

PRIMEROS PASOS EN EL USO DE UN MODELO PGAS EN EL METODO DE LOS ELEMENTOS DE BORDE. FORMULACION PARA FLUJO DE STOKES CON SCALAPACK

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

^a Centro de Investigación de Métodos Computacionales (CIMEC)

Universidad Nacional del Litoral (UNL) - CONICET

Predio Conicet-Santa Fe, Colectora Ruta Nac 168 / Paraje El Pozo, Santa Fe (S3000GLN), Argentina

e-mail: gusadrr@yahoo.com.ar, 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, Neuquén (Q8300IBX), Argentina

e-mail: (sssarraf, ezequiel.jose.lopez)@gmail.com

^c Grupo de Investigación en Métodos Numéricos en Ingeniería (GIMNI)

Universidad Tecnológica Nacional (UTN), Facultad Regional Santa Fe (FRSF)

Lavaysse 610, Santa Fe (S3000GLN), Argentina

e-mail: lbattaglia@santafe-conicet.gob.ar

Palabras Clave: flujo reptante, flujo tridimensional, método de elementos de borde, técnica de colocación, ponderación de Galerkin, ScaLAPACK.

Resumen. El espacio de direcciones globales particionado (o PGAS, por *Partitioned Global Address Space*) es un modelo de programación paralela propuesto para entornos con memoria distribuida, tales como en los *clusters* especializados en computación numérica intensiva de alto rendimiento (o HPC, por *High Performance Computing*). Dicho modelo es la base, entre otros, de los lenguajes de programación multi-paradigma UPC (*Unified Parallel C*) y del coarray fortran (CAF), los cuales proporcionan una extensión para efectuar comunicaciones *one-side*. En particular, CAF es una extensión incluida en el estándar Fortran 2008. En un trabajo anterior (e.g. D'Elía *et al.*, "Primeros pasos en el uso de un modelo PGAS en el método de los elementos de borde", *Mecánica Computacional*, vol. XXXII, 2013), se han presentado las primeras etapas de una codificación basada en coarray Fortran de un código computacional por BEM (*Boundary Element Method*), con énfasis en la etapa de factorización LU del sistema de ecuaciones. Para contar con una subrutina de referencia (o de contraste), en este trabajo se expone el ensamblado del sistema de ecuaciones por BEM en un entorno distribuido, donde la solución del sistema de ecuaciones lineales con matriz densa, cuadrada, y regular, se basa en la librería de HPC de dominio público ScaLAPACK (<http://www.netlib.org>), apta, en particular, para computadoras personales con memoria primaria tanto compartida como distribuida. Como caso particular se consideran problemas de flujo reptante (o de Stokes) alrededor de cuerpos tridimensionales, modelados tanto con ponderación de Galerkin (o GBEM, por *Galerkin Boundary Element Method*), como por colocación al centroide de los elementos.

1. INTRODUCCION

Las soluciones de las ecuaciones integrales de borde (o BIE, por *Boundary Integral Equation*) típicamente son obtenidas numéricamente mediante variantes del método de elementos de borde (BEM, por *Boundary Element Method*), usando colocación, o bien con el *Galerkin Boundary Element Method* (GBEM) (e.g. ver [Sauter y Schwab, 2011](#)). Existen diversas aplicaciones prácticas de las BIE, e.g. microflujos en dispositivos *lab-on-chip* ([Lepchev y Weihs, 2010](#)), micromezcladores con flujo oscilatorio ([Shipman et al., 2007](#)), microventiladores en electrónica ([Galvis et al., 2012](#)), o en *micro-electro-mechanical systems* (MEMS, e.g. ver [Wang et al., 2006](#); [Méndez et al., 2008](#); [Berli y Cardona, 2009](#)), entre otros casos. En [D'Elía et al. \(2011\)](#) se describe una técnica de cuadratura cuádruple para funciones de Green tensoriales y con una singularidad débil, apta para GBEM, mientras que en [D'Elía et al. \(2014\)](#); [Sarraf et al. \(2014b\)](#) se emplea una BIE indirecta en velocidades (CIV-BIE, por *Completed Indirect Velocity-BIE*), de Fredholm y de segunda clase, cuyo espectro del operador integral es completado para eliminar los modos rígidos, con la cual se obtiene el valor puntual de la tracción en flujos reptantes alrededor de cuerpos rígidos tridimensionales con esquinas y bordes afilados, donde la solución de la CIV-BIE se obtiene con colocación (BEM) o ponderación de Galerkin (GBEM).

Por otro lado, en los últimos años han ganado aceptación los lenguajes de programación paralela de alto nivel basados en el modelo PGAS (por *Partitioned Global Address Space*, e.g. ver [Diaz et al. \(2012\)](#)), aplicables en computadoras paralelas con memoria primaria tanto compartida como distribuida. El modelo PGAS provee comunicación *one-side*, y es la base de las propuestas basadas en UPC (por *Unified Parallel C*, <http://upc.lbl.gov>), o en el Coarray Fortran (CAF o Fortran 2008, <http://www.j3-fortran.org>). En esta línea, por ejemplo, en [D'Elía et al. \(2013\)](#); [D'Elía et al. \(2013\)](#) se presentaron resultados preliminares en el uso de los modelos PGAS. Empero, dado que los modelos PGAS existentes no previeron desde el inicio la actual hibridización del *hardware* ([Luszczek et al., 2014](#); [Schreiber, 2014](#)), también se ha propuesto el OpenACC (<http://www.openacc.org>) como una alternativa para uniformizar la programación en paralelo, en particular de las computadoras personales con CPU/GPGPU heterogéneos (GPGPU, por *General Purpose computing on Graphics Processing Units*). Al igual que en OpenMP (<http://www.openmp.org>), se introducen directivas en el código fuente para identificar las porciones que pueden ser aceleradas mediante directivas. A diferencia de OpenMP en sus versiones anteriores a la 4.0, el código puede iniciarse en la CPU o en la GPU.

No obstante, un recurso mucho más clásico es el empleo de la librería ScaLAPACK (e.g. ver <http://www.netlib.org>) orientada a HPC y de dominio público. Cuenta con una licencia tipo BSD (*Berkeley Software Distribution*), i.e. una licencia de *software* libre permisiva. La ScaLAPACK está orientada a los sistemas lineales densos y, en particular, apta para computadoras personales constituidas únicamente con una o varias CPU, ya sea con memoria primaria compartida (procesadores multinúcleo), o bien distribuida (e.g. la disponible en un *cluster Beowulf*). Dado que el Sistema de Ecuaciones Lineales (SEL) obtenido con BEM/GBEM clásico involucra en general una matriz del sistema que es densa, cuadrada, y regular, el uso de ScaLAPACK es un recurso interesante como una subrutina de referencia (o de contraste) para obtener la solución numérica por un método directo, en computadoras homogéneas, tanto personales como las ensambladas según un *cluster Beowulf*. Este trabajo se concentrará en esta última alternativa, con el propósito de contar con una subrutina de referencia para cuando posteriormente se implemente con PGAS/OpenACC, lo cual resulta una estrategia de frecuente uso en la literatura ([Luszczek et al., 2014](#); [Tondi et al., 2012](#); [Badia et al., 2011](#)).

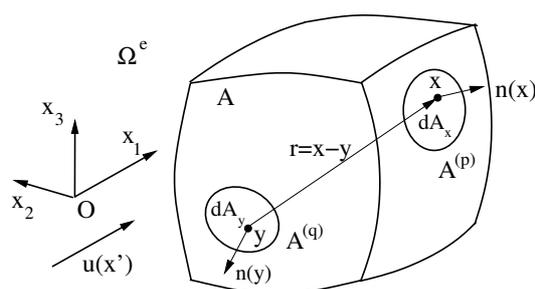


Figura 1: Una superficie cerrada y suave por partes A con un dominio exterior Ω^e : punto de observación \mathbf{x} , punto fuente \mathbf{y} , posición relativa $\mathbf{r} = \mathbf{x} - \mathbf{y}$, vectores $\mathbf{n}(\mathbf{x})$, $\mathbf{n}(\mathbf{y})$, y las áreas diferenciales $dA_{\mathbf{x}}$, $dA_{\mathbf{y}}$.

