

IMPLEMENTACIÓN DE UNA FORMULACIÓN RÁPIDA DE MULTIPOLOS APLICADOS AL MÉTODO DE LOS ELEMENTOS DE CONTORNO

Marco Dondero^a, Adrián P. Cisilino^a y Georgios Stavroulakis^b

^aDivisión Soldadura y Fractomecánica – INTEMA, Facultad de Ingeniería
Universidad Nacional de Mar del Plata, Mar del Plata, Argentina mdondero@fi.mdp.edu.ar,
cisilino@fi.mdp.edu.ar, <http://www.intema.gov.ar>

^bDepartamento de Producción Ingeniería y Gerencia, Universidad Técnica de Creta
Chania, Creta, Grecia, gestavr@dpem.tuc.gr, <http://users.isc.tuc.gr/~gestavroulakis/>

Keywords: Fast Multipole Method, Boundary Element Method, variable compleja,
bidimensional, ecuación de Laplace.

Resumen. Se presenta en este trabajo una descripción completa del algoritmo de Multipolos Rápidos combinado con el Método de los Elementos de Contorno (FMBEM, *en inglés*) y su implementación para un problema de potencial estacionario. Se usa la alternativa de una formulación usando variable compleja. El algoritmo implementado es luego validado y calibrado para resolver problemas que consisten en el estudio del campo de temperaturas en un sólido bidimensional con múltiples agujeros. La validación se realiza comparando los resultados de la solución por FMBEM con los obtenidos utilizando una formulación directa de BEM para el mismo problema. El proceso de calibración comprende la determinación de los parámetros cantidad de elementos por celda, número de términos en las expansiones y tolerancia del resolvidor iterativo. El primer parámetro involucra directamente al tamaño de la celda y la cantidad de vecinos cercanos que un elemento posee; este mostró ser el más sensible con la calidad y el tiempo de solución.

1 INTRODUCCIÓN

El Fast Multipole Method (FMM) está considerado entre los 10 algoritmos más importantes desarrollados durante el siglo XX. Fue desarrollado por [Rokhlin y Greengard \(1985, 1987, 1988\)](#) a mediados de los 80s, y tomó casi una década a la comunidad científica descubrir su potencial para combinarlos con el Método de los Elementos de Contorno (BEM). El FMM combinado con el BEM permite resolver problemas de varios millones de incógnitas en horas en una computadora de escritorio. Esto abre el camino hacia más aplicaciones obstaculizadas por la falta de eficiencia en el proceso de solución.

La clave del FM-BEM es la reducción del costo computacional del cálculo de los coeficientes de la matriz densa (costo cuadrático) y su resolución (costo cúbico en el caso de revolvedores lineales) a cuasi-lineal. Esta reducción se logra agrupando los elementos de contorno en celdas y expandiendo las soluciones fundamentales en series multipolo ([Liu, 2006](#) y [Nishimura 2002](#)). Las celdas poseen una estructura jerárquica tipo árbol en la que las divisiones más pequeñas (hojas) contienen un número de elementos prescripto. De esta forma se logra traducir las interacciones nodo-nodo (o elemento-elemento) a celda-celda, con el producto matriz por vector calculado utilizando las expansiones multipolo. Finalmente se utiliza un resolvidor iterativo (GMRES) que prescinde del almacenamiento de la matriz de coeficientes (matrix free).

2 EL MÉTODO DE LOS ELEMENTOS DE CONTORNO DIRECTO

Se describe a continuación brevemente el método de elementos de contorno para problemas de potencial en dos dimensiones en régimen estacionario gobernado por la ecuación de Laplace

$$\nabla^2 \phi(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in V; \quad (1)$$

bajo las condiciones de contorno

$$\begin{aligned} \phi(\mathbf{x}) &= \bar{\phi}(\mathbf{x}), \quad \forall \mathbf{x} \in S_1; \\ q(\mathbf{x}) &\equiv \frac{\partial \phi}{\partial n}(\mathbf{x}) = \bar{q}(\mathbf{x}), \quad \forall \mathbf{x} \in S_2; \end{aligned} \quad (2)$$

donde ϕ es el campo de potencial en el dominio V , $S = S_1 \cup S_2$ el contorno de V , n la normal exterior y las cantidades con barra son valores conocidos, ver [Figura 1](#).

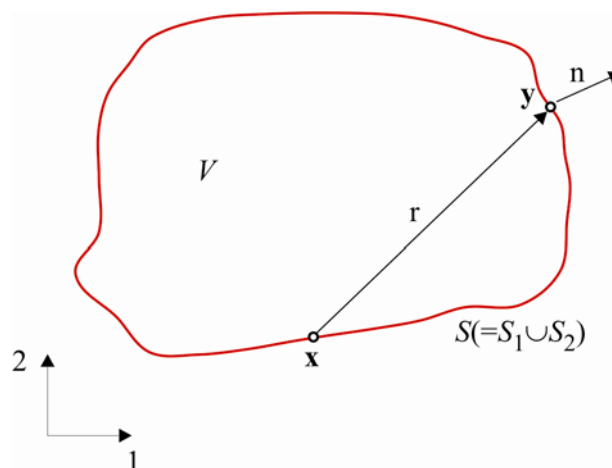


Figura 1: Dominio V , contorno S y variables del sistema.

La solución se puede escribir como

$$\phi(\mathbf{x}) = \int_S [G(\mathbf{x}, \mathbf{y})q(\mathbf{y}) - F(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})] dS(\mathbf{y}), \quad \forall \mathbf{x} \in V; \quad (3)$$

donde $G(\mathbf{x}, \mathbf{y})$ y $F(\mathbf{x}, \mathbf{y})$ son las funciones de Green para problemas en 2D

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi} \ln\left(\frac{1}{r}\right), \quad y \quad (4)$$

$$F(\mathbf{x}, \mathbf{y}) = \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} = \frac{1}{2\pi r} \frac{\partial r}{\partial n}. \quad (5)$$

Haciendo que los puntos de colocación \mathbf{x} pertenezcan al contorno S (Figura 1), se llega a la formulación clásica de BEM (Brebbia 1992), conocida también como BIE por su denominación en inglés *boundary integral equation*

$$C(\mathbf{x})\phi(\mathbf{x}) = \int_S [G(\mathbf{x}, \mathbf{y})q(\mathbf{y}) - F(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})] dS(\mathbf{y}), \quad \forall \mathbf{x} \in S. \quad (6)$$

Si la superficie es suave en el punto de colocación \mathbf{x} , el coeficiente $C(\mathbf{x})=1/2$.

Se discretiza el contorno S utilizando N elementos constantes dados por segmentos rectos con un nodo en el centro (Figura 2).

