} P(\xi) + \frac{R \ddot{R}}{\dot{R}^2} U(\xi) \Omega(\xi) + [U(\xi) - \xi] U'(\xi) \Omega(\xi) + P'(\xi) = 0 \quad (7.16)$$

$$\frac{\dot{\rho}_0}{\rho_0} \frac{R}{\dot{R}} (1 - \gamma) P(\xi) + \frac{\rho'_0 R}{\rho_0} U(\xi) (1 - \gamma) P(\xi) + 2 \frac{R \ddot{R}}{\dot{R}^2} P(\xi) + \dots \quad (7.17)$$

$$\dots + [U(\xi) - \xi] \left(P'(\xi) - \gamma P(\xi) \frac{\Omega'(\xi)}{\Omega(\xi)} \right) = 0 \quad (7.18)$$

Para que la solución exhiba un comportamiento auto-similar, se debe cancelar la dependencia temporal y espacial en estas ecuaciones. La dependencia temporal se elimina si la posición del frente de onda R y la densidad inicial del medio $\rho_0(r,t)$ exhiben un comportamiento en ley de potencias respecto del tiempo. Esto quiere decir que

$$R = R_0 t^\alpha \quad y \quad \rho_0(r, t) = \rho_0(r) t^\beta \quad (7.19)$$

Si además se debe eliminar la dependencia espacial en las ecuaciones (7.15), (7.16) y (7.17), la cual se manifiesta en la presencia del término $\rho'_0 R / \rho_0$, siendo $\rho'_0 = \partial \rho_0 / \partial r$, entonces ρ_0 debe ser una ley de potencia de la posición r , es decir

$$\rho_0(r) = \hat{\rho} r^\delta = \hat{\rho} \xi^\delta R^\delta \quad (7.20)$$

De esta forma se tiene que

$$\frac{\rho'_0(r) R}{\rho_0} = \frac{\rho'_0(\xi) R}{\rho_0 R} = \frac{\delta}{\xi} \quad (7.21)$$

Sin embargo, si el problema involucra un segundo parámetro físico tal como la energía de la explosión, entonces se tiene un comportamiento auto-similar en el cual la definición del parámetro ξ debe estar relacionada con este parámetro físico[Sed59]. Por ejemplo, si tengo un mayor aporte de energía, entonces se producirá un movimiento más rápido, correspondiente a valores más grandes de \dot{R} . De esta manera, si está especificada la energía aportada al sistema, la misma debe intervenir en la definición de R y por ende de ξ . Así se puede definir, a menos de una constante de proporcionalidad, la posición del frente R como una combinación adimensional de las cantidades físicas conocidas, además de r y t .

El problema de la explosión puntual posee solamente restricciones que involucran cantidades físicas. Estas son la energía de la explosión E_x y la densidad inicial del medio circundante, la cual se asume que es constante $\rho_0 = \text{cte}$ ($\delta = 0$ en la ecuación 7.20). Luego, la posición radial del frente de onda esférico es definida como una ley de potencias respecto del tiempo

$$R(t) \propto \left(\frac{E_x}{\rho_0} \right)^{1/5} t^{2/5} \quad (7.22)$$

donde la constante de proporcionalidad depende de la ecuación de estado del gas. Teniendo en cuenta las expresiones anteriores y reemplazando en el sistema de ecuaciones (7.15), (7.16) y (7.17) se llega al sistema de ecuaciones diferenciales ordinarias planteado por Taylor y Sedov

$$[U(\xi) - \xi] \xi \Omega'(\xi) + [\xi U'(\xi) + 2U(\xi)] \Omega(\xi) = 0 \quad (7.23)$$

$$-\frac{3}{2} U(\xi) \Omega(\xi) + [U(\xi) - \xi] U'(\xi) \Omega(\xi) + P'(\xi) = 0 \quad (7.24)$$

$$-3\Omega(\xi) P(\xi) + [U(\xi) - \xi] [\Omega(\xi) P'(\xi) - \gamma P(\xi) \Omega'(\xi)] = 0 \quad (7.25)$$

La integración de este sistema de ODEs entre $0 \leq \xi \leq 1$ se puede realizar utilizando un método de integración para problemas de valores iniciales tal como el método Runge-Kutta de cuarto orden. Para ello se deben especificar las condiciones iniciales para cada una de las variables $V = (U, \Omega, P)^T = (V_1, V_2, V_3)^T$. Estas condiciones se obtienen a partir de asumir que la

onda de choque es fuerte ($M \gg 1$). Las ecuaciones de Rankine-Hugoniot que permiten relacionar las variables del flujo y termodinámicas a uno y otro lado de una onda de choque para el caso de un gas perfecto (relación constante de calores específicos del gas) pueden escribirse como

$$M_2^2 = \frac{2 + (\gamma - 1)M_1^2}{2\gamma M_1^2 - (\gamma - 1)} \quad (7.26)$$

$$\frac{\rho_1}{\rho_2} = \frac{\gamma - 1}{\gamma + 1} + \frac{2}{(\gamma + 1)M_1^2} \quad (7.27)$$

$$\frac{p_2}{p_1} = \frac{2\gamma M_1^2}{\gamma + 1} - \frac{\gamma - 1}{\gamma + 1} \quad (7.28)$$

Luego, en el límite de choque fuerte las relaciones (7.26), (7.27) y (7.28) permiten expresar las variables del flujo inmediatamente detrás de la onda de choque de la siguiente manera

$$U(\xi = 1) = \frac{2}{\gamma + 1}, \quad \Omega(\xi = 1) = \frac{\gamma + 1}{\gamma - 1}, \quad P(\xi = 1) = \frac{2}{\gamma + 1} \quad (7.29)$$

Reordenando el sistema de Ecs.(7.23), (7.24) y (7.25) de la siguiente manera

$$V_1' = \frac{3V_2V_3 - \frac{3}{2}V_2^2V_1[V_1 - \xi] - 2\gamma/\xi V_1V_2V_3}{\gamma V_2V_3 - [V_1 - \xi]^2V_2^2} \quad (7.30)$$

$$V_2' = \frac{-[\xi V_1' + 2V_1]V_2}{[V_1 - \xi]\xi} \quad (7.31)$$

$$V_3' = \frac{3}{2}V_2V_1 - [V_1 - \xi]V_2V_1' \quad (7.32)$$

puedo expresar a $V_1' = f_1(V_1, V_2, V_3, \xi)$, $V_2' = f_2(V_1, V_2, V_3, \xi)$ y $V_3' = f_3(V_1, V_2, V_3, \xi)$. Llevando a cabo la integración numérica del sistema con el método de Runge-Kutta de cuarto orden (asumiendo que $\gamma = 1.4$), se obtienen los perfiles auto-similares para las variables del flujo que se presentan en la figura 7.38.

Finalmente, para determinar la posición del frente de onda esférico en el tiempo resulta necesario determinar el valor de la constante de proporcionalidad en la Ec.(7.22). Si se indica con Q a la constante de proporcionalidad, se puede evaluar la energía total en el perfil auto-similar, que debe ser igual a la energía liberada por la explosión E_x . Para ello, se debe plantear la integral de la energía por unidad de volumen en el diferencial de volumen $dV = 4\pi r^2 dr$, donde la energía del flujo está dada por

$$E = e + \frac{\rho}{2}u^2 \quad (7.33)$$

$$= C_v \rho T + \frac{\rho}{2}u^2 \quad (7.34)$$

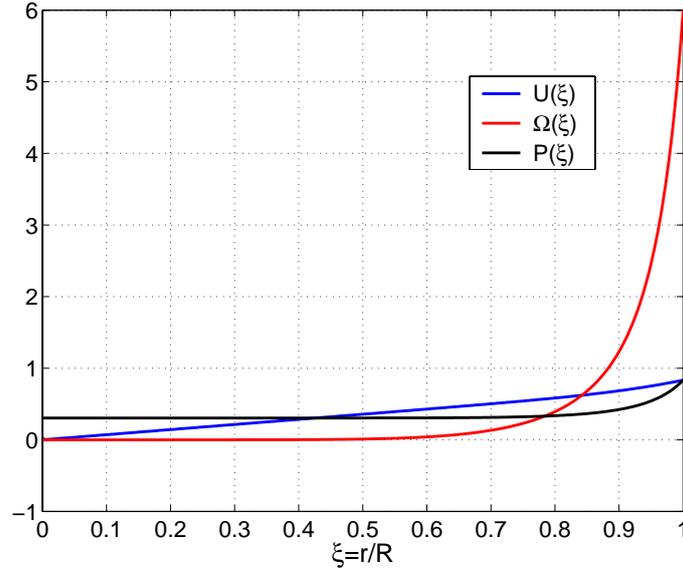


Figura 7.38: Perfiles auto-similares para las variables del flujo Ω , U , P en el problema de Taylor-Sedov esférico.

Teniendo en cuenta la ecuación de estado del gas y la relación que existe entre los calores específicos C_v , C_p y la constante del gas \mathfrak{R} , se llega a

$$E = \frac{P}{\gamma-1} + \frac{\rho}{2}u^2 \quad (7.35)$$

Luego, la energía de la explosión puede expresarse de la siguiente manera

$$E_x = \int E dV \quad (7.36)$$

$$= \int_0^R \left(\frac{P}{\gamma-1} + \frac{\rho u^2}{2} \right) 4\pi r^2 dr \quad (7.37)$$

Haciendo el cambio de variables de r a ξ , teniendo en cuenta las Ecs.(7.22) y (7.13), y despejando para la constante de proporcionalidad Q

$$Q = \left(\frac{16\pi}{25} \int_0^1 \left(\frac{P(\xi)}{\gamma-1} + \frac{\Omega(\xi)U(\xi)^2}{2} \right) \xi^2 d\xi \right)^{-1/5} \quad (7.38)$$

la que por integración numérica de los perfiles auto-similares $P(\xi)$, $\Omega(\xi)$, $U(\xi)$, se determina que posee el valor $Q \simeq 1.1653$ [Hut00, Tho02].

