

GENERACIÓN DE SUB-DOMINIOS LOCALES DE INTERPOLACIÓN EN UN MÉTODO SIN MALLA

GENERATION OF LOCAL INTERPOLATION SUB-DOMAINS IN MESHLESS METHODS

Nicolás Ipinza Carrasco^{1,2} Franco Perazzo Maggi¹ Juan Aranda Pastén^{1,2} Luis Pérez Pozo^{1,2}

Recibido 15 de junio de 2006, aceptado 8 de mayo de 2007

Received: June 15, 2006 Accepted: May 8, 2007

RESUMEN

Una de las principales ventajas de utilizar un método sin malla es la posibilidad de realizar análisis numéricos sin la necesidad de efectuar un proceso de partición o subdivisión del dominio en elementos más pequeños. Sin embargo, la utilización de puntos o partículas para discretizar un cuerpo requiere de una estrategia para seleccionar aquellos puntos que formarán parte de los sub-dominios de interpolación local de la solución, también llamados “nubes”. El presente trabajo muestra el desarrollo e implementación de una técnica para la generación de los sub-dominios de interpolación para un método sin malla.

Para validar la correcta implementación de la técnica propuesta, se llevarán a cabo simulaciones en el campo de la elasticidad lineal de sólidos 3D mediante un programa numérico basado en la formulación teórica del Método de Puntos Finitos (MPF).

Palabras clave: Métodos sin malla, método de puntos finitos, nubes de interpolación local.

ABSTRACT

The absence of grids in meshless methods removes the burden associated to them in the modelling process. Nevertheless an adequate strategy for the selection of points which will be used in the interpolation sub domains (clouds) is required. This work presents the development and application of such a strategy. Corresponding validations are presented through simulations in the field of 3D linear elasticity, performed with a meshless program based in the Finite Point Method technique.

Keywords: Meshless methods, finite point method, local interpolation of clouds.

INTRODUCCIÓN

El desarrollo de métodos numéricos sin malla o libre de malla plantea una forma alternativa de solución en algunos problemas de la mecánica computacional, que tradicionalmente pueden ser resueltos mediante técnicas numéricas como el Método de Elementos Finitos (MEF) o el de Volúmenes Finitos. La ausencia en estos métodos de una malla de elementos o de elementos disminuye el tiempo invertido en la preparación de la información necesaria para el cálculo y tiene ventajas sobre todo en geometrías 3D donde es necesario contar con un generador

de malla eficiente y robusto [11]. La manera de realizar la interpolación local de la función aproximada y la forma de obtener el sistema de ecuaciones diferenciales discretas que gobiernan el problema ha dado lugar a dos clases de métodos sin malla, la primera basada en *smooth particle hydrodynamics procedures* [19] y la segunda basada en *generalized finite difference techniques* [11, 17], para mayor referencia y detalles de otros métodos sin malla ver [15].

En el Método de Puntos Finitos (MPF) [2-4], la aproximación local se obtiene mediante la técnica estándar de mínimos cuadrados ponderados, utilizándose

¹ Departamento de Mecánica. Universidad Técnica Federico Santa María. Avenida España 1680. Valparaíso, Chile. E-mail: nicolas.ipinza@usm.cl, franco.perazzo@usm.cl

² Aula UTFSM-CIMNE. Departamento de Mecánica. Universidad Técnica Federico Santa María. Avenida España 1680. Edif. C-246. Fono: (56-32) 2654133. Valparaíso, Chile. E-mail: juan.arandap@usm.cl, luis.perez@usm.cl

colocación puntual para obtener el sistema de ecuaciones discretas, es decir, se utiliza el conjunto de ecuaciones diferenciales directamente, sin necesidad de construir una forma débil o realizar integraciones algunas sobre el dominio [18]. La consistencia y convergencia del método ha sido analizada con anterioridad por los autores, verificándose un correcto comportamiento del método para problemas tanto de mecánica de fluidos como de sólidos, fundamentalmente en 2D [4, 5, 7, 9 y 14]. En el presente trabajo se analiza y desarrolla una técnica para la generación de sub-dominios locales de interpolación o “nubes” en 3D, basada en una triangulación local de Delaunay. Esta importante etapa comprende desde la modelación geométrica de la pieza a analizar, hasta la generación de nubes para todo el dominio.

Los ejemplos desarrollados en geometrías 3D demuestran un correcto comportamiento de la solución numérica del MPF cuando se utiliza la técnica de generación de nubes propuesta en este trabajo.

FORMULACIÓN DEL MÉTODO DE PUNTOS FINITOS

Sea Ω_I el subdominio de interpolación de una función $u(\mathbf{x})$ y s_j con $j=1,2,\dots,n$ una colección de n puntos con coordenadas $\mathbf{x}_j \in \Omega_I$. El subíndice I en las expresiones identifica aquel punto de la nube, donde se requiere evaluar la aproximación, también denominado “nodo estrella”. La función incógnita $u(\mathbf{x})$ puede ser aproximada en el interior de Ω_I por

$$u(\mathbf{x}) \cong \hat{u}(\mathbf{x}) = \sum_{l=1}^m p_l(\mathbf{x}) \alpha_l = \mathbf{p}^T(\mathbf{x}) \alpha$$

$$\forall \mathbf{x}_I \in \Omega, \forall \mathbf{x} \in \Omega_I \quad (1)$$

donde $\alpha^T = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_m]$, y el vector $\mathbf{p}(\mathbf{x})$, llamado “base de interpolación”, contiene típicamente monomios que aseguran en el espacio de coordenadas una base completa. Por ejemplo en 3D, las bases más usadas son las siguientes:

$$p^T = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2]$$

$$p^T = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2, \dots, x^2y, x^2z, xy^2, xyz, xz^2, y^2z, yz^2]$$

$$p^T = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2, x^3, \dots, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3] \quad (2)$$

con $m = 10, 17$ y 20 respectivamente.

