PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 09-10

Estudio inicial del modelo MOHID

Ignacio Barreto Pablo Ezzatti Mónica Fossati

2009

Estudio inicial del modelo MOHID
Barreto, Igancio; Ezzatti, Pablo; Fossati, Mónica
ISSN 0797-6410
Reporte Técnico RT 09-10
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay, mayo de 2009

Estudio inicial del modelo MOHID

Ignacio Barreto[1], Pablo Ezzatti[1], Mónica Fossati[2]

[1] Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay
{ibarreto, pezzatti}@fing.edu.uy

[2] Instituto de Mecánica de los Fluidos, Facultad de Ingeniería, Universidad de la República, Uruguay
mfossati@fing.edu.uy

Resumen

Este trabajo presenta el estudio preliminar del modelo numérico tridimensional MOHID. Este modelo simula el flujo en diversos cuerpos de agua, tanto ríos, como estuarios y océanos. Actualmente el modelo está siendo utilizado por un grupo de investigación perteneciente al grupo Hidráulica Fluvial y Marítima del Instituto de Mecánica de Fluidos e Ingenieria Ambiental para estudiar diferentes procesos en el Río de la Plata.

En particular se aplica el MOHID para estudiar la hidrodinámica del Río de la Plata con gran precisión en su resolución utilizando la metodología de modelos encajados. A través de esta metodología, es posible anidar grillas de resolución espacial creciente, forzando los modelos locales con resultados de aplicaciones de mayor escala. De esta forma, el modelo permite estudiar áreas cada vez más cercanas a la región de interés, a partir del traspaso de las condiciones de borde del modelo "padre" hacia la grilla anidada ("hijo"). Por otra parte, se está desarrollando la modelación del Atlántico Sur acoplada con un modelo atmosférico, para generar y propagar las oscilaciones meteorológicas hacia el Río de la Plata. Este tipo de aplicaciones requieren de una gran capacidad de cálculo para resolver los sistemas de ecuaciones resultantes. En consecuencia, siguiendo el camino de otros usuarios de este tipo de modelos, se propone trabajar en el diseño y la implementación de versiones paralelas del código, posibilitando su ejecución en plataformas paralelas de memoria distribuida.

En general, este tipo de modelos comienzan siendo programas pequeños, que resuelven alguna problemática en particular, que evolucionan con el tiempo en base al trabajo de muchos investigadores (generalmente más preocupados por la situación a modelar que por el modelo en sí mismo). Esta situación causa serios problemas en cuanto al diseño y documentación de los modelos. Entonces, el objetivo de este primer documento de trabajo es intentar unificar la diversa información obtenida del modelo sobre la base de distintos métodos: estudio de la documentación, interpretación del código fuente, deducciones basándose en experimentos con las ejecuciones, etc. Se pretende además que el documento sirva de insumo para los trabajos futuros planteados (paralelizar).

Índice

1 *Introducción*

Este trabajo presenta el estudio preliminar del modelo numérico tridimensional MOHID [25]. Este modelo simula el flujo en diversos cuerpos de agua, tanto ríos, como estuarios y océanos. La importancia en el estudio del modelo radica en que actualmente el MOHID está siendo utilizando por un grupo de investigación perteneciente al grupo Hidráulica Fluvial y Marítima del Instituto de Mecánica de Fluidos e Ingenieria Ambiental (IMFIA) para simular diferentes procesos en el Río de la Plata [1].

En particular se aplica el MOHID para estudiar la hidrodinámica del Río de la Plata con gran precisión en su resolución utilizando la metodología de modelos encajados. A través de esta metodología, es posible anidar grillas de resolución espacial creciente, forzando los modelos locales con resultados de aplicaciones de mayor escala. De esta forma, el modelo permite estudiar áreas cada vez más cercanas a la región de interés, a partir del traspaso de las condiciones de borde del modelo “padre” hacia la grilla anidada (“hijo”). Por otra parte, se está desarrollando la modelación del Atlántico Sur acoplada con un modelo atmosférico, para generar y propagar las oscilaciones meteorológicas hacia el Río de la Plata [2]. Este tipo de aplicaciones requieren de una gran capacidad de cálculo para resolver los sistemas de ecuaciones resultantes. En consecuencia, siguiendo el camino de otros usuarios de este tipo de modelos, se propone trabajar en el diseño y la implementación de versiones paralelas del código, posibilitando su ejecución en plataformas paralelas de memoria distribuida mediante el paradigma de pasaje de mensajes. De acuerdo a la información relevada, mediante la paralelización se espera lograr una significativa mejora de los tiempos de procesamiento, alcanzando valores de speedup que permitan efectuar simulaciones reales del Río de la Plata en tiempos de ejecución razonables. La primera etapa del trabajo de mejora del desempeño implica el estudio del MOHID, logrando un grado de comprensión importante del modelo.

En general, este tipo de modelos comienzan siendo programas pequeños, que resuelven alguna problemática en particular, que evolucionan con el tiempo en base al trabajo de muchos investigadores (generalmente más preocupados por la situación a modelar que por el modelo en sí mismo). Esta situación causa o genera serios problemas en cuanto al diseño y documentación de los modelos. Entonces, el objetivo de este primer documento es intentar unificar la diversa información obtenida en base a distintos métodos (estudio de la documentación, interpretación del código fuente, deducciones en base a experimentos con las ejecuciones, etc.). Es importante destacar que este documento no intenta ser una versión final, sino solo una puesta a punto que sirva de insumo para los trabajos futuros planteados, en particular, la aplicación al modelo de estrategias de alto desempeño.

En el resto del documento se presenta diversa información del MOHID que fue relevada hasta el momento, enfocándose en los ítems de interés para las aplicaciones realizadas en el IMFIA.