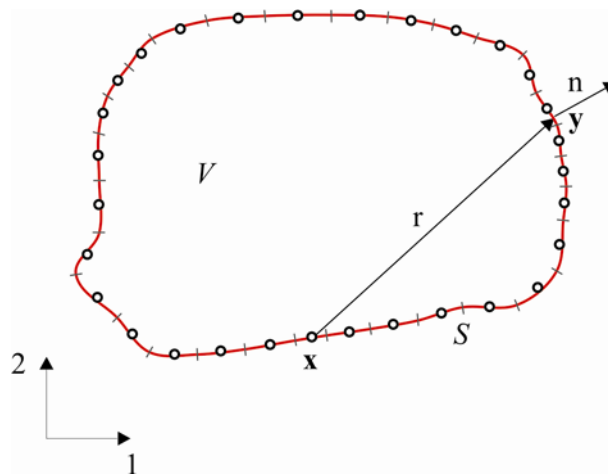


Figura 2: Discretización del contorno S usando elementos constantes.

Situando el punto de colocación \mathbf{x} en el nodo i , se obtiene la forma discreta de BIE a partir de la ecuación (6). De esta forma resulta

$$\frac{1}{2}\phi_i = \sum_{j=1}^N [g_{ij}q_j - f_{ij}\phi_j], \quad \text{para } i=1, 2, \dots, N; \quad (7)$$

$$g_{ij} = \int_{\Delta S_j} G(\mathbf{x}, \mathbf{y}) dS(\mathbf{y}), \quad f_{ij} = \int_{\Delta S_j} F(\mathbf{x}, \mathbf{y}) dS(\mathbf{y}), \quad \text{para } i, j=1, 2, \dots, N; \quad (8)$$

donde ϕ_j y q_j ($j=1,2,\dots,N$) son los valores nodales de ϕ y q en el elemento ΔS_j (Figura 2), respectivamente.

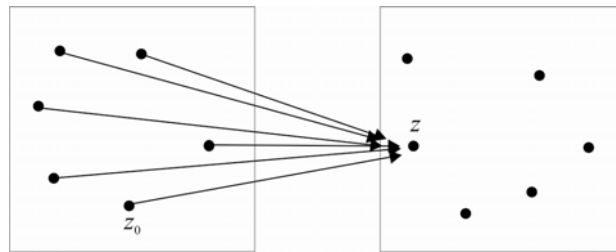
Aplicando las condiciones de contorno (2) en cada nodo al sistema de ecuaciones anterior, y reordenando luego sus columnas para agrupar las incógnitas del problema se llega a

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{b}, \quad (9)$$

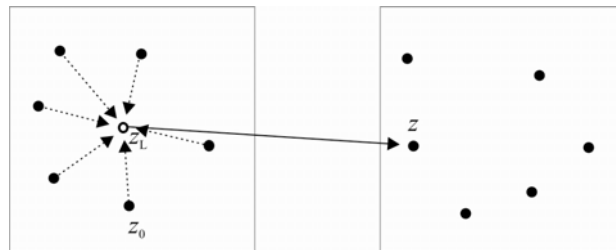
donde \mathbf{A} es la matriz de coeficientes, $\boldsymbol{\lambda}$ el vector que agrupa los valores de ϕ y q desconocidos y \mathbf{b} el vector de valores conocidos. La construcción de la matriz \mathbf{A} requiere operaciones de orden cuadrático $O(N^2)$ y su espacio de almacenamiento es también $O(N^2)$ dado que es una matriz llena. La solución del sistema de ecuaciones (7) utilizando eliminación de Gauss utiliza operaciones de $O(N^3)$. Es por esto el BEM directo, en este enfoque convencional, es en general lento e ineficiente para problemas de gran escala.

3 EL FAST MULTIPOLE METHOD (FMM)

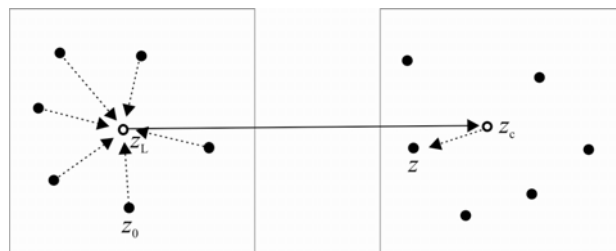
El método FMM propuesto por Rokhlin (1985) otorga una técnica para calcular rápidamente el producto matriz vector mencionado en el punto anterior con operaciones de $O(N \log(N))$ de acuerdo a Aluru (1996). La idea: traducir las interacciones nodo-nodo (o elemento-elemento) a celda-celda, como se ve en la Figura 3; es decir reemplazar las interacciones directas $\mathbf{z}_0 - \mathbf{z}$ por las expandidas con el método FMM. Las celdas poseen estructura jerárquica (árbol) donde las más pequeñas (hojas) poseen un número especificado de elementos. En FMM el sistema de ecuaciones se resuelve mediante métodos iterativos como el GMRES donde se utilizan las expansiones multipolo para calcular el producto del lado izquierdo de (9), es decir el producto matriz por vector prueba (la posible solución).



Evaluación convencional de la contribución de las partículas distantes.



Los *momentos multipolo* (líneas punteadas) representan a las partículas distantes.



Evaluación utilizando momentos multipolo y expansión local (a la derecha).

Figura 3: Aceleración mediante *momentos* entre partículas distantes (izquierda) y partículas locales (derecha).

3.1 Formulación del BEM utilizando notación compleja

Continuando con el ejemplo de potencial en dos dimensiones, se describe en lo que sigue las expansiones necesarias para el FMM. Se usará notación compleja por conveniencia, \mathbf{x} e \mathbf{y} se reemplazan por $z_0 = x_1 + \mathbf{i}x_2$ y $z = y_1 + \mathbf{i}y_2$ ($\mathbf{i} = \sqrt{-1}$) respectivamente (Figura 4).

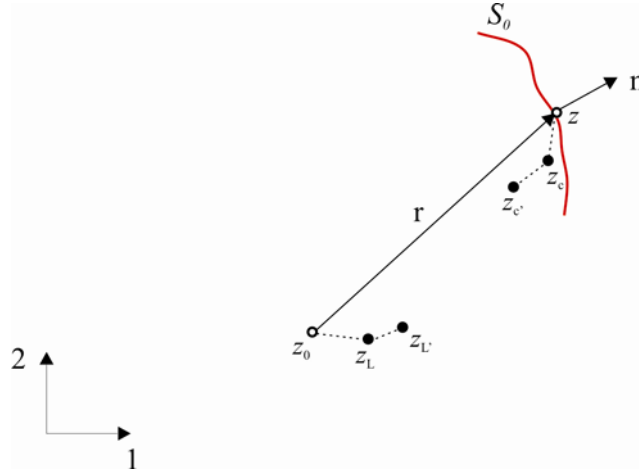


Figura 4: Notación compleja y los puntos de las expansiones multipolo.