7.4.2. Solución numérica

Setup del problema

Se propone resolver el flujo generado por una explosión esférica en una atmósfera con condiciones uniformes. La explosión es causada por la liberación de energía en forma localizada en una zona del dominio computacional centrada en el origen $(0, 0, 0)$ cuyo radio es $R_0 \simeq 0.25\text{m}$. La energía liberada por la explosión es tal que la presión alcanza el valor $p_{blast} = 10^5 \cdot p_{atm}$, siendo p_{atm} el valor de la presión en la atmósfera. Se considera que la atmósfera donde ocurre la explosión se encuentra en condiciones estándar, es decir que la densidad del aire es $\rho_0 = 1.225\text{kg/m}^3$ y la presión $p_{atm} = 101325\text{Pa}$. El gas se encuentra en reposo en todo el dominio.

También se asume que la explosión inicial (no simulada) es una transformación que ocurre instantáneamente a volumen constante. De esta manera, la densidad del gas en el volumen que representa al foco de la explosión es igual a la densidad ambiente ρ_0 . Se debe tener en cuenta que la condición inicial utilizada para el cálculo de la solución es un salto abrupto de la presión y la temperatura al pasar del interior al exterior de la esfera de radio $R_0 \simeq 0.25\text{m}$. Dado que en los instantes iniciales los perfiles de la solución no poseen la forma de los perfiles autosimilares, en realidad la comparación de la posición de la onda de choque esférica en el tiempo obtenida mediante las simulaciones y la solución analítica son realizadas a partir de un cierto instante para el cual se observa que la solución numérica sí desarrolla un perfil similar al analítico.

La ecuación de estado que relaciona las variables termodinámicas del gas es la ecuación de estado para gases perfectos. El coeficiente isoentrópico del gas es igual a $\gamma = 1.4$ y el calor específico a volumen constante $c_v = 717\text{J}/(\text{Kg K})$. Finalmente, la constante del aire es $\mathfrak{R}_{aire} = 287\text{J}/(\text{Kg K})$.

El problema es resuelto utilizando una malla no estructurada de tetraedros, sobre un dominio esférico de radio $R_{ext} = 5\text{m}$. La figura 7.39 presenta un corte de la malla base en el plano $X - Z$, en la cual se observa el campo de presiones para la condición inicial y el detalle de la malla en el foco de la explosión. La malla base es generada con el software netgen[SGG04], y presenta una mayor concentración de elementos hacia el centro del dominio con un gradiente en el tamaño de los mismos igual a $\nabla h = 0.2$. El mallador permite prescribir el tamaño aproximado de los elementos en las cercanías de un punto, en este caso el centro de la explosión. El tamaño que se prescribe es $h \simeq 0.03\text{m}$, con el objeto de que la zona inicial de la explosión esté representada con precisión. Así la malla base está constituida aproximadamente por 421.000 elementos y 76.500 nodos.

La única condición de borde que se impone al problema es la fijación de la presión al valor ambiente $p|_{\Gamma} = p_{atm}$ en la superficie esférica Γ que limita al dominio computacional. Se puede utilizar esta condición de borde dado que el problema se deja evolucionar hasta un instante de tiempo para el cual el frente de la onda de choque esférica se encuentra suficientemente lejos del contorno con esta condición de borde. Las ecuaciones de Euler en 3-D son resueltas

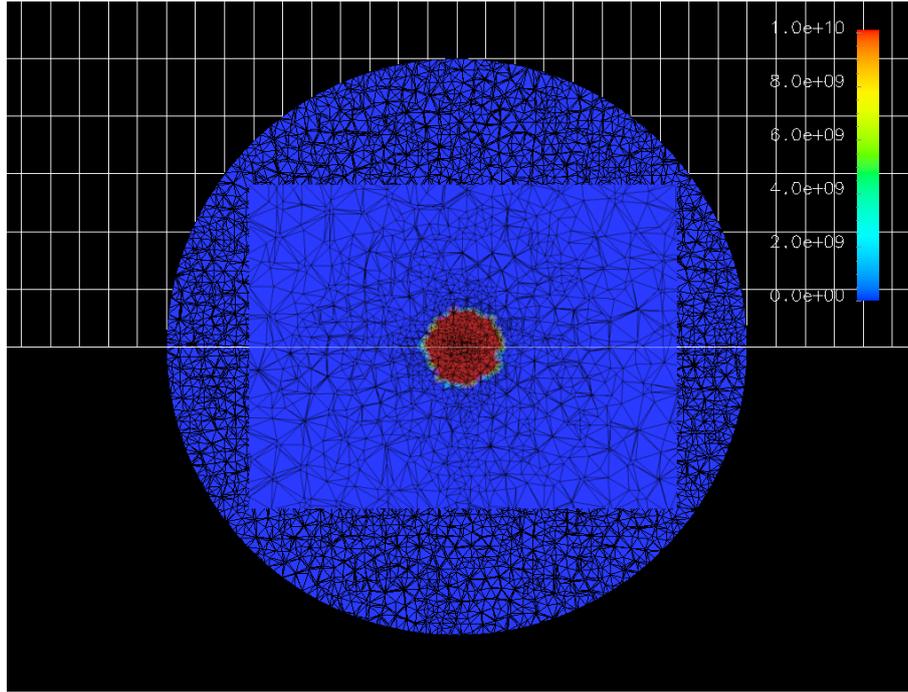


Figura 7.39: Detalle de la condición inicial para la presión en un corte de la malla base sobre el plano $X - Z$ para el problema de Taylor-Sedov esférico.

con PETSc-FEM utilizando 15 procesadores del *cluster* Aquiles, utilizando un esquema de integración temporal Backward-Euler.

Solución adaptativa

El paso de tiempo es estimado cada vez que se adapta la malla. Dadas las características del problema, se elige un Courant variable para estimar el paso de tiempo. En los primeros pasos de la simulación (hasta aprox. $t \simeq 1.0e - 5s$) se considera $Cou = 0.8$ dado que la intensidad de la onda de choque es mayor. Luego, la restricción en el paso de tiempo puede ser relajada, prescribiéndose un Courant igual a $Cou = 2$. Las regiones del dominio computacional que requieren ser refinadas / desrefinadas son seleccionadas en función de la magnitud del gradiente elemental de la densidad $\nabla\rho(T)$. Esto quiere decir que si la densidad en el elemento T es tal que

$$\| \nabla\rho(T) \| \cdot h(T) > c_1 \cdot \max_{T_i \in \tau} (\| \nabla\rho(T_i) \| \cdot h(T)) \quad (7.39)$$

entonces el elemento T es seleccionado para ser refinado. El valor de la constante $c_1 \simeq 0.15$ en todo el procedimiento de cálculo y el tamaño del elemento es calculado como el radio de la esfera circunscripta en el elemento.

Se prescriben dos niveles de refinamiento como máximo y la frecuencia de adaptación de la malla se estima en 10 pasos de tiempo, teniendo en cuenta el criterio mencionado en el capítulo 6. El tiempo final para la simulación es de $t_f \simeq 0.001s$. Este tiempo final se estima en función de la solución que proporciona la ecuación 7.22 para la posición de la onda de choque, teniendo en cuenta que la misma debe encontrarse lejos del contorno donde se mantiene fijada la presión.

Solución sin adaptación de la malla

En este caso la solución se resuelve sobre la malla base utilizada en el procedimiento adaptativo. De manera semejante a la solución obtenida en forma adaptativa, hasta el instante (aprox. $t \simeq 4.0e - 5$ s) la solución se calcula con $Cou = 0.6$, para luego mantenerse constante e igual a $Cou = 2.0$ hasta el tiempo final $t_f \simeq 0.001$ s.

7.4.3. Resultados

Se analiza la posición de la onda de choque esférica en el tiempo y se comparan los resultados obtenidos en las simulaciones con y sin el procedimiento de adaptación de mallas con aquél que se obtiene de evaluar la Ec.(7.22), teniendo en cuenta el valor de la constante de proporcionalidad dado por la ecuación (7.38). Para el caso de las soluciones obtenidas numéricamente, la posición de la onda de choque esférica se determina a partir de considerar la solución a lo largo de una línea en la dirección radial y asumiendo que la onda de choque ha llegado a la posición indicada cuando el incremento en la densidad ambiente es mayor al 5%. De esta manera, se confecciona el gráfico de la figura 7.40, donde se observa que las soluciones obtenidas con y sin adaptación de la malla prácticamente no difieren una de otra. Sin embargo, tal como se observa en la figura 7.40 y se resume en la tabla 7.3, la posición de la onda de choque para las soluciones obtenidas en las simulaciones presentan un retraso si se las compara con la posición dada por la Ec.(7.22).

t[seg]	$(R_{analit} - R_{adapt})/R_{analit}$	$(R_{analit} - R_{no-adapt})/R_{analit}$
0.0001	0.06	0.08
0.00025	0.12	0.09
0.00045	0.16	0.12

Cuadro 7.3: Posición de la onda de choque para distintos instantes de tiempo - Comparaciones entre las soluciones numéricas y analítica.

Sólo a los fines ilustrativos, la figura 7.41 presenta las caras expuestas de la malla adaptada para el instante de tiempo $t = 0.645$ ms. Dado que se utilizan dos niveles de refinamiento, las caras que se observan en dicha figura corresponden a los elementos pertenecientes al primer nivel de refinamiento. Para este paso de tiempo, la malla posee 2341440 elementos y 427972 vértices.

Las figuras 7.42 presentan el estado para la densidad y la malla para el instante $t = 0.645$ ms. En la figura 7.42.a) se puede ver que tal como lo indica el perfil analítico para la densidad de la figura 7.38, la mayor parte del material barrido por la onda de choque esférica es comprimido y se encuentra formando una cáscara delgada inmediatamente por detrás de la onda de choque. Por otro lado, en la figura 7.42.b) se observa que la onda de choque es capturada efectivamente por el procedimiento de adaptación de la malla. Las figuras 7.43 presentan el campo de presiones y del número de Mach para el mismo instante. El Mach del flujo detrás de la onda es

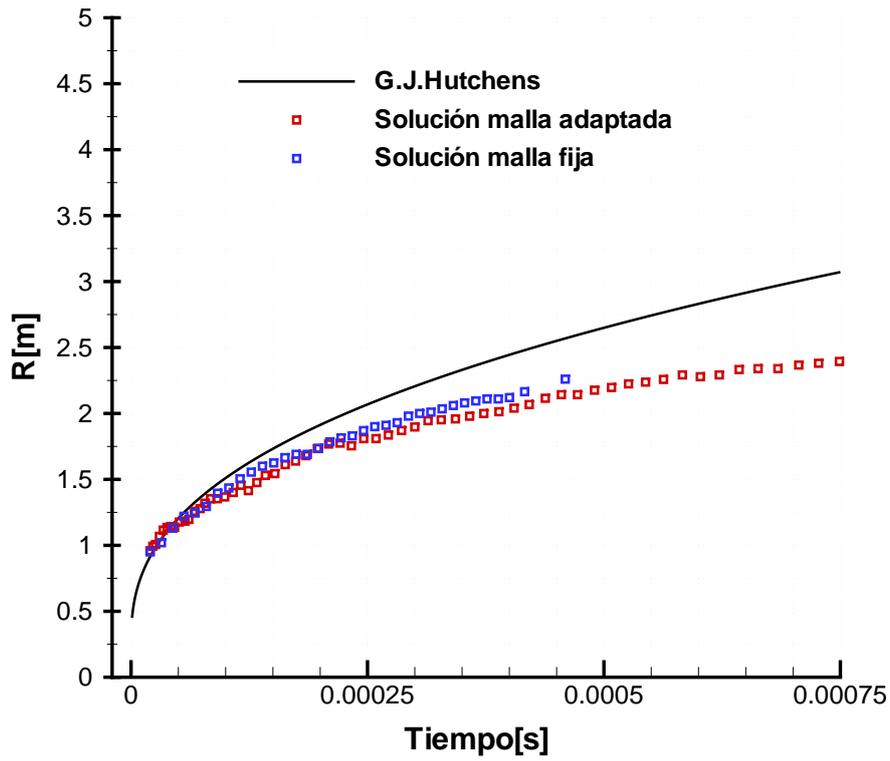


Figura 7.40: Posición de la onda de choque en función del tiempo - Comparación entre la solución teórica dada por la Ec.(7.22) y las soluciones obtenidas con y sin el procedimiento de adaptación de la malla.

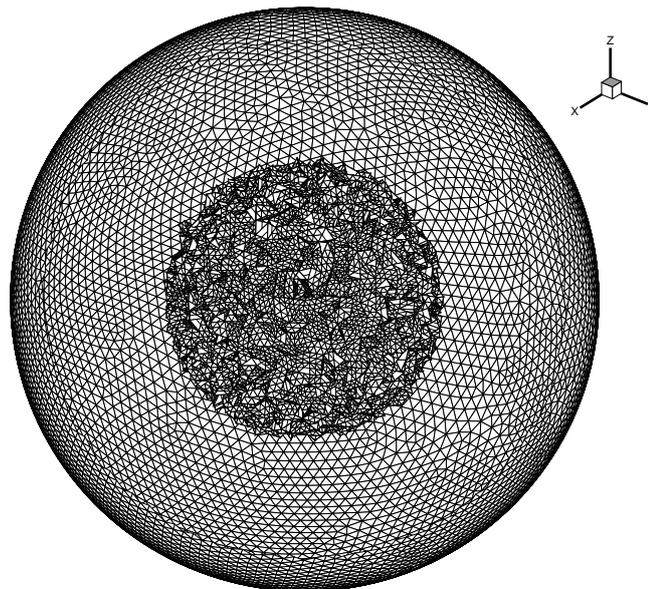


Figura 7.41: Caras expuestas de la malla adaptada para $t = 0.645$ ms.

subsónico mientras que la presión es prácticamente uniforme en la misma región. Observar que el valor de la presión en el centro de la explosión es aproximadamente igual a la mitad del valor máximo de la presión inmediatamente detrás de la onda de choque, lo cual concuerda con la solución autosimilar para la presión dada por la figura 7.38. Teniendo en cuenta la posición de la onda de choque para el instante $t = 0.5\text{ms}$, se puede indicar que en promedio, hasta ese instante, la onda de choque ha viajado a una velocidad aproximada de 3000m/seg.

Finalmente, a los fines de evaluar la capacidad del procedimiento adaptativo para mejorar la precisión con la cual se resuelve el campo de movimiento, se presentan los perfiles de densidad y número de Mach para distintos instantes de tiempo sobre el eje Z en la figura 7.44. Se observa para ambas variables que en los primeros instantes de tiempo ($t \simeq 10\mu\text{s}$) la onda de choque es resuelta con mayor precisión (y su intensidad es mayor) si se la compara con la solución obtenida sobre la malla no adaptada. El refinamiento permite resolver la onda de choque en una zona más delgada al tiempo que la fuerte expansión que se genera por detrás de la onda de choque también es resuelta sin presentar oscilaciones en la solución. Estas oscilaciones espúrias pueden atribuirse a la falta de estabilización de la solución. Sin embargo, no debe confundirse este hecho con el hecho de pensar que la solución adaptada es más difusa, ya que tal cual se observa, tanto la onda de choque como la expansión son resueltas más abruptamente con este procedimiento. Finalmente, observar que el salto en el número de Mach que ocurre en las curvas correspondientes para la solución sin adaptar en la zona de la expansión no se presenta cuando el flujo se resuelve adaptativamente.

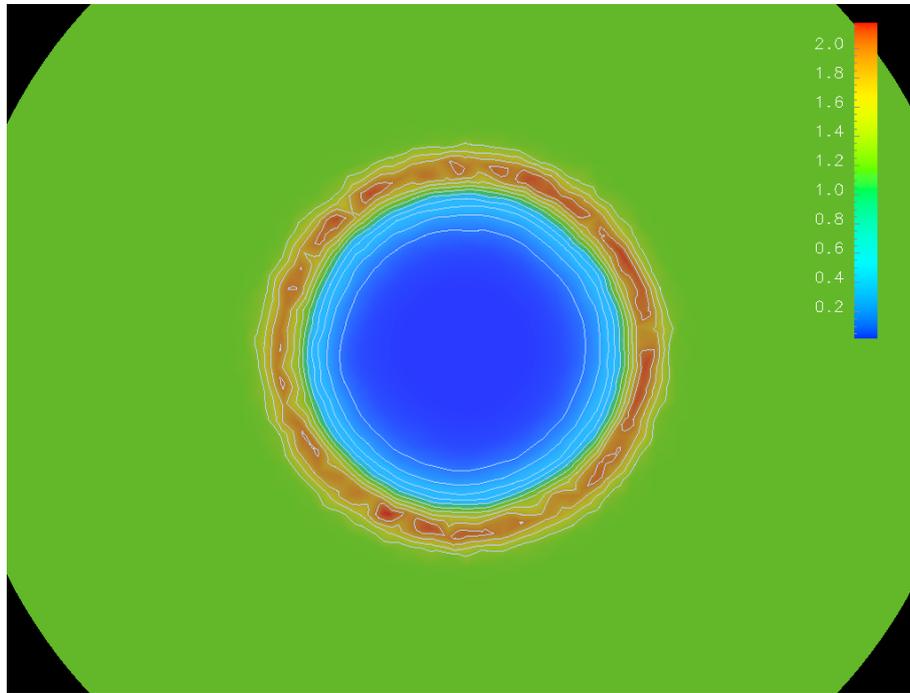
Calidad de las mallas

La calidad de las mallas obtenidas durante el cálculo adaptativo de la solución se analiza utilizando la métrica de Liu (ver ecuación 2.14). Además, tal como se menciona en el capítulo 2, se asume que la distribución de los valores de la métrica de calidad para los elementos de la malla responden a una distribución normal. Luego, la tabla 7.4 resume los datos estadísticos para la malla base y para dos mallas adaptadas.

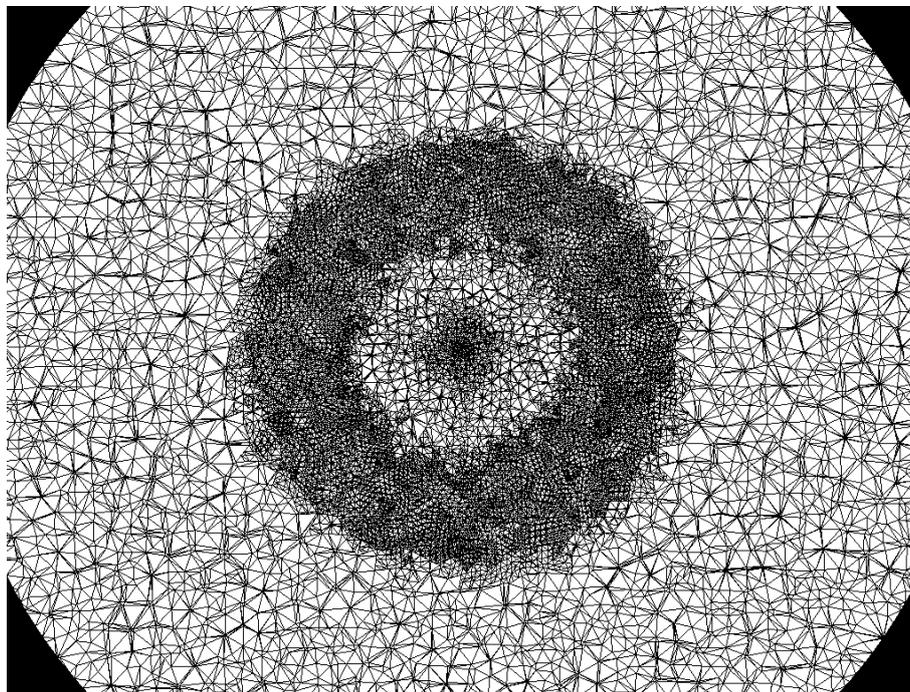
STEP	$\bar{\eta}$	η_{min}	η_{std}
0 (malla base)	0.8911	0.5416	0.0561
550	0.8841	0.5416	0.0571
1240	0.8833	0.5416	0.0573

Cuadro 7.4: Datos estadísticos sobre la calidad de las mallas para el problema de la explosión esférica.

