

**XXIV CONGRESO LATINOAMERICANO DE HIDRÁULICA
PUNTA DEL ESTE, URUGUAY, NOVIEMBRE 2010**

MEJORA DEL DESEMPEÑO COMPUTACIONAL DEL MODELO RMA-11

Pablo Ezzatti¹, Mónica Fossati², Ismael Piedra-Cueva², Gerardo Ares¹

*Universidad de la República, ¹Instituto de Computación, ²Instituto de Mecánica de los Fluidos e Ingeniería Ambiental.
J. Herrera y Reissig 565, Montevideo, Uruguay. pezzatti@fing.edu.uy, mfossati@fing.edu.uy, ismaelp@fing.edu.uy,
gares@fing.edu.uy*

RESUMEN:

Uno de los modelos numéricos en elementos finitos más utilizados para el estudio del transporte de sustancias y la calidad de agua en estuarios y ríos es el RMA-11. Desarrollado por I. King, este modelo trabaja acoplado con los modelos hidrodinámicos RMA-2 y/o RMA-10. Una de las desventajas del modelo es el alto costo computacional, es decir, el elevado tiempo de cálculo del modelo. En este trabajo se presenta el estudio preliminar de inclusión de estrategias de computación de alto desempeño al modelo RMA-11 realizado para mitigar el problema del tiempo de cálculo. En este sentido, se diseñó y desarrolló una modificación del modelo centrada en la etapa de resolución de los sistemas lineales. La propuesta implementada se evalúa por un lado según la mejora del costo computacional y además desde el punto de vista numérico utilizando como casos de prueba diversas aplicaciones del modelo en el Río de la Plata. Los resultados obtenidos muestran que los valores numéricos se mantienen pero se reportaron mejoras significativas en los tiempos de cómputo lo que representa el éxito de la implementación.

ABSTRACT:

RMA-11 is a numerical model widely used for studying the transport of constituents and water quality in rivers and estuaries. This model accepts velocities and depths inputs from hydrodynamic models like RMA-2 and RMA-10. When applied to large water systems like the Río de la Plata, RMA-11 demands long execution times to compute a simulation. This paper presents the analysis of the computational efficiency for the RMA-11 and introduces a proposal for improving the efficiency by using high performance computing techniques. The experimental results obtained when solving representative test cases show a significant improvement on the performance, achieving significant gains in computational speed: the execution times of the implemented version decreased up to one third of the time of the original implementation.

PALABRAS CLAVES:

Desempeño computacional, RMA-11, sistema lineales.

INTRODUCCIÓN

En las últimas décadas, el desarrollo de herramientas numéricas para simular la dinámica del movimiento de fluidos ha emergido como un área de intenso trabajo. En Uruguay, en el Instituto de Mecánica de los Fluidos e Ingeniería Ambiental (IMFIA) de la Facultad de Ingeniería de la Universidad de la República, se vienen desarrollando desde hace varios años tareas de investigación y de modelación del Río de la Plata y su Frente Marítimo. En un principio se construyeron modelos basados en la utilización de esquemas de diferencias finitas y luego con el fin de obtener mejoras en diversas características del modelo se realizó el pasaje del paradigma de diferencias finitas al de elementos finitos (MEF). Se utiliza el conjunto de modelos RMA, incluyendo el modelo hidrodinámico bidimensional integrado en vertical RMA-2, el modelo hidrodinámico baroclínico tridimensional RMA-10 y el modelo de transporte de sustancias RMA-11 que utiliza el cálculo hidrodinámico de los dos anteriores. Aplicaciones de estos modelos hidrodinámicos en su variante bidimensional han mostrado una excelente capacidad de simulación. Se han estudiado aspectos específicos de la circulación en el Río de la Plata, por ejemplo el comportamiento del campo de salinidades y su variación estacional (Fossati y Piedra-Cueva, 2008), la dinámica de sedimentos o el transporte de sustancias contaminantes descargadas en dicho cuerpo de agua (Lorenzo et al, 2002; Piedra-Cueva y Fossati, 2007).

Las cualidades de descripción de dichos modelos, su capacidad de trabajar con grilla irregular y con distintos tipos de elementos, y su facilidad para representar zonas costeras, han permitido la obtención de resultados sumamente alentadores en diversas simulaciones realizadas. En contraposición a la mejora de los resultados numéricos, se incrementan de forma abrupta los costos computacionales (tiempo de procesamiento). Esto implica un gran obstáculo para su utilización y para aplicar mejoras al modelo, como incrementar la precisión en el cálculo y expandir eficientemente el modelo a tres dimensiones. Teniendo en cuenta este problema, en trabajos previos se desarrolló una variante del modelo hidrodinámico RMA-10 utilizando técnicas de alto desempeño (Ezzatti y Piedra-cueva, 2005). Los resultados obtenidos en el mismo mostraron mejoras ampliamente satisfactorias en el desempeño computacional.