Considerar primero, la siguiente integral del *kernel* G de BIE (6)

$$\int_{S_0} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) dS(\mathbf{y}), \quad (10)$$

donde S_0 es un subgrupo de S y está lejos del punto de colocación \mathbf{x} (Figura 4).

Introduciendo ahora la notación compleja se escribe

$$G(\mathbf{x}, \mathbf{y}) = \text{Re}\{G(z_0, z)\}, \quad (11)$$

donde

$$G(z_0, z) = -\frac{1}{2\pi} \ln(z_0 - z). \quad (12)$$

Así, la integral en (10) es la parte real de

$$\int_{S_0} G(z_0, z) q(z) dS(z). \quad (13)$$

Análogamente se escribe en forma compleja la integral del *kernel* F de BIE (6)

$$\int_{S_0} F(z_0, z) \phi(z) dS(z), \quad (14)$$

donde

$$F(z_0, z) = \frac{\partial G}{\partial n} = (n_1 + in_2)G' = n(z)G' \quad \text{y} \quad G' \equiv \frac{\partial G}{\partial z}. \quad (15)$$

De esta manera el *kernel* $F(\mathbf{x}, \mathbf{y})$ es la parte real de $\text{Re}\{F(z_0, z)\}$ en notación compleja.

3.2 Conceptos del Fast Multipole Method

Se describen ahora varios conceptos importantes para el método FMBEM:

- *Expansión Multipolo (momentos)*

Siendo z_c un punto cercano al punto de campo z , es decir $|z - z_c| \ll |z_0 - z_c|$, se escribe

$$G(z_0, z) = -\frac{1}{2\pi} \ln(z_0 - z) = -\frac{1}{2\pi} \left[\ln(z_0 - z_c) + \ln\left(1 - \frac{z - z_c}{z_0 - z_c}\right) \right]. \quad (16)$$

Aplicando la siguiente serie de Taylor, $\ln(1 - \zeta) = -\sum_{k=1}^{\infty} \frac{\zeta^k}{k}$, para $|\zeta| < 1$, se obtiene

$$G(z_0, z) = -\frac{1}{2\pi} \sum_{k=0}^{\infty} O_k(z_0 - z_c) I_k(z - z_c), \quad (17)$$

donde $I_k(z) = \frac{z^k}{k!}$, para $k \geq 0$; $O_k(z) = \frac{(k-1)!}{z^k}$, para $k \geq 1$; y $O_0(z) = -\ln(z)$. Tener en cuenta que las derivadas de $I_k(z)$ y $O_k(z)$ satisfacen $I'_k(z) = I_{k-1}(z)$, para $k \geq 1$; y $O'_k(z) = -O_{k+1}(z)$, para $k \geq 0$.

Ahora, la integral de G en (13) se evalúa como

$$\begin{aligned} \int_{S_0} G(z_0, z) q(z) dS(z) &= \frac{1}{2\pi} \int_{S_0} \left[\sum_{k=0}^{\infty} O_k(z_0 - z_c) I_k(z - z_c) \right] q(z) dS(z) \\ &= \frac{1}{2\pi} \sum_{k=0}^{\infty} O_k(z_0 - z_c) M_k(z_c) \end{aligned}, \quad (18)$$

donde

$$M_k(z_c) = \int_{S_0} I_k(z - z_c) q(z) dS(z), \quad (19)$$

son los *momentos* alrededor de z_c , independientes del punto de colocación z_0 y solo se necesitan calcular una sola vez.

Considerando al *kernel* F , recordando la definición en (15) y usando (17) se obtiene

$$G' = \frac{1}{2\pi} \sum_{k=1}^{\infty} O_k(z_0 - z_c) I_{k-1}(z - z_c), \quad (20)$$

y la integral en (14) queda dada por

$$\int_{S_0} F(z_0, z) \phi(z) dS(z) = \frac{1}{2\pi} \sum_{k=1}^{\infty} O_k(z_0 - z_c) N_k(z_c), \quad (21)$$

donde

$$N_k(z_c) = \int_{S_0} n(z) I_{k-1}(z - z_c) \phi(z) dS(z). \quad (22)$$

Éstos son los *momentos* de la integral del *kernel* F , similares a los (19) para la integral del *kernel* G . Notar la diferencia en los momentos N , ya que $N_0=0$.

- *Traslación moment-to-moment (M2M)*

Si el punto z_c es movido a $z_{c'}$ (Figura 4) se aplica la siguiente traslación

$$M_k(z_{c'}) = \int_{S_0} I_k(z - z_{c'}) q(z) dS(z) = \int_{S_0} I_k((z - z_c) + (z_c - z_{c'})) q(z) dS(z). \quad (23)$$

Usando la formula binomial

$$(a+b)^n = \sum_{m=0}^n \binom{n}{m} a^m b^{n-m}, \quad \text{donde } \binom{n}{m} = \frac{n!}{(n-m)!m!} = \binom{n}{n-m}, \quad (24)$$

se llega a

$$M_k(z_{c'}) = \sum_{l=0}^k I_{k-l}(z_c - z_{c'}) M_l(z_c). \quad (25)$$

En esta *traslación M2M* hay un número finito de términos y se aplica tanto a las integrales del *kernel* G como a las del *kernel* F .

- *Expansión Local y Traslación moment-to-local (M2L)*

Sea el punto z_L cercano al punto de colocación z_0 (Figura 4), es decir $|z_0 - z_L| \ll |z_c - z_L|$. De la *expansión multipolo* (18), se tiene

$$f(z_0) \equiv \int_{S_0} G(z_0, z) q(z) dS(z) = \frac{1}{2\pi} \sum_{k=0}^{\infty} O_k(z_0 - z_c) M_k(z_c). \quad (26)$$

Para obtener la expresión de la expansión local se expande $f(z_0)$ alrededor de z_L usando series de Taylor

$$f(z_0) = \sum_{l=0}^{\infty} f^{(l)}(z_L) I_l(z_0 - z_L). \quad (27)$$

Ahora evaluando la (26) en z_L

$$f^{(l)}(z_L) = \frac{(-1)^l}{2\pi} \sum_{k=0}^{\infty} O_{l+k}(z_L - z_c) M_k(z_c), \quad (28)$$

y reemplazando en (27), se obtiene la *expansión local*

$$\int_{S_0} G(z_0, z) q(z) dS(z) = \sum_{l=0}^{\infty} L_l(z_L) I_l(z_0 - z_L), \quad (29)$$

donde los coeficientes están dados por la siguiente *traslación M2L*:

$$L_l(z_L) = \frac{(-1)^l}{2\pi} \sum_{k=0}^{\infty} O_{l+k}(z_L - z_c) M_k(z_c). \quad (30)$$

Esta *traslación M2L* se aplica también a las integrales del *kernel F*.

- *Traslación Local-to-local (L2L)*

Si el punto z_L es movido a $z_{L'}$ (Figura 4), se obtiene la siguiente expresión usando la *local expansion* (29) anterior:

$$\int_{S_0} G(z_0, z) q(z) dS(z) = \sum_{l=0}^{\infty} L_l(z_L) I_l(z_0 - z_L) = \sum_{l=0}^{\infty} L_l(z_L) I_l((z_0 - z_{L'}) + (z_{L'} - z_L)). \quad (31)$$

Aplicando la fórmula binomial (24), se obtiene

$$\int_{S_0} G(z_0, z) q(z) dS(z) = \sum_{l=0}^{\infty} L_l(z_{L'}) I_l(z_0 - z_{L'}), \quad (32)$$

donde los coeficientes están dados por la *traslación L2L*:

$$L_l(z_{L'}) = \sum_{m=l}^{\infty} I_{m-l}(z_{L'} - z_L) L_m(z_L). \quad (33)$$

La *traslación L2L* se aplica también a las integrales del *kernel F*.

3.3 Algoritmos y procedimientos del FMM

Paso 1. Discretización.

Discretizar la frontera S de la misma manera que en BEM convencional usando elementos constantes, ver [Figura 2](#).

Paso 2. Determinar la estructura de árbol de la malla.

- Construir un cuadrado que circunscriba el contorno S . Este cuadrado es la celda de nivel $l=0$ que contiene a todos los elementos de la malla.

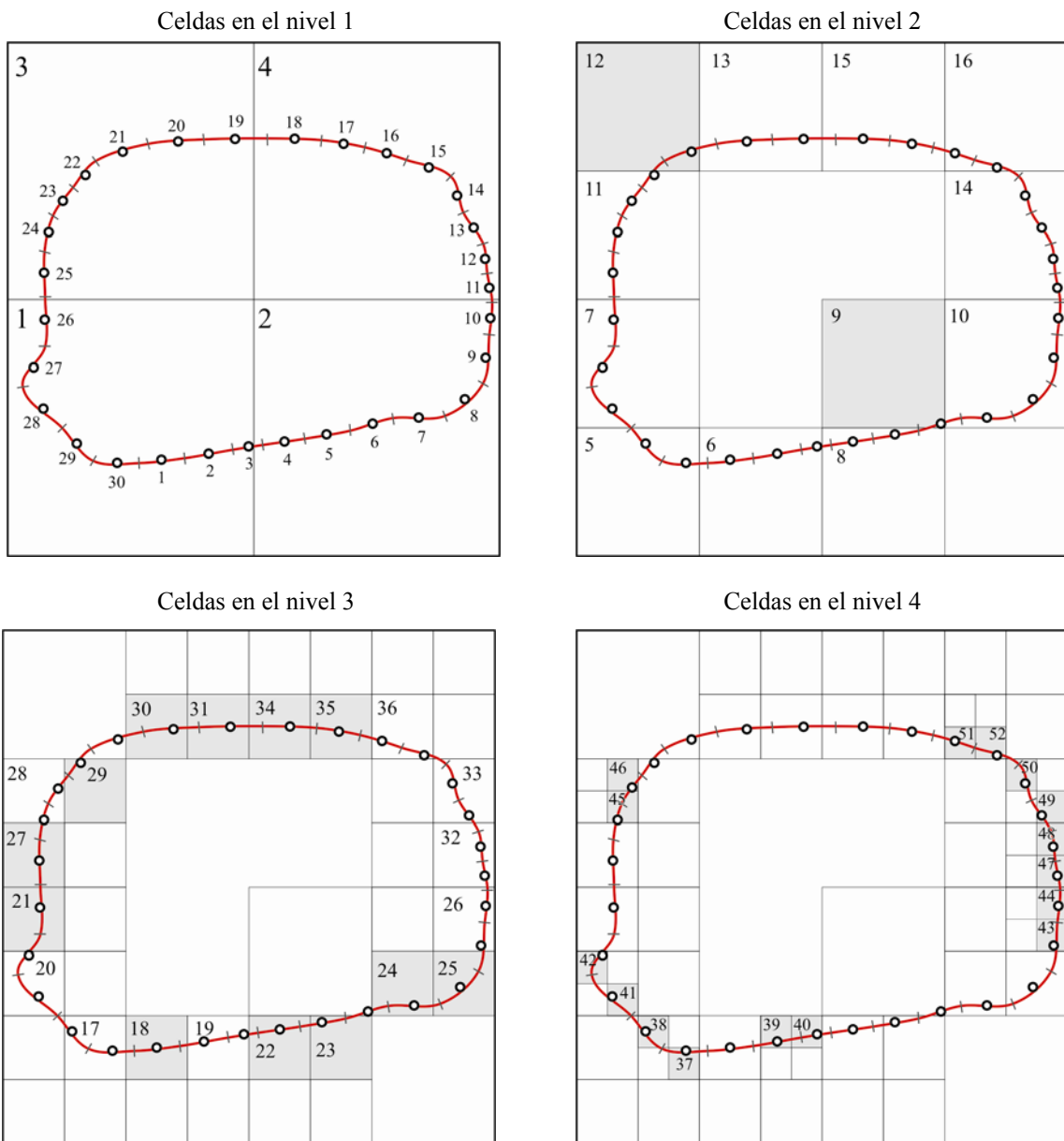


Figura 5: Construcción de la estructura tipo árbol (Paso 2). Numeración correlativa de las celdas comenzando desde la celda 0 en el nivel 0. Las *hojas* están sombreadas en gris.

- Dividir la celda de nivel $l=0$ (celda *padre*) en 4 cuadrados iguales (cuyo lado mide la

mitad del de la celda padre). Cuando una de estos nuevos cuadrados contiene al menos un elemento de la malla se lo considera ‘una celda’ (celda *hijo*) de nivel $l=1$. Se dice que un elemento de contorno pertenece a una celda C si su nodo está en C . (ver Figura 5)

- Continuar con el proceso, dividiendo ahora las celdas de nivel l para obtener las celdas de nivel $l=2$, y así sucesivamente hasta que la cantidad de elementos de contorno pertenecientes a una celda alcance el mínimo especificado.

