

# Desarrollo y evaluación de algoritmos para combinar mallas de elementos finitos con grillas del método de red de vórtices inestacionario con topología arbitraria

Martín E. Pérez Segura<sup>1</sup>, Sergio Preidikman<sup>1,2</sup> y Mauro S. Maza<sup>1,2</sup>

<sup>1</sup>Dpto. de Estructuras, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba, Córdoba, Argentina.

<sup>2</sup>CONICET - Consejo Nacional de Investigaciones Científicas y Técnicas, Av. Rivadavia 1917, Bs. As., Argentina.

Fecha de recepción del manuscrito: 21/10/2014

Fecha de aceptación del manuscrito: 18/02/2015

Fecha de publicación: 16/03/2015

**Resumen**—La motivación de este trabajo proviene de una dificultad real que afecta, en particular, a los problemas de aeroelasticidad computacional enfocados en un campo de aplicación relativamente reciente: el análisis de alas batientes. Esta área, inspirada en la biología, trata el complejo movimiento que se produce durante el vuelo de diversas criaturas (insectos principalmente) con el fin de utilizar sus resultados en el diseño de dispositivos de utilidad práctica. Se analiza aquí la etapa de preproceso requerida en la resolución de estos problemas de interacción fluido-estructura. Esta etapa comprende la vinculación entre las distintas discretizaciones que surgen de abordar los problemas con un enfoque multifísico y posibilita la transferencia de información entre mallas estructurales y grillas aerodinámicas. Dicha transferencia es de capital importancia en el campo de la aeroelasticidad computacional, donde los métodos de interacción pueden fácilmente transformarse en el factor que controla la precisión de la simulación aeroelástica. En este sentido, se presentan algoritmos eficientes y robustos para la vinculación topológica de las mallas/grillas, dividiendo la tarea en tres etapas, considerando diferentes opciones para dos de ellas. A su vez estos algoritmos son implementados y evaluados utilizando mallas/grillas correspondientes a casos de interés práctico. Los resultados obtenidos muestran resultados alentadores en cuanto al incremento en la eficiencia de los programas de transferencia de datos, permitiendo formular conclusiones y recomendaciones de aplicación general.

**Palabras clave**—Multifísica. Interacción Fluido-Estructura. Aeroelasticidad Computacional. Mallas Estructurales. Grillas Aerodinámicas. Algoritmos.

**Abstract**— The motivation for this work arises from the real issue that affects particularly problems of computational aeroelasticity related to flapping wings applications, which are considered a recent inquired field. This subject, inspired in biology, studies the complex phenomena that occurs during some creatures' flight (insects mostly) in order to apply its results to practical cases. Here, a preprocess stage required amongst the solution of the fluid-structure interaction problems is analyzed. This stage involves the linking between the different discretizations that emerge from treating the problems with a multiphysics approach, and enables the data transference between structural meshes and aerodynamic grids. The previously stated data transference gains importance since the precision of the simulations may easily depend on the interaction methods applied. Therefore, efficient algorithms which contribute to the topological entailment of grids and meshes are presented, dividing the process into three phases and considering different options for the last two. Thereby, the algorithms are implemented and tested for real applications. Results show significant improvements in data transference software efficiency, hence conditions are set to formulate conclusions and recommendations with general suitability.

**Keywords**— Multiphysics. Fluid-Structure Interaction. Computational Aeroelasticity. Structural Meshes. Aerodynamic Grids. Algorithms.

## INTRODUCCIÓN

La utilización de nuevos materiales en el ámbito ingenieril, en particular materiales compuestos como son las resinas reforzadas con fibras, y el impulso a la búsqueda de soluciones para problemas de concepción innovadora que requieren el aprovechamiento máximo de las capacidades de estos materiales obligan a sofisticar las

técnicas de análisis de manera de describir en mayor detalle los fenómenos físicos relacionados.

Esto conlleva la aparición de modelos matemáticos de gran complejidad que raras veces poseen solución analítica para casos de interés ingenieril. Es aquí donde aparecen los métodos numéricos como técnicas que permiten hallar soluciones aproximadas de valor práctico. Para su aplicación, el dominio a analizar es discretizado en pequeños subdominios, apareciendo así grillas o mallas que los definen.

En el caso de fenómenos muy complejos, que abarcan varias áreas de la ciencia que se han estudiado tradicionalmente de manera independiente, surge la posibilidad de abordar el problema con un enfoque

Dirección de contacto:

Martín Eduardo Pérez Segura, Ituzaingó 946, Córdoba-Capital, X5000 IFT, +00543516619395, martin.eduardo.perez.segura@gmail.com.

particionado. Este permite utilizar modelos diferentes, cada uno altamente especializado y probado para resolver una de las partes que componen el problema. Requiere, a su vez, del diseño y programación de un método de interacción que, si bien no resulta una tarea trivial, es más asequible que el desarrollo y codificación de un método que permita atacar el problema completo, como un solo sistema.

La utilización de un enfoque particionado en conjunto con métodos numéricos implica la aparición ya no de una, sino de dos o más mallas o grillas en las que se encuentran definidas variables del problema. Algunas de estas variables son compartidas por las discretizaciones o están en estrecha relación entre ellas. Esto está vinculado directamente a la interdependencia de los fenómenos individuales que componen el problema, que obliga al tratamiento conjunto. Dicha interdependencia se incorpora en el modelo a través de un mecanismo eficiente de pre-procesamiento de datos que deriva en vinculación de las mallas o grillas, lo cual no resulta una tarea trivial ya que estas discretizaciones pueden ser muy diferentes, no solo en su topología, sino también en su naturaleza. En el caso de la aeroelasticidad computacional, la partición más habitual es aquella que ataca por separado el problema aerodinámico del estructural/dinámico.

En este trabajo se estudia el caso particular en el que el modelo aerodinámico es el *Método de Red de Vórtices Inestacionario* (ULVM) y la estructura se modela utilizando el *Método de Elementos Finitos* (FEM). Se presentan diferentes estrategias que permiten resolver parte del problema topológico mencionado anteriormente: determinar en cuál de los elementos finitos de la *malla estructural* (ME) se encuentra cada nodo (o punto de control) perteneciente a la *grilla aerodinámica* (GA). En general la GA y la ME tienen topologías muy diferentes. La topología de las GA depende de consideraciones puramente aerodinámicas y es independiente de la discretización de la estructura, generalmente se compone de elementos cuadriláteros, con un nodo en cada vértice, y un nodo adicional, denominado punto de control, en el centro geométrico de cada elemento. Mientras que, una ME, en general, obedece únicamente a consideraciones de la dinámica estructural.

La transferencia de datos entre grillas y mallas es de capital importancia particularmente en el campo de la aeroelasticidad computacional, donde los métodos de interacción entre las mismas pueden fácilmente transformarse en el factor que controla la precisión de la simulación aeroelástica. Luego, es precisamente aquí donde encuentra sitio el presente trabajo que busca contribuir con medios computacionales para hacer más eficaz y eficiente el pre-procesamiento de datos que requiere dicha interacción, orientada al análisis de alas batientes.

### CONCEPTOS PRELIMINARES

Se exponen a continuación algunos conceptos básicos relacionados con el *Método de los Elementos Finitos* sobre los que se basan algunos desarrollos posteriores.

#### Dominio de origen

Un método para la definición de elementos que resulta simple, sistemático y de fácil implementación computacional es el uso del *dominio de origen* (también

conocido como método del elemento maestro). Este mecanismo consiste en utilizar un dominio auxiliar que será de naturaleza similar a aquel en el cual se pretende el elemento (dominio del elemento). Luego, se utiliza una transformación afín o mapeo que lo vincula con el dominio del elemento. De esta manera, a partir de un elemento simple, de geometría sencilla, puede definirse otro acorde a las necesidades del problema, tal y como se ilustra en la Fig. 1.

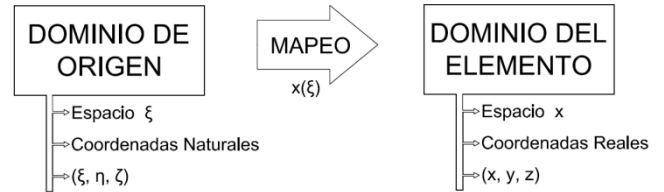


Fig. 1: Método del dominio de origen.

Existen dos casos clásicos de dominios de origen que engloban casi la totalidad de los elementos comúnmente utilizados en el FEM. Éstos son el *Cuadrilátero Bilineal* y el *Hexaedro Trilineal* que son utilizados para la formación de elementos en dos y tres dimensiones respectivamente.

#### Cuadrilátero bilineal

Se define el dominio de origen como un cuadrilátero regular de bordes rectos de dos unidades de longitud determinados por cuatro nodos identificados con las *coordenadas naturales* “ $\zeta$ ” y “ $\eta$ ”, en el plano  $\mathbb{R}^2$  (denominado “*espacio  $\zeta$* ”), según:

$$\xi_a = \begin{Bmatrix} \xi_a \\ \eta_a \end{Bmatrix}, \quad a = 1,2,3,4 \quad (1)$$

donde “ $a$ ” representa la cantidad de nodos del elemento y, siendo:

$$\xi_1 = \begin{Bmatrix} -1 \\ -1 \end{Bmatrix}, \xi_2 = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}, \xi_3 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}, \xi_4 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \quad (2)$$

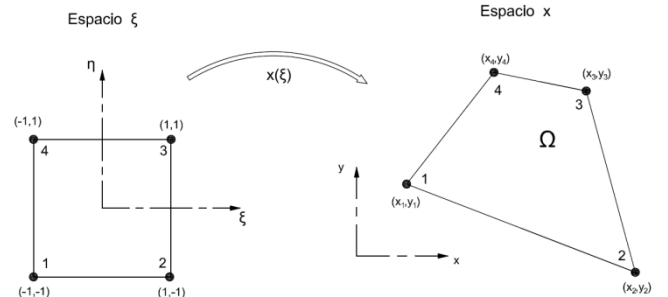


Fig. 2: Cuadrilátero bilineal.

Según se indica en la Fig. 2, se pretende un cambio de coordenadas a través de un mapeo del tipo

$$x(\xi) = \sum_{a=1}^4 N_a(\xi) x_a, \quad (3)$$

para transformar el cuadrilátero del “*espacio  $\zeta$* ” en un elemento real en el plano  $\mathbb{R}^2$  (denominado “*espacio  $x$* ”) pero en coordenadas “ $x$ ” e “ $y$ ”, definido por cuatro nodos dados por:

$$x_a = \begin{Bmatrix} x_a \\ y_a \end{Bmatrix}, \quad a = 1,2,3,4. \quad (4)$$

Las funciones  $N_a$  se determinan a partir de asumir que las coordenadas del “*espacio  $x$* ” pueden expresarse como expansiones bilineales de las coordenadas del “*espacio  $\zeta$* ” y, además, que los nodos a través del mapeo deben coincidir. Esto es:

$$x(\xi, \eta) = \alpha_0 + \alpha_1\xi + \alpha_2\eta + \alpha_3\xi\eta \quad (5)$$

$$x(\xi, \eta) = \alpha_0 + \alpha_1\xi + \alpha_2\eta + \alpha_3\xi\eta \quad (6)$$

$$x(\xi_a) = x_a, \quad a = 1,2,3,4. \quad (7)$$

**Hexaedrotrilineal**

El hexaedro trilineal es una generalización del cuadrilátero bilineal y se desarrolla con un procedimiento análogo. No obstante, en este caso se trabaja en tres dimensiones y el dominio de origen se define como un hexaedro regular (cubo) con aristas que poseen dos unidades de longitud, y en cuyo centro geométrico se encuentra el centro de coordenadas del “espacio  $\zeta$ ”. En este caso, el mapeo se expresa como sigue

$$x(\xi) = \sum_{a=1}^8 N_a(\xi) x_a \quad (8)$$

donde la relación de coordenadas se establece como una expansión trilineal del tipo:

$$x(\xi, \eta, \zeta) = \alpha_0 + \alpha_1\xi + \alpha_2\eta + \alpha_3\zeta + \alpha_4\xi\eta + \alpha_5\eta\zeta + \alpha_6\xi\zeta + \alpha_7\xi\eta\zeta. \quad (9)$$

**El método de degeneración**

Este método permite obtener un elemento alternativo a partir de un dominio de origen que no es conforme con él. De este modo, si se trabaja en dos dimensiones, se utilizará siempre como dominio de origen el cuadrilátero bilineal, y análogamente con el hexaedro trilineal en el espacio.

El método de degeneración consiste en la coalescencia de los nodos del elemento mediante la modificación de las funciones de forma  $N_a$ , que conforman el mapeo. Esto permite “suprimir” aristas o bordes para alterar la geometría y lograr un elemento de naturaleza diferente a aquel que le dio origen. La Fig. 3 presenta un ejemplo donde se construye un prisma triangular.

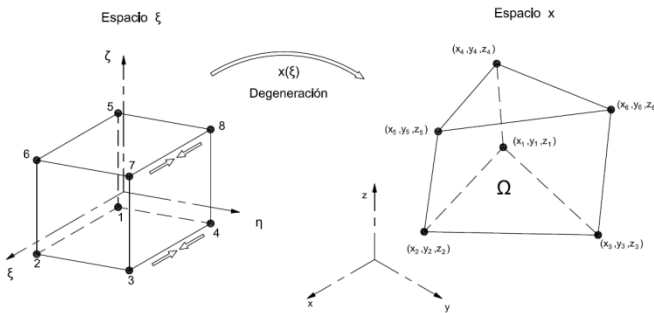


Fig. 3: Método de degeneración para prisma triangular.

**Elementos de orden superior**

La clave para la construcción de estos elementos yace en las expresiones que vinculan las coordenadas del “espacio  $\zeta$ ” con las del “espacio  $x$ ”: ya no se impondrá una expansión lineal (bilineal o trilineal) sino una del orden deseado para el elemento. La aplicación del dominio de origen es perfectamente válida para generar elementos de orden superior siempre que se tenga en cuenta que el elemento auxiliar deberá ser del orden deseado. En estos casos, los elementos cuentan, en general, con mayor cantidad de nodos, lo que permite imponer las condiciones necesarias para determinar los coeficientes a utilizar en los mapeos.

**EL PROGRAMA DE VINCULACIÓN**

Se pretende un código que tome como input las mallas del FEM y las grillas del UVLM, y genere una *matriz de localización* que indique en qué elemento de la ME se encuentra cada punto de control y cada nodo de la GA, tal como se diagrama en la Fig. 4. El código deberá cumplir con tres tipos de funciones: las destinadas a la recolección de datos provenientes de las mallas y grillas, las que seleccionan y organizan en grupos los elementos de la ME para su posterior evaluación, y las que consisten en la localización propiamente dicha. Luego, el código se divide en tres módulos identificables con las tareas asignadas a los conjuntos de funciones: *El Módulo de Ingreso*, *El Módulo de Selección* y *El Módulo de Localización*.



Fig. 4: Diagrama de flujobásico del programa de vinculación.

*El Módulo de Ingreso* tiene por objetivo, procesar los datos de entrada provenientes del mallador, y poner a disposición de los módulos siguientes variables internas consistentes en parámetros, como ser, las particularidades de los elementos, sus funciones de forma, matrices jacobianas, entre otros.

*El Módulo de Selección* tiene por finalidad realizar una preselección de los elementos donde es probable encontrar cada punto que requiere ser situado, buscando reducir virtualmente al mínimo el número de evaluaciones a realizar. La metodología de operación de este módulo pretende ser relativamente simple y consiste principalmente en nociones topológicas y geométricas. Aquí cobra importancia el ordenamiento que se da a los elementos preseleccionados, teniendo en cuenta que las evaluaciones en el *Módulo de Localización* se realizan en una secuencia establecida.

*El Módulo de Localización* tiene por propósito evaluar los elementos que el *Módulo de Selección* indique y determinar en cuál de ellos se encuentra el punto en análisis. El resultado de cada ciclo de evaluaciones del tercer módulo debe ser biunívoco, ya que, sin dar lugar a la ambigüedad, cada punto de la GA se corresponderá con uno y sólo un elemento de la ME. Los métodos utilizados en este módulo son más complejos que aquellos usados en el *Módulo de Selección*. Aquí se recurre a las propiedades mismas de los elementos, sus sistemas de construcción y otros recursos del análisis numérico.

**El módulo de ingreso**

*El Programa de Vinculación* está pensado como una función autónoma y debe ser capaz de generar por sí misma los datos y variables necesarios para su funcionamiento. El encargado de esta tarea es el *Módulo de Ingreso* que, independientemente del procedimiento que siga, deberá: identificar los tipos de elementos involucrados, generar matrices de procesamiento de datos de la ME y la GA, crear variables con la cantidad de elementos y nodos, y cargar vectores de funciones de forma. Las matrices de

procesamiento de datos hacen referencia a dos arreglos en particular, el *Arreglo de Nodos y Coordenadas* que enlista las coordenadas de cada uno de los nodos de una malla, y el *Arreglo IEN*, también conocido como matriz de conectividades, que relaciona los índices locales y globales de los nodos de cada elemento.

El *Módulo de Ingreso* toma como input los archivos de datos de la ME y la GA provenientes del mallador y su funcionamiento se basa en la utilización de funciones de lectura. Por otro lado, los parámetros que son recurrentes (como es el caso de las funciones de forma, por ejemplo) se almacenan en funciones tipo menú externas que pueden ser ejecutadas a conveniencia.

**El módulo de selección**

Este módulo no es estrictamente necesario desde el punto de vista operativo, pero su implementación genera incrementos considerables en la eficiencia. Además, es quizás el más versátil, ya que se prioriza los aspectos geométricos y topológicos de la malla del FEM para la preselección de elementos. Se presentarán cinco alternativas que van desde las más sencillas, y quizás por eso menos eficientes pero más robustas, hasta algunos de complejidad relativamente elevada que han mostrado resultados alentadores.

En la mayoría de los casos, la variable de salida del *Módulo de Selección* será una matriz, la cual será transferida al módulo siguiente. La *Matriz de Selección* asigna una fila a cada uno de los puntos a localizar y enlista, de forma ordenada, el conjunto de elementos que fueron preseleccionados para dicho punto, como se ve en la Fig. 5.

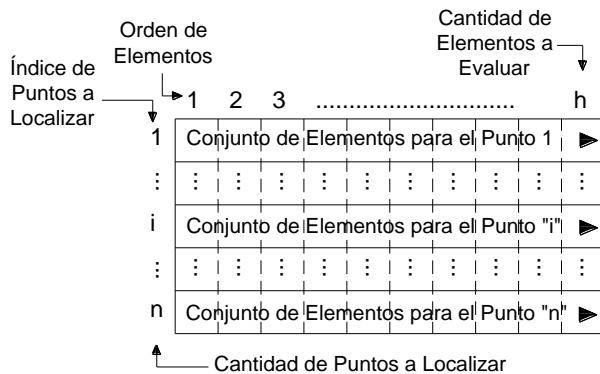


Fig. 5: Matriz de selección genérica.

**Selección: El algoritmo de lista y sus variantes**

El *Algoritmo de Lista* es el más trivial de todos y equivale, prácticamente, a prescindir del *Módulo de Selección*. Este algoritmo utiliza una lista de los elementos según sus índices globales para evaluarlos uno a uno, sin importar qué punto se esté localizando.

El método de *Lista Aleatoria Simple* es una sutil modificación al caso anterior. Surge de incorporar el concepto del orden de los elementos seleccionados como una forma de reducir el tiempo de ejecución. Iniciar la lista en el elemento número 1 y continuar de forma ordenada, para todos los puntos no resulta del todo práctico. Por lo tanto, se designa aleatoriamente una secuencia de elementos a evaluar, de manera que todos los elementos sean evaluados virtualmente la misma cantidad de veces,

evitando priorizar los que posean índices bajos. El término "Simple" hace referencia a que la secuencia aleatoria es generada una única vez y se utiliza la misma para todos los puntos.

Otra variante es el método de *Lista Aleatoria Múltiple*, que presenta la particularidad, a diferencia del caso anterior, de generar un ordenamiento aleatorio de elementos distinto para cada punto que se desee localizar.

**Selección: El algoritmo de centros de masa**

Este algoritmo concibe a los elementos como una disposición de masas puntuales de igual valor ubicadas sobre cada uno de sus nodos, permitiendo calcular, para cada elemento, un punto designado como centro de masa, cuyo vector de coordenadas se obtiene de la ec.(10).

$$X_{CM} = \frac{\sum_{i=1}^a x_i}{a} \tag{10}$$

donde  $x_i$  es el vector de coordenadas de cada uno de los nodos del elemento.

El procedimiento consiste en calcular los centros de masa de todos los elementos de la ME para luego calcular la distancia entre cada uno de ellos y el punto que se desea localizar. Hecho esto, se seleccionan, en orden creciente respecto de las distancias, los nueve elementos más próximos al punto en análisis para conformar el conjunto que se transferirá al módulo siguiente. La Fig. 6 ilustra un ejemplo plano de lo antes planteado.

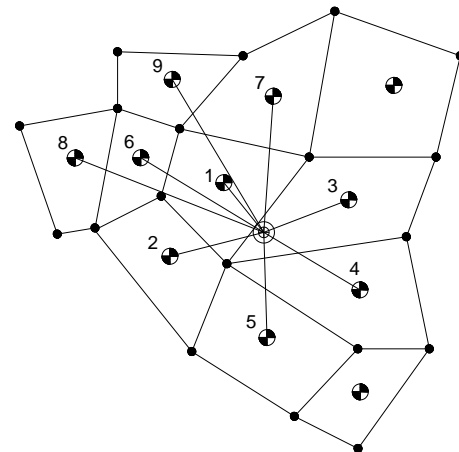


Fig. 6: Centros de masa más próximos a un punto a localizar.

Optar por considerar sólo nueve elementos puede resultar arbitrario; de hecho, el número de elementos elegidos puede incrementarse o disminuirse según el criterio que se aplique. No obstante, considerar nueve resulta un intermedio entre un juicio conservador y la intención de reducir el número de evaluaciones.

**Selección: El algoritmo ESUP**

El *Algoritmo ESUP* recoge su nombre de la frase en inglés "Elements Surrounding Points" (elementos que rodean puntos) y toma sus bases en el trabajo de R. Löhner [2]. Este método consiste en generar un arreglo que permita, dado un nodo de la ME, identificar los elementos que lo rodean, es decir, aquellos que lo contienen, como los identificados en el caso ejemplo de la Fig. 7.

El criterio de selección del *Algoritmo ESUP* consiste en encontrar, para cada punto a localizar, el nodo de la ME más próximo y seleccionar los elementos que rodean a

dicho nodo para la evaluación posterior. La cantidad de estos elementos varía sensiblemente dentro de una malla de FEM, resultando en la imposibilidad de pre-dimensionar una matriz que almacene estos datos, generando una pérdida importante de eficiencia. La solución a este inconveniente también es propuesta por Löhner en su trabajo y consiste en el uso de *Listas Vinculadas* ("Linked Lists").

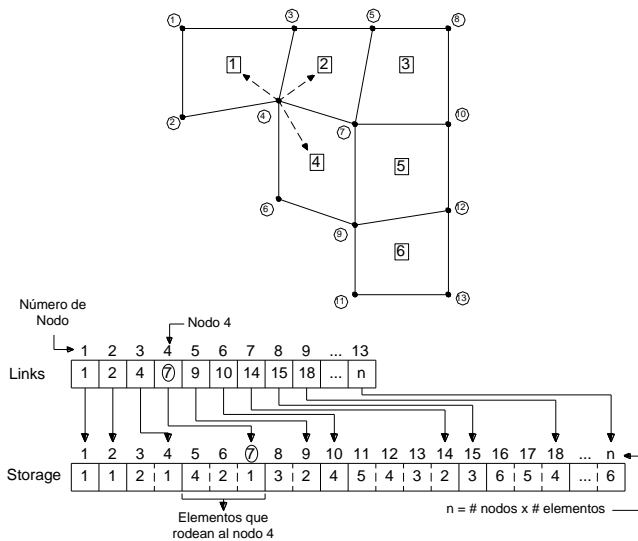


Fig. 7: Listas vinculadas para el algoritmo ESUP, ejemplo plano.

Las *listas vinculadas*, presentadas en el ejemplo de la Fig. 7, son una herramienta que permite reemplazar el uso de matrices cuando no es posible conocer a priori la cantidad de datos a almacenar. Este sistema cuenta con dos vectores filas que se relacionan entre sí. En el primero, denominado "Storage", se almacena la información propiamente dicha. En el segundo, denominado "Links", se encuentran marcadores que indican dónde comienzan y dónde terminan los datos de una entidad particular. Es decir, en el vector "Links", el índice de posición se condice con los nodos de la malla y los valores que contiene indican la última posición del vector "Storage", donde se almacena un elemento que rodea al nodo indicado por el índice de posición.

**Selección: El algoritmo ESUP ordenado por centros de masa**

Este último algoritmo es una combinación de los dos anteriores, *ESUP* y *Centros de Masa*, y constituye la variante más compleja de las que aquí se presentan. El sistema consiste en evaluar de forma ordenada según el criterio de *Centros de Masa* a los elementos que son seleccionados con el criterio de *ESUP*.

**El módulo de localización**

El objetivo de este módulo es identificar biunívocamente el elemento de la ME que contenga a cada punto de la GA, para poder confeccionar así la *Matriz de Localización* que será el resultado final del *Programa de Vinculación*, tal y como en el ejemplo de la Fig. 8. Para lograrlo, son dos los métodos que se proponen, el primero toma la forma de un método numérico clásico, el método de Newton-Raphson. Mientras que el segundo recurre al concepto de las

*Coordenadas de Área*, también conocidas como *Coordenadas Triangulares*.

Estos sistemas deberán ejecutarse múltiples veces por cada punto que se desee localizar y deberán acoplarse apropiadamente con los módulos anteriores.

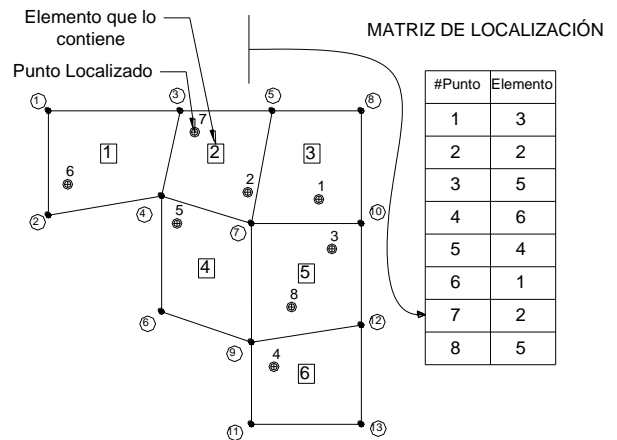


Fig. 8: Matriz de localización.

**Localización: El algoritmo de Newton-Raphson**

Se plantea que si un punto en el "espacio ζ" pertenece al Dominio de Origen, su transformación, a través del mapeo, será interior al elemento en el "espacio x", y viceversa. Así, la condición de pertenencia puede expresarse en términos de las coordenadas del punto ξ<sub>p<sub>i</sub></sub> (en dos dimensiones) como

$$\xi_{p_i} \in \Omega^d \Leftrightarrow \begin{cases} (-1 < \xi_{p_i} < 1) \\ (-1 < \eta_{p_i} < 1) \end{cases} \quad (10)$$

Determinar la pertenencia o no con este criterio implica resolver un sistema de ecuaciones para realizar el mapeo ξ(x), inverso al mencionado en el esquema de la Fig. 1. Es aquí donde se utiliza el *Método de Newton-Raphson*. La elección de este método para la solución del sistema de ecuaciones se justifica en la diversidad de tipos de funciones de forma que pueden presentarse.

Reordenando la ec.(8) puede escribirse, para un punto arbitrario que pertenece a un elemento con n nodos,

$$X_p - \sum_{a=1}^n N_a(\xi) x_a = 0 \quad (11)$$

resultando en un sistema de ecuaciones en el cual X<sub>p</sub> es el punto que se desea localizar, x<sub>a</sub> los nodos del elemento que se evalúa y N<sub>a</sub>(ξ) las funciones de forma. Quedando sólo como incógnitas las coordenadas del punto ξ que, luego de ser encontradas deberán contrastarse con las condiciones de pertenencia.

Definiendo una función vectorial

$$F_e(\xi, X_p) = X_p - \sum_{a=1}^n N_a(\xi) x_a = 0 \quad (12)$$

y expandiendo en serie la función F<sub>e</sub> según la variable ξ, puede proponerse un esquema iterativo de solución

$$F_e(\xi_{k+1}, X_p) = F_e(\xi_k, X_p) + \left. \frac{\partial F_e(\xi, X_p)}{\partial \xi} \right|_{\xi_k} \Delta \xi_k + H_2 \quad (13)$$

donde, H<sub>2</sub> representa los términos de orden superior (desde segundo orden en adelante) que completarían la igualdad pero que son despreciados y, siendo "np" la cantidad de iteraciones fijadas a priori, se escribe

$$\xi_{k+1} = \xi_k + \Delta\xi_k \text{ con } k = 1, 2, 3 \dots np \quad (14)$$

Además, la derivada parcial de la función  $F_e$  se identifica con la matriz jacobiana de la transformación de coordenadas, es decir

$$\frac{\partial F_e(\xi, X_p)}{\partial \xi} = [J_e(\xi, X_p)] = [Xa] * [N'] \quad (15)$$

siendo  $[Xa]$  la matriz de coordenadas de los nodos del elemento y  $[N']$  la matriz de funciones de forma derivadas, esto es

$$[Xa] = \begin{bmatrix} x_1 & \dots & x_a \\ y_1 & \dots & y_a \\ z_1 & \dots & z_a \end{bmatrix} \quad (16)$$

$$[N'] = \nabla N_a = \begin{bmatrix} \nabla N_1 \\ \vdots \\ \nabla N_a \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_1}{\partial \eta} & \frac{\partial N_1}{\partial \zeta} \\ \vdots & \vdots & \vdots \\ \frac{\partial N_a}{\partial \xi} & \frac{\partial N_a}{\partial \eta} & \frac{\partial N_a}{\partial \zeta} \end{bmatrix}. \quad (17)$$

En la convergencia, debido a la definición de  $F_e$ , se cumplirá que

$$F_e(\xi_{k+1}, X_p) = F_e(\xi_k, X_p) + \left. \frac{\partial F_e(\xi, X_p)}{\partial \xi} \right|_{\xi_k} \Delta\xi_k = 0. \quad (18)$$

Reemplazando y ordenando resulta

$$[J_e(\xi_k, X_p)] * \Delta\xi_k = -F_e(\xi_k, X_p), \quad (19)$$

expresión que permite calcular  $\Delta\xi_k$  fijando un valor inicial de  $\xi_k$  para luego continuar con el proceso iterativo.

### Reducción de elementos

La *Reducción de Elementos* es un complemento pensado para implementar en cualquier sistema de localización que utilice las funciones de forma de los elementos en su proceso, como por ejemplo, el *Algoritmo de Newton-Raphson*. Conceptualmente consiste en recurrir a los elementos incompletos o de transición que tienen la particularidad de prescindir de los nodos "interiores" respecto de sus contrapartes, los elementos completos.

Un elemento completo cuenta con una serie de nodos interiores que no modifican su geometría y, sin embargo, incrementan el número de funciones de forma. La reducción en la cantidad de nodos implica un menor número de funciones de forma, significando una disminución en la cantidad de operaciones del método. Así, este fenómeno permite aumentar la eficiencia del código.

### Localización: El algoritmo de coordenadas de área

El algoritmo que se trata en esta sección es de aplicación exclusiva a elementos de primer orden, debido a que es condición necesaria que las aristas o bordes del elemento sean rectos. Las coordenadas de área permiten posicionar biunívocamente un punto dentro de un plano triangular con una terna ordenada. Cada valor de la terna corresponde al área de cada triángulo que surge de unir el punto interno con cada uno de los vértices (expresada de manera relativa al área total), como puede apreciarse en la Fig. 9.

La característica más importante de este sistema es la condición que se establece entre los elementos de la terna

$$a_1 + a_2 + a_3 = A \quad (20)$$

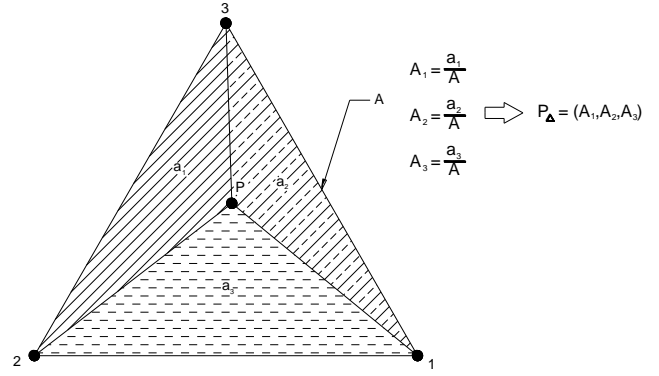


Fig. 9: Coordenadas de área para un punto arbitrario "P".

La condición anterior se cumplirá siempre que el punto "P" sea interno al triángulo, caso contrario la igualdad no verificará. Este fenómeno permite diferenciar un punto interno de uno externo, aunque, propuesto de esta manera, el método sólo aplica a elementos triangulares planos. No obstante, es posible extenderlo a otros casos, incluso en tres dimensiones. Por ejemplo, para un elemento cuadrilátero se requerirán cuatro coordenadas, es decir, se deberá calcular el área de cuatro triángulos, como muestra la Fig. 10.

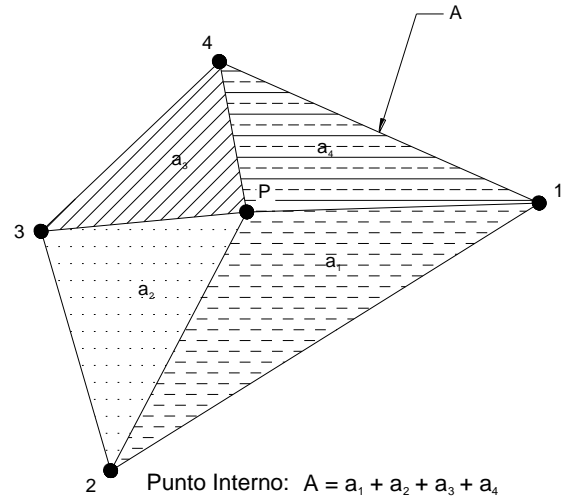


Fig. 10: Coordenadas de área para un elemento cuadrilátero.

Los elementos definidos en el espacio tridimensional requieren un tratamiento diferente pero que no pierde de vista el concepto propuesto. En estos casos, se calculan volúmenes y la cantidad de éstos dependerá del tipo de elemento. Se tratarán aquí los elementos hexaedros, puesto que representan la versión más general de los elementos espaciales.

Primeramente debe plantearse que, de acuerdo con la Fig. 11, un elemento hexaedro puede dividirse en seis fragmentos. Estas partes son pentaedros (pirámides) que tienen por base cada una de las caras del elemento original y como cúspide un punto arbitrario, "P". La suma de los volúmenes de los seis subelementos será igual al volumen total si y sólo si, el punto "P" es interno al elemento. Se tendrá entonces que

$$v_1 + v_2 + v_3 + v_4 + v_5 + v_6 = V. \quad (21)$$

Contrariamente, si el punto es externo al elemento, se verifica que

$$v_1 + v_2 + v_3 + v_4 + v_5 + v_6 > V. \quad (22)$$

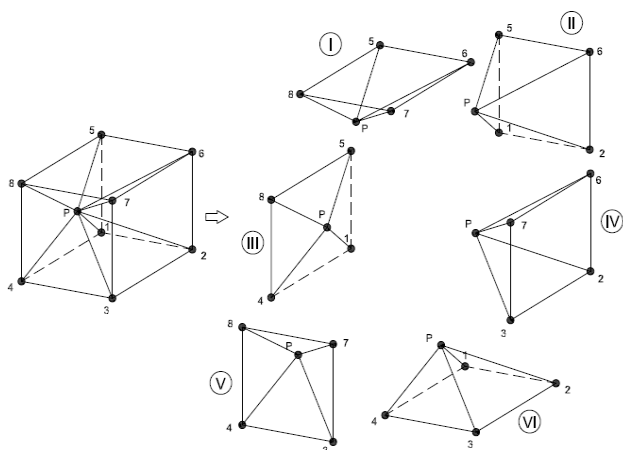


Fig. 11: División de un elemento hexaedro con el punto arbitrario "P".

En segundo lugar, es necesario calcular el volumen total del hexaedro prescindiendo el punto "P". Para ello será necesario dividirlo en porciones cuyos volúmenes puedan ser calculados de manera sencilla. Esta división consiste en definir seis subelementos tetraedros, según Fig. 12.

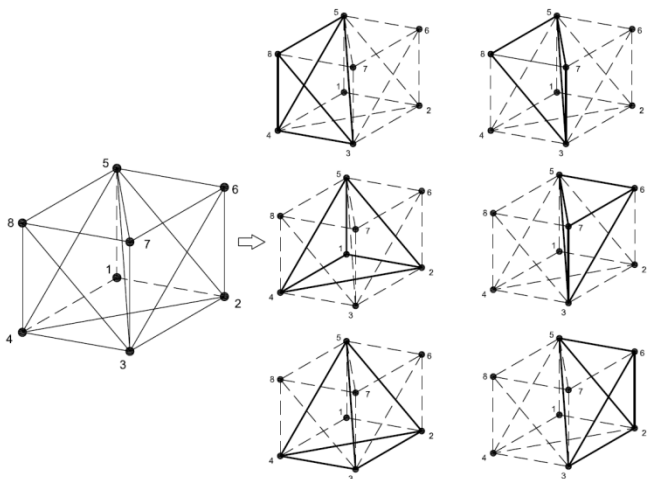


Fig. 12: División de un elemento hexaedro para cálculo de volumen total.

El *Algoritmo de Coordenadas de Áreas* requiere, como se dijo, el cálculo del valor de áreas y volúmenes, de elementos y de porciones de éstos. Para realizar esta tarea, se cuenta únicamente con las coordenadas de los nodos de cada elemento dadas por las Matrices de Procesamiento de Datos. Es por ello que se recurre a un principio del álgebra de vectores que utiliza el producto vectorial y el producto mixto como herramientas para calcular características geométricas.

**Puntos sin localizar**

Es posible que algunos puntos de la GA que no sean ubicados dentro de un elemento de la ME del conjunto preseleccionado. Esta situación puede deberse a dos casos en los cuales el *Módulo de Localización*, cualquiera sea el algoritmo que utilice, finalizará sus ciclos sin asignar un elemento al punto en cuestión, a saber:

- El punto a localizar no se encuentra dentro de la ME.
- El elemento que contiene al punto no fue incluido en la preselección.

En ese caso, la Matriz de Localización contará con un cero en vez de designar un elemento para el nodo en cuestión.

Ante la situación anterior, deberá evaluarse si es posible que el punto a localizar realmente sea externo a la ME. De no ser así, deberá ampliarse el conjunto dado por el *Módulo de Selección* mediante el cambio del algoritmo utilizado (por ejemplo, utilizar para esos nodos el *Algoritmo de Lista*).

**Evaluaciones**

Cualquier algoritmo que se desee implementar debe contar con dos cualidades, a saber, eficacia y eficiencia. Para evaluar la eficacia de los métodos se utilizarán contextos relativamente sencillos, con mallas estructurales de pocos elementos y una cantidad reducida de puntos a localizar. En cuanto a la eficiencia, se estudian contextos más reales y, por ende, más complejos. Por otro lado y dado que cada módulo cumple una función determinada y distinta, es necesario definir qué parámetros se considerarán para la evaluación en cada caso.

Los testeos sobre los distintos algoritmos se llevarán a cabo con la ayuda de mallas especialmente confeccionadas para tal fin. Los casos más sencillos permiten obtener resultados sin lidiar con tiempos de ejecución excesivamente largos, y comprenden los presentados en las Fig. 13 a Fig. 15. Las mallas más complejas, en contraparte, presentan ejemplos más realistas y son más exigentes con los códigos, permitiendo realizar análisis más profundos. En todos los casos se generan modelos de grillas aerodinámicas coherentes con las mallas estructurales para poder efectuar las evaluaciones.

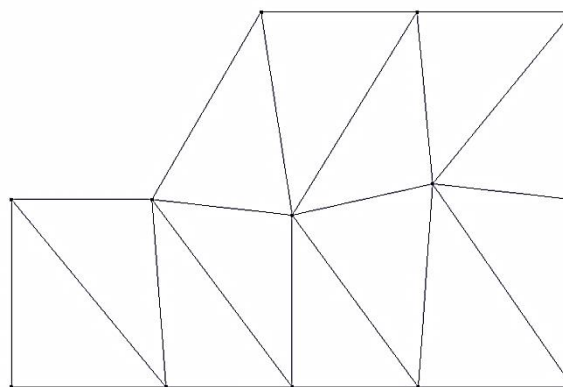


Fig. 13: Malla de prueba plana, CASO A.

Las mallas más interesantes de analizar los modelos de alas de insectos que fueron generados a partir del trabajo de S. Combes [3]. Se consideraron cuatro especies diferentes, Pepsis, Ischnura, Eristalis y Pieris, según su nombre científico, mostradas en las Fig. 16 a Fig. 20.

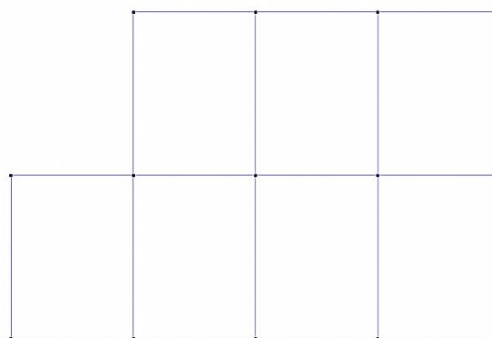


Fig. 14: Malla de prueba plana, CASO B.



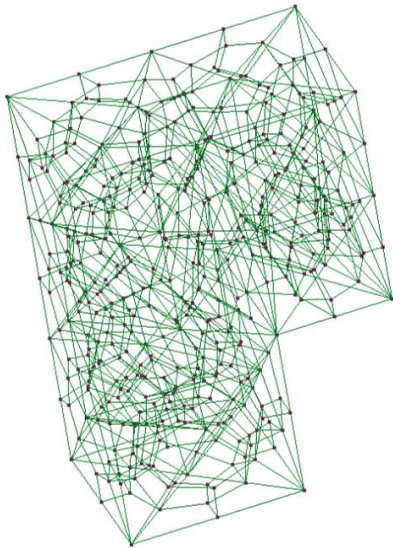


Fig. 15: Malla de prueba tridimensional, CASO C.

la GA. Los tiempos medidos, que se muestran en la Fig. 20, fueron adimensionalizados respecto del tiempo medio (la media aritmética de todos los tiempos de ejecución contabilizados), y se expresan como un parámetro comparativo, el tiempo de ejecución adimensional:

$$\tau_{ej} = \frac{\text{Tiempo Medido}}{\text{Tiempo Medio}} \quad (23)$$

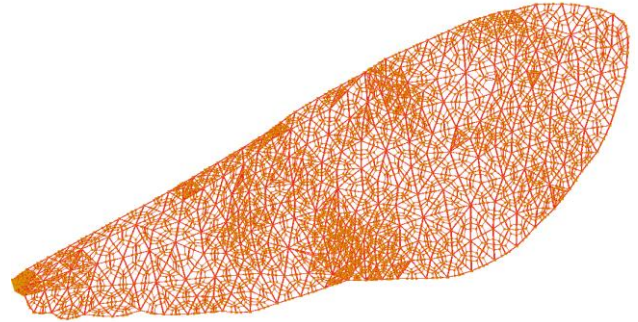


Fig. 19: Malla de prueba, modelo ala de Pepsis.

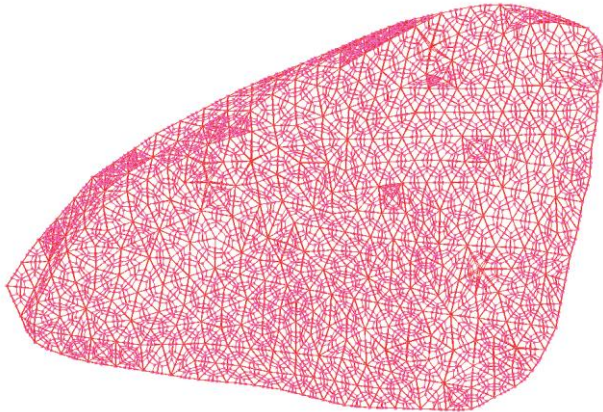


Fig. 16: Malla de prueba, modelo ala de Pieris.

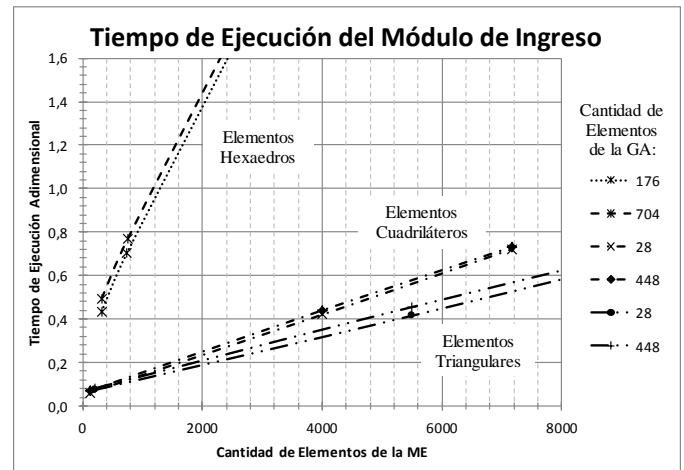


Fig. 20: Tiempo de ejecución del Módulo de Ingreso.

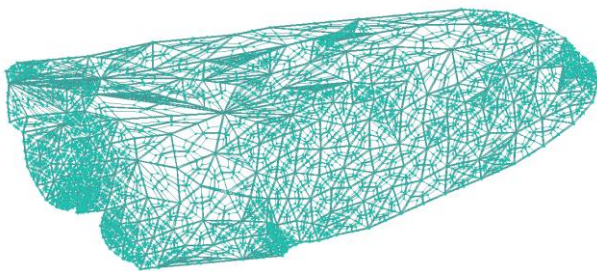


Fig. 17: Malla de prueba, modelo ala de Eristalis.

El incremento en la dimensión de las mallas (ME y GA) conlleva un aumento en el tiempo de ejecución. Sin embargo, un análisis más detallado indica que la influencia de la dimensión de la ME es mucho más marcada que su contraparte referida a la GA. Este fenómeno encuentra motivos en que el Módulo de Ingreso sólo confecciona para la grilla del UVLM el *Arreglo de Nodos y Coordenadas*, mientras que, para la ME debe además generar el *Arreglo IEN*.

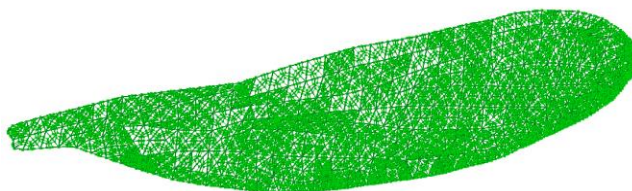


Fig. 18: Malla de prueba, modelo ala de Ischnura.

### Evaluación del Módulo de Ingreso

La medición de performance del *Módulo de Ingreso* se realizó a partir de la toma de tiempos de procesamiento para las mallas de prueba antes presentadas (Fig. 13 a Fig. 19). El objetivo de la evaluación fue obtener la relación entre el tiempo de ejecución y las dimensiones de la ME y

### Evaluación del Módulo de Selección

En este caso hay dos aspectos fundamentales. Por un lado es importante determinar el tiempo necesario para realizar la preselección de elementos para cada nodo de la GA. Por el otro, el ordenamiento que realice este módulo de los elementos preseleccionados tendrá gran influencia en el tiempo de ejecución del módulo siguiente (localización), pues puede modificar la cantidad de evaluaciones necesarias para localizar efectivamente un nodo en un elemento.

A su vez, los algoritmos de selección son analizados en dos grupos. La división es debida a la gran diferencia que existe en el desempeño entre el *Algoritmo de Lista* y sus



variantes y los restantes métodos. Para los primeros, se midió el tiempo total de ejecución del Programa de Vinculación utilizando el Algoritmo de Newton-Raphson para efectuar las localizaciones y se compararon los resultados con el Algoritmo de Centros de Masa en condiciones análogas. Los resultados se exponen en la Fig. 21, donde puede observarse que la performance del Algoritmo de Lista y sus tres variantes es similar y muy inferior a la correspondiente al Algoritmo de Centros de Masa.

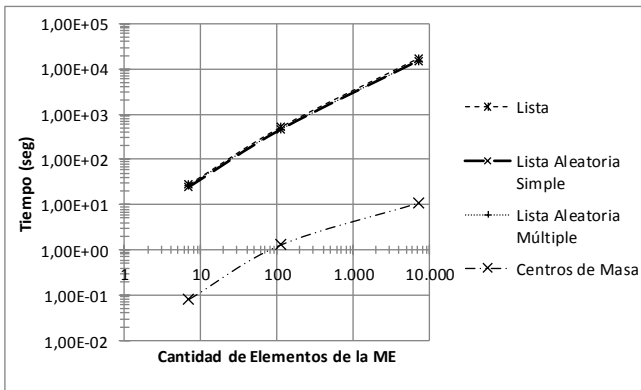


Fig. 21: Tiempo de ejecución, algoritmos de selección por “fuerza bruta”.

Tal y como fueron descritos oportunamente, el Algoritmo de Lista y sus variantes constituyen métodos conocidos como de fuerza bruta y se caracterizan por ser extremadamente simples. Tienen a su favor una gran robustez, por lo que pueden considerarse virtualmente a prueba de fallos. A pesar esto, son altamente ineficientes, lo que hace poco recomendable su aplicación general a modelos de interés práctico.

Los restantes algoritmos destinados al Módulo de Selección permiten un análisis algo más profundo. En este caso, el tiempo de ejecución se computa como un parámetro adimensional obtenido según la ec. (23), y corresponde al tiempo requerido para realizar la preselección de un conjunto de elementos a evaluar para cada punto a localizar (de la GA). Estos valores se encuentran en la Fig. 22. En ella puede observarse que el algoritmo más rápido para realizar la preselección es el de Centros de Masa. El algoritmo ESUP necesita aproximadamente el doble de tiempo y el ESUP Ordenado por Centros de Masa es el más lento, existiendo una gran diferencia con los otros dos.

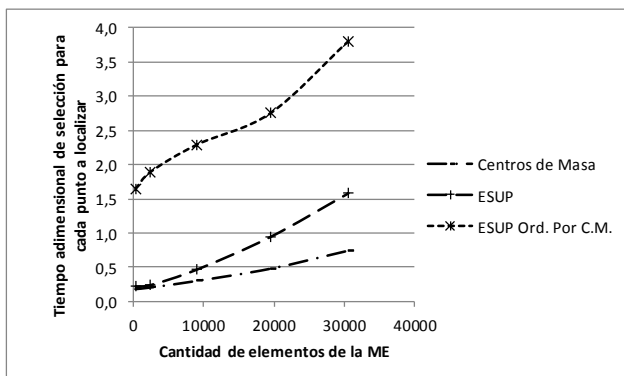


Fig. 22: Tiempo de preselección de elementos para cada punto de la GA.

En cuanto a la calidad del ordenamiento de los elementos preseleccionados, se midió cuántos elementos fue necesario evaluar para localizar cada nodo de la GA. En la Fig. 23 se

observan los promedios correspondientes a cada algoritmo. Aquí, el algoritmo ESUP Ordenado por Centros de Masa es el que provee mejor performance, siendo poco más eficiente que el de Centros de Masa. El algoritmo ESUP provee la peor performance en este sentido, lo que se corresponde con el hecho de que es el único que no cuenta con un criterio de ordenamiento de los elementos preseleccionados.

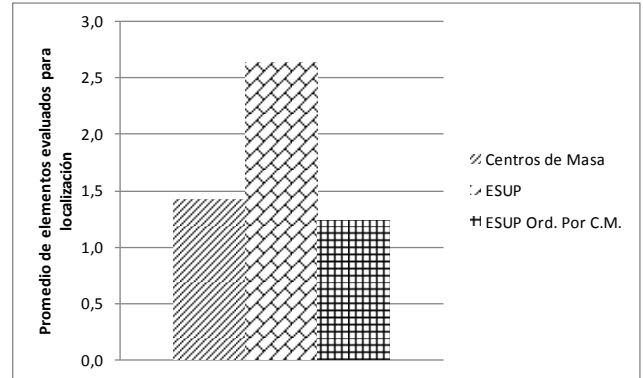


Fig. 23: Precisión en la selección, promedio de elementos evaluados.

### Evaluación del Módulo de Localización

Para las comparaciones se utilizó el valor medio del tiempo necesario para evaluar un elemento. Se analizó cada tipo de elemento por separado. En la Fig. 24 se observan los resultados obtenidos. Como la reducción de elementos no es aplicable a elementos cuadriláteros y triangulares de primer orden, no se muestran datos en estos casos.

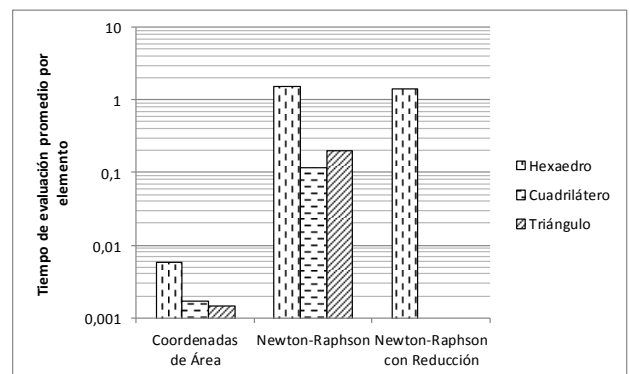


Fig. 24: Tiempo de ejecución por cada elemento evaluado.

Las cifras marcan una notable diferencia entre la velocidad del Algoritmo de Coordenadas de Área y las dos variables del Algoritmo de Newton-Raphson. Otra diferencia, quizás menos significativa, es la que existe al aplicar la Reducción de Elementos al Algoritmo de Newton-Raphson, con mejoras en los tiempos rondan el 10%.

### CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se implementaron diferentes algoritmos que permiten la vinculación a nivel topológico de grillas aerodinámicas provenientes del Método de Red de Vórtices Inestacionario y mallas estructurales correspondientes al Método de Elementos Finitos. Dicha vinculación consiste en la determinación, para cada punto de la grilla aerodinámica, del elemento finito que lo contiene.

Las herramientas computacionales generadas fueron utilizadas para procesar datos de casos de prueba simples y casos de aplicación práctica con un nivel de complejidad

elevado. Los resultados obtenidos se utilizaron, en conjunto con mediciones de tiempo, para evaluar y comparar los diferentes métodos propuestos entre sí. Esto permitió generar recomendaciones claras para implementaciones de este tipo.

Los resultados del análisis muestran, en primer lugar, que es conveniente realizar el proceso en dos etapas, una de preselección de elementos y una de localización propiamente dicha. La principal desventaja de esta estrategia radica en la posibilidad de que el elemento que contiene a un nodo no se encuentre dentro de la lista de elementos preseleccionados. Sin embargo, debe considerarse lo siguiente:

- es probable que esto no ocurra si se utilizan mallas de buena calidad (respecto de la regularidad en forma y tamaño de los elementos); y
- llegado el caso, el problema tiene solución simple aplicando, solo para los puntos en cuestión, alguno de los algoritmos más robustos (aunque menos eficientes).

Con respecto a la etapa de preselección, es importante que el algoritmo elegido cuente con algún criterio de ordenamiento de elementos, ubicando en primer lugar aquellos para los que la probabilidad de contener el punto analizado sea mayor. Esto disminuye de manera importante la cantidad de evaluaciones a realizar en la segunda etapa, aumentando notablemente su velocidad.

De los algoritmos implementados para esta etapa, el de *Lista* y sus variantes resultan ser los menos eficientes pero los más robustos. A su vez, no existen diferencias sustanciales de eficiencia entre sí, siendo el algoritmo de *Lista* el de más fácil implementación. Se recomienda su utilización para localizar puntos conflictivos. De los otros tres algoritmos, el de *Centros de Masas* es claramente el más ventajoso pues, si bien genera un ordenamiento que requiere algunas evaluaciones más que el *ESUP Ordenado por Centros de Masas*, es mucho más eficiente para realizar las preselecciones.

Con respecto a la etapa de localización propiamente dicha, es altamente recomendable utilizar algoritmos basados en coordenadas de área siempre que sea posible. Para elementos de orden superior, utilizar la estrategia de Reducción mejora la eficiencia.

Como trabajos futuros se plantean las siguientes líneas de investigación:

- utilización de información de conectividad de la grilla aerodinámica para aumentar la eficiencia de la etapa de preselección;
- búsqueda de nuevos algoritmos para mejorar la eficiencia de la etapa de localización para elementos de orden superior;
- articulación de los métodos implementados con la determinación de las coordenadas locales de los puntos a localizar dentro del elemento que los contiene;
- análisis de aplicabilidad de los métodos implementados a otros tipos de elementos,

entre otras.

## REFERENCIAS

- [1] Thomas J. R. Hughes (1987), "The Finite Element Method: Linear Static and Dynamic Finite Element Analysis," Prentice-Hall Inc., New Jersey, USA.
- [2] Reinald Löhner (2008), "Applied CFD Techniques: An Introduction Based on Finite Element Methods," John Wiley & Sons, New Jersey, USA.
- [3] Stacey A. Combes (2002), "Wing Flexibility and Design for Animal Flight," PhD Thesis, University Of Washington, USA.
- [4] Romesh C. Batra (2006), "An Introduction to the Finite Element Method," Department of Engineering Science & Mechanics, Virginia Polytechnic Institute & State University, Blacksburg, USA.
- [5] C. Geuzaine, J. F. Remacle (2003), "Gmsh Reference Manual," Free Software Foundation Inc., USA.
- [6] Amar Khennane (2013), "Introduction to Finite Element Analysis Using MATLAB and Abaqus", CRC Press Taylor & Francis Group LLC, USA.
- [7] A.Kaveh (2014), "Computational Structural Analysis and Finite Element Methods", Springer International Publishing, Switzerland.