Se observa que la calidad mínima no disminuye, lo cual se debe a que el elemento de peor calidad de la malla nunca fue refinado. Se concluye entonces que en este caso η_{min} no es suficiente para caracterizar la calidad de las mallas obtenidas con el refinamiento. Por otro lado, se ve que la calidad media $\bar{\eta}$ y la desviación estándar η_{std} permiten detectar pequeños cambios en

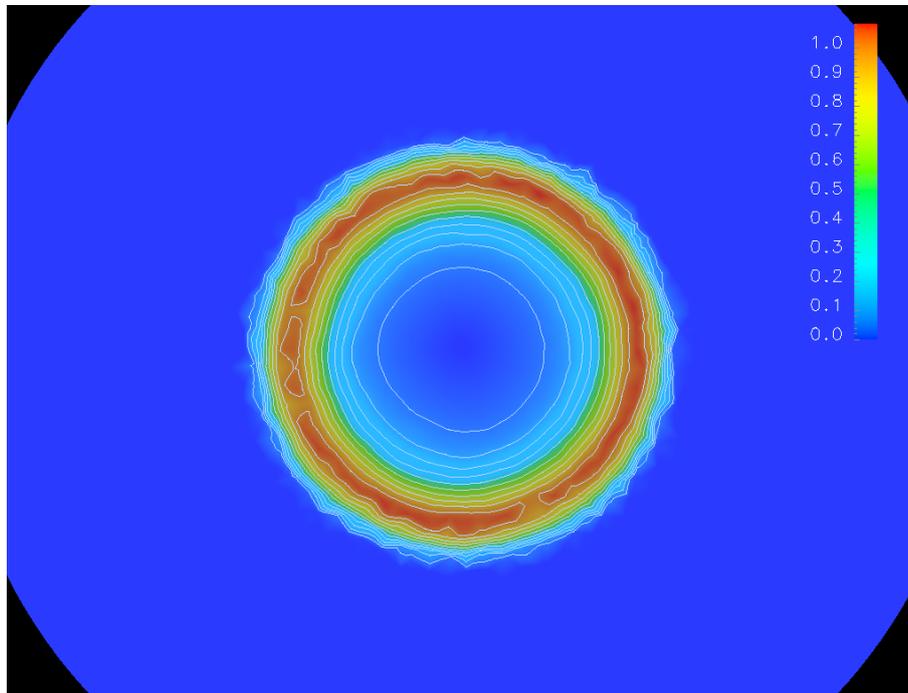


(a) Isolíneas de densidad.

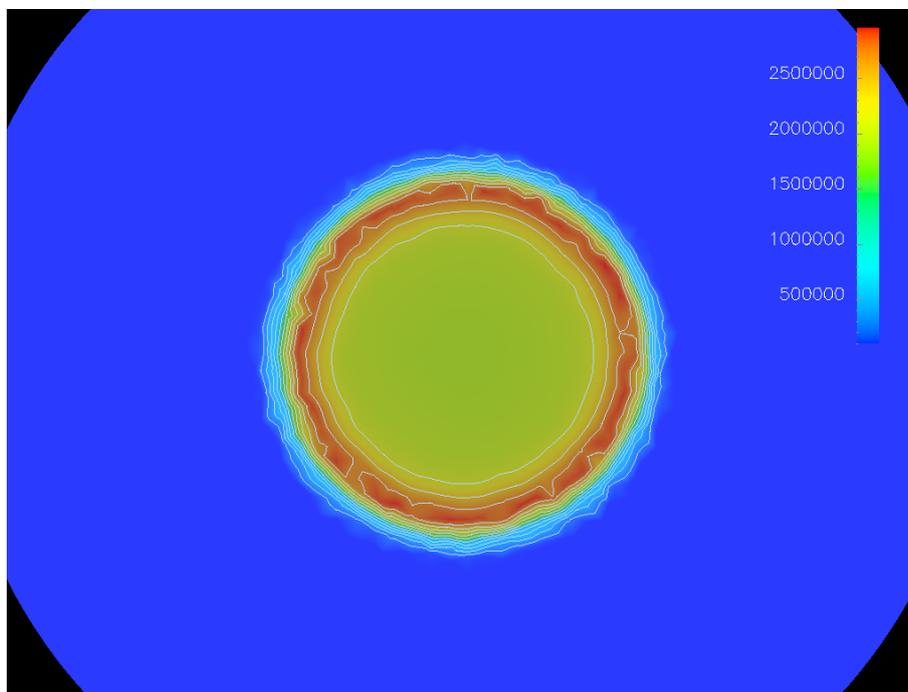


(b) Corte de la malla adaptada.

Figura 7.42: Solución en el plano XZ para la densidad y malla adaptada - $t = 0.645\text{ms}$.



(a) Isolneas de Mach.



(b) Isolneas de Presión.

Figura 7.43: Solución en el plano XZ para el número de Mach y la presión - $t = 0.645\text{ms}$

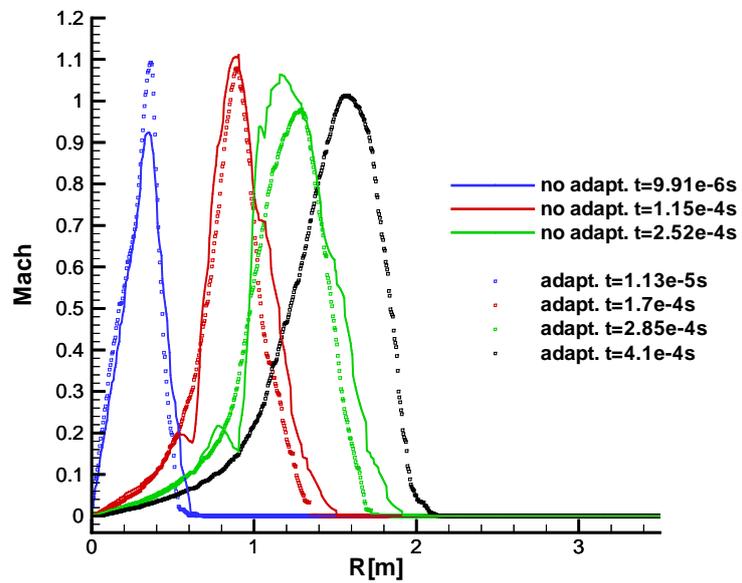
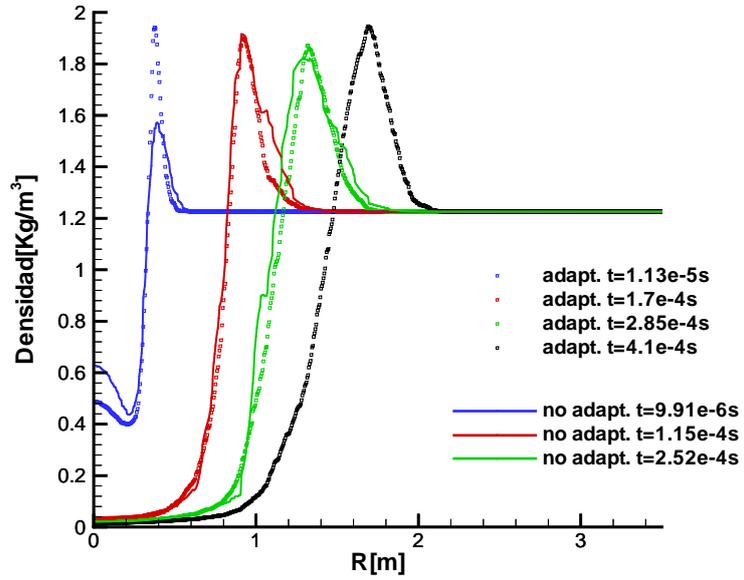


Figura 7.44: Perfiles de densidad y Mach sobre el eje Z para distintos instantes de tiempo.

la calidad de las mallas adaptadas. En función de estos valores se concluye, para este problema, que la calidad de las mallas adaptadas es prácticamente igual a la de la malla base ya que la disminución en la calidad media es menor al 1 % mientras que el incremento en la desviación es aproximadamente del 2 %. La figura 7.45 presenta un histograma con la distribución de la métrica η para la malla adaptada en el paso de tiempo 1240. Se superpone en la misma figura la curva que representa a la distribución normal.