La función incógnita $u(\mathbf{x})$ puede ser evaluada en los n puntos de la nube Ω_I , obteniendo

$$\mathbf{u}^h = \begin{Bmatrix} u_1^h \\ \vdots \\ u_n^h \end{Bmatrix} \cong \begin{Bmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_n \end{Bmatrix} = \begin{Bmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_n^T \end{Bmatrix} \alpha = \mathbf{C} \alpha \quad (3)$$

donde $u_j^h = u(\mathbf{x}_j)$ son las incógnitas, pero los valores buscados $\hat{u}_j = \hat{u}(\mathbf{x}_j)$ son los valores aproximados y $\mathbf{p}_j = \mathbf{p}(\mathbf{x}_j)$.

En una aproximación mediante elementos finitos el número de puntos en el subdominio se escoge de forma que $m = n$. En este caso, \mathbf{C} es una matriz cuadrada y el procedimiento conduce a las funciones estándares del método de elementos finitos.

Si $n > m$, la aproximación utilizada no se puede adaptar a todos los valores de u_j^h . El problema puede ser resuelto determinando los valores de \hat{u} , que minimicen la suma de las distancias al cuadrado o error en cada punto, ponderado por una función $\omega(\mathbf{x}_I - \mathbf{x}_j)$, es decir

$$\mathbf{J}_I = \sum_{l=1}^n \omega(\mathbf{x}_I - \mathbf{x}_j) (u_j^h - \hat{u}(\mathbf{x}_j))^2 \quad \text{mínimo} \quad (4)$$

La función de ponderación utilizada en el MPF corresponde a la función de Gauss normalizada. Reemplazando la aproximación (1) en (4) y minimizando respecto de α , se obtiene

$$\alpha = \mathbf{C}_I^{-1} \mathbf{u}^h, \quad \mathbf{C}_I^{-1} = \mathbf{A}_I^{-1} \mathbf{B}_I \quad (5)$$

donde las componentes de $\mathbf{A}(\mathbf{x}_I) = \mathbf{A}_I$ (matriz de momentos) y $\mathbf{B}(\mathbf{x}_I) = \mathbf{B}_I$ son

$$[\mathbf{A}_I]_{ij} = \sum_{k=1}^n p_i(\mathbf{x}_k) \omega(\mathbf{x}_I - \mathbf{x}_k) p_j(\mathbf{x}_k) \quad i, j = 1, \dots, m$$

$$[\mathbf{B}_I]_{ij} = p_i(\mathbf{x}_j) \omega(\mathbf{x}_I - \mathbf{x}_j) \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (6)$$

Substituyendo α de la ecuación (5) en (1), la aproximación final adopta la siguiente forma.

$$\hat{u}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{C}_I^{-1} \mathbf{u}^h \quad \forall \mathbf{x} \in \Omega_I \quad (7)$$

con las funciones de forma

$$N_j^i(x) = \mathbf{p}^T(\mathbf{x})\mathbf{C}_j^{-1} \quad i = 1, \dots, n \quad (8)$$

Se debe notar que de acuerdo con el carácter de mínimos cuadrados de la aproximación

$$u(\mathbf{x}_j) \cong \hat{u}(\mathbf{x}_j) \neq u_j^h \quad (9)$$

es decir, los valores locales de la función aproximada no coinciden con los valores nodales de la función incógnita. De todas formas \hat{u} es una aproximación válida, con la cual se busca satisfacer la ecuación diferencial y condiciones de contorno, siendo u_j^h simplemente parámetros desconocidos.

CONCEPTO DE GENERACIÓN DE NUBES

Dentro de todo método numérico libre de malla hay tres procesos básicos presentes: proceso de interpolación o aproximación local, proceso de ponderación y finalmente discretización de la función objetivo. Se puede observar en [8] que las etapas citadas determinan la importancia de la generación de sub-dominios locales o “nubes de puntos” y definen en gran manera la calidad del cálculo posterior.

En un método libre de malla, el dominio de análisis debe quedar descrito completamente a través de nodos. Una vez obtenida la totalidad de los puntos, tanto del contorno como del dominio interior, se procede con el proceso propio de generación de nubes.

En la figura 1 se presenta un ejemplo del concepto de nubes de puntos.

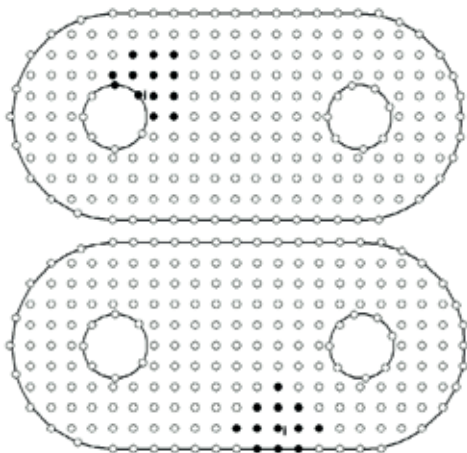


Figura 1. Concepto de nubes de puntos.

TÉCNICAS DE GENERACIÓN DE NUBES

Los investigadores dedicados a los métodos libres de malla han propuesto diversas técnicas geométricas de generación de nubes, [6, 11-13, 16-17]. La metodología seguida en el presente trabajo está basada en la investigación de Löhner [14].

Para aplicar la técnica propuesta de generación de nubes, se requieren dos aspectos fundamentales, por un lado la lista de puntos que discretizan la geometría y por otro la triangulación de la superficie.

Una vez obtenida esta información se procede a determinar las nubes de puntos. Una técnica fácil de implementar está basada en el criterio de mínima distancia [12], en el cual los puntos que conforman la nube quedan dentro de un círculo de radio R_1 y centrado en el nodo estrella, *a priori* no resulta fácil definir el valor de R_1 de modo de conseguir un número suficiente de puntos; sin embargo, el mayor inconveniente es conseguir una nube adecuada, o lo más simétrica posible; ver figura 2a.