2. FORMULACION INTEGRAL Y NUMERICA EN FLUJO DE STOKES

La formulación CIV-BIE no es otra cosa que la Formulación Integral (FI) de Power-Miranda / Hebeker. La misma consiste en una BIE que describe el flujo de Stokes exterior a una superficie A cerrada, rígida y suave a trozos en \mathbb{R}^3 . Esta BIE no posee modos rígidos, es de tipo indirecta y de segunda clase, y su término fuente es función de la velocidad no perturbada, e.g. ver Pozrikidis (1997); Power y Miranda (1987); Hebeker (1986). En cuanto a las formulaciones numéricas para resolver la CIV-BIE, las mismas comprenden un método de colocación, y otro de ponderación de Galerkin. Ambas técnicas usan un doble lazo anidado sobre los elementos $p, q = 1, 2, \dots, E$, donde E es el número de paneles o elementos de la malla BEM, y los puntos \mathbf{x} , \mathbf{y} están vinculados a los elementos p, q , respectivamente (ver Fig. 1). Estos puntos corresponden al punto de observación (\mathbf{x}) y fuente (\mathbf{y}).

Tanto la formulación integral CIV-BIE, orientada al flujo de Stokes, como los métodos numéricos por colocación y por ponderación de Galerkin, son resumidos en otro trabajo también presentado en este congreso (Sarraf et al., 2014a).

3. SCALAPACK

Los *clusters* del CIMEC emplean *software* de libre distribución disponible en el sistema operativo **Linux**, en particular, las distribuciones **Fedora**. La instalación de la clásica librería ScaLAPACK en distribuciones Linux es descrita con gran detalle en muchas partes (e.g. ver <http://www.netlib.org>). Es posible instalar librerías ya construídas para distribuciones Linux específicas, así como también el esquema tradicional de *configure-make-install*. En la Fig. 2 se muestra una jerarquía de las capas de *software* que participan en la ScaLAPACK (adaptado de Choi et al., 1996) en donde:

- **BLACS** (*Basic Linear Algebra Communication Subprograms*, <http://www.netlib.org>): este paquete ya viene incluido dentro de la versión 2.0.2 (2012-05-01) de ScaLAPACK;
- **LAPACK** (*Linear Algebra PACKage*, <http://www.netlib.org>);
- **BLAS** (*Basic Linear Algebra Subprograms*, <http://www.netlib.org>);
- **MPI** (*Message Passing Interface*, <http://www.mpi-forum.org>): es el estándar *ipso facto* empleado para comunicación entre procesadores, y es el único actualmente previsto en ScaLAPACK. En particular, se ha optado por la distribución **OpenMPI**.

La denominación de subrutinas globales en la Fig. 2 se refiere a aquellas cuyos argumentos incluyen arreglos distribuidos en un esquema bidimensional (2D) de la forma bloque-cíclico, mientras que las subrutinas locales se refieren a aquellas cuyos argumentos incluyen arreglos contenidos únicamente dentro de cada proceso.

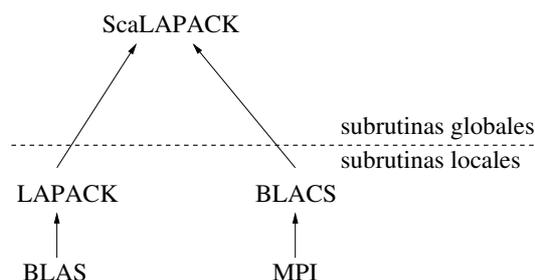


Figura 2: Jerarquía de las capas de *software* que participan en la ScaLAPACK (Choi et al., 1996).

4. SCALAPACK EN EL CODIGO BEM/GBEM

A los efectos de simplificar la presentación, sólo se considera el caso de la subrutina por colocación (BEM) secuencial. La misma se la copia con un nuevo nombre, y se introducen los siguientes cambios:

- Inicializar BLACS (*Basic Linear Algebra Communication Subprograms*, detallada en <http://www.netlib.org>);
- Alocar los arreglos *distribuidos* (o globales) del SEL, i.e. las matrices del sistema \mathbf{A}_D , del término independiente \mathbf{B}_D , y de la solución \mathbf{X}_D . Esto exige la definición de un tamaño de bloque n_b conveniente a los efectos de una mayor velocidad de cómputo, y que se define mediante un algoritmo heurístico;
- Opcionalmente activar llamadas a la librería MPI para mediciones estadísticas o de consumos de recursos;
- En una primera implementación, el doble lazo en los índices p, q sobre los E elementos de la malla BEM no se lo modifica, excepto en la etapa de la inserción de las contribuciones elementales, tanto en la matriz de coeficientes distribuida \mathbf{A}_D como en el término independiente \mathbf{B}_D . Para insertar la contribución $A_{ij}^{(p,q)}$ de los elementos p, q se ha introducido una versión modificada de la subrutina PDELSET provista en ScaLAPACK, denominada aquí como PDELADD, pues PDELSET asigna un valor en la matriz distribuida pero no permite acumular. La subrutina introducida PDELADD, como es usual, primero chequea si la contribución $A_{ij}^{(p,q)}$ pertenece a la porción local de la matriz de coeficientes distribuida \mathbf{A}_D , y, sólo en ese caso, la acumula. En forma análoga para la contribución elemental $B_{ij}^{(p,q)}$ de la matriz del término independiente \mathbf{B}_D . En una segunda implementación, por razones de eficiencia, PDELADD podría mejorarse para acumular una submatriz en lugar de un escalar mediante el empleo de las rutinas globales de combinación disponibles en BLACS;
- Una vez ensamblado el SEL distribuido $\mathbf{A}_D \mathbf{X}_D = \mathbf{B}_D$, se hacen las llamadas a la factorización LU y subsiguiente resolución del SEL, ambas provistas por ScaLAPACK;
- Se copia la solución distribuida \mathbf{X}_D hallada por ScaLAPACK a un arreglo local \mathbf{X}_L ;
- Se desalocan todos los arreglos distribuidos y se cierran los entornos BLACS y MPI;
- Se continúa con el cómputo secuencial usual.

En la presentación del trabajo se expondrán ejemplos numéricos de validación, con uno o más procesos, y usando diferentes números de nodos en alguno de los *clusters* del CIMEC.

5. CONCLUSIONES

Hay que tener ciertas precauciones con el uso alternativo de las librerías ScaLAPACK pre-compiladas, e.g. aquellas instaladas con recursos tales como *yum*, en el sentido de asegurar

consistencia entre las versiones de los compiladores de `mpicc/mpifort` que fueron empleados en las mismas, con los empleados en el código de aplicación. Por otra parte, ha sido útil introducir la subrutina PDELADD para realizar la acumulación de las contribuciones elementales en la matriz distribuida del sistema. Además, en las etapas previas, fue valioso contar con *tutorials* para el uso de la factorización LU disponible en ScaLAPACK (e.g. [Chepurniy, 2013](#)). En el futuro se preve repetir las modificaciones realizadas en la subrutina por colocación al caso de ponderación de Galerkin, y de aprovechar ambas variantes en el análisis de la fuerza de amortiguamiento en micro-resonadores inmersos en un flujo de Stokes, y discretizados con una malla BEM con un relativo elevado número de grados de libertad. Tal tipo de análisis resultará de interés en MEMS, los cuales son ampliamente utilizados en sensores biológicos o químicos, en donde la resolución del sensor resulta influenciada por la disipación de energía ocasionada por el amortiguamiento viscoso ([Magrab, 2012](#)).

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, proyecto PIP 112 2011 00978), Universidad Nacional del Litoral (UNL, Argentina, proyecto CAI+D 2009–III-4–2), Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, proyecto PICT 2492–10), 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 Xfig y L^AT_EX.

REFERENCIAS

- Badia J.M., Movilla J.L., Climente J.I., Castillo M., Marqués M., Mayo R., Quintana-Ortiz E.S., y Planelles J. A parallel solver for huge dense linear systems. *Comput. Phys. Commun.*, 182(11):2441–2442, 2011.
- Berli C. y Cardona A. On the calculation of viscous damping of microbeam resonators in air. *J. of Sound and Vibration*, 327(1-2):249–253, 2009.
- Chepurniy N. LAPACK and ScaLAPACK Examples, (https://www.sharcnet.ca/help/index.php/LAPACK_and_ScaLAPACK_examples#ScaLAPACK_EXAMPLE). 2013.
- Choi J., Demmel J., Dhillon I., Dongarra J., Ostrouchov S., Petitet A., Stanley K., Walker D., y Whaley R.C. ScaLAPACK: a portable linear algebra library for distributed memory computers - design issues and performance. *Computer Physics Communications*, 97(1-2):1–15, 1996.
- D’Elía J., Battaglia L., Cardona A., y Storti M. Full numerical quadrature of weakly singular double surface integrals in Galerkin boundary element methods. **International Journal for Numerical Methods in Biomedical Engineering**, 27(2):314–334, 2011.
- D’Elía J., Battaglia L., Cardona A., Storti M., y Ríos Rodríguez G. Galerkin boundary elements for a computation of the surface tractions in exterior Stokes flows. **Journal of Fluids Engineering- Trans ASME**, 2014. To appear.
- D’Elía J., Dalcín L., Sarraf S., López E., Battaglia L., Ríos Rodríguez G., y Sonzogni V. Primeros pasos en el uso de un modelo PGAS en el método de los elementos de borde. En C.G. Garino, A. Mirasso, y M. Storti, editores, *Mecánica Computacional*, volumen XXXII, páginas 961–969. 2013.
- D’Elía J., Dalcín L., Sarraf S., López E., Battaglia L., Ríos Rodríguez G., y Sonzogni V. Use of the PGAS model for high performance computing in Beowulf Clusters. En C. Garcia Garino

- y M. Printista, editores, *HPCLaTAm 2013*, páginas 210–215. High-Performance Computing Latin America Community, Mendoza, Argentina, 2013.
- Diaz J., Muñoz Caro C., y Niño A. A survey of parallel programming models and tools in the multi and many-core era. *IEEE Trans. Parallel Distr. Syst.*, 23(8):1369–1388, 2012.
- Galvis E., Yarusevych S., y Culham J.R. Incompressible laminar developing flow in microchannels. *J Fluids Eng- Trans ASME*, 134(1):014503–014503–4, 2012.
- Hebeker F.K. Efficient boundary element methods for three-dimensional exterior viscous flow. *Num. Meth. PDE*, 2(4):273–297, 1986.
- Lepchev D. y Weihs D. Low Reynolds number flow in spiral microchannels. *J Fluids Eng- Trans ASME*, 132(7):071202, 2010.
- Luszczek P., Kurzak K., y Dongarra J. Looking back at dense linear algebra software. *J. Parallel Distrib. Comput.*, 74(7):2548–2560, 2014.
- Magrab E.B. *Vibration of Elastic Systems with applications to MEMS and NEMS*. Springer, 2012.
- Méndez C., Paquay S., Klapka I., y Raskin J.P. Effect of geometrical nonlinearity on MEMS thermoelastic damping. *Nonlinear Anal. R. World Appl.*, 10(3):1579–1588, 2008.
- 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.
- Pozrikidis C. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press, 1997.
- Sarraf S., López E., Battaglia L., Ríos Rodríguez G., y D'Elía J. Amortiguamiento viscoso en un microresonador en flujo reptante simulado con elementos de borde. En *Mecánica Computacional*. 2014a.
- Sarraf S., López E., Ríos Rodríguez G., y D'Elía J. Validation of a Galerkin technique on a boundary integral equation for creeping flow around a torus. **Computational and Applied Mathematics**, 33(1):63–80, 2014b.
- Sauter S.A. y Schwab C. *Boundary element methods*. Springer, Springer-Verlag Berlin Heidelberg, 2011.
- Schreiber R. A few bad ideas on the way to the triumph of parallel computing. *J. Parallel Distrib. Comput.*, 74(7):2544–2547, 2014.
- Shipman T.N., Prasad A.K., Davidson S.L., y Cohee D.R. Particle image velocimetry evaluation of a novel oscillatory-flow flexible chamber mixer. *J Fluids Eng- Trans ASME*, 129(2):179–187, 2007.
- Tondi R., Cavazzoni C., Danecek P., y Morelli A. Parallel, large, dense matrix problems: Application to 3D sequential integrated inversion of seismological and gravity data. *Comput. Geosci.*, 48(0):143–156, 2012.
- Wang Z., Ghoniem N., Swaminarayan S., y LeSar R. A parallel algorithm for 3D dislocation dynamics. *J. Comp. Physics*, 219(2):608–621, 2006.