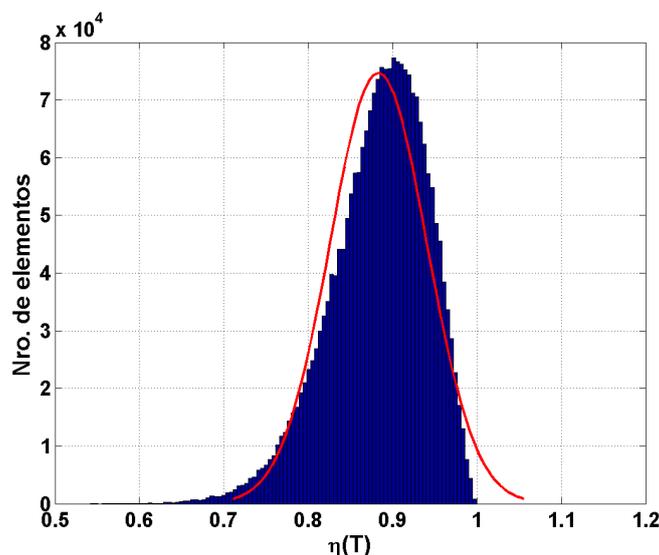


Figura 7.45: Histograma con la distribución de $\eta(T)$ para los elementos de la malla adaptada $\text{step}=1240$.

7.4.4. Costo relativo de la estrategia de adaptación de mallas

Se realiza un análisis sobre el tiempo requerido para realizar la adaptación de la malla en relación al tiempo necesario para resolver las ecuaciones del flujo en el problema de la explosión esférica, teniendo en cuenta que el flujo es resuelto en paralelo utilizando 15 procesadores del *cluster* de PC, mientras que la adaptación de la malla se lleva a cabo secuencialmente. En lo que se refiere al tiempo de adaptación de la malla (t_{adapt}) se incluyen todas las tareas necesarias para adaptar la malla y generar los archivos requeridos para reiniciar el cálculo de la solución. Estas tareas incluyen la selección de los elementos a refinar, el refinamiento propiamente dicho, la actualización de las condiciones de borde, la proyección del último estado computado sobre la malla nueva y la estimación del paso de tiempo (ver diagrama 6.9). Por otro lado, dentro del tiempo requerido por el *solver* de las ecuaciones (t_{solver}) se incluye el *overhead* o costo adicional para reiniciar la corrida cada n_{step} pasos de tiempo además del tiempo necesario para avanzar la solución. La gráfica de la figura 7.46 presenta los resultados obtenidos para la relación t_{adapt}/t_{total} , donde $t_{total} = t_{adapt} + t_{solver}$. Se observa que dicha relación se mantiene prácticamente constante y que el procedimiento de adaptación de la malla demanda aproximadamente el 4 % del tiempo total requerido para resolver el problema. Dados los comentarios

que Waltz[Wal04] realiza sobre este aspecto, se concluye que el procedimiento de adaptación de la malla tiene un costo relativo pequeño en relación al tiempo requerido para resolver el problema y que se encuentra en el rango de valores típicos para códigos de solución adaptativos parcialmente paralelizados (ver sección 6.3). Por lo tanto, a menos del costo adicional en el que se incurre cada vez que se reinicia el cálculo de la solución, el procedimiento de adaptación en sí mismo permite utilizar una frecuencia de adaptación de la malla elevada.

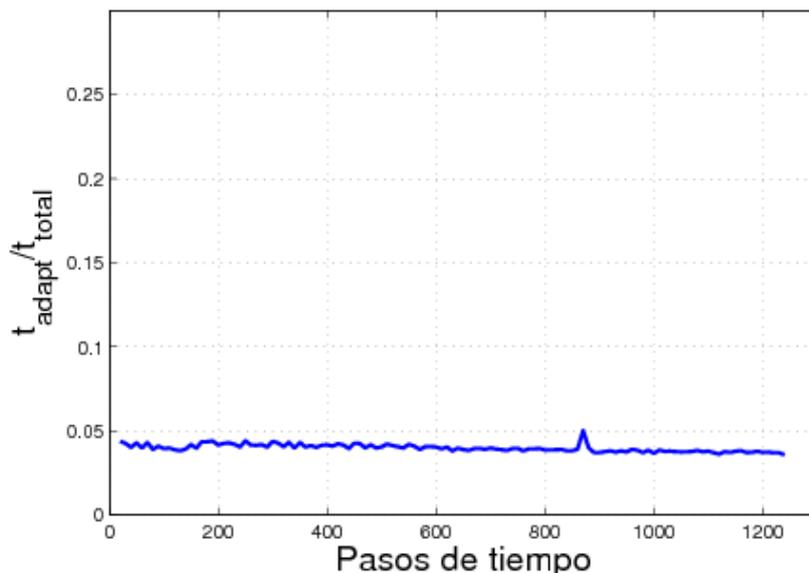


Figura 7.46: Costo relativo para la adaptación de la malla en el problema de la explosión esférica.

7.4.5. Escalabilidad del algoritmo de refinamiento

Una característica importante de todo algoritmo es su escalabilidad, haciendo referencia con ello al comportamiento del algoritmo frente al incremento en el tamaño del problema que debe resolver. Para un algoritmo de refinamiento de mallas, el tamaño del problema está dado por el número de elementos que deben ser refinados / desrefinados. Un comportamiento ideal del algoritmo implicaría una relación lineal entre el número de elementos que son procesados y el tiempo de cálculo requerido. Con el objeto de analizar la escalabilidad del algoritmo de refinamiento de mallas implementado en esta tesis, se evalúa el tiempo reloj que requiere la rutina de refinamiento de la malla en un problema práctico tal como el de la explosión esférica de Taylor-Sedov presentado anteriormente. Ya que la onda de choque esférica incrementa su superficie a medida que el tiempo evoluciona, el tamaño del problema crece con dicha evolución. La figura 7.47 permite mostrar que el procedimiento de refinamiento presenta un comportamiento casi lineal, al menos en el rango de número de elementos refinados que se presenta ($50.000 < Ne_{le_{ref}} < 240.000$). Realizando un ajuste por mínimos cuadrados, se determina que

la recta que mejor ajusta a los puntos datos relevados en las mediciones, la cual se indica en trazo rojo continuo, posee una pendiente $\alpha = 1.2285e - 4$ (segundos por elemento), por lo tanto se puede inferir que una medida aproximada de la velocidad de refinamiento es de 8140 tetraedros por segundo en un procesador Intel Pentium 4 3.0GHz.

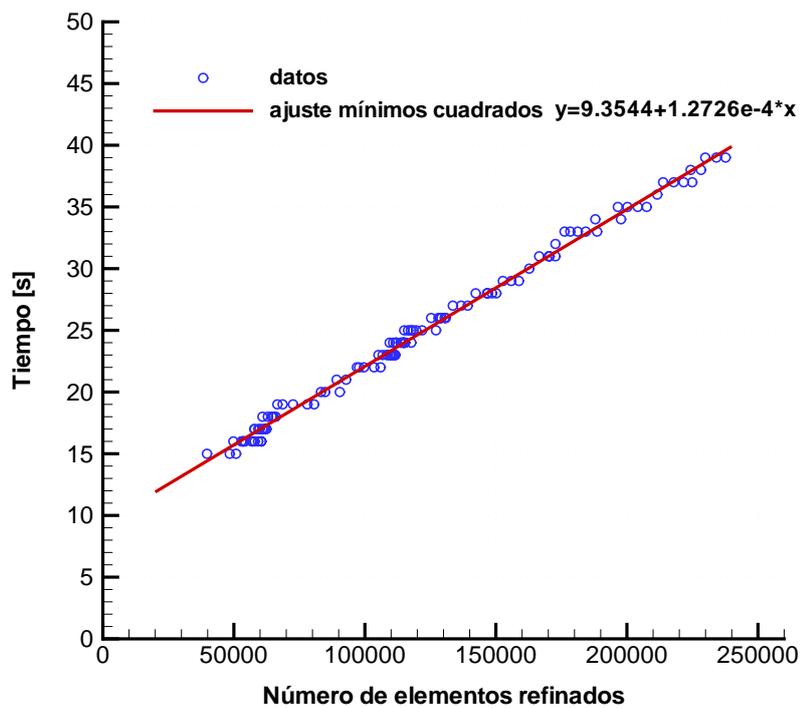


Figura 7.47: Escalabilidad del algoritmo de refinamiento.

Capítulo 8

Conclusiones, Contribuciones y Trabajos a Futuro

8.1. Conclusiones

Se ha desarrollado una estrategia que permite realizar adaptación tipo-h de mallas no estructuradas de elementos finitos bidimensionales y tridimensionales, orientada a la resolución de flujos compresibles. El código que implementa a dicha estrategia permite mejorar la precisión en el cálculo de la solución utilizando eficientemente los recursos computacionales disponibles y sin comprometer la calidad de las mallas adaptadas. Se comprueba que las discontinuidades en el flujo son resueltas de manera precisa. También se observa en la práctica que el esquema de adaptación para problemas no estacionarios no introduce inestabilidades en la solución, aún cuando los parámetros de estabilización utilizados para resolver el flujo son de menor magnitud que los utilizados cuando el problema se resuelve sobre una malla fija. Tampoco se observa que el refinamiento agregue difusión numérica a la que de por sí introduce el método SUPG + *shock capturing* utilizado. A pesar de ello, cuestiones de este tipo requieren de un análisis teórico para ser respondidas desde un punto de vista formal, por lo cual es necesario trabajar en este aspecto.

Por otro lado, se considera que si bien los indicadores de error que utilizan el gradiente de la solución son efectivos, fáciles de utilizar y permiten capturar correctamente las discontinuidades en el flujo, requieren un ajuste ad-hoc por parte del usuario que puede atentar contra la aplicación efectiva del procedimiento de adaptación. Estos métodos para indicar la regiones a refinar son muy populares, pero es necesario contar con estimadores de error elaborados desde el punto de vista formal para poder evaluar los beneficios reales en cuanto a la reducción del error local en la solución debido a la adaptación de la malla. En este sentido, el estimador de error basado en la reconstrucción del gradiente permite detectar otras regiones del dominio que deben ser refinadas además de las regiones con gradientes más grandes. Sin embargo, resulta necesario fijar criterios en cuanto a cómo evaluar los procedimientos de estimación de error para poder determinar cuál es el más conveniente.