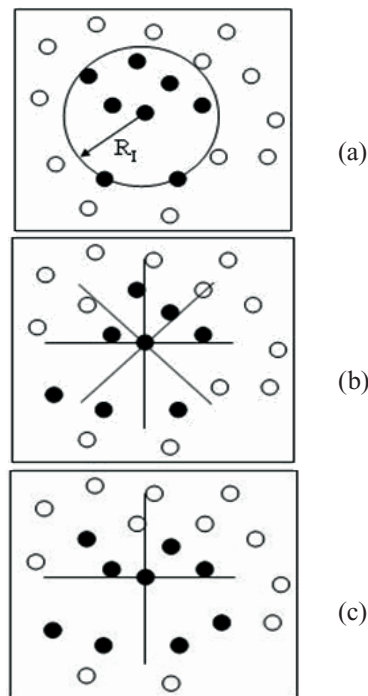


Figura 2. Generación local de subdominios, a) criterio mínima distancia, b) criterio de ocho segmentos y c) criterio de cuatro cuadrantes

Una segunda técnica denominada por sus autores como “eight segment criterion” [11], consiste en subdividir el entorno que rodea al nodo estrella con ocho segmentos

y seleccionar por cada segmento el nodo más cercano; ver figura 2b.

Un tercer método denominado como “four quadrant criterion” [17], consiste en utilizar un sistema cartesiano de ejes en el nodo estrella y escoger los dos más cercanos a éste por cada cuadrante. En estos dos últimos métodos se pueden conseguir nubes de mejor calidad, sin embargo, deben ser modificados para la obtención de los sub-dominios para nodos de la superficie y cercanos a ésta, ver figura 2c.

Triangulación de Delaunay

La técnica empleada en el presente trabajo se basa en una triangulación local de Delaunay, la cual es ampliamente utilizada debido a lo equilibrada de la distribución obtenida, esta permite conectar el nodo estrella con sus vecinos más próximos mediante tetraedros (caso 3D) [20].

Los pasos importantes de esta técnica son:

1. Definir un sistema local de ejes cartesianos, centrado en el nodo estrella, para identificar aquellos puntos más próximos al origen por cada octante, ver figura 3a.
2. Formar con el nodo estrella los tetraedros iniciales; ver figura 3b.
3. Para cada tetraedro se genera una esfera circunscrita; ver figura 3c.
4. Se verifica la NO existencia de puntos interiores adicionales dentro de la esfera; ver figura 3c. En caso de existir un punto interior se elimina el tetraedro inicial y se crean dos nuevos tetraedros con el punto más cercano al nodo estrella; ver figura 3d.
5. Se repite este proceso para todos los puntos pertenecientes al dominio local; ver figura 3e.
6. Finalmente se escogen los puntos más cercanos o que aportan más información al nodo estrella; ver figura 3f.

GENERACIÓN DE NUBES Y VECTORES NORMALES

A continuación se detallan las etapas involucradas en la generación de nubes 3D.

Generación inicial de puntos que discretizan la geometría

La generación inicial de puntos que discretizan todo el dominio del problema y la triangulación de la superficie se obtienen a partir del programa comercial de Pre y Post proceso GIDTM.

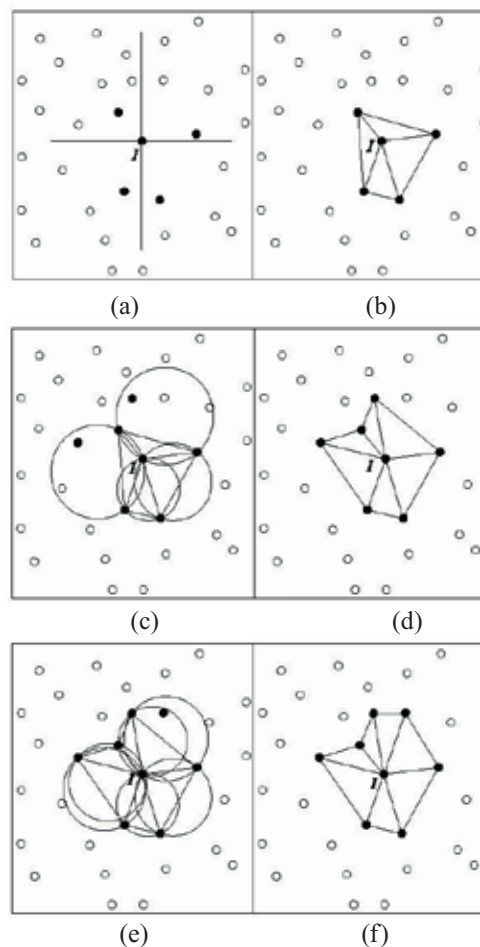


Figura 3. Generación local de subdominios mediante la triangulación de Delaunay.

Cálculo de la región local de búsqueda de puntos para cada subdominio de interpolación o “nube”

La región local de búsqueda de puntos es un cubo de lado L , centrado en el nodo estrella que se calcula de la siguiente forma:

$$L^3 = V = \frac{n}{\rho} \quad (10)$$

donde V es el volumen de la región local, n es el N° de puntos contenidos en ella (este se considera constante e igual a un número arbitrario de puntos, el cual se define previamente) y ρ la densidad de puntos de esta región. Se considera que la densidad de puntos es constante para todo el volumen de la geometría, por tanto este parámetro también es constante para cada una de las regiones locales. Se calcula en base a la cantidad de puntos totales de la discretización y el volumen del paralelepípedo que encierra a la geometría del problema.

Búsqueda de puntos cercanos

La búsqueda para los puntos cercanos se realiza usando la región local calculada anteriormente tal como se muestra en las figuras 4a y 4b. Luego, dentro de cada región se verifica la cantidad de puntos; si el número de puntos cercanos encontrados es demasiado pequeño, la región de búsqueda se agranda en un 30%. Por el contrario, si demasiados puntos fueron encontrados, la región de búsqueda se reduce en un 15%. Se repite este procedimiento, hasta que un número aceptable de puntos cercanos sea encontrado.

Búsqueda de caras cercanas

La búsqueda de caras cercanas se realiza usando la región local utilizada en la etapa anterior. Mediante el análisis de vértices y baricentros de las caras respectivas, se procede a verificar si estos puntos pertenecen o no a la región local respectiva; ver figura 4b.

Filtrando caras cercanas

Luego de almacenadas las caras correspondientes a cada nodo estrella, se deben eliminar aquellas que el nodo estrella *i*, “no pueda ver”. Esto se puede observar en la figura 4c, donde sólo quedan las caras que cumplen con la condición descrita.

Filtrando puntos cercanos

La búsqueda para los puntos cercanos puede entregar algunos ubicados en el lado “incorrecto” de un límite según lo mostrado en la figura 4c. Estos puntos son llamados puntos “inválidos”. Esta situación sucede con frecuencia para las esquinas agudas, zonas cóncavas y en general para cualquier geometría compleja con nubes densas de puntos. Las caras cercanas obtenidas previamente se pueden utilizar para filtrar los puntos más lejanos. Por lo tanto, cualquier punto *j* que al crear el segmento de recta **punto *j*: punto *i*** intersekte una de las caras cercanas, se quita de la lista, dejando sólo los puntos en el lado “correcto” del límite; ver figura 4d.

Triangulación de Delaunay

Se generan los tetraedros mediante la técnica de Delaunay dentro del dominio local de búsqueda.

Una vez obtenida una nube preliminar de puntos alrededor del nodo estrella, se determina la cantidad requerida de puntos que conforman ésta. Este ítem se relaciona con el tamaño de la base de interpolación usada; ver ecuación (2).

La técnica descrita anteriormente permite generar nubes para problemas en geometrías complejas, con presencia de concavidades a lo largo del contorno de la pieza.

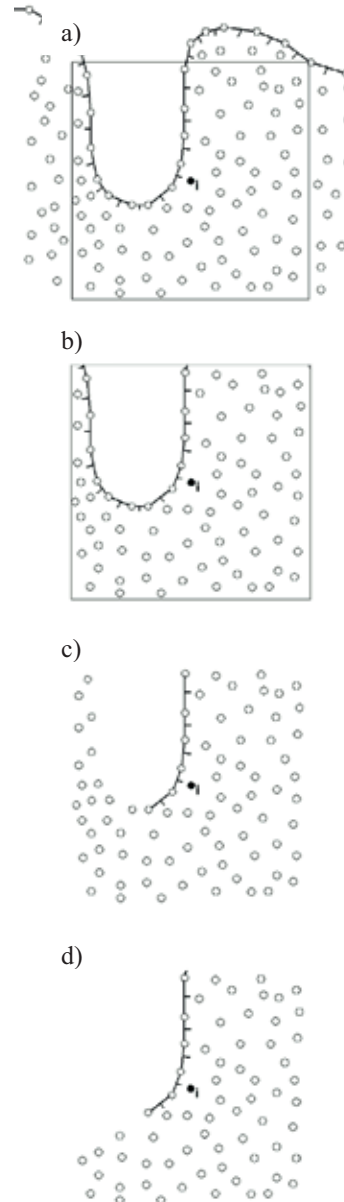


Figura 4. Esquema de generación de sub-dominio local para nodo estrella *i*.

Cálculo del vector normal perteneciente a cada punto de la superficie

Una vez obtenidos los puntos pertenecientes a la nube, se procede a calcular los vectores normales correspondientes a cada uno de los puntos de la superficie de la discretización,

para ello se utiliza la triangulación de la superficie obtenida a partir del programa GIDTM. En primer lugar se determinan los triángulos vecinos al punto donde se desea determinar el vector normal; ver figura 5. Luego se calcula el vector normal de cada uno de los triángulos vecinos y se procede a calcular el vector del punto de interés de la siguiente forma

$$\hat{n}_i = \sum \frac{\hat{n}_c}{\|\hat{n}_c\|} \quad (11)$$

donde, \hat{n}_i es el vector normal del punto de interés y \hat{n}_c el vector normal de cada uno de los triángulos alrededor de este punto.

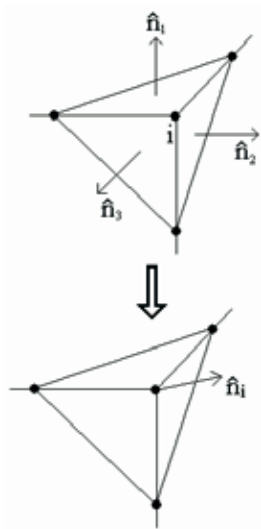


Figura 5. Cálculo de vector normal para puntos superficie.

IMPLEMENTACIÓN DEL GENERADOR DE NUBES PROPUESTO

Para implementar el generador de nubes 3D, se necesita la siguiente información geométrica de la pieza a analizar:

1. La lista de puntos que describen la geometría, tanto del contorno como del dominio.
2. La lista de triángulos que definen la superficie del contorno que encierra el cuerpo a analizar.

Es importante poner énfasis en este último punto y mencionar que para el desarrollo de la técnica de generación de nubes propuesta es necesaria una triangulación o mallado de la superficie que representa el contorno de la geometría, ya que de esta manera se definen con precisión

los límites del cuerpo tridimensional. Esto no quiere decir bajo ninguna circunstancia que la técnica transgreda la filosofía “meshless”, ya que la técnica numérica de cálculo sólo considera los nodos que discretizan la geometría [14]. Los triángulos que definen los contornos sólo son utilizados en la correcta generación de nubes que se necesita.

El trabajo se ha llevado a cabo teniendo como generador de la geometría al software de Pre y Post proceso GIDTM. Es sobre la base de la información que puede entregar éste que se elaboraron las distintas etapas de la generación de nubes. En la figura 6 se muestra una geometría 3D particular, con su respectiva discretización del contorno mediante triángulos.

Una vez obtenidos los puntos que discretizan la geometría y los triángulos que describen el contorno, corresponde la etapa de búsqueda de puntos y caras cercanas a cada nodo estrella, tanto del contorno, como del dominio.

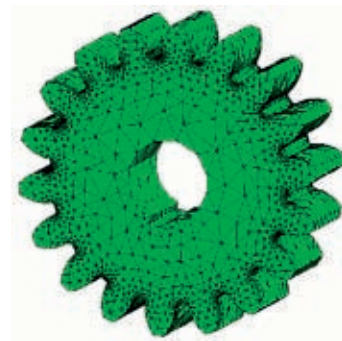


Figura 6. Contorno de rueda dentada recta, discretizada a través de triángulos.

Luego de obtener los puntos y caras cercanos como se muestra en la figura 4b, hay que filtrar los puntos “inválidos” correspondientes a esta primera búsqueda.

Para la eliminación de puntos inválidos se deben filtrar las caras que el nodo estrella “no puede ver”. Se utiliza para ello el siguiente algoritmo [10]:

DO: triángulos almacenados

- Obtener vector normal del triángulo apuntando hacia el exterior \hat{n}_Δ
 - Producto cruz entre dos lados del triángulo, cuidando de que apunte hacia el exterior. Luego normalizar el vector obtenido.
- Obtener vector que une al nodo estrella i con el baricentro del triángulo $\vec{v}_{Ba \rightarrow i}$

- Obtener producto punto entre \hat{n}_Δ y $\vec{v}_{Ba \rightarrow i}$
- Obtener ángulo α entre \hat{n}_Δ y $\vec{v}_{Ba \rightarrow i}$
 $\alpha = \text{COS}^{-1} \left(\hat{n}_\Delta \cdot \vec{v}_{Ba \rightarrow i} / \left(\|\hat{n}_\Delta\| * \|\vec{v}_{Ba \rightarrow i}\| \right) \right)$
- Verificar si
 $\alpha \geq 90^\circ$ Se mantiene el triángulo.
 $\alpha < 90^\circ$ Se elimina el triángulo.

END DO

En la figura 7 se puede observar esta técnica de eliminación de caras en una analogía 2D.

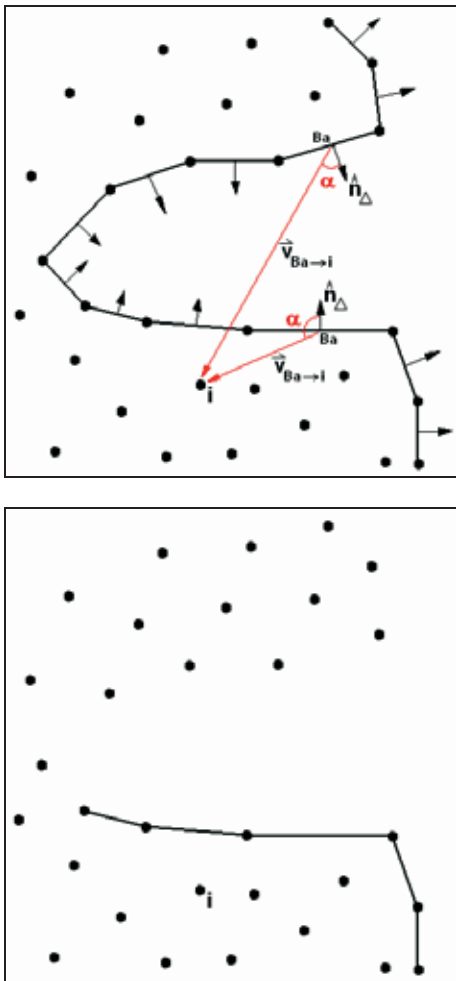


Figura 7. Eliminación de caras que el nodo estrella “no puede ver”.

Luego de esta etapa, se procede a eliminar los puntos inválidos de la nube preliminar. La técnica de eliminación se basa en el segmento de recta que une el nodo estrella con cada punto a analizar. Si esta intersecciona alguna de las caras cercanas al nodo estrella, se dispone la eliminación

del punto respectivo. El detalle del algoritmo [10] es el siguiente:

DO: puntos j de la nube preliminar

Obtener segmento de recta \vec{L}_{ij} , uniendo el nodo estrella i , con nodo j examinado.

DO: Triángulos almacenados (vértices a, b, c)

Obtener vector normal del triángulo \hat{n}_Δ

Calcular el punto P , que intersecciona el plano del triángulo, con la proyección de la recta \vec{L}_{ij}

IF: Punto P no pertenece al segmento de recta \vec{L}_{ij}

CYCLE

Verificar si punto P pertenece a la región del triángulo

Calcular los puntos medios de cada lado del triángulo M_{ab}, M_{ac}, M_{bc}

Obtener tres vectores uniendo el punto P , con los puntos medios M de cada lado, $\overline{MP}_{ab}, \overline{MP}_{ac}, \overline{MP}_{bc}$.

Calcular los tres vectores normales interiores a cada lado del triángulo $\hat{n}_{ab}, \hat{n}_{ac}, \hat{n}_{bc}$.

Calcular producto punto para cada lado respectivo

$$\left(\overline{MP}_{ab} \cdot \hat{n}_{ab} \right), \left(\overline{MP}_{ac} \cdot \hat{n}_{ac} \right), \left(\overline{MP}_{bc} \cdot \hat{n}_{bc} \right).$$

$$\text{IF: } \left[\left(\overline{MP}_{ab} \cdot \hat{n}_{ab} \geq 0 \right) \text{ y } \left(\overline{MP}_{ac} \cdot \hat{n}_{ac} \geq 0 \right) \text{ y } \left(\overline{MP}_{bc} \cdot \hat{n}_{bc} \geq 0 \right) \right]$$

THEN

Punto P pertenece a la región del triángulo

ELSE

Punto P no pertenece a la región del triángulo

END IF

IF: Punto P pertenece a la región del triángulo

THEN

Eliminar punto j

EXIT DO

END IF

END DO

END DO

En la figura 8 se ilustra globalmente el proceso de eliminación de caras y puntos inválidos de la nube del nodo i .