Además de la representación de las variables hidrodinámicas, corrientes y elevación de la superficie libre, es de interés para los investigadores del IMFIA realizar estudios de transporte de sustancias y calidad de agua. Para esto se puede utilizar el modelo de transporte RMA-11 (King, 1997), que trabaja de forma acoplada con los resultados obtenidos con los modelos hidrodinámicos RMA-2 ó RMA-10 (dependiendo si se trabaja con modelos hidrodinámicos barotrópicos o baroclínicos tridimensionales). El modelo RMA-11 es un modelo de transporte de sustancias y calidad de agua que utiliza una discretización en elementos finitos, y que permite realizar simulaciones unidimensionales, bidimensionales y tridimensionales del sistema en forma aislada o combinada. Algunos de los constituyentes que pueden representarse con el modelo son sedimentos cohesivos y no cohesivos, oxígeno disuelto, DBO, los componentes del ciclo del nitrógeno y del fósforo, coliformes, temperatura y salinidad.

La estructura de este trabajo se describe a continuación. En primer lugar se presentan los objetivos del trabajo y posteriormente se describe el modelo numérico RMA-11. La metodología empleada para la incorporación de técnicas de computación de alto desempeño y los casos de prueba desarrollados se describen posteriormente. Luego, se desarrollan las características de la propuesta realizada, y su evaluación posterior. Por último, se ofrecen las conclusiones arribadas en el trabajo y las líneas de trabajo futuro planteadas.

OBJETIVO

Disminuir los tiempos de cómputo que insume el modelo de transporte de sustancias RMA-11 aplicando estrategias de alto desempeño computacional. Esto posibilita realizar simulaciones de mayor porte o mejorar la precisión del mismo.

EL MODELO NUMÉRICO

Características Generales

El modelo RMA-11 (King, 1997) es un modelo utilizado en ingeniería para estudiar la calidad de agua de diversos cuerpos como estuarios, lagos, ríos y bahías. Está basado en una discretización en elementos finitos y permite realizar simulaciones unidimensionales, bidimensionales y tridimensionales del sistema. El modelo fue diseñado para resolver la ecuación de advección – difusión que gobierna el transporte de una sustancia en un cuerpo de agua. En el modelo se pueden representar diversos constituyentes, como salinidad, temperatura, oxígeno disuelto, nitrógeno, entre otros, para lo cual deben incluirse también las relaciones de decaimiento o crecimiento en función de la concentración. El campo de velocidades y profundidades en la zona de estudio es ingresado al modelo en diferentes formatos, entre otros *ASCII* y binarios. En particular, el modelo está diseñado para recibir las salidas producidas por el modelo hidrodinámico bidimensional RMA-2 (King, 1998) y el modelo tridimensional de flujo estratificado RMA-10 (King, 1993).

Algunas de las características que distinguen al RMA-11 sobre otros modelos similares son la capacidad de manejar configuraciones de borde irregulares, tamaños variable de elementos y la capacidad de representar zonas inundables poco profundas. El modelo puede representar tanto condiciones estacionarias como situaciones dinámicas. El campo de velocidades que se utiliza para resolver las ecuaciones puede ser constante y uniforme o puede ser el resultado de interpolaciones temporales de los datos contenidos en los archivos proporcionados por los modelos hidrodinámicos.

Para incrementar la eficiencia computacional el RMA-11 permite representar más de quince constituyentes al mismo tiempo. Para cada uno de los constituyentes debe definirse por separado la carga, condiciones iniciales y de decaimiento. El modelo permite simular la temperatura considerando el intercambio de calor con la atmósfera, los ciclos de los nutrientes nitrógeno y fósforo, la relación DBO-DO, el crecimiento de algas, los sedimentos cohesivo y no cohesivos como arenas, y otros constituyentes pasivos arbitrarios. En el caso que se simule el transporte de sedimento en suspensión cohesivo el modelo tiene incluido un modelo de fondo para representar los procesos de interacción entre la columna de agua y fondo, como la erosión, deposición y consolidación.

Otra característica importante del modelo RMA-11 es que su implementación resuelve las ecuaciones dinámicas mediante un esquema puramente implícito. Esto tiene la gran ventaja práctica de admitir pasos de tiempo relativamente grandes aún cuando se trabaje con una configuración de estado dinámico y se esté utilizando mallas de alta resolución.

Código del modelo

La implementación del modelo RMA-11 se compone de diversos módulos. La mayoría de las rutinas están codificadas en Fortran 77 mientras que algunos pocos módulos agregados recientemente están implementados en Fortran 90. Se dispone de versiones probadas del modelo para varios sistemas operativos, entre los que se destacan Linux, Unix y Windows (NT, XP y Server). En la Figura 1 se presenta el Algoritmo (pseudo-código) simplificado del modelo.

```
1: Inicialización
2: for 1:cantPasosSimulación do
3:   Calcular estructuras auxiliares
4:   for 1 : cantPasosNewton-Raphson do
5:     Resolver sistema lineal para calcular aporte
6:   end for
7:   Modificar las condiciones iniciales
8:   Registrar resultados
9: end for
10: Finalización
```

Figura 1.- Pseudo-código del RMA-11.

La primera etapa del Algoritmo (paso 1) corresponde a una componente de inicialización del modelo. En esta etapa se obtienen la grilla de cálculo y los parámetros globales, como son la cantidad máxima de pasos y el criterio de convergencia del método de resolución de sistemas no lineales, la cantidad de pasos de tiempo que se desea simular, etc. Luego, como se puede observar en el Algoritmo, entre los pasos 2 y 9 el programa se compone de un ciclo. En cada iteración del ciclo se efectúan los cálculos necesarios para realizar la simulación de un paso de tiempo. Esto implica resolver un sistema de elementos finitos no lineal, que utiliza de condición inicial el resultado calculado por el paso anterior del ciclo, exceptuando el primer paso en el cual se utilizan los valores ingresados en la etapa de inicialización.

El método de Newton-Raphson es utilizado para resolver los sistemas de ecuaciones no lineales (pasos 4 al 6 del Algoritmo), lo que implica la resolución de un sistema de ecuaciones lineales en cada paso de la iteración. El modelo RMA-11 permite la definición del criterio numérico de convergencia de la iteración de Newton-Raphson, así como establecer la cantidad máxima de iteraciones permitidas del método, siendo ambos valores parámetros obtenidos en la etapa de inicialización.

En las distintas aplicaciones dinámicas de MEF es necesario realizar en cada paso de tiempo (o en cada paso de la iteración de Newton-Raphson si se trabaja con un problema no lineal) el cálculo de los valores de los nodos de los distintos elementos de la malla. Estos valores se utilizan para la generación, luego de una etapa de ensamblado, de la matriz de rigidez. En el modelo RMA-11 se dispone de la rutina FRONTALL (y varias sub-rutinas) para realizar estas tareas que se corresponden al paso 5 del Algoritmo 1. Para efectivizar el cálculo de los coeficientes se utilizan distintas sub-rutinas. La sub-rutina utilizada se determina en tiempo de ejecución dependiendo del tipo y forma del elemento que se esté procesando. Para el cálculo de los valores de los nodos es necesario calcular integrales utilizando una cuadratura de Gauss.

La estrategia de resolución de los sistemas lineales utilizada es el método frontal, presentada por Irons en 1970 (Irons, 1970) y extendida para matrices no simétricas por Hood en 1976 (Hood, 1976). Los métodos frontales son versiones de la eliminación Gaussiana, relacionadas directamente con los elementos finitos y diseñadas en un principio para disminuir las necesidades de memoria de las estrategias estándar de resolución. Su estrategia de trabajo, que se basa en ir cargando en la memoria la matriz de a trozos (los distintos frentes), permitió a las distintas implementaciones de los métodos resolver grandes sistemas lineales en el hardware disponible en los años 70. En el caso de simular varios constituyentes en forma simultánea el sistema lineal posee varios vectores incógnitas. Esta situación es transparente en la etapa de factorización (factorización LU) pero es necesario hacer tantas sustituciones como sistemas a resolver.

Posteriormente, en los pasos 7 y 8 del algoritmo, se modifican las condiciones iniciales del siguiente paso de la iteración y se registran los resultados obtenidos respectivamente. Por último, el paso 10 del Algoritmo 1 corresponde a una etapa de finalización de la aplicación. Algunas de las tareas que se destacan en esta etapa son el registro de resultados finales, la liberación de recursos computacionales y la clausura de archivos.

METODOLOGÍA

Inicialmente se evalúan los tiempos de ejecución insumidos por las distintas etapas del modelo RMA-11 identificando las etapas de mayor consumo. Posteriormente se propone una mejora del desempeño computacional del modelo aplicando estrategias de alto desempeño.

Para lograr mejorar el desempeño computacional del modelo RMA-11 fue necesario realizar un estudio de los tiempos insumidos por las distintas rutinas originales, para luego diseñar las posibles variantes y por último comparar las distintas propuestas implementadas con la versión original. Esta comparación se realizó tanto desde el punto de vista del desempeño computacional como de los resultados numéricos obtenidos, permitiendo por un lado evaluar las posibles mejoras en el desempeño computacional alcanzadas y por otro lado validar los resultados numéricos obtenidos con la nueva versión. Para realizar las evaluaciones de desempeño computacional y de resultados numéricos fue necesario disponer de un conjunto de casos de prueba.

La mejora se concentra en la rutina encargada del cálculo de los coeficientes y posterior factorización de los sistemas lineales que se resuelven en cada paso del método de Newton-Raphson utilizado por el RMA-11 para resolver los sistemas no lineales derivados del modelo de elementos finitos. La modificación propuesta consiste en separar las etapas de cálculo, generación y factorización de la matriz de cada sistema lineal. En una primera etapa se calculan todos los coeficientes de la matriz, almacenándolos en una estructura auxiliar de tipo tabla de dispersión. Luego, se genera una matriz dispersa en algún formato preestablecido. Por último, se factoriza y se resuelve el sistema utilizando la biblioteca de uso público MULTifrontal Massively Parallel sparse direct Solver versión 4.6.3 (MUMPS) (Amestoy et al, 2001), en su modalidad secuencial. Durante sus diferentes etapas la biblioteca MUMPS necesita de las rutinas de los Basic Linear Algebra Subprograms (BLAS). En nuestro caso se utilizó la implementación de K. Goto optimizada para procesadores Pentium IV bajo sistemas operativos Linux de dicha biblioteca.

CASOS DE PRUEBA

Se utilizan seis casos de prueba que difieren en la cantidad de elementos y nodos con las que se discretiza el dominio, así como también en la cantidad de constituyentes que se simulan. Hay tres grillas distintas del Río de la Plata, denominadas M1, M2 y M3, que difieren en la cantidad de elementos, y consecuentemente nodos, utilizados para representar el dominio. La diferencia en las grillas se encuentra principalmente en la resolución utilizada para representar la zona costera de la ciudad de Montevideo, la capital de Uruguay. El dominio del modelo se muestra en la Figura 2(a), y un detalle de las grillas M1, M2 y M3 en la zona de Montevideo se muestran en la Figura 2(b), 2(c) y 2(d), respectivamente.

Además, para cada una de las grillas se implementó el modelo RMA-11 para simular un constituyente (S) y seis constituyentes (O). El primer caso corresponde a la simulación del transporte de sedimento en suspensión cohesivo. El segundo corresponde al estudio del oxígeno disuelto en el cual se simulan también la DBO y los componentes del ciclo de nitrógeno (nitrógeno orgánico, amoníaco, nitrato y nitrito). El modelo hidrodinámico para todas estas grillas ha sido validado utilizando resultados empíricos, niveles de agua y corrientes, medidos en diversas zonas del Río de la Plata (Piedra-Cueva y Fossati, 2007; Fossati y Piedra-Cueva, 2008)

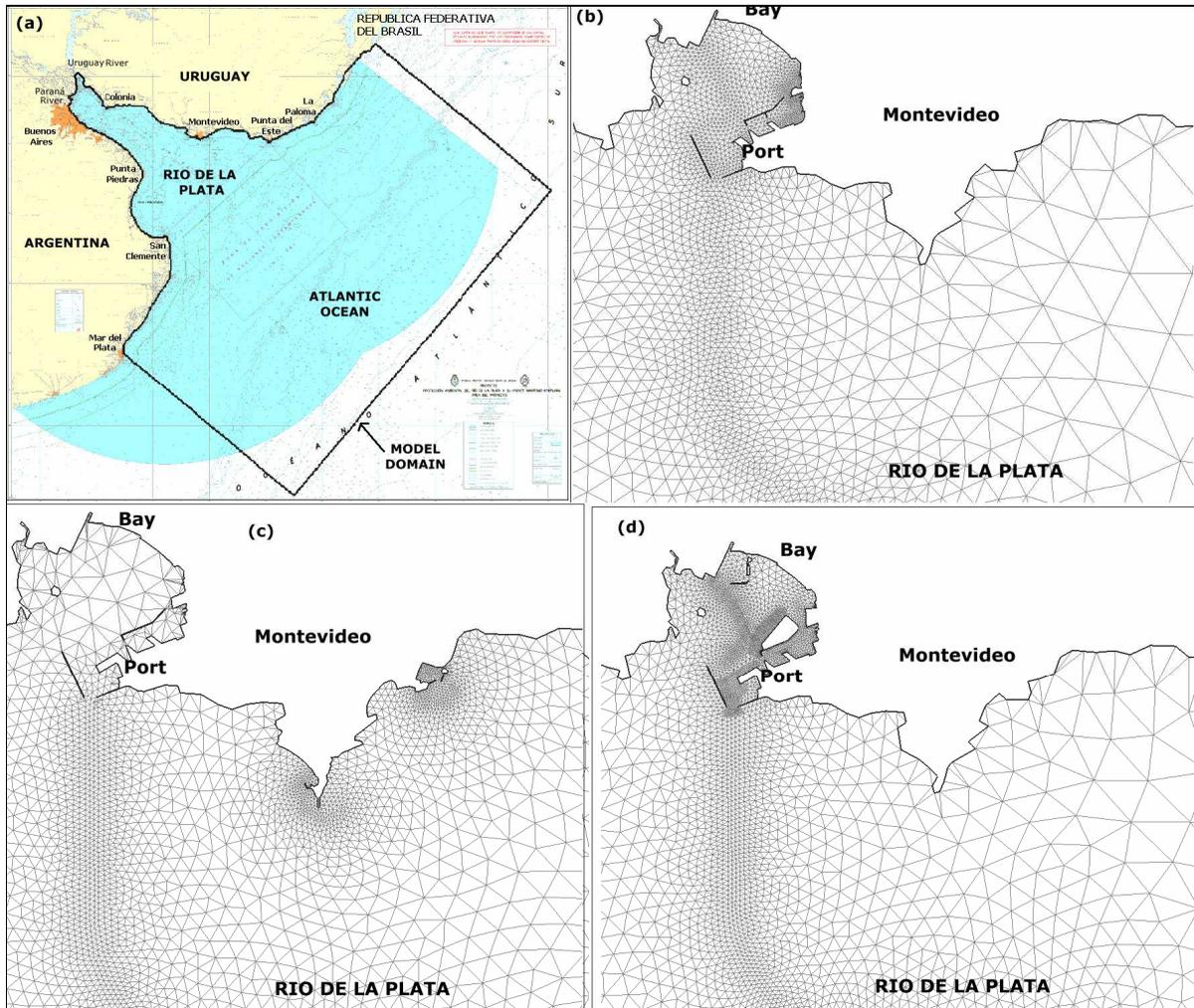


Figura 2.- (a) Ubicación del área de estudio y dominio del modelo. Detalle de la grilla de alta resolución en la zona costera de Montevideo (b) M1 (c) M2 y (d) M3.

La Tabla 1 resume las principales características de los distintos casos de prueba utilizados en el trabajo, detallándose cantidad de elementos y cantidad de nodos de la malla utilizada, cantidad de ecuaciones, cantidad de coeficientes no nulos presentes en las matrices asociadas y la cantidad de constituyentes a simular.

Tabla 1.- Principales características de los casos de prueba

Caso de Prueba	Número de elementos	Número de nodos	Número de ecuaciones	Cantidad de coeficientes no nulos	Número de constituyentes simulados
M1-S	7462	15526	13770	154340	1
M1-O	7462	15526	15460	173190	6
M2-S	10499	21618	21610	243850	1
M2-O	10499	21618	21560	243080	6
M3-S	12150	24960	24900	281270	1
M3-O	12150	24960	24900	281180	6

EVALUACIÓN DEL COSTO COMPUTACIONAL

En esta sección se presentan los resultados obtenidos al evaluar los tiempos de ejecución del modelo RMA-11 en su implementación original. Es importante hacer notar que para cada una de las versiones que se evalúan en este trabajo, original y propuestas, se realizó una etapa preliminar en la cual se buscó optimizar las opciones del compilador, buscando los mejores tiempos de cómputo sin perder calidad en los resultados. Como consecuencia, los tiempos de ejecución que se presentan en lo que resta del trabajo son los resultantes de aplicar a las distintas versiones utilizadas las optimizaciones de compilación. En cuanto a la configuración del criterio de parada del método Newton- Raphson se ha utilizado en forma general como cota superior de la iteración 4 pasos.

Tiempos de ejecución de las distintas etapas del modelo

Para realizar esta etapa del trabajo se utilizó el caso de prueba denominado caso M3-O que cuenta con 12150 elementos y 24960 nodos de cálculo donde se simulan 6 constituyentes. Todas las pruebas se realizaron en un computador Dell, denominando Obras12, que dispone de dos procesadores Pentium IV de 2.8 GHz, 1 GB RAM y sistema operativo Linux distribución Debian 4.1.1. Se empleó el compilador Fortran Intel 9.0.

En la Tabla 2 se resumen los tiempos de ejecución evaluados para las distintas etapas del modelo RMA-11 especificadas en el Algoritmo. Considerando los resultados resumidos en la Tabla 2, se puede concluir que la mayor fuente de consumo de tiempo de cómputo es la etapa de cálculo y resolución de las matrices de rigidez. Esta etapa se realiza en cada paso del método Newton-Raphson para la resolución de los sistemas no lineales y es efectuada por la rutina FRONTALL.

Tabla 2.- Tiempos de ejecución de las distintas etapas del modelo RMA-11.

Etapas	Tiempo [seg]	Desviación Estándar
Inicialización	0.59	0.01
Calculo de estructuras auxiliares	0.20	0.01
Resolución sistema lineal para calcular aporte	3.70	0.19
Modificación condiciones iniciales y registro resultados	0.42	0.01
Finalización	0.29	0.00

Es importante considerar que los tiempos de ejecución de las etapas de inicialización y finalización son independientes de la cantidad de pasos a simular. Esto implica que al trabajar con simulaciones de periodos importantes su influencia en el tiempo total será todavía menor que en el caso evaluado. Precisamente en el caso evaluado, los valores muestran que los tiempos de ejecución de las etapas mencionadas no son significativas, siendo más cortos que los requeridos para efectuar un paso de la iteración del método de Newton-Raphson. Algo similar ocurre con las etapas de cálculo de las estructuras auxiliares y modificación de las condiciones iniciales que se ejecutan una única vez por paso de tiempo.

Tiempos de ejecución de la rutina FRONTALL

Debido a que la rutina FRONTALL resultó ser la más costosa se realizó una segunda instancia de estudio orientada a evaluar los tiempos de cómputo de las distintas etapas de la rutina.

Como se mencionó anteriormente, la rutina FRONTALL utiliza el método frontal para la resolución de los sistemas lineales. El método no necesita realizar el cálculo de todos los coeficientes de la matriz de rigidez para comenzar a efectuar la factorización, sino que su modalidad de trabajo es ir evaluando los distintos elementos (calculando los coeficientes que aportan) y factorizar los distintos frentes (subconjunto de filas y columnas de la matriz de rigidez, que poseen todos los coeficientes calculados), de forma de utilizar una menor cantidad de memoria. Sin embargo, para intentar identificar las zonas críticas del código, se delimitaron y calcularon los tiempos de los tres fragmentos lógicos que a priori posee la rutina: el cálculo de los coeficientes, la factorización de los distintos frentes de la matriz dispersa y la resolución. Para calcular los tiempos de cada una de las etapas lógicas de la rutina FRONTALL se efectuó la sumatoria de los tiempos de ejecución de las partes que las componen. Por ejemplo, el tiempo de factorización se calculó como la sumatoria de los tiempos de ejecución para factorizar cada uno de los frentes.

En la Tabla 3 se puede observar la relación de tiempos de cómputo de las tres partes identificadas de la rutina FRONTALL. Los tiempos presentados son el promedio de 10 ejecuciones. En base a los tiempos calculados se puede concluir que la factorización es en forma categórica la etapa más costosa de la rutina FRONTALL.

Tabla 3.- Tiempos de ejecución de las distintas etapas lógicas de la rutina FRONTALL.

Etapas	Tiempo [seg]	Desviación Estándar
Cálculo de coeficientes	0.19	0.01
Factorización	3.51	0.19
Sustitución	0.00	0.00
Total	3.70	0.20

PROPUESTA

El trabajo presenta una aplicación de estrategias de computación de alto desempeño al modelo RMA-11 y en particular en la búsqueda de mejoras de la rutina FRONTALL que, como se muestra en la sección anterior, es la etapa crucial del código en cuanto a costo computacional del modelo.

Las técnicas de alto desempeño computacional se basan por un lado en explotar al máximo las capacidades de cómputo del hardware disponible y por otro en utilizar estrategias que minimicen las necesidades de cómputo para la resolución del problema. Para lograr estos objetivos generalmente se utilizan métodos específicos para la resolución de las distintas problemáticas. Estos métodos se basan en artificios que facilitan la utilización de la menor cantidad de recursos, en muchos casos teniendo en cuenta las distintas características de las arquitecturas de computadoras utilizadas (memorias, procesadores, conexiones, etc).

La propuesta utilizada en este trabajo consiste en cambiar la metodología de trabajo que utiliza la rutina FRONTALL. Como primera fase, se establece una etapa de generación de la matriz de rigidez y luego se utiliza un algoritmo para factorizar y resolver los sistemas lineales. Debido a las importantes dimensiones de las matrices de rigidez se hace necesario utilizar estrategias de almacenamiento disperso para su manejo. Es importante aclarar que existen diversas bibliotecas para la resolución de sistemas lineales involucrados en los MEF que permiten ingresar la matriz en formato elemental, es decir ir agregando los coeficientes asociados a cada elemento. Si se utiliza una biblioteca con las características mencionada no es necesario disponer de una etapa de generación de la matriz en formato disperso. Sin embargo, por lo general los tiempos de ejecución de este tipo de estrategias son superiores a los obtenidos al utilizar estrategias que trabajan directamente sobre la estructura de datos.

La lógica e implementación de la nueva rutina FRONTALL se puede dividir (a diferencia de la rutina FRONTALL original) en tres fases independientes que se describen a continuación: cálculo de los coeficientes, generación de la matriz y resolución del sistema.

- Cálculo de los coeficientes: calcula todos los coeficientes de la matriz aportados por los distintos elementos. Los valores calculados se cargan en una estructura auxiliar.
- Generación de la matriz: en base a los resultados de los cálculos de la etapa previa, filtrando y reordenando distintos coeficientes se genera la matriz dispersa en algún formato.
- Resolución del sistema: se realiza la factorización y resolución del sistema lineal.

Cálculo de los coeficientes

La etapa de cálculo de los coeficientes se compone de una iteración sobre los elementos de la malla. Para cada elemento se calcula el aporte a los coeficientes de la matriz de cada uno de los nodos para cada una de las variables. Se utilizan las rutinas originales del sistema, (COEFXX), que calculan los coeficientes y los almacenan en una estructura local.

A continuación se almacenan los valores anteriormente calculados en una estructura auxiliar de tipo tabla de dispersión abierta (Aho et al, 1983) para almacenar globalmente toda la matriz. La estrategia de las tablas de dispersión (comúnmente denominadas hash) se basa en dividir el conjunto de datos a almacenar en una cantidad específica de clases (también denominadas cubetas). Para esta tarea es necesario la definición de una función de dispersión la cual para cada objeto del conjunto de datos devuelve una única clase. Al utilizar hash abiertos cada cubeta de la tabla de dispersión dispone de una lista de elementos. En esta propuesta la función de dispersión utilizada queda definida por la fila del coeficiente a almacenar como se presenta en la Ecuación 1 y la lista de las cubetas se compone de varios nodos en donde cada uno de ellos es una dupla en el que se dispone del indicador de columna y el valor.

$$f_{\text{dispersion}}(\text{coef}_{ij}) = i \quad [1]$$

Para cargar los datos se aplica la función de dispersión a cada coeficiente (por ej. coef_{ij}), y una vez obtenida la cubeta (la i en el ej.) de la tabla de dispersión, se recorre la lista que tiene asociada la cubeta buscando si ya existe alguna entrada cuyo valor de columna sea igual al de la columna del coeficiente a almacenar (j en el ej.). En caso afirmativo se suma al valor de la entrada el valor del coeficiente calculado. En caso contrario se agrega un nuevo nodo al final de la lista con el valor e indicador de columna correspondiente.

Al finalizar esta etapa de la rutina se cuenta con todos los coeficientes de la matriz en un formato disperso simple. Sin embargo, por la forma de construcción pueden existir entradas de fila y columna que posean valores nulos.

Generación de la matriz

La segunda etapa de la rutina propuesta consiste en llevar a cabo diversas tareas de depuración, que se realizan para poder invocar al método de resolución que se esté utilizando. Entre otras tareas de depuración se pueden filtrar los coeficientes nulos que se hayan generado, ordenar los coeficientes y/o generar las estructuras auxiliares de forma de disponer de la matriz en el formato necesario para la siguiente etapa. Si bien conceptualmente esta etapa es adaptable para cualquier solver, se desarrolló una versión particular para la biblioteca utilizada. En este caso se filtran los coeficientes

nulos y no se establece ningún orden (más que el dado por la agrupación de los coeficientes de las mismas filas), ya que la biblioteca utilizada en la etapa de factorización no lo requiere.