Una celda sin hijos (es decir aquella que contiene la cantidad de elementos mínima especificada) se llama *hoja* (sombreadas en la Figura 5 y en la Figura 6). Usando este procedimiento, ilustrado paso a paso en la Figura 5 para el caso en que se especificó la cantidad mínima de elementos por celda igual a uno, se obtiene la estructura de árbol que se ilustra en la (Figura 6).

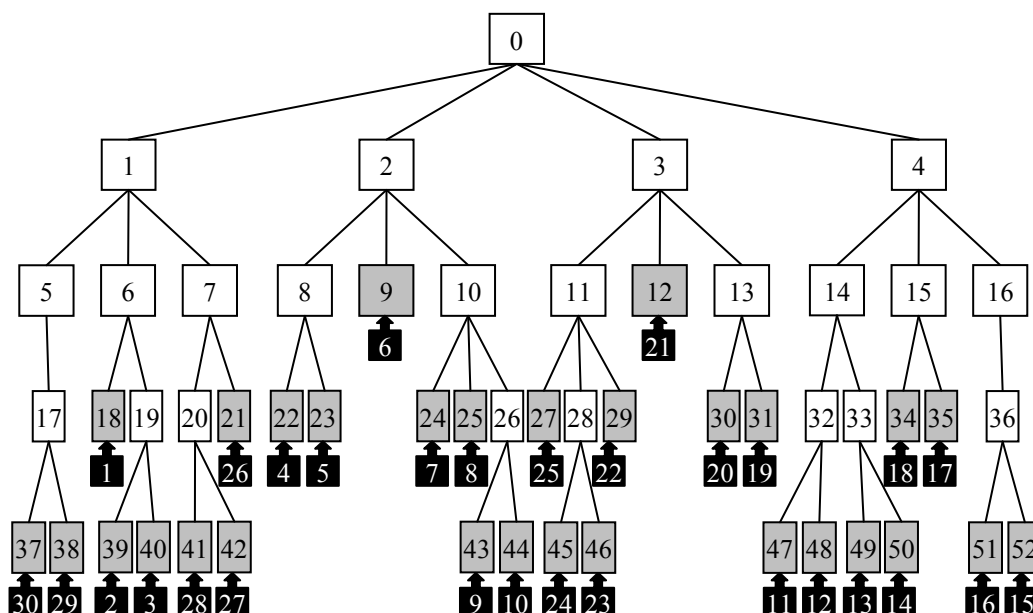


Figura 6: Árbol (*quad-tree*) con las celdas (cuadrados) y elementos del contorno (flechas negras). Las *hojas* son las celdas sombreadas en gris, y les corresponde un elemento a cada una.

Paso 3. Cálculo de los momentos multipolo (*Upward pass*)

Calcular los momentos multipolo para todas las celdas. El proceso comienza por las celdas C hojas usando (19) y (22) y considerando z_c como el centro de las celdas C (ver Figura 7a). Luego para las celdas C no-*hojas* del nivel l , los momentos multipolo se calculan sumando los momentos multipolo de sus hijos luego de aplicar la *traslación M2M* de (25). La *traslación M2M* traslada los momentos desde el centro del *hijo* de C al centro de C . Este procedimiento se repite a partir $l \geq 2$ incrementando el nivel l (ver Figura 7b).

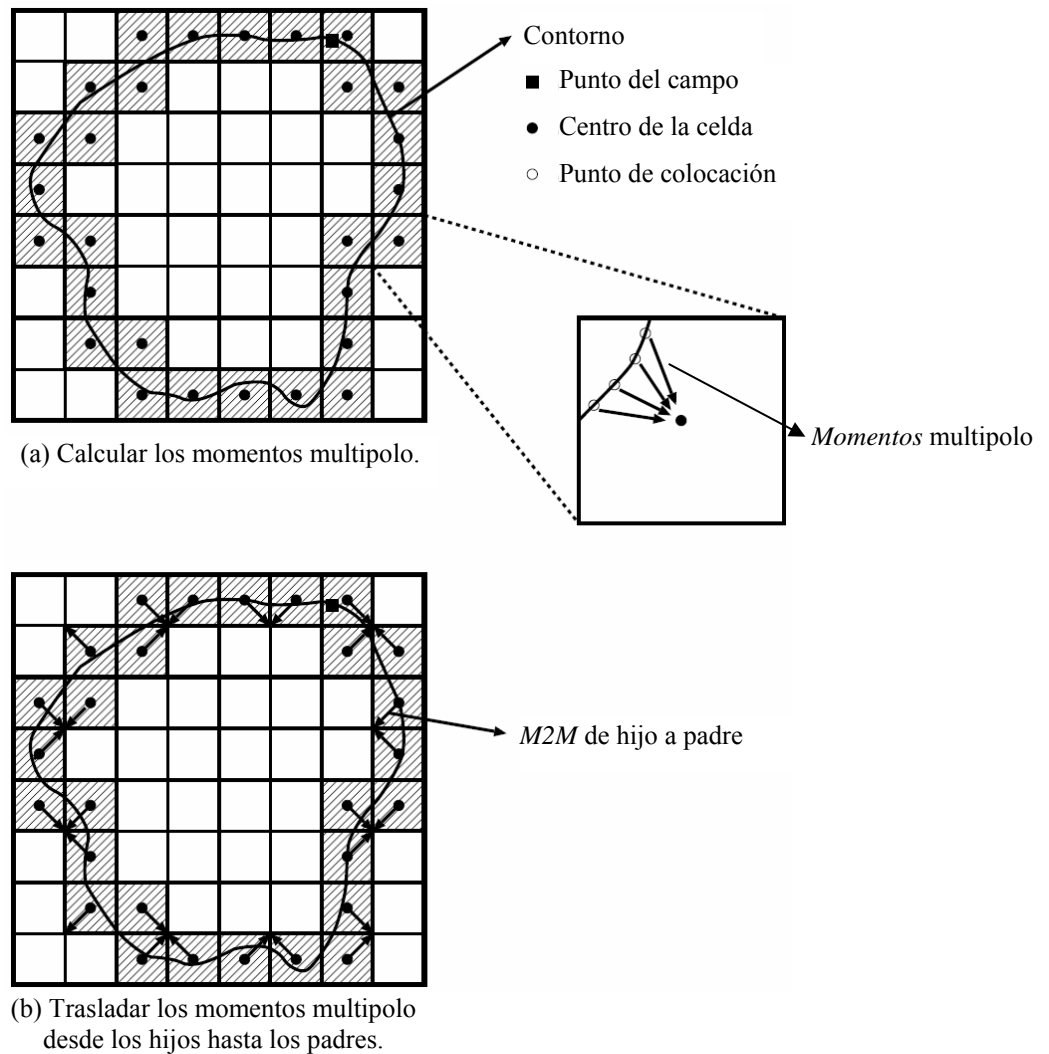


Figura 7: Momentos multipolo y traslación M2M (Paso 3).

Paso 4. Cálculo de los coeficientes de las expansiones locales (Downward pass)

Se definirán unos conceptos importantes primero. Se dice que dos celdas son “*adyacentes en el nivel l* ” si ambas están en el nivel l y comparten al menos un vértice. Dos celdas son “*bien separadas en el nivel l* ” si no son adyacentes en el nivel l , pero sus padres lo son en el nivel $l-1$. La lista de todas las celdas bien separadas de una celda C en el nivel l se denomina “*lista de interacción*” de C .

Ahora se calcula la expansión local asociada a una celda C . La misma representa la suma de: (a) las contribuciones de los elementos de contorno en las celdas de la lista de interacción de C , y (b) las contribuciones de todos los elementos de contorno en las celdas que no son adyacentes al padre de C . Las contribuciones (a) se calculan usando la *traslación M2L* (30) (ver [Figura 8a](#)) y las contribuciones (b) se hacen trasladando la expansión local desde el dentro del padre de C al centro de C usando la *traslación L2L* (33) (ver [Figura 8b](#)). Este proceso se repite comenzando desde $l=2$ y aumentando l hasta llegar a las hojas. En el nivel 2, no existen contribuciones (b) ya que no hay celdas bien separadas en el nivel 1.

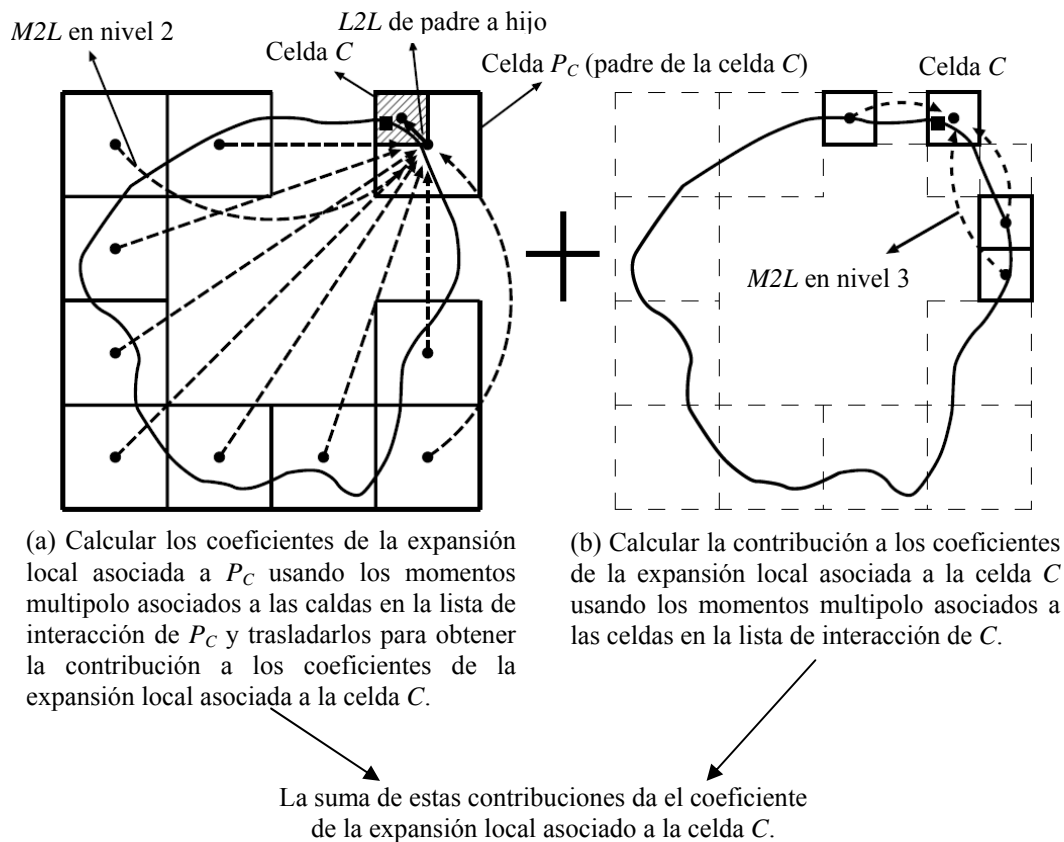


Figura 8: Traslaciones $M2L$ y $L2L$ (Paso 4).

Paso 5. Evaluación de las integrales en BIE.

Calcular el lado derecho de (6). Las contribuciones de los elementos que pertenecen a la celda que contiene al punto de colocación z_0 y la de los elementos que pertenecen a las celdas adyacentes a C se calculan como en BEM directo (ver Figura 9a). La contribución de todo el resto se hace usando la *expansión local* (29). Esto se realiza utilizando los coeficientes de la *expansión local* calculados en el paso 4 y cambiando en punto de expansión desde en centro de C al punto de colocación z_0 (ver Figura 9b).

Paso 6. Iteraciones de la solución.

Actualizar el vector desconocido en $\mathbf{A}\boldsymbol{\lambda}=\mathbf{b}$ correspondiente a la BIE (6) y continuar con el Paso 3 para la multiplicación matriz vector hasta que la solución converja dentro de una tolerancia dada.