Las evaluaciones de performance realizadas a través de mediciones de tiempo y el moni-

toreo de recursos requeridos por la implementación de la estrategia de adaptación en lenguaje C++ permiten determinar que la etapa de cálculo del flujo demanda un esfuerzo computacional notablemente mayor que la de adaptación de la malla, aún en el caso de esta tesis en donde la adaptación se realiza secuencialmente. Por lo tanto, se puede concluir en este sentido que en el caso de problemas no estacionarios resulta conveniente adaptar la malla cada pocos pasos de tiempo. También se ha mostrado que el uso de la estrategia de adaptación permite reducir el tiempo de cálculo. Sin embargo, dicha reducción es dependiente del problema que se resuelve.

La implementación de la estrategia de adaptación en lenguaje C++ y haciendo uso de las librerías MPICH2, STL y Boost tiene como objeto facilitar la continuidad en el desarrollo del trabajo realizado según la lista de trabajos a futuro que se plantean en la correspondiente sección.

8.2. Contribuciones

Las contribuciones del trabajo desarrollado en el transcurso de la tesis son las siguientes

- La estrategia desarrollada y su implementación amplían la capacidad del *software* PETSc-FEM[SNP⁺08], permitiendo resolver problemas de mayor tamaño y complejidad en cuanto a los fenómenos físicos que puedan presentar.
- La estrategia desarrollada permite adaptar mallas no estructuradas de elementos finitos sin comprometer la calidad geométrica de la mismas.
- La extensión a 3-D del criterio de nodo 1-irregular permite obtener un algoritmo de refinamiento simple, que escala casi linealmente con el tamaño del problema de adaptación.
- La implementación modular de la estrategia facilita su aplicación a otra clase de problemas, modificando en principio sólo el procedimiento de estimación del error en la solución aproximada.

8.3. Trabajos a futuro

Evidentemente el desarrollo de un *software* de adaptación de mallas es una tarea compleja y multidisciplinaria, que involucra varios aspectos, entre ellos el análisis riguroso del método de elementos finitos utilizado para resolver las ecuaciones de gobierno con el objeto de obtener un procedimiento de estimación del error adecuado, el análisis de las estructuras de datos más convenientes para representar la malla y sus adyacencias, los algoritmos que operan sobre las estructuras de datos, y la evaluación de la calidad de la malla. Considerando las dificultades encontradas y las exigencias a un código de tales características, se considera que los aspectos más urgentes a atender en el futuro son:

- Revisar la estructura de datos utilizada para representar las entidades geométricas de la malla y sus adyacencias.
- Analizar otros procedimientos para estimar el error cuando la ecuación de advección-difusión posee término advectivo dominante. En este sentido se cree que sería valioso realizar algunas experiencias más con el estimador de error basado en la reconstrucción del gradiente.
- Además, se deberían probar / introducir esquemas de refinamiento que contemplen la posibilidad de refinar en una dirección preferencial.
- Dada la necesidad de resolver problemas de mayor tamaño, se propone desarrollar un código que implemente una estrategia similar a la desarrollada pero aprovechando el potencial del cálculo en paralelo. Esto involucra que la revisión de la estructura de datos que se plantea en el primer ítem de esta lista presente características adecuadas para el cálculo distribuido. La implementación del código en paralelo se realizaría utilizando la librería MPICH2 [Arg07] y ParMetis [Kar05].
- En problemas no estacionarios, analizar la posibilidad de realizar adaptación en la variable temporal.
- Iniciar un análisis teórico que permita responder si el esquema de adaptación propuesto modifica la estabilidad del esquema SUPG.
- Incorporar la estrategia de *untangling* y *smoothing* desarrollada en [LNS06] para resolver el problema de inversión de los elementos cuando se proyectan vértices sobre contornos curvos.

Apéndice A

Propiedades asociadas a la métrica η

Se demuestra la relación que existe entre la métrica η con el volumen v y las longitudes l_{ij} de las aristas de un tetraedro. La independencia de la métrica con respecto del nodo de referencia utilizado para calcularla ha quedado demostrada en la sección 2.1.1, a partir de mostrar que la matriz Jacobiana A es nodalmente invariante.

En primera instancia se definen los elementos necesarios para llevar a cabo la demostración. Sea el tetraedro regular R definido por los vértices (r_0, r_1, r_2, r_3) de coordenadas

$$\begin{aligned}r_0 &= (-\sqrt{3}a/2, 0, 0) \\r_1 &= (0, -a/2, 0) \\r_2 &= (0, a/2, 0) \\r_3 &= (-\sqrt{3}a/6, 0, -\sqrt{6}a/3)\end{aligned}$$

que se presenta en la figura A.1. Sea T un tetraedro cualquiera definido por los vértices (t_0, t_1, t_2, t_3) . Se indica con R a la matriz Jacobiana del tetraedro R referenciada al vértice r_0 y con T a la matriz Jacobiana del tetraedro T referenciada a su vértice t_0 . Luego,

$$R = [r_1 - r_0 | r_2 - r_0 | r_3 - r_0] \quad (\text{A.1})$$

es decir que

$$R = a \begin{bmatrix} \sqrt{3}/2 & \sqrt{3}/2 & \sqrt{3}/3 \\ -1/2 & 1/2 & 0 \\ 0 & 0 & \sqrt{6}/3 \end{bmatrix}, \quad R^{-1} = \frac{1}{a} \begin{bmatrix} 1/\sqrt{3} & -1 & -1/\sqrt{6} \\ 1/\sqrt{3} & 1 & -1/\sqrt{6} \\ 0 & 0 & 3/\sqrt{6} \end{bmatrix} \quad (\text{A.2})$$

Bajo la condición de que ambos T y R deben tener el mismo volumen v , es decir

$$v = \frac{1}{6} \det(R) \quad (\text{A.3})$$

se obtiene el valor de a

$$a = (6\sqrt{2}v)^{1/3} \quad (\text{A.4})$$

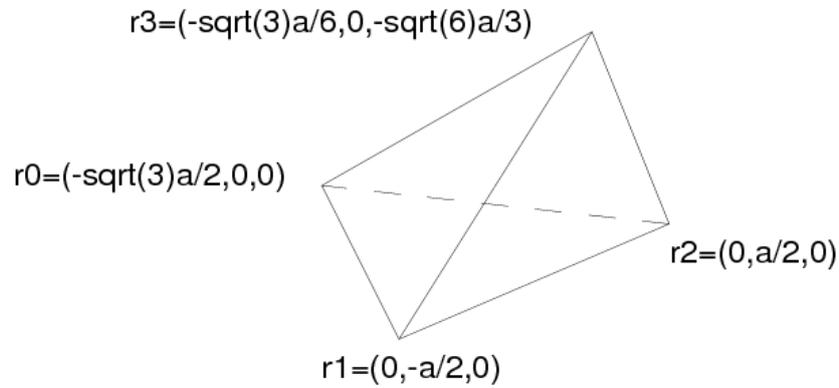


Figura A.1: Tetraedro regular R

Además se denota con el escalar $d_{ij} = (t_j - t_i)^T \cdot (t_j - t_i)$ al cuadrado de la longitud de la arista definida por los nodos ij .

■ **Relación con el volumen y longitudes de las aristas**

Para ello se debe desarrollar tanto la expresión del $\det(A(\mathbf{R}, \mathbf{T}))$ como de la traza $Tr(A(\mathbf{R}, \mathbf{T}))$. Por definición (ver sección 2.1.1) se sabe que

$$A(\mathbf{R}, \mathbf{T}) = \mathbf{M}^T(\mathbf{R}, \mathbf{T})\mathbf{M}(\mathbf{R}, \mathbf{T}) \quad (\text{A.5})$$

$$= (\mathbf{R}^{-1})^T \mathbf{T}^T \mathbf{T} \mathbf{R}^{-1} \quad (\text{A.6})$$

y por propiedad del determinante

$$\det A(\mathbf{R}, \mathbf{T}) = \det(\mathbf{M}^T \mathbf{M}) \quad (\text{A.7})$$

$$= \det(\mathbf{M}^T) \det(\mathbf{M}) \quad (\text{A.8})$$

$$= \det(\mathbf{M})^2 \quad (\text{A.9})$$

pero dado que se impone la condición de que R y T deben tener el mismo volumen, entonces

$$\det(\mathbf{M}) = \det(\mathbf{T}) \det(\mathbf{R}^{-1}) \quad (\text{A.10})$$

$$= 1 \quad (\text{A.11})$$

Con lo cual se concluye que

$$\det(A(\mathbf{R},\mathbf{T})) = 1 \quad (\text{A.12})$$

Por otra parte

$$T^T T = \begin{bmatrix} d_{01} & (d_{01} + d_{02} - d_{12})/2 & (d_{01} + d_{03} - d_{13})/2 \\ (d_{01} + d_{02} - d_{12})/2 & d_{02} & (d_{02} + d_{03} - d_{23})/2 \\ (d_{01} + d_{03} - d_{13})/2 & (d_{02} + d_{03} - d_{23})/2 & d_{03} \end{bmatrix} \quad (\text{A.13})$$

Luego de Ecs.(A.2) y (A.13) se obtiene

$$A(\mathbf{R},\mathbf{T}) = \begin{bmatrix} (2d_{01} + 2d_{02} - d_{12})/3 & \dots & \dots \\ \dots & d_{12} & \dots \\ \dots & \dots & \frac{1}{2}(d_{03} + d_{13} + d_{23}) - \frac{1}{6}(d_{01} + d_{02} + d_{12}) \end{bmatrix} \quad (\text{A.14})$$

de donde

$$Tr(A(\mathbf{R},\mathbf{T})) = \frac{1}{2a^2}(d_{01} + d_{02} + d_{03} + d_{12} + d_{13} + d_{23}) \quad (\text{A.15})$$

$$= \sum_{i,j=0,i \neq j}^3 l_{ij}^2 \quad (\text{A.16})$$

Reemplazando los valores para el $\det(A)$ y la $Tr(A)$ en la Ec.(2.13)

$$\eta(\mathbf{T}) = \frac{3\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}}{(\lambda_1 + \lambda_2 + \lambda_3)} \quad (\text{A.17})$$

se obtiene

$$\eta(\mathbf{T}) = \frac{3\sqrt[3]{1}}{\frac{1}{2a^2} \sum_{i,j=0,i \neq j}^3 l_{ij}^2} \quad (\text{A.18})$$

$$= \frac{6a^2}{\sum_{i,j=0,i \neq j}^3 l_{ij}^2} \quad (\text{A.19})$$

Reemplazando el valor de a dado por la Ec.(A.4) en esta última ecuación se llega a la Ec.(2.14)

$$\eta(\mathbf{T}) = \frac{6(6\sqrt{2}\nu)^{2/3}}{\sum_{i,j=0,i \neq j}^3 l_{ij}^2} \quad (\text{A.20})$$

$$= \frac{12(3\nu)^{2/3}}{\sum_{i=1}^6 l_{ij}^2} \quad (\text{A.21})$$

Apéndice B

Sistema de ecuaciones diferenciales ordinarias de Taylor-Sedov

En este apéndice se deriva el sistema de ecuaciones diferenciales ordinarias a partir del cual se obtienen los perfiles auto-similares que describen el comportamiento de las variables del flujo en el problema de la onda explosiva esférica o problema de Taylor-Sedov. A tal fin, se parte de las ecuaciones (7.10), (7.11) y (7.12) y se hace uso de las definiciones para las variables del flujo dadas por las ecuaciones (7.13) y sus derivadas (7.14).

Para operar sobre la ecuación de continuidad (7.10) es necesario calcular las siguientes derivadas parciales

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho_0}{\partial t} \Omega(\xi) + \rho_0 \left(-\xi \frac{\dot{R}}{R} \Omega'(\xi) \right) \quad (\text{B.1})$$

$$\frac{\partial \rho}{\partial r} = \frac{\partial \rho_0}{\partial r} \Omega(\xi) + \rho_0 \frac{1}{R} \Omega'(\xi) \quad (\text{B.2})$$

$$\frac{\partial u}{\partial r} = \frac{\dot{R}}{R} U'(\xi) \quad (\text{B.3})$$

Reemplazando las ecuaciones (B.2), (B.1) y (B.3) en la Ec.(7.10) se obtiene la siguiente

$$\begin{aligned} \dot{\rho}_0 \Omega(\xi) - \rho_0 \xi \frac{\dot{R}}{R} \Omega'(\xi) + \dot{R} U(\xi) \left(\rho_0' \Omega(\xi) + \rho_0 \frac{\Omega'(\xi)}{R} \right) + \dots \\ \dots + \frac{\dot{R}}{R} U'(\xi) \rho_0 \Omega(\xi) + 2 \frac{\rho_0 \Omega(\xi) \dot{R} U(\xi)}{\xi R} = 0 \end{aligned} \quad (\text{B.4})$$

Multiplicando por el factor $R/(\rho_0 \dot{R})$ y agrupando términos se llega a la Ec.(7.15). De manera similar, introduciendo las derivadas parciales

$$\frac{\partial u}{\partial t} = \dot{R}U(\xi) - \xi \frac{\dot{R}^2}{R} U'(\xi) \quad (\text{B.5})$$

$$\frac{\partial p}{\partial r} = \frac{\partial \rho_0}{\partial r} \dot{R}^2 P(\xi) + \rho_0 \frac{\dot{R}^2}{R} P'(\xi) \quad (\text{B.6})$$

en la ecuación de momento (7.11) se llega a que

$$\dot{R}U(\xi) - \xi \frac{\dot{R}^2}{R} U'(\xi) + \frac{\dot{R}^2}{R} U(\xi) U'(\xi) + \frac{1}{\rho_0 \Omega(\xi)} \left[\rho_0' \dot{R}^2 P(\xi) + \rho_0 \frac{\dot{R}^2}{R} P'(\xi) \right] = 0 \quad (\text{B.7})$$

Multiplicando por el factor $R\Omega(\xi)/(\dot{R}^2)$ se obtiene la ecuación (7.16). Finalmente, si se tiene en cuenta que el cuadrado de la velocidad del sonido c_s se expresa en función de las definiciones para p y ρ dadas por las ecuaciones (7.13) como

$$c_s^2 = \gamma \dot{R}^2 P(\xi) / \Omega(\xi) \quad (\text{B.8})$$

y que la derivada parcial de la presión respecto del tiempo está dada por

$$\frac{\partial p}{\partial t} = \dot{\rho}_0 \dot{R}^2 P(\xi) + 2\rho_0 \dot{R} \ddot{R} P(\xi) - \xi \rho_0 \frac{\dot{R}^3}{R} U'(\xi) \quad (\text{B.9})$$

Reemplazando las ecuaciones (B.8), (B.9), (B.6), (B.2) y (B.1) en la ecuación de la energía (7.12) se obtiene la siguiente

$$\begin{aligned} & \dot{\rho}_0 \dot{R}^2 P(\xi) + 2\rho_0 \dot{R} \ddot{R} P(\xi) - \xi \rho_0 \frac{\dot{R}^3}{R} P'(\xi) + \dot{R}U(\xi) \left[\rho_0' \dot{R}^2 P(\xi) + \rho_0 \frac{\dot{R}^2}{R} P'(\xi) \right] - \dots \\ & \dots - \gamma \dot{R}^2 \frac{P(\xi)}{\Omega(\xi)} \left[\dot{\rho}_0 \Omega(\xi) - \xi \rho_0 \frac{\dot{R}}{R} \Omega'(\xi) \right] - \gamma \dot{R} \dot{R}^2 \frac{P(\xi) U(\xi)}{\Omega(\xi)} \left[\rho_0' \Omega(\xi) + \rho_0 \frac{\Omega'(\xi)}{R} \right] = 0 \quad (\text{B.10}) \end{aligned}$$

Multiplicando por el factor $R/(\rho_0 \dot{R}^3)$ y agrupando términos se llega a la ecuación (7.17).

Bibliografía

- [AM06] S. Arya and D.M. Mount. *ANN: Approximate Nearest Neighbors Library*. Department of Computer Science, University of Maryland, 2006.
- [Arg07] Argonne National Laboratories, Mathematics and Computer Science Division, <http://www.mcs.anl.gov/research/projects/mpich2/>. *MPICH2, A freely available, portable implementation of the MPI Standard*, 2007.
- [BBJW94] J. Bell, M.J. Berger, Saltzman J., and M. Welcome. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J.Sci.Stat.*, 1(15):127–138, 1994.
- [BC89] M.J. Berger and P. Collela. Local Adaptive Mesh Refinement for Shock Hydrodynamics. *Journal of Computational Physics*, (82):64–84, 1989.
- [Ber98] M. Berzins. Mesh Quality: A Function of Geometry, Error Estimates or Both? Technical report, School of Computer Studies, University of Leeds, 1998.
- [BH80] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov Galerkin methods for advection dominated flows. In *Third Internat. Conf. of Finite Element Methods in Fluid Flow*, Banff, Canada, 1980.
- [BH82] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.
- [BJW00] M. Berzins, P.K. Jimack, and M. Walkley. Mesh quality and moving meshes for 2D and 3D unstructured mesh flow solvers. Technical report, School of Computer Studies, University of Leeds, 2000.
- [BL02] T.J. Barth and M.G. Larson. A posteriori error estimation for high order godunov finite volume methods on unstructured meshes. In Herbin and Kröner, editors, *Finite Volumes for Complex Applications III*, volume 5, pages 41–63. Hermes Science Pub., London, 2002.
- [Bod97] E.P. Boden. *An Adaptive Gridding Technique for Conservation Laws on Complex Domains*. PhD thesis, Cranfield Institute of Technology, UK, 1997.

- [Boo07] Boost Software, <http://www.boost.org>. *Boost C++ Libraries*, 1998-2007.
- [BPR92] I. Babuska, L. Plank, and R. Rodriguez. Basic problems of a posteriori error estimation. *Comput. Meth. Appl. Mech. Engrg.*, 101:97–112, 1992.
- [BR78] I. Babuska and W.C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J.Numer.Anal.*, 15:736–754, 1978.
- [Bry51] A.E. Bryson. An Experimental Investigation of Transonic Flow Past a Two-Dimensional and Circular-arc Sections using a Mach-Zehnder Interferometer. Report 1094, NACA, 1951.
- [BS97] M. Berzins and W. Speares. A 3D Unstructured Mesh Adaption Algorithm for Time Dependent Shock Dominated Problems. *Int. Journal for Num. Methods in Fluids*, 25:81–104, 1997.
- [CH07] C. Carstensen and J. Hu. A unifying theory of a posteriori error control for non-conforming finite element methods. *Numer. Math.*, pages 473–502, 2007.
- [Cod93] R. Codina. A discontinuity-capturing crosswind-dissipation for the finite element solution of the advection-diffusion equation. *Comput. Mehods Appl. Mech. Engrg.*, pages 325–342, 1993.
- [CS04] M. Cooper and J.E. Sheperd. The effect of transient nozzle flow on detonation tube impulse. In *40th Joint Propulsion Conference and Exhibit*, Fort Lauderdale, FL, July 2004. AIAA/ASME/SAE/ASEE, AIAA.
- [DH03] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley Sons Ltd, 2003.
- [DLGC98] J. Dompierre, P. Labbe, F. Guibault, and R. Camerero. Proposal of Benchmarks for 3D Unstructured Tetrahedral Mesh Optimization. *Proceedings of the 7-th International Meshing RoundTable '98*, pages 459–478, 1998.
- [DVLG05] J. Dompierre, M.G. Vallet, P. Labbe, and F. Guibault. An analysis of simplex shape measures for anisotropic meshes. *Computer Methods in Applied Mechanics and Engineering*, 194:4895–4914, 2005.
- [Eat07] John W. Eaton. *Octave, versión 2.9.9*. <http://www.gnu.org/software/octave>, 1996-2007.
- [Eck00] Bruce Eckel. *Thinking in C++*, volume 1. Prentice Hall Inc., Upper Saddle River, New Jersey 07458, second edition, 2000. <http://www.mindview.net/ThinkingInCPP1e.html>.

- [Eck04] Bruce Eckel. *Thinking in C++*, volume 2. Prentice Hall Inc., Upper Saddle River, New Jersey 07458, second edition, 2004. <http://www.mindview.net/ThinkingInCPP2e.html>.
- [FLS⁺97] J.E. Flaherty, R.M. Loy, M.S. Shepard, B.K. Szymanski, J.D. Teresco, and L.H. Ziantz. Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of Three-Dimensional Conservation Laws. *Journal of parallel and distributed computing*, 47:139–152, 1997.
- [FLÖ⁺98] J.E. Flaherty, R.M. Loy, C. Özturan, M.S. Shepard, B.K. Szymanski, J.D. Teresco, and L.H. Ziantz. Parallel Structures and Dynamic Load Balancing for Adaptive Finite Element Computation. *Applied Numerical Mathematics*, 26:241–263, 1998.
- [Fre74] D.A. Freiwald. Approximate Spherical Blast Theory and Laser-Initiated Pellet Microexplosions. Report LA-5641-MS, Los Alamos, June 1974.
- [Hir90] Ch. Hirsh. *Numerical Computation of internal and external Flows*, volume 2 of *Computational Methods for Inviscid and Viscous Flows*. John Wiley and Sons, Chichester, 1990.
- [HM86] T.J.R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: Iv. A discontinuity-capturing operator for multidimensional advective-diffusive systems. *Comput. Mehods Appl. Mech. Engrg.*, 58(3):329–336, 1986.
- [Hut00] G.J. Hutchens. Approximate near-field blast theory: A generalized approach. *Journal of Applied Physics*, 88(6):3654–3658, 2000.
- [Joh95] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, 1995.
- [KAKY06] K. Kato, T. Aoki, S. Kubota, and M. Yoshida. A numerical scheme for strong blast wave driven by explosion. *IJNMF*, 51:1335–1353, 2006.
- [Kar05] Karypis Laboratories, <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>. *ParMetis, Parallel Graph Partitioning and Fill-reducing Matrix Ordering*, 1997-2005.
- [Knu00] P. Knupp. Algebraic mesh quality metrics. *SIAM Journal of Scientific Computing*, April 2000.
- [KV93] Y. Kallinderis and P. Vijayan. Adaptive refinement-coarsening scheme for three-dimensional unstructured meshes. *AIAA Journal*, 31(8):1440–1447, 1993.
- [Lan98] C. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.

- [LB92] R. Löhner and J.D. Baum. Adaptive h-refinement on 3D unstructured grids for transient problems. *Int. Journal for Num. Methods in Fluids*, 14:1407–1419, 1992.
- [LJ94] A. Liu and B. Joe. On the shape of tetrahedra from bisection. *Mathematics of Computation*, 63(207):141–154, July 1994.
- [LJ96] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. *Mathematics of Computation*, 65(215):1183–1200, 1996.
- [LNS06] E. López, N.M. Nigro, and M. Storti. Untangling y Smoothing simultáneos de mallas., 2006.
- [Mav95] D.J. Mavriplis. Unstructured mesh generation and adaptivity. Report 95-26, ICASE - NASA Langley Research Centre, April 1995.
- [MK06] M. Möller and D. Kuzmin. Adaptive mesh refinement for high-resolution finite element schemes. *Int. J. Numer. Meth. Fluids*, 52:1197–1203, 2006.
- [NS04] A. Nagórka and N. Sczygiol. Implementation Aspects of a Recovery-Based Error Estimator in Finite Element Analysis. *Lecture Notes in Computer Science*, 3019/2004:722–729, 2004.
- [OE04] C. Omgba-Essama. *Numerical Modelling of Transient Gas-Liquid Flows (Application to Stratified and Slug Flow Regimes)*. PhD thesis, Cranfield Institute of Technology, UK, 2004.
- [PE01] S. Prakash and C.R. Ethier. Enhanced Error Estimator for Adaptive Finite Element Analysis of 3D Incompressible Flows. *Comput. Methods in Applied Mechanics and Engineering*, 190:5413–5426, 2001.
- [PPS05] A. Plaza, M.A. Padrón, and J.P. Suárez. Non-degeneracy Study of the 8-tetrahedra Longest-edge Partition. *Applied Numerical Mathematics*, 55:458–472, 2005.
- [Qui91] J.J. Quirk. *An Adaptive Grid Algorithm for Computational Shock Hydrodynamics*. PhD thesis, Cranfield Institute of Technology, UK, 1991.
- [Rem03] J. Remacle. Transient Adaptive Discontinuous Galerkin Method with Anisotropic Meshes. In *Mesh Quality and Dynamic Meshing*, pages 72–75, Livermore, CA, January 2003. Sandia National Laboratories.
- [RKS] J. Remacle, B.K. Karamete, and M.S. Shepard. Algorithm Oriented Mesh Database. Technical report, Rensselaer Polytechnic Institute, Troy, NY, USA.
- [Sed59] L.I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, New York, 1959.

- [SGG04] J. Schöberl, H. Gerstmayr, and R. Gaisbauer. *NETGEN - automatic mesh generator*. JKU Linz, Austria, 2004.
- [Sha53] Ascher H. Shapiro. *The Dynamics and Thermodynamics of Compressible Fluid Flow*, volume 1. John Wiley Verlag, 1953.
- [Sil06] Silicon Graphics, Inc., <http://www.sgi.com/tech/stl>. *Standard Template Library*, 1993-2006.
- [SMS06] R. Stallman, R. McGrath, and P.D. Smith. *GNU Make, A Program for Directing Recompilation*. GNU, 2006.
- [SNP⁺08] M. Storti, N. Nigro, R. Paz, L. Dalcin, L. Battaglia, E. Lopez, and G.A. Ríos Rodríguez. *PETSc-FEM, A General Purpose, Parallel, Multi-Physics FEM Program*. CIMEC-CONICET-UNL, 1999-2008.
- [SNPD08] M.A. Storti, N.M. Nigro, R.R. Paz, and L.D. Dalcín. Dynamic boundary conditions in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 197:1219–1232, 2008.
- [Sol54] George E. Solomon. Transonic flow past cone cylinders. Report 1242, NACA, 1954.
- [Sta96] M.L. Staten. Selective refinement of two and three-dimensional finite element meshes. Master’s thesis, Department of Civil Engineering, Brigham Young University, 1996.
- [Sül98] E. Sülli. A posteriori error analysis and adaptivity for finite element approximations of hyperbolic problems. In Ohlberger Kröner and Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, volume 5 of *Lecture Notes in Computational Science and Engineering*, pages 122–194. Springer-Verlag, Heidelberg, 1998.
- [Tay46] G.I. Taylor. The air wave surrounding an expanding sphere. In *Mathematical and Physical Sciences*, volume 186 of *Series A*, pages 273–292. Royal Society of London, Royal Society, September 1946.
- [Tay50a] G.I. Taylor. The Formation of a Blast Wave by a Very Intense Explosion. I. Theoretical Discussion. In *Mathematical and Physical Sciences*, volume 201 of *Series A*, pages 159–174. Royal Society of London, Royal Society, March 1950.
- [Tay50b] G.I. Taylor. The Formation of a Blast Wave by a Very Intense Explosion. II. The Atomic Explosion of 1945. In *Mathematical and Physical Sciences*, volume 201 of *Series A*, pages 175–186. Royal Society of London, Royal Society, March 1950.

- [Tho02] Kip Thorne. Applications of classical physics, 2002. www.pma.caltech.edu/Courses/ph136/yr2002.
- [Usn98] Asif S. Usmani. An h-adaptive SUPG-FEM solution of the pure advection equation. *Applied Numerical Mathematics*, 26:193–202, 1998.
- [Ver98] R. Verfürth. A posteriori error estimators for convection-diffusion equations. *Numer. Mathematik*, 80:641–663, 1998.
- [VLB96] B. Vanderstraeten, M. Lefebvre, and J. Berghmans. A simple blast wave model for bursting spheres based on numerical simulation. *Journal of Hazardous Materials*, 46:145–157, 1996.
- [Vol00] J. Volker. A numerical study of a posteriori error estimators for convection-diffusion equations. *Comput. Methods in Applied Mechanics and Engineering*, 190:757–781, 2000.
- [VW52] W.G. Vincenti and C.B. Wagoner. Transonic Flow past a Wedge Profile with Detached Bow Wave. Report 1095, NACA, 1952.
- [Wal04] J. Waltz. Parallel Adaptive Refinement for Unsteady Flow Calculations on 3D Unstructured Grids. *Int. J. Numer. Meth. Fluids*, 46:37–57, 2004.
- [YK05] M.P. Young and O.J. Kwon. A Parallel Unstructured Dynamic Mesh Adaption Algorithm for 3-D Unsteady Flows. *International Journal for Numerical Methods in Fluids*, 48:671–690, 2005.
- [ZZ87] O.C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Meth. Engineering*, 24:337–357, 1987.
- [ZZ92a] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. part 1: the recovery techniques. *Int. J. Numer. Meth. Engineering*, 33:1331–1364, 1992.
- [ZZ92b] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. part 2: error estimates and adaptivity. *Int. J. Numer. Meth. Engineering*, 33:1365–1382, 1992.