Una vez definida la nube corresponde verificar la cantidad de nodos que la conforman, teniendo en cuenta el tamaño de la base de interpolación que se utilizará. El criterio para añadir o eliminar nodos es la distancia entre éstos y el nodo estrella.

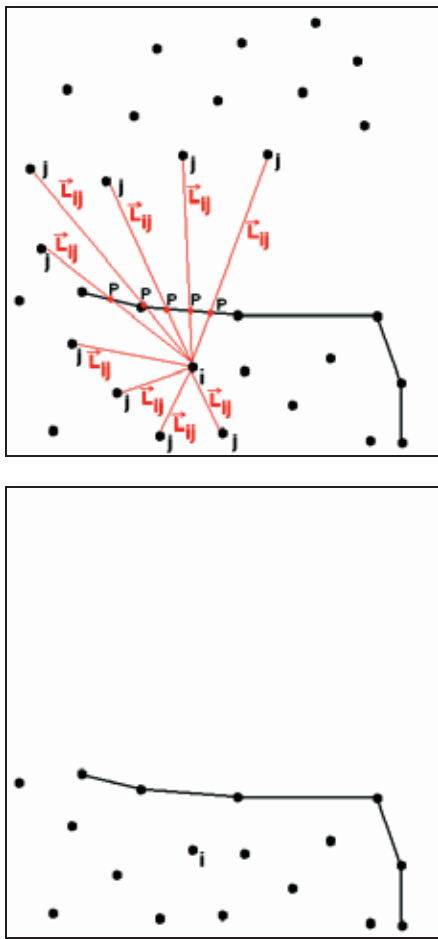


Figura 8. Proceso de eliminación de nodos inválidos de la nube del nodo estrella i .

EJEMPLO DE GENERACIÓN DE NUBES

A continuación se ilustrará un ejemplo para visualizar el tipo de sub-dominio generado con el algoritmo propuesto.

Ménsula

Este elemento tridimensional fue utilizado como ejemplo de aplicación del generador de nubes en el espacio. La discretización fue llevada a cabo mediante 4.473 nodos, como se ilustra en la figura 9. La figura 10 muestra la nube correspondiente al nodo estrella 633, la cual también está ubicada en una zona cóncava de la pieza. Como se puede observar, la nube generada muestra un buen comportamiento en cuanto a presencia de puntos inválidos.

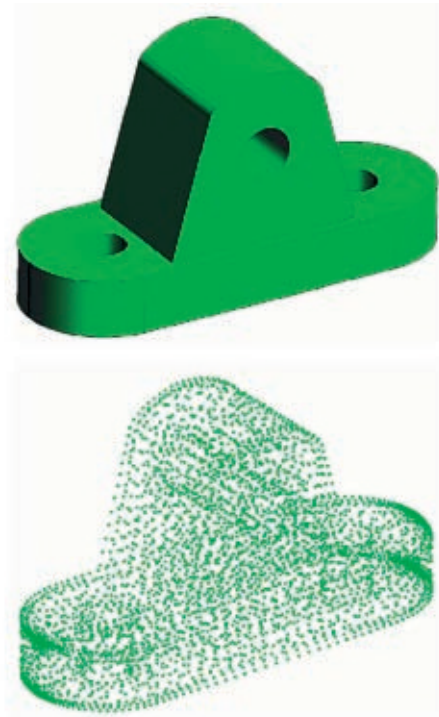


Figura 9. Ménsula. Modelo geométrico y discretización en puntos.

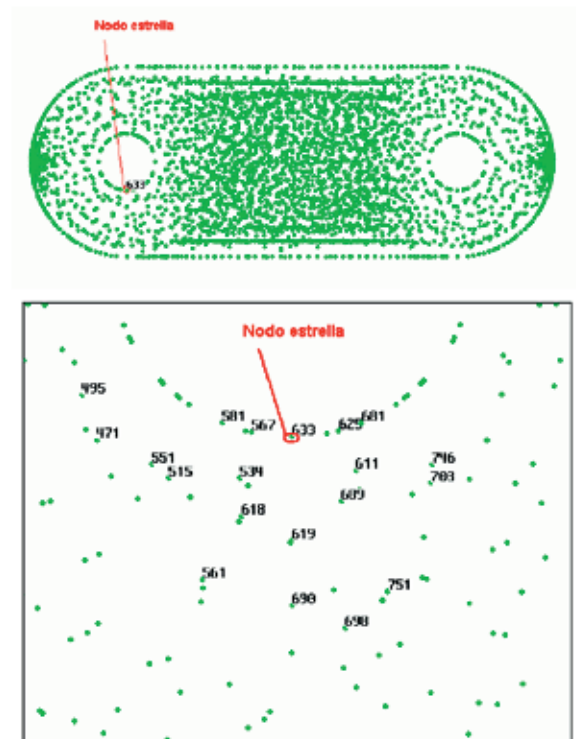


Figura 10. Nube del nodo estrella 633. Vista alzada de la geometría.

EJEMPLOS NUMÉRICOS

Se desarrollaron dos ejemplos numéricos para validar las nubes generadas con el algoritmo propuesto. Estos ejemplos fueron analizados mediante un código sin malla (MPF), el cual recibe las nubes generadas para luego ensamblar y resolver el sistema de ecuaciones de elasticidad.

Cilindro empotrado en tracción simple

Los datos de este ejemplo son los que se ilustran en la figura 11. Según esto, el máximo desplazamiento en la dirección del esfuerzo se determina con

$$\delta = \frac{4 * L * P}{E * \pi * d^2} \quad (12)$$

Reemplazando los valores mostrados en la figura 11, se tiene que

$$\delta = \frac{4 * 0.3 * 10^6}{2.07e11 * \pi * 0.1^2} = 1.8453 * 10^{-04} [m] \quad (13)$$

Al modelar esta geometría con tan sólo 738 puntos y posteriormente generar las nubes, se obtienen los resultados ilustrados en las figuras 12 y 13.

Se puede verificar que el error cometido respecto al máximo desplazamiento en la dirección del esfuerzo, es de 0,42%, mientras que los esfuerzos en el eje z se presentan constantes. El esfuerzo en el eje z corresponde a un valor constante e igual a $1.273 * 10^{08}$ [Pa].

Ménsula

Esta pieza mecánica se analizó con un esfuerzo en dirección longitudinal en el orificio central. También se prescribe un desplazamiento nulo, para poder simular presencia de pernos sujetadores en los lugares correspondientes. Esta condición, y la discretización en triángulos y puntos de la geometría, se muestran en la figura 14.

Para este ejemplo, la ménsula se discretizó en 7.697 puntos.

Los resultados numéricos obtenidos se ilustran en las figuras 15-19. En la figura 15 se puede contrastar la solución en cuanto a desplazamiento, en la dirección del esfuerzo aplicado, cuando se utiliza el MPF y el MEF. En las figuras 16-19, se muestran los contornos de desplazamiento y los esfuerzos, además de la geometría deformada respectivamente.

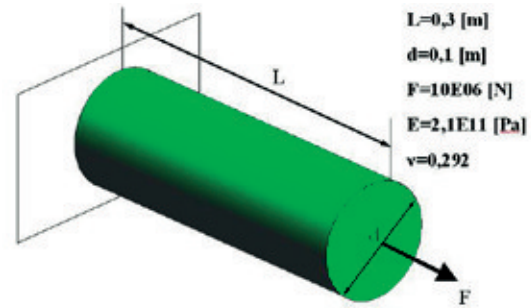


Figura 11. Cilindro empotrado en tracción simple. Datos geométricos, material y condiciones de carga.

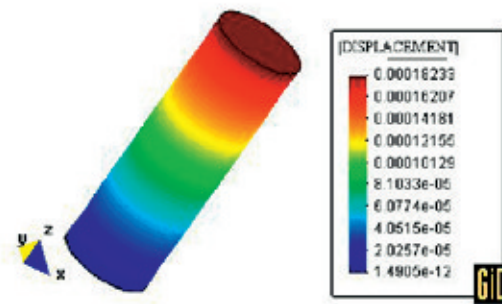


Figura 12. Cilindro empotrado en tracción: contornos de desplazamiento en la dirección del esfuerzo aplicado.

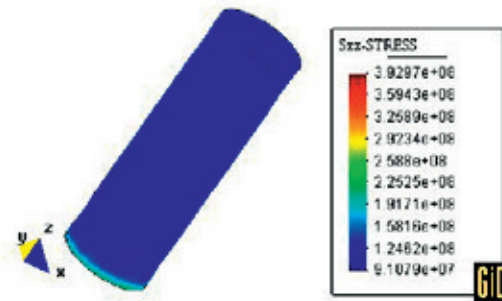


Figura 13. Cilindro empotrado en tracción: contornos de esfuerzo en la dirección del esfuerzo aplicado

Otro aspecto interesante por investigar corresponde a la calidad de las nubes generadas en los contornos de Neumann, ya que es en estos puntos donde se concentran los mayores porcentajes del error, de la solución numérica que se obtiene con el MPF.

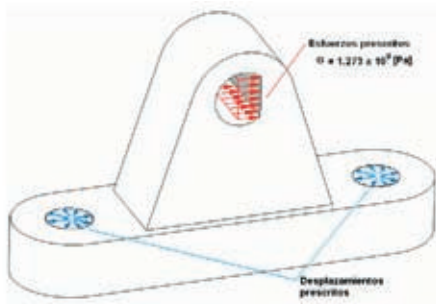


Figura 14. Ménsula: Condiciones de contorno y discretización en triángulos y puntos.

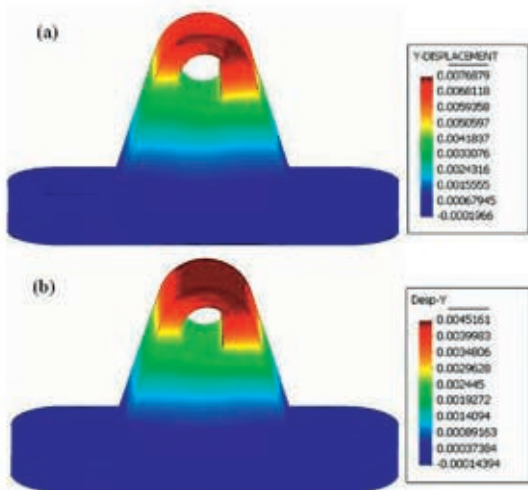


Figura 15. Comparación en contornos de desplazamiento en la dirección del esfuerzo aplicado, entre a) MPF y b) MEF.

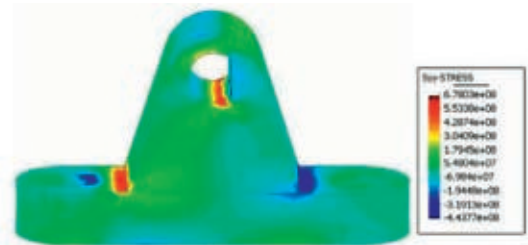


Figura 16. Contornos de esfuerzo en dirección del esfuerzo aplicado, obtenidos con MPF.