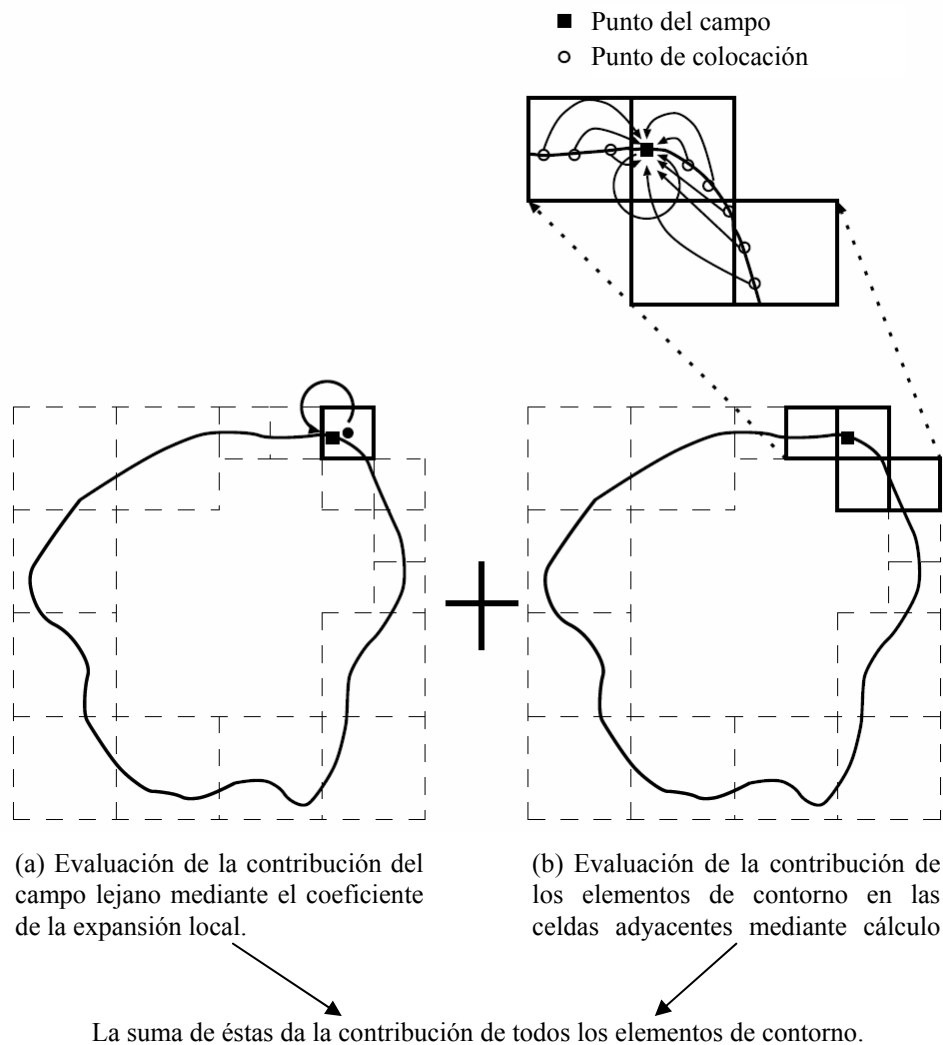


Figura 9: Evaluación de la contribución total (Paso 5).

3.4 Implementación

A continuación se describe la implementación actual del código en Fortran 90 para resolver problemas de potencial en 2D con Fast Multipole.

La discretización se hace usando elementos constantes. Las integrales no singulares, singulares o cercanas a singulares se realizan en forma analítica. El código no hace uso de integración numérica en ningún caso. El esquema de cálculo tipo *matrix-free* sigue el propuesto por de Liu y Nishimura (2006). El programa desarrollado utiliza un resolvidor iterativo GMRES cuyo código fuente se obtuvo de la página web *netlib* (<http://www.netlib.org/>). Para el esquema iterativo del resolvidor GMRES se proveen dos rutinas: una rutina para el producto matriz por vector *matvec* y otra *msolve* para calcular la matriz del pre-condicionador. La primera se realiza utilizando las expansiones multipolo ya descritas, donde se produce la reducción del tiempo de cálculo, y la última se calcula evaluando directamente las soluciones fundamentales entre todos los elementos dentro de cada hoja, generando así una matriz diagonal bloque. El pre-condicionador es de vital importancia para la convergencia del método.

4 RESULTADOS

Se desarrollo un código para problemas de potencial estacionario en dos dimensiones usando Fortran 90 y considerando su futura paralelización. Esta implementación es validada y calibrada usando un caso de aplicación que consisten en el estudio del campo de temperaturas en un sólido bidimensional con múltiples agujeros. La geometría del sólido es una placa cuadrada con un arreglo ordenado de 10 x 10 agujeros y condiciones de contorno 0°C y 100°C en sus lados inferior y superior respectivamente. El perfil térmico obtenido se puede ver en la [Figura 10](#).

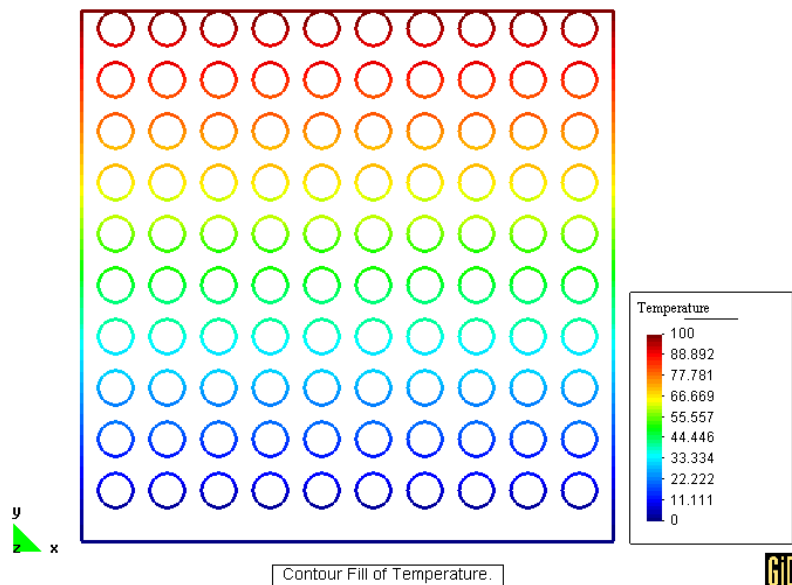


Figura 10: Perfil térmico obtenido para la placa con agujeros.

4.1 Validación

La validación se realiza comparando los resultados de la solución por FMBEM con los obtenidos utilizando una formulación directa de BEM para el mismo problema.

El primer paso consiste en determinar el número de elementos requeridos para el BEM directo en un estudio de convergencia ([Figura 11](#)), una malla de 4400 elementos resulta adecuada.

Se mide, luego, la calidad (menor energía almacenada) de la solución para FMBEM utilizando siempre la misma malla de 4400 elementos. Al aumentar el número de elementos por celda se llega a un valor idéntico de energía que el del método directo, como se ve en la [Figura 11](#).

4.2 Calibración

El proceso de calibración comprende la determinación de los parámetros cantidad de elementos por celda, número de términos en las expansiones y tolerancia del resolvidor iterativo. El primer parámetro involucra directamente al tamaño de la celda y la cantidad de vecinos cercanos que un elemento posee; este mostró ser el más sensible con la calidad y el tiempo de solución como puede verse en la [Figura 12](#). El valor del error (comparado con la solución por BEM directo en cada nodo) es la media de toda la placa y las barras de error representan la dispersión con respecto a la media. Para más de 10 elementos por celda el error

relativo se reduce notablemente y su dispersión casi desaparece. Se observa un incremento en el tiempo de CPU al pasar de 200 elementos por celda ya que el número de vecinos cercanos (donde se utiliza integración directa como en BEM) es tan grande que el método se convierte en un BEM directo (imaginemos como extremo una única celda conteniendo a todos los elementos).

Finalmente la [Figura 13](#) muestra el speed-up del FMBEM frente a BEM directo

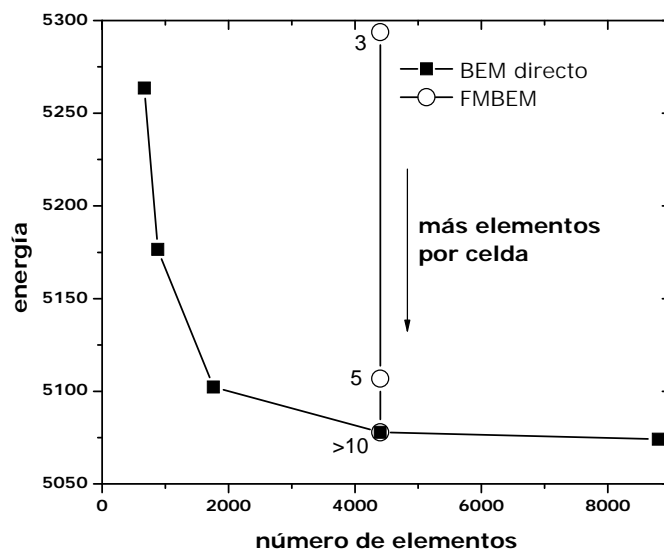


Figura 11: Energía almacenada en función del número de elementos.

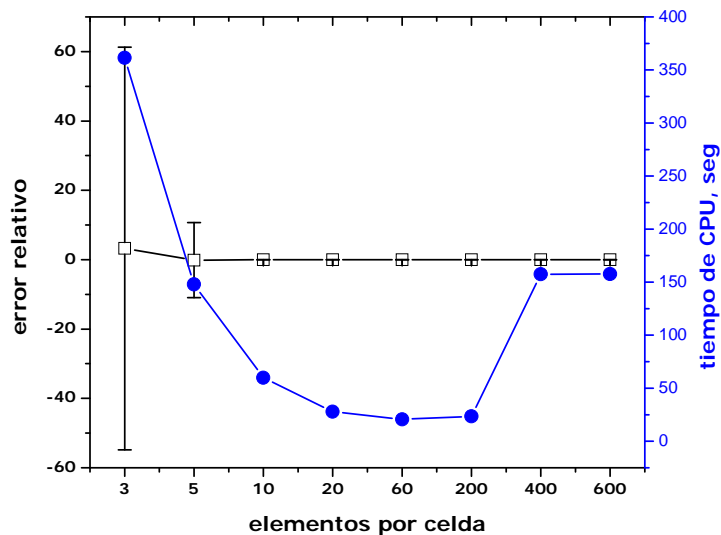


Figura 12: Error relativo y tiempo de CPU vs. elementos por celda para la malla de 4400 elementos.

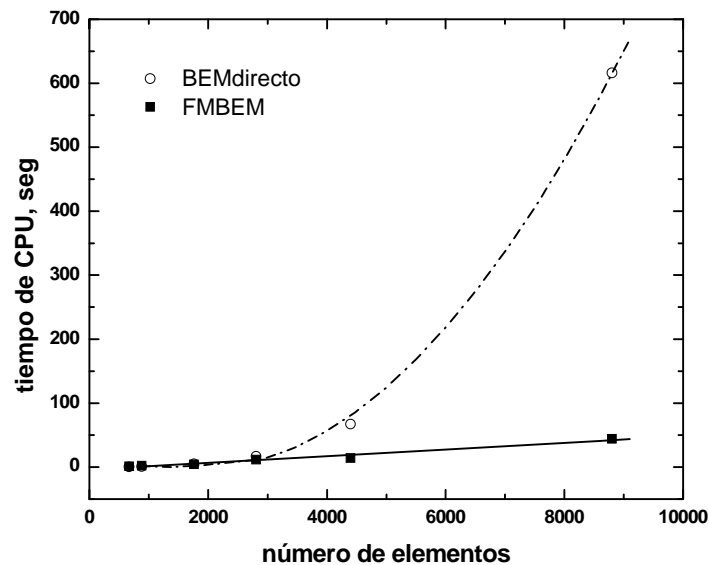


Figura 13: Comparación del tiempo de CPU usado por BEM directo y FMBEM.

5 CONCLUSIONES

En el presente trabajo se implementó el método de los Elementos de Contorno acelerado con el Fast Multipole Method para problemas de potencial en dos dimensiones. Observando el tiempo de CPU usado por ambos BEM y FMBEM, se percibe la amplia ventaja de su utilización y la puerta que este último abre hacia problemas de mayor tamaño. De los estudios de validación y calibración, observando las figuras de error y tiempo vs. elementos por celda se manifiesta la existencia de un rango óptimo donde la calidad de solución no va en desmedro del tiempo de ejecución. La calidad de la solución por FMBEM puede mejorarse utilizando elementos de mayor orden, pero esto puede no ser del todo ventajoso para la eficiencia en problemas de mayor escala, ya que el código utiliza integración analítica en todas las integrales (singulares y no singulares).

6 AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto PICT 12-14114 de la Agencia Nacional de Producción Científica y Tecnológica y el proyecto ALFA ELBENET “Europe-Latin America Boundary Element Method” auspiciado por la Unión Europea.

REFERENCIAS

- Aluru S., Greengard's n-body algorithm is not $O(n)$. *SIAM J. Scientific Computing*, 17: 773–776, 1996.
- Brebbia C.A. and Dominguez J., “Boundary Elements, An Introductory Course”, Comp. Mech. Pub. McGraw-Hill Book Company, Great Britain, (ISBN 1-85312-160-6), 1992.
- Greengard L. F., Rokhlin V., A fast algorithm for particle simulations, *J Comput Phys*, 73(2):325–48, 1987.
- Greengard L. F., The rapid evaluation of potential fields in particle systems, *Cambridge: The MIT Press*, 1988.
- Liu Y. J. y Nishimura N., The fast multipole boundary element method for potential problems: a tutorial, *Engineering Analysis with Boundary Elements*, 30(5):371-381, 2006.
- Nishimura N., Fast multipole accelerated boundary integral equation methods, *Appl Mech*

Rev, 55(4):299-324, 2002.
Rokhlin V. Rapid solution of integral equations of classical potential theory, *J Comput Phys*,
60:187–207, 1985.