2 Revisión Histórica

El modelo MOHID fue desarrollado por investigadores del MARETEC [3] (Centro de Ambiente y Tecnologías Marítimas) perteneciente al Instituto Superior Técnico (IST) [4] de la Universidad Técnica de Lisboa. MOHID es un sistema de modelación en tres dimensiones de la dinámica de diversos cuerpos de agua como ríos, estuarios y océanos. El nombre MOHID es un acrónimo de MOdelo HIDrodinámico (en portugués MOdelo HIDrodinâmico).

La primera versión del modelo fue desarrollada en el año 1985. En sus comienzos, el modelo permitía realizar simulaciones únicamente en dos dimensiones. Utilizaba el paradigma de diferencias finitas [5] para la discretización del dominio. Estas primeras versiones del modelo estaban implementadas en Fortran 77.

En el año 1995 se introdujo la primera versión tridimensional del modelo en el trabajo de Santos [6]. Esta versión se basaba en un modelo de coordenadas verticales llamado double Sigma [7]. Con la intención de generalizar el uso de diferentes sistemas de coordenadas es que en el año 1999 se pasa a trabajar con el paradigma de volúmenes finitos. Trabajo presentado por Martins [8].

Luego se introdujeron varias mejoras al MOHID, siendo una de las más destacadas el aporte de Leitão [9] que integró los modelos de transporte euleriano y lagangiano en 3D.

Posteriormente, debido al agregado de mayores funcionalidades y la capacidad de trabajar con modelos en tres dimensiones, se hizo necesario el cambio de paradigma de programación y consecuentemente del lenguaje utilizado. Se migró el código para adaptarse al paradigma de orientación a objetos, siendo Fortran 95 el lenguaje elegido para la nueva implementación. Este cambio permitió el diseño basado en clases (si bien este lenguaje no soporta por completo el concepto de clases, esta versión permite aplicar diversos conceptos de este paradigma de programación).

Actualmente, MOHID está integrado por diferentes herramientas como lo son MOHID Water, MOHID Soil y MOHID Land. Estas herramientas pueden ser usadas para estudiar el ciclo del agua en un ambiente integrado, permitiendo simular procesos físicos y biogeoquímicos tanto en el agua como en los sedimentos.

Desde el punto de vista informático se puede catalogar como un sistema de tamaño considerablemente grande. Actualmente MOHID está compuesto por más de 60 módulos que implican más de 300 mil líneas de código.

A continuación se muestran algunos de los estudios más destacados realizados hasta el momento que utilizaron el modelo MOHID.

Un resumen de estudios de estuarios como los de Minho, Lima, Douro, Mondego, Tejo, Sado, Mira, Arade y Guadiana y lagunas costeras tales como Ria de Aveiro y Ria Formosa se pueden encontrar en el trabajo de Martins [10]. El modelo fue también utilizado en algunos ríos de Galicia (Río de Vigo) como se muestra en los trabajos realizados por Taboada [11], Montero [12], así como la aplicación de MOHID al Río de Pontevedra en el estudio elaborado por Taboada y Villarreal [13].

Como se puede ver, MOHID se ha aplicado en la costa Atlántica y en la península Ibérica. Por otro lado, se han modelado algunos estuarios que se encuentran en el resto de Europa, por ejemplo, el estuario de Western Scheldt que se encuentra en el suroeste de los Países Bajos, el estuario de Gironde en Francia, todas estas investigaciones realizadas por Cancino y Neves [14].

MOHID ha sido aplicado más recientemente a diversas reservas de agua dulce en Portugal, como Monte Novo, Roxo y Alqueva, según el trabajo de Braunschweig[15], con el objetivo de estudiar el flujo y calidad del agua en esos lugares.

3 Descripción general de MOHID

A continuación se presentan algunas de las principales características del framework MOHID.

El modelo utiliza una aproximación de volúmenes finitos para discretizar el dominio del problema. En el Anexo A se puede encontrar una pequeña explicación del paradigma de volúmenes finitos y en el libro de Ferziger y Peric [16] se presenta una cobertura exhaustiva del tema. La aproximación basada en volúmenes finitos permite usar coordenadas verticales genéricas, dependiendo del proceso principal del área a estudiar o investigar. En el caso de MOHID se puede utilizar el modelo para realizar simulaciones en una, dos o tres dimensiones.

El diseño de MOHID, en las últimas versiones, está basado en el paradigma de orientación a objetos; mientras que para la implementación se utilizó el lenguaje Fortran 95. El diseño establece una jerarquía de módulos permitiendo así que cada módulo se encargue de manejar la información específica referente a su propósito.

Hay versiones del modelo implementadas para correr tanto sobre sistemas operativos Windows como sobre Linux. Además ofrece lineamientos para poder extender el modelo para ser utilizado en otros sistemas operativos.

MOHID posee una interfaz gráfica (MOHID GUI) para utilizar los ejecutables pero solamente es posible ejecutarlo en el sistema operativo Windows, ya que la interfaz está implementada sobre la plataforma Microsoft .NET. Por otro lado, se permite invocar el modelo a través de la línea de comandos.

Una de las principales características de MOHID es que implementa la metodología de modelos encajados. A través de esta estrategia es posible anidar grillas de resolución espacial de manera creciente, forzando los modelos locales con resultados de aplicaciones de mayor escala. De esta forma, el modelo permite estudiar áreas cada vez más cercanas a la región de interés, a partir del pasaje de las condiciones de borde del modelo "padre" hacia el modelo anidado ("hijo"), aumentando en general la resolución. Una particularidad importante del modelo es que permite utilizar diferentes pasos de tiempo en los diferentes modelos encajados, restringiendo únicamente que los pasos de tiempo del modelo padre sean múