

TECNICAS ARBOL EN INTERACCIONES DE n PARTICULAS ORIENTADAS A ELEMENTOS DE BORDE. EJEMPLOS NUMERICOS

Jorge D'Elía^a, Sofía Sarraf^{a,b}, Laura Battaglia^{a,c} y Ezequiel López^{a,b}

^a Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC)
Instituto de Desarrollo Tecnológico para la Industria Química (INTEC)
Universidad Nacional del Litoral - CONICET
Güemes 3450, 3000-Santa Fe, Argentina, e-mail: jdelia@intec.unl.edu.ar
web page: <http://www.cimec.org.ar>

^b Departamento de Mecánica Aplicada, Facultad de Ingeniería, Universidad Nacional del Comahue,
CONICET, Buenos Aires 1400, 8300-Neuquén, Argentina
e-mail: ([sssarraf](mailto:sssarraf@gmail.com), [ezequiel.jose.lopez](mailto:ezequiel.jose.lopez@gmail.com))@gmail.com

^c Grupo de Investigación en Métodos Numéricos en Ingeniería (GIMNI)
Universidad Tecnológica Nacional, Facultad Regional Santa Fe
Lavaysse 610, 3000-Santa Fe, Argentina
e-mail: lbattaglia@santafe-conicet.gob.ar

Palabras clave: técnicas rápidas para n partículas, árbol descendente de Barnes-Hut, método de elementos de borde, flujo reptante estacionario.

Resumen. En un trabajo anterior (D'Elia *et al.*, "Técnicas árbol en interacciones de n partículas orientadas a elementos de borde", en Mecánica Computacional, vol. XXVI, pp. 992-999, Córdoba, Argentina, octubre de 2007) se describieron las primeras etapas en una adaptación de los denominados algoritmos rápidos para simulaciones de interacciones entre n partículas, todas contra todas, en una variante jerárquica (o en árbol), para su uso complementario en el método de los elementos de borde (*Boundary Element Method*, BEM). En la presente contribución se muestra su uso en el caso de flujo de Stokes en régimen estacionario alrededor de cuerpos rígidos tridimensionales, mediante una formulación integral de contorno en las velocidades capaz de representar una fuerza y cupla no nulas sobre el cuerpo. El modelo BEM empleado es de bajo orden, para elementos triangulares planos. Se muestran ejemplos numéricos, incluyendo casos con soluciones analíticas, cuerpos con aristas y vértices, y cuerpos con geometrías deliberadamente intrincadas.

1. INTRODUCCION

Como es conocido en un método de paneles, o de elementos de borde (BEM, por *Boundary Element Method*), en sus formulaciones más clásicas, los elementos interactúan todos contra todos, lo cual conduce a matrices de influencia llenas (densas), cuyo cómputo involucra un costo computacional $O(n^2)$, donde n puede ser ya sea el número de elementos E o bien el número de nodos N , de la malla de paneles. Por otra parte, en los métodos numéricos que incluyen interacciones entre n partículas (Marzouk y Ghoniem, 2005; Bowers et al., 2007), todas contra todas, cada interacción involucra típicamente alguna función de Green proporcional a una potencia inversa de la distancia, en donde además cada partícula tiene asociado un número constante de atributos geométricos y físicos. A partir de un análisis de las interacciones, se proponen formulaciones rápidas, esto es, con un costo computacional mucho menor que $O(n^2)$. La utilidad práctica de las mismas se evidencia cuando se las entrelaza con los métodos iterativos para resolver de manera aproximada el sistema de ecuaciones relacionado con la formulación integral asociada con el BEM.

Entre las formulaciones rápidas se cuenta la del árbol recursivo descendente auto-adaptativo del algoritmo de Barnes y Hut (1986), el cual involucra la construcción de una estructura de datos en la forma de un árbol de octantes (*octree*) en el caso de un problema tridimensional (3D). Este algoritmo rápido, propuesto inicialmente para simulaciones computacionales en astrofísica (Ahn y Lee, 2008), ha sido extendido en otras aplicaciones, entre ellas, electromagnetismo (Aronsson et al., 2009), ecuación de Helmholtz (Aronsson et al., 2010), apantallamiento electrostático (Li et al., 2009), fractura en elasticidad tridimensional (Benedetti et al., 2008), dislocaciones dinámicas en deformaciones plásticas (Wang et al., 2006), e interacción de vórtices fluidos (Winkel et al., 2012). Es de notar también que el uso de los *octrees* es de creciente interés cuando se los combina en las GP-GPU (*General-Purpose computing on Graphics Processing Units*), como en Bédorf et al. (2012), Yaseen y Yaohang Li (2012) y Gaburov et al. (2010).

Kumar y Graham (2012) Lindbo y Tornberg (2010)

En un trabajo anterior (D'Elía et al., 2007) se describieron las primeras etapas de una implementación, sin embargo, no se mostraron resultados numéricos. En este trabajo se presenta una implementación de un algoritmo rápido para BEM, basada en el árbol recursivo descendente auto-adaptativo de Barnes y Hut (1986). Si bien el código computacional GBEM resuelve también la ecuación escalar de Laplace, en el presente trabajo sólo se considerará la ecuación vectorial de Stokes estacionaria. En los ejemplos numéricos se mostrará el flujo de Stokes en régimen estacionario alrededor de un cuerpo rígido en 3D, descrito a través de una ecuación integral de borde (BIE, por *Boundary Integral Equation*) en las velocidades, de tipo Indirecta y Completada (CIV-BIE, por *Completed Indirect Velocity BIE*, Ingber y Mammoli (1999)), en donde el campo solución es la densidad superficial de doble capa ψ , que es modificada para eliminar los modos rígidos, y tal que pueda representar una fuerza y cupla no nulas sobre el cuerpo. El modelo BEM empleado es de bajo orden, para elementos triangulares planos.

2. ARBOL DESCENDENTE DE BARNES-HUT AUTO-ADAPTATIVO

Un algoritmo jerárquico construye primero una estructura de datos de tipo árbol con raíz (Storti et al., 2012) para la representación jerárquica del espacio tridimensional \mathbb{R}^3 en cubos, y luego se calculan las interacciones recorriendo el árbol, posiblemente en forma parcial.

Cada vértice en un árbol con raíz tiene asociado un número de nivel, que es igual a la longitud del único camino que lo une con el vértice raíz. En consecuencia, el vértice raíz tiene el número de nivel cero, mientras que a medida que los vértices quedan más alejados de la raíz, el número

de nivel aumenta. Los vértices sin hijos son las hojas del árbol y tienen los números de nivel más elevados.

Cada partícula sólo interactúa con una parte del árbol, pues aquellas partes de la nube de puntos que queden más alejadas de la partícula considerada interactúan con la misma a través de los niveles más bajos del árbol, i.e. más cerca del vértice raíz, mientras que las partes de la nube más cercanas a la partícula considerada interactúan en los niveles más altos del árbol, es decir, más cerca de las hojas.

La construcción del árbol, naturalmente recursiva, se puede hacer, ya sea en forma ascendente desde las hojas hasta la raíz [Xue \(1998\)](#), o bien en forma descendente desde la raíz hacia las hojas, como en la versión original de [Barnes y Hut \(1986\)](#) propuesta como una mejora al de [Appel \(1985\)](#).

El algoritmo recursivo de Barnes-Hut es inherentemente adaptativo pues no hace ninguna suposición sobre la distribución espacial de las partículas en el cubo raíz. En el código computacional GBEM se ha implementado el algoritmo de Barnes-Hut mejorado por [Amor López \(1997\)](#), en el cual las partículas son ingresadas de a una. La construcción del árbol T puede resumirse con el siguiente pseudo código, que es una adaptación del dado por [Demmel \(1996\)](#) y que ya fue esbozado en [D'Elía et al. \(2007\)](#):

```

1 procedure hacer-el-arbol (X, T) {
2   !entra: X coordenadas de la malla BEM
3   !sale : T arbol jerarquico
4   n = size (X, dim=1) ! numero de particulas
5   d = size (X, dim=2) ! dimension del espacio
6   inicia-arbol (X, T, raiz)
7   for i = 1 to n do
8     ingresa-particula (X, T, raiz, i)
9 }

```

donde $\mathbf{X} \in \mathbb{R}^{n \times 3}$ es la matriz de las coordenadas de los puntos en el espacio \mathbb{R}^3 , mientras que n es el número de partículas.

Cada nueva partícula añadida recorre el árbol actual, desde la raíz hasta el vértice del nivel más bajo que puede contenerla, actualizando el área, el centroide, y las densidades dipolar y monopolar de los vértices por los cuales pasa. La actualización de dichos valores a medida que se recorren los vértices tiene la ventaja parcial de evitar una etapa adicional para calcularlos. Para cada partícula i añadida, con $i = 1, 2, \dots, E$, se realizan los siguientes pasos:

- Si la celda asociada con el vértice visitado v_m está vacío, entonces la partícula i se añade directamente;
- Si no, se hacen las siguientes tareas:
 - Al vértice visitado v_m se le agregan $r = 8$ vértices hijos;
 - Se divide la celda espacial del vértice visitado v_m , un cubo con longitud de arista L_m y centroide x_m dados, en $r = 8$ octantes iguales, que son numerados en alguna forma conveniente;
 - Se reubican las dos partículas, la pre-existente y la recién llegada, en los vértices hijos que les correspondan a cada una, siempre que no coincidan en un mismo hijo;
 - Si las dos partículas coincidieran en un mismo hijo, entonces se procede por recursión hasta ubicar una por celda.

Esta construcción recursiva de Barnes-Hut descendente da lugar a un *octree* en \mathbb{R}^3 , el cual es una estructura de datos muy bien conocida para representar datos irregulares. La construcción

recursiva de Barnes-Hut descendente se puede resumir con el siguiente pseudo código, adaptado de Demmel (1996) y ya mostrado en D'Elía et al. (2007):

```

1 procedure ingresa-particula (X,T,m,i) {
2   ! Intenta agregar la particula i en el vertice v_m del
3   ! arbol T, donde X son las coordenadas de la malla BEM.
4   ! Por construccion cada hoja tendra 1 o 0 particula.
5   select case (T (m).nroparticulas)
6   case (0) ! es una hoja sin ocupar
7     almacenar la particula i de coordenada x en el vertice v_m
8   case (1) ! es una hoja ya ocupada
9     agregar r-hijos dividiendo el cubo del vertice v_m
10    mover la particula preexistente al hijo correspondiente
11    actualizar centroide y area
12    identificar el hijo c donde hay que colocar la particula i
13    ingresa-particula (X,T,c,i)
14  case (2:) ! tiene 2 o mas particulas
15    identificar el hijo c donde hay que colocar la particula i
16    ingresa-particula (X,T,c,i)
17  end select
18 }

```

La actualización de los atributos geométricos y físicos de los vértices por donde pasa la partícula ingresada se realiza de la siguiente forma, asumiendo el caso vectorial para la ecuación de Stokes. Supongamos que, en una dada etapa, el área S_u , el centroide \mathbf{x}_u , y la densidad \mathbf{W}_u dipolar o monopolar, asociados al vértice u están dados por

$$\begin{aligned}
 S_u &= \sum_{i=1}^U S_i ; \\
 \mathbf{x}_u &= \frac{1}{S_u} \sum_{i=1}^U S_i \mathbf{x}_i ; \\
 \mathbf{W}_u &= \sum_{i=1}^U \mathbf{W}_i ;
 \end{aligned} \tag{1}$$

donde U es el número total de partículas contenidas en el subárbol que empieza en el vértice u . Al añadir una nueva partícula i a la celda, los nuevos valores del área, posición y densidad de la celda se actualizan con

$$\begin{aligned}
 S_{u+1} &= S_u + S_i ; \\
 \mathbf{x}_{u+1} &= \frac{1}{S_{u+1}} (\mathbf{x}_u S_u + \mathbf{x}_i S_i) ; \\
 \mathbf{W}_{u+1} &= \mathbf{W}_u + \mathbf{W}_i ;
 \end{aligned} \tag{2}$$

donde S_i , \mathbf{x}_i y \mathbf{W}_i son el área, la posición y la densidad, respectivamente, de la partícula añadida al vértice. En la versión descendente del algoritmo de Barnes-Hut, la construcción del árbol finaliza cuando se ingresaron las n partículas, donde cada hoja tiene 0 o 1 partícula.

De lo anterior se desprende la necesidad de un test para decidir cuándo una celda está lo suficientemente lejos de una partícula dada. Existen diversas propuestas, todas basadas en consideraciones geométricas vinculadas con el diámetro de la celda y las distancias relativas. El algoritmo de Barnes-Hut considera que una celda está bien separada si $D/L < \theta$, donde D es algún diámetro de la celda, $L = \|\mathbf{x} - \mathbf{y}\|_2$ es la distancia euclídea entre la posición \mathbf{x} de la partícula y el centroide de la celda \mathbf{y} , mientras que θ es un cierto valor umbral prefijado. Un

umbral bajo sugiere que una partícula tiene que estar muy distante de la celda antes de aceptar una expansión multipolar, con lo cual el costo computacional aumenta, pues el árbol tiene que ser atravesado hasta sus niveles más profundos, haciendo que se calculen un mayor número de interacciones directas entre partículas. En el límite $\theta \rightarrow 0$ se recupera el caso todas contra todas, con un costo computacional $O(n^2)$, mientras que cuando $0 \ll \theta \leq 1$ hay menos interacciones directas.

En cuanto al *solver* iterativo, el árbol jerárquico aparece cuando hay que evaluar el producto matriz vector $\mathbf{q} = \mathbf{A}\mathbf{p}$, o su traspuesta, resumido en el siguiente pseudo código, también adaptado de Demmel (1996),

```

1 function q = daxpy (X, T, p) {
2   n = length (p) ; raiz = root-arbol (T)
3   q = 0
4   for i = 1 to n do
5     q (i) = influencia (X,T,raiz,i)
6 }

```

en donde

```

1 function f = influencia (X,T,r,i) {
2   ! influencia f sobre la partícula i causada por
3   ! subarbol de raiz r
4   f = 0
5   x = X (i,:) ; y = T (r).baricentro
6   D = T (r).diametro ; L = ||x - y||_2
7   select case (T (r).nroparticulas)
8     case (1) ! 1 partícula
9       f = aproximacion-BEM (x,y)
10    case (2:) ! 2 o mas partículas
11      if (D/L < theta) then
12        f = aproximacion-BEM (x,y)
13      else
14        forall (hijos c de r) do f = f + influencia (X,T,c,i)
15      end if
16    end select
17 }

```

Nótese que es también un algoritmo recursivo que recorre el árbol, posiblemente de forma parcial, pues no necesariamente debe llegar hasta todas las hojas.

3. EJEMPLOS NUMERICOS

Se considera el flujo reptante alrededor de una esfera y de otros cuerpos rígidos, cuyos centroides coinciden con el origen de coordenadas cartesianas. En el caso de la esfera, los resultados son comparados con las soluciones analíticas para tres tipos de flujo entrante. Los números de nodos N y de elementos E de cada malla empleada en el caso de la esfera son resumidas en la Tabla 1, donde las mallas 1, 2, y 3 son estructuradas.

En todos los ejemplos numéricos mostrados en este trabajo se ha empleado el valor umbral $\theta = 0.15$. Con este valor, el porcentaje de coeficientes que se han calculado con expresiones de campo lejano ha fluctuado entre el 50 y el 70 %. En cuanto al *solver* iterativo, se ha empleado GMRES sin preconditionamiento (Saad y Schultz, 1986).

En todos los casos, se aplica la regla de integración Q_{21} , lo que significa que se usan 2 puntos de Gauss-Legendre (GL) en cada coordenada de integración en la autointegral y en las interacciones cercanas, y 1 punto en las interacciones lejanas (D'Elía et al., 2011).

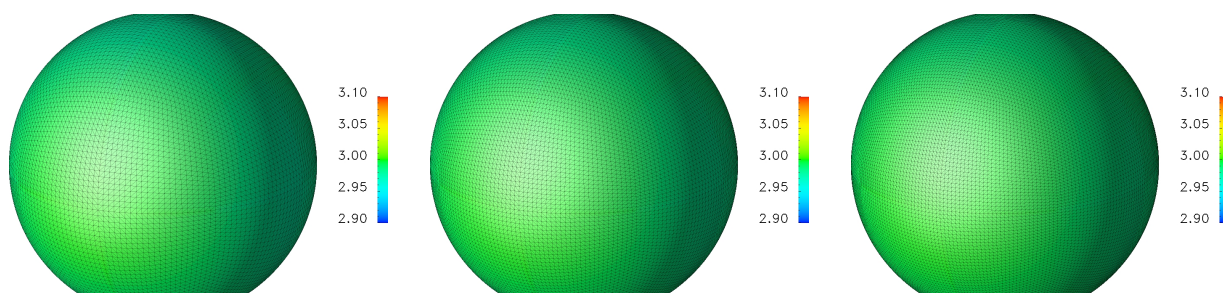


Figura 1: Tracción adimensional τ_1 sobre la superficie de la esfera usando BEM jerárquico basado en el algoritmo de Barnes-Hut, con la regla de cuadratura Q_{21} . Mallas BEM con refinamiento uniforme 1, 2 y 3, de izquierda a derecha.

Malla	1	2	3
N	10 586	15 002	20 186
E	21 168	30 000	40 000

Tabla 1: Flujo reptante estacionario alrededor de una esfera de radio \mathcal{R} . Número de nodos N y número de elementos E en cada malla. Las mallas 1-3 son estructuradas.

3.1. Esfera unitaria

En el caso de la esfera, se dispone de soluciones analíticas para diversos tipos de flujos incidentes. En particular, se consideran tres distribuciones de flujo no perturbado incidente: uniforme, cortante y paraboloidal (Power y Miranda, 1987). Las expresiones analíticas para la velocidad no perturbada, la fuerza y el torque en cada caso son resumidas en la Tabla 2 (Guazzelli, 2003; Dhont, 1996).

En las simulaciones numéricas se adoptan los siguientes valores: densidad del fluido $\rho = 1 \text{ kg/m}^3$, viscosidad cinemática $\nu = 1 \text{ m}^2/\text{s}$, velocidad no perturbada $U_\infty = 10^{-3} \text{ m/s}$ a lo largo de la dirección x_1 , y radio de la esfera $\mathcal{R} = 1 \text{ m}$. Como longitud típica en el cálculo de la fuerza, el torque y la tracción adimensionales se emplea el diámetro de la esfera, i.e. $L = 2\mathcal{R} = 2 \text{ m}$.

El valor exacto de la tracción sobre la superficie de esfera bajo el efecto de un flujo uniforme es igual a la constante $\mathbf{t} = (3/2)\mu U_\infty/\mathcal{R} \mathbf{e}_1^0$, donde \mathbf{e}_1^0 es el versor cartesiano en la dirección x . La tracción adimensional $\tau_1 = t_1/(\mu U_\infty L^{-1})$ se muestra en la Fig. 1 para las mallas con refinamiento uniforme 1, 2 y 3.

Flujo incidente	Velocidad no perturbada \mathbf{u}	Fuerza \mathbf{D}	Torque \mathbf{C}
uniforme	$(U_\infty, 0, 0)$	$(6\pi\mu U_\infty \mathcal{R}, 0, 0)$	$\mathbf{0}$
cortante	$U_\infty(x_2, -x_1, 0)/\mathcal{R}$	$\mathbf{0}$	$(0, 0, 8\pi\mu U_\infty \mathcal{R}^2)$
paraboloidal	$U_\infty(x_1^2 + x_2^2, 0, 0)/\mathcal{R}^2$	$(4\pi\mu U_\infty \mathcal{R}, 0, 0)$	$\mathbf{0}$

Tabla 2: Flujo reptante estacionario alrededor de una esfera de radio \mathcal{R} . Expresiones analíticas de la fuerza y torque para tres tipos de flujos incidentes (Guazzelli, 2003; Dhont, 1996).

3.2. Geometrías intrincadas o con aristas

Se consideran algunas geometrías intrincadas o con aristas. Para el caso con aristas se elige un cubo de lado $L = 1 \text{ m}$, mientras que en los restantes se seleccionaron ciertos ejemplos de los disponibles en el repositorio “180 wrapped tubes” (Edelsbrunner, 2012), algunos de los cuales

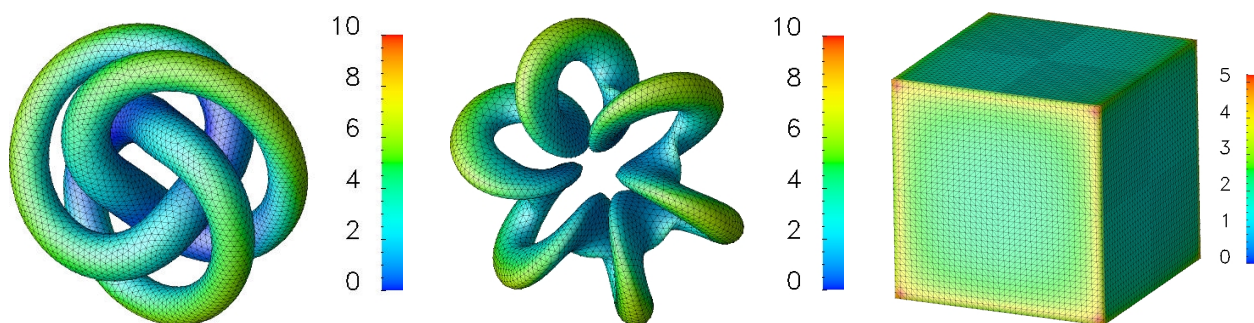


Figura 2: Geometrías deliberadamente intrincadas o con aristas: *speed* (izq.) y *torsion* (cen.) (Edelsbrunner, 2012), y cubo de arista unitaria (der.). Tracción adimensional τ_1 bajo flujo paralelo $U_\infty = (10^{-3}, 0, 0)$ m/s, usando BEM jerárquico basado en el algoritmo de Barnes-Hut.

pueden considerarse aproximaciones a fibras retorcidas u organismos microscópicos. Las geometrías fueron generadas usando el mallador NETGEN (Schöberl, 2012). En todos los casos, las superficies no se tocan entre sí, y encierran un volumen finito positivo. Las coordenadas nodales mínimas y máximas son $-1/2$ y $+1/2$, respectivamente, mientras que la velocidad no perturbada es $U_\infty = 10^{-3}$ m/s a lo largo de la dirección x_1 .

Adoptando la nomenclatura dada en Edelsbrunner (2012) en los casos (i-ii), se tienen los siguientes ejemplos, incluyendo los números de nodos y de elementos de las mallas respectivas:

- i) *Speed* (caso (3,3/3)): 6 471 nodos y 12 942 paneles;
- ii) *Torsion* (caso (1,6)): 5 051 nodos y 10 102 paneles;
- iii) Cubo unitario: 8 866 nodos y 17 328 paneles;

Los campos de la tracción adimensional τ_1 obtenidos con el algoritmo de Barnes-Hut son mostrados en la Fig. 2. La tracción adimensional τ_1 exhibe el comportamiento esperado en la distribución y en el máximo valor en cada caso, con un incremento en la intensidad cerca de las aristas, y en magnitudes que no difieren significativamente de los correspondientes a la esfera.

4. DISCUSION

En los ejemplos numéricos de trabajos anteriores (D'Elía et al., 2008, 2009), referidos al problema de flujo reptante alrededor de un cuerpo rígido tridimensional, se habían empleado técnicas de colocación y de Galerkin. De lo analizado en trabajos previos y de los ejemplos aquí presentados, se puede inferir como regla aproximada que los tamaños prácticos que se pueden analizar en una computadora con 16 GB de memoria RAM, y cuando se emplea doble precisión, están en el orden de:

- O (6 kpaneles): técnica de colocación, matriz del sistema densa, solver directo;
- O (12 kpaneles): ponderación de Galerkin, matriz del sistema densa, solver directo;
- O (36 kpaneles): algoritmo de Barnes-Hut, equivalente a una matriz rala, solver iterativo;

siempre que el sistema de ecuaciones resulte bien condicionado, lo cual en BEM es dependiente de la forma de la geometría, como ocurre con todos los ejemplos mostrados.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina), Universidad Nacional del Litoral (UNL, Argentina, proyecto

CAI+D 2009–III–4–2), Agencia Nacional de Promoción Científica y Tecnológica (ANCyT, Argentina, proyectos PICT 2492–10, PICT 1141–07), EU-IRSES (PIRSSES-GA-2009-246977), y ha sido parcialmente realizado con los recursos del *Free Software Foundation/GNU-Project*, tales como GNU–Linux–OS, GNU–GFortran, GNU–Octave, GNU–Git, GNU–Doxygen, y GNU–GIMP, así como otros recursos de código abierto, tales como NETGEN, ParaView, OpenDX, Xfig y \LaTeX .

REFERENCIAS

- Ahn C. y Lee S.H. A new treecode for long-range force calculation. *Computer Physics Communications*, 178(2):121–127, 2008.
- Amor López M. *Divide and Conquer Algorithms on Processor Meshes*. Ph.D. thesis, Dpto de Electrónica y Sistemas, Universidad de Santiago de Compostela, 1997. Advisors: Francisco Argüello Pedreira and Juan López Gómez.
- Appel A.W. An efficient program for many-body simulations. *SIAM J. Sci. Stat. Comput.*, 6(1):85–103, 1985.
- Aronsson J., Butt K., Jeffrey I., y Okhmatovski V. The Barnes-Hut hierarchical center-of-charge approximation for fast capacitance extraction in multilayered media. *IEEE Transactions on Microwave Theory and Techniques*, 58(5):1175–1178, 2010.
- Aronsson J., Jeffrey I., y Okhmatovski V. Generalization of the Barnes-Hut algorithm for the Helmholtz equation in three-dimensions. *IEEE Antennas and Wireless Propagation Letters*, 8:425–428, 2009.
- Barnes J. y Hut P. A hierarchical $O(n \log n)$ force-calculation algorithm. *Nature*, 324:446–449, 1986.
- Bédorf J., Gaburov E., y Zwart S.P. A sparse octree gravitational N-body code that runs entirely on the GPU processor. *Journal of Computational Physics*, 231(7):2825–2839, 2012.
- Benedetti I., Aliabadi M.H., y Dav G. A fast 3D dual boundary element method based on hierarchical matrices. *International Journal of Solids and Structures*, 45(7–8):2355–2376, 2008.
- Bowers K.J., Dror R.O., y David E. Shaw D.E. Zonal methods for the parallel execution of range-limited n-body simulations. *Journal of Computational Physics*, 221:303–329, 2007.
- D'Elía J., Battaglia L., Cardona A., y Storti M. Full numerical quadrature of weakly singular double surface integrals in Galerkin boundary element methods. *Int. J. for Num. Meth. in Biomedical Engng.*, 27(2):314–334, 2011. Doi:10.1002/cnm.1309.
- D'Elía J., Battaglia L., y Storti M. Técnicas árbol en interacciones de n partículas orientadas a elementos de borde. In Elaskar, Pilotta, y Torres, editors, *Mecánica Computacional*, vol. XXVI, pages 992–999. Córdoba, 2007.
- D'Elía J., Battaglia L., Storti M., y Cardona A. Galerkin boundary integral equations applied to three dimensional Stokes flows. In Cardona, Storti, y Zuppa, editors, *Mecánica Computacional*, vol. XXVII, pages 2397–2410. San Luis, 2008.
- D'Elía J., Battaglia L., Storti M., y Cardona A. Galerkin boundary integral equations applied to three dimensional Stokes flows. In C. Bauza, P. Lotito, L. Parente, y M. Venere, editors, *Mecánica Computacional*, vol. XXVIII, pages 1453–1462. Tandil, 2009.
- Demmel D. Fast hierarchical methods for the n-body problem. Parts 1 and 2. Course CS267, lecture 24. 1996. <http://www.cs.berkeley.edu/~demmel/cs267/lecture26/lecture26.html/>.
- Dhont J.K.G. *An Introduction to Dynamics of Colloids*. Elsevier, 1996.
- Edelsbrunner H. 180 wrapped tubes. Technical report, Dep. of Computer Science, Duke University, Durham, <http://www.cs.duke.edu/~edels/Tubes/>, 2012.

- Gaburov E., Bédorf J., y Zwart S.P. Gravitational tree-code on graphics processing units: implementation in CUDA. *Procedia Computer Science*, 1(1):1119–1127, 2010.
- Guazzelli E. Microhidrodinámica. Cours de base sur les suspensions, 2003. Course notes, Groupe Ecoulements de Particules (GEP), IUSTI, Polytech Marseille.
- Ingber M.S. y Mammoli A.A. A comparison of integral formulations for the analysis of low Reynolds number flows. *Eng. Anal. with Boundary Elements*, 23:307–315, 1999.
- Kumar A. y Graham M.D. Accelerated boundary integral method for multiphase flow in non-periodic geometries. *J. Comp. Physics*, 231(20):6682–6713, 2012.
- Li P., Johnston H., y Krasny R. A cartesian treecode for screened Coulomb interactions. *Journal of Computational Physics*, 228(10):3858–3868, 2009.
- Lindbo D. y Tornberg A.K. Spectrally accurate fast summation for periodic Stokes potentials. *J. Comp. Physics*, 229(23):8994–9010, 2010.
- Marzouk Y.M. y Ghoniem A.F. K-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical N-body simulations. *Journal of Computational Physics*, 207:493–528, 2005.
- Power H. y Miranda G. Second kind integral equation formulation of Stokes flows past a particle of arbitrary shape. *SIAM J. Appl. Math.*, 47(4):689–698, 1987.
- Saad Y. y Schultz M. Generalized minimum residual method (GMRES). *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.
- Schöberl J. NETGEN 4.3 mesher. Technical report, University of Linz, Austria, <http://sourceforge.net/projects/netgen-mesher/>, 2012.
- Storti M., D'Elía J., Paz R., Dalcín L., y Pucheta M. Algoritmos y Estructuras de Datos (notas y transparencias en formato pdf). Cátedra AED, Dpto de Informática, Facultad de Ingeniería y Ciencias Hídricas (FICH), Universidad Nacional del Litoral, 2012.
- Wang Z., Ghoniem N., Swaminarayan S., y LeSar R. A parallel algorithm for 3D dislocation dynamics. *Journal of Computational Physics*, 219(2):608–621, 2006.
- Winkel M., Speck R., Hübner H., Arnold L., Krause R., y Gibbon P. A massively parallel, multi-disciplinary Barnes-Hut tree code for extreme-scale N-body simulations. *Computer Physics Communications*, 183(4):880–889, 2012.
- Xue G. A cost optimal parallel algorithm for computing force field in n-body simulations. In Hsu y Kao, editors, *COCOON 98, LNCS 1949*, pages 95–104. Springer Verlag, Berlin, 1998.
- Yaseen A. y Yaohang Li Y. Accelerating knowledge-based energy evaluation in protein structure modeling with Graphics Processing Units. *Journal of Parallel and Distributed Computing*, 72(2):297–307, 2012.