Resolución

La última etapa de la nueva rutina se encarga de la factorización y resolución del sistema lineal de ecuaciones. Para esta tarea se utiliza la versión serial de la biblioteca de uso libre MULTifrontal Massively Parallel sparse direct Solver (MUMPS) 4.6.3 (Amestoy et al, 2001), biblioteca que implementa los métodos multifrontales. Dado que la biblioteca MUMPS necesita disponer de la biblioteca BLAS en el trabajo se utilizó la implementación de K. Goto optimizada para procesadores Pentium IV bajo sistemas operativos Linux.

RESULTADOS

En esta sección se presentan los estudios llevados a cabo con el objetivo de evaluar la versión de la rutina FRONTALL propuesta. Los estudios se realizaron sobre los casos de prueba descritos en la Sección 5.1 utilizando en forma exclusiva el computador obras12, descrito anteriormente.

En primer término se presenta la evaluación de los resultados numéricos de la propuesta aplicada sobre el caso M3-O realizando una simulación de 544 pasos de tiempo. Luego, se comparan los tiempos de cómputo al realizar simulaciones con la versión original del modelo RMA-11 y con la versión propuesta.

Resultados numéricos

No es necesario profundizar en detalles sobre la necesidad de que la versión propuesta logre obtener resultados numéricos similares a los que se obtienen con la versión original. Es importante resaltar la dificultad de alcanzar dicho objetivo, considerando que las matrices involucradas en la resolución del modelo RMA-11 aplicado al Río de la Plata son altamente mal condicionadas, generando que pequeñas perturbaciones en los cálculos puedan implicar grandes cambios en los resultados numéricos en pocos pasos de simulación.

Con el objetivo de realizar la validación desde el punto de vista numérico de la propuesta se comparó el resultado obtenidos de efectuar una simulación de 544 pasos de tiempo (equivalente a 136 horas) para el caso M3-O con la versión propuesta y la original.

La comparación numérica se realiza únicamente en el último paso de tiempo ya que en ese instante se arrastran todos los errores cometidos anteriormente. En este sentido, en la Tabla 4 se presenta la distancia, medida con la norma 2, entre los resultados obtenidos en el paso de tiempo 544 por la versión original y la propuesta. También se presenta la mayor diferencia por constituyente encontrada.

Tabla 4.- Diferencias en los resultados numéricos de la versión original y propuesta..

Constituyente	Norma2 de la diferencia	Máxima diferencia
OD	4.53×10^{-04}	5.15×10^{-05}
DBO	1.10×10^{-04}	3.05×10^{-05}
Norg	7.19×10^{-05}	2.86×10^{-06}
NH ₃	7.50×10^{-06}	1.91×10^{-06}
NO ₂	2.30×10^{-07}	8.94×10^{-08}
NO ₃	3.15×10^{-07}	4.47×10^{-08}

Estudiando la Tabla 4, se puede afirmar que ambas versiones obtuvieron resultados sumamente similares, ya que las diferencias en los resultados numéricos finales en ninguna ocasión superan el valor 1.00×10^{-03} .

Evaluación de tiempos de cómputo

En cuanto a la evaluación de los tiempos de cómputo, se realizó la comparación por un lado de los tiempos de ejecución del modelo original y de la variante propuesta, y por otro lado de los tiempos de las rutinas FRONTALL original y la propuesta.

Los tiempos de ejecución de realizar una simulación de 544 pasos de tiempo para el caso M3-O utilizando el modelo en la versión original y la propuesta obtenidos fueron 124.7 y 56.1 minutos, respectivamente. Se puede observar que el tiempo de ejecución de la versión propuesta es menor a la mitad del tiempo implicado por la versión original.

En cuanto a los tiempos que insume la nueva rutina FRONTALL en la Tabla 5 se presentan los tiempos de ejecución promedio de cada una de las tres etapas especificadas en la sección de diseño de la versión propuesta, así como el tiempo de ejecución de la rutina en su totalidad.

Tabla 5.- Tiempo de cálculo de las distintas etapas de la rutina FRONTALL propuesta.

Etapas	Tiempo [seg]	Desviación Estándar
Cálculo de coeficientes	0.66	0.02
Reordenamiento	0.00	0.00
Resolución	0.68	0.02
Total	1.34	0.02

En la Tabla 6 se comparan los tiempos de ejecución de la rutina FRONTALL original y propuesta para los distintos casos de prueba. Detallándose el promedio del tiempo de ejecución de la rutina original y propuesta y las desviaciones estándar asociadas.

Tabla 6.- Tiempos de ejecución de la FRONTALL original y propuesta para los diferentes casos de prueba.

Caso de Prueba	Original [seg]	Desviación Estándar	Propuesta [seg]	Desviación Estándar
M1-S	0.55	0.00	0.56	0.00
M1-O	1.01	0.00	0.91	0.00
M2-S	2.54	0.12	0.91	0.01
M2-O	3.80	0.13	1.34	0.03
M3-S	2.36	0.06	1.07	0.02
M3-O	3.70	0.19	1.34	0.02

Estudiando los resultados de la Tabla 6 se puede observar que la rutina FRONTALL original implica hasta casi 3 veces el tiempo que insume la rutina propuesta. Además, si comparamos el tiempo de ejecución de la etapa de factorización de la rutina original con la suma de los tiempos de cargar la estructura auxiliar, el reordenamiento, más el tiempo de factorización y resolución del sistema lineal de la rutina FRONTALL propuesta, es posible concluir que la relación de costo es de aproximadamente 4 veces.

Otra característica positiva, no documentada, de la versión propuesta es que utiliza una cantidad menor de memoria RAM para efectuar las mismas simulaciones que la versión original.

CONCLUSIONES y TRABAJO FUTURO

En el artículo se presenta el estudio del modelo numérico RMA-11 y una propuesta de modificación de la estrategia de resolución de los sistemas lineales obtenidos al aplicar el método de Newton-Raphson para resolver los sistemas no lineales. El modelo ha sido aplicado con éxito para representar el transporte de sustancias en el Río de la Plata.

En el trabajo se propone, a diferencia de la versión original, separar la generación de la matriz de rigidez de la resolución propiamente dicha del sistema. Para llevar adelante dicho objetivo se expone la utilización de una estructura auxiliar de tipo tabla de dispersión, de forma de almacenar temporalmente los coeficientes y luego invocar una etapa de reordenamiento y consolidación de coeficientes para generar la matriz en algún formato disperso preestablecido. Por último, se resuelve el sistema de ecuaciones mediante el uso de la biblioteca MUMPS 4.6.3 en su versión secuencial, utilizando la implementación de la biblioteca BLAS de K. Goto optimizada para la maquina que se utilizó. Se compararon los resultados numéricos obtenidos por ambas versiones al efectuar una simulación de porte mediano. Los resultados muestran que con la nueva versión se obtienen valores de igual calidad que la versión original ya que las diferencias entre los resultados de ambas nunca superaron el valor 1.00×10^{-03} .

En cuanto al desempeño computacional, evaluado en tiempo de ejecución, la versión propuesta mostró una superioridad con respecto a la versión original ya que las pruebas efectuadas mostraron que la versión original necesita, en los casos estudiados, hasta casi tres veces más de tiempo que la versión propuesta para efectuar la misma simulación. En las aplicaciones del RMA-11 en el Río de la Plata deben simularse generalmente períodos largos de tiempo, cercanos al año, por lo que la versión propuesta del modelo reduce considerablemente los plazos de tiempo necesarios actualmente para estos estudios.

La mejora del desempeño computacional del RMA-11 alcanzada permite ampliar su utilización, posibilitando en tiempos de ejecución razonables, mejorar la definición de las grillas y hacer viables simulaciones de períodos de tiempos que antes eran prohibitivos desde este punto de vista. Sin embargo, el afán por simular dominios de dimensiones mayores y utilizando mejor resolución, motiva el estudio de arquitecturas de hardware tipo cluster como línea de trabajo futuro.

REFERENCIAS BIBLIOGRÁFICAS

Aho, A., Hopcroft, J. and Ullman, J. (1983). Data Structures and Algorithms. Addison-Wesley, 1983.

Amestoy P, Duff, I, Koster J, and J-Y. (2001) L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, Vol 23 N 1:15–41.

Cathy A; Anderson Jamie D; Weinberg, Marca; Lawrence and Jonathan R Randall. (2002). Biological and economic implications of sacramento watershed management options. In *Journal of the American Water Resources Association*, April 2002.

Ezzatti P, y Piedra-Cueva I. (2005). Mejora del desempeño computacional del RMA-10. *VIII Congreso Argentino de Mecánica Computacional (MECOM05)* Argentina.

Fossati, M. y Piedra-Cueva, I. (2008). Numerical modelling of residual flow and salinity in the Río de la Plata. *Applied Mathematical Modelling*, 32:1066–1086, June 2008.

Hood P. (1976). Frontal solution program for unsymmetric matrices. *International Journal for Numerical Methods in Engineering*, 10:379–400, 1976.

Irons, B. (1970). A frontal solution program for finite element analysis. *International Journal for Numerical Methods in Engineering*, 2, 2:5–32, 1970.

King, I. (1993). RMA-10, a finite element model for three-dimensional density stratified flow. Report prepared in co-operation with Australian Water and Coastal Studies for the Sydney Deepwater Outfalls Environmental Monitoring Program Post Commissioning Phase, 1993.

King, I. (1997). Rma-11: A three-dimensional finite element model for water quality in estuaries and streams version 2.5. Technical report, Dept. of Civil and Environmental Engineering, University of California Davis, California.

King, I. (1998). RMA-2, a two dimensional finite element model for flow in estuaries and streams. Report Dept. of Civil Engineering. University of California. Davis, California USA, 1998.

Lorenzo E., Piedra-Cueva I. y Fossati M. (2002). Modelación numérica del futuro emisario punta lobos (Montevideo). In *XX Congreso Latinoamericano de Hidráulica*, La Habana, Cuba, Octubre 2002.

Piedra-Cueva I. y Fossati, M. (2007). Residual currents and corridor of flow in the Río de la Plata. *Applied Mathematical Modelling*, 31:564–577, March 2007.