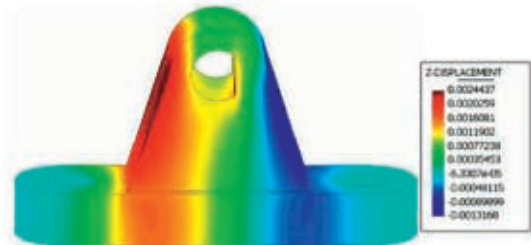


Figura 17. Contornos de desplazamiento en dirección transversal a la dirección del esfuerzo aplicado, obtenidos con MPF.



Figura 18. Geometría deformada (MPF) Discretización en puntos.

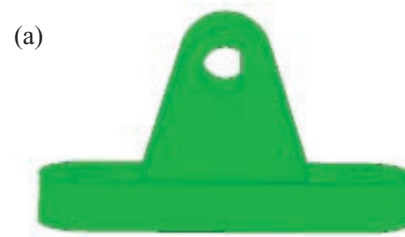


Figura 19. Proceso de deformación de la pieza (MPF). (Continúa).



Figura 19. Proceso de deformación de la pieza (MPF).

CONCLUSIONES

En cuanto a la generación de sub-dominios de interpolación o nubes, se puede observar que para geometrías reales y complejas, el algoritmo propuesto cumple eficazmente los requerimientos de nubes sin presencia de puntos inválidos.

Mediante ejemplos geométricos se comprobaron las virtudes del código propuesto para la generación de nubes, en especial en piezas sólidas con particularidades geométricas cóncavas. Considerando que las nubes son generadas para problemas en 3D, y que además presentan un correcto comportamiento geométrico, se visualiza como futura línea de investigación, un estudio sobre la utilización eficiente de las estructuras de datos para la generación de nubes.

AGRADECIMIENTOS

Los autores agradecen el soporte entregado para la realización del presente trabajo a CONICYT mediante el proyecto Fondecyt 1040371 y a la DGIP de la UTFSM a través del proyecto USM 250521.

De la misma forma los autores agradecen al Centro Internacional de Métodos Numéricos en Ingeniería (CIMNE), de Barcelona, España, la utilización del Pre y Post procesador GID™.

REFERENCIAS

- [1] E. Oñate, S. Idelsohn and O.C. Zienkiewicz. "Finite point method in computational mechanics". Research Report. N° 67. CIMNE. Barcelona, España. 1995.
- [2] E. Oñate, S. Idelsohn, O.C. Zienkiewicz and T. Fisher. "A finite point method for analysis of fluid flow problems". Proceedings of the 9th Int. Conference on Finite Element Methods in Fluids. Venize, Italy, pp. 15-21. 1995.
- [3] E. Oñate and S. Idelsohn. "A mesh free finite point method for advective-diffusive transport and fluid flow problems". Computational Mechanics. N° 21, pp. 283-292. 1998.
- [4] E. Oñate, F. Perazzo and J. Miquel. "Advances in stabilized finite point method for structural mechanics". Report N° 164 CIMNE. Barcelona, España. 1999.
- [5] E. Oñate, F. Perazzo and J. Miquel. "A finite point method for elasticity problems". Computers & Structures. Vol. 79, pp. 2151-2163. 2001.
- [6] E. Oñate and S. Idelsohn. "A finite point method for incompressible flow problems". Computing and Visualization in Science. 2000.
- [7] F. Perazzo, J. Miquel and E. Oñate. "El método de puntos finitos para problemas de la dinámica de sólidos". Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería. Vol. 20 N° 3. Universitat Politècnica de Catalunya. Barcelona, España. 2004.
- [8] F. Perazzo. "Una metodología numérica sin malla para la resolución de las ecuaciones de elasticidad, mediante el método de Puntos Finitos". Tesis para optar al grado de doctor. Universitat Politècnica de Catalunya. Barcelona, España. 2002.
- [9] F. Perazzo, S. Oller, J. Miquel and E. Oñate. "Avances en el método de Puntos Finitos para la mecánica de sólidos". Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería. Universitat Politècnica de Catalunya. Vol. 22 N° 2, pp. 153-167. Barcelona, España. 2006.
- [10] N. Ipinza. "Desarrollo e implementación de la capacidad de modelar geometrías 3D en el método sin malla de Puntos Finitos". Tesis para optar al título de Ingeniero Mecánico. Departamento de Mecánica. Universidad Técnica Federico Santa María. Santiago, Chile. 2005.
- [11] N. Perrone and R. Kao. "A general finite difference method for arbitrary meshes". Computers & Structures. Vol. 5, pp. 45-58. 1975.

- [12] P.S. Jensen. "Finite difference technique for variable grids". *Computers & Structures*. Vol. 2, pp. 17-29. 1972.
- [13] R. Löhner. "Some Useful Data Structure for the Generation of Unstructured Grids". *Communications and Applied Numerical Methods*. 1988.
- [14] R. Löhner, C. Sacco, E. Oñate and S. Idelsohn. "A finite point method for compressible flow". *International Journal for Numerical Methods in Engineering*. Vol. 53, pp. 1765-1779. 2002.
- [15] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl, "Meshless Method: An overview and recent developments". *Computer Method in Applied Mechanics and Engineering*. Vol. 139, pp. 3-48. 1996.
- [16] T. Liszka, C.A. Duarte and W. Tworzydło. "Hp-Meshless cloud method". *Computers Methods in Applied Mechanics and Engineering*. Vol. 139, pp. 263-288. 1996.
- [17] T. Liszka and J. Orkisz. "The finite difference method at arbitrary irregular grids and its application in applied mechanics". *Computers & Structures*. Vol. 11, pp. 83-95. 1980.
- [18] O. Zienkiewicz y R. Taylor. "El método de los elementos finitos". *Las bases*. Quinta edición. Vol. 1. 2000.
- [19] L. Lucy. "A numerical approach to the testing of the fission hypothesis". *The Astromical Journal*. Vol. 82 N° 12, pp. 1013-1024. 1977.
- [20] N. Weatherill, K. Morgan and O. Hassan. "An introduction to mesh generation". *John Wiley & Sons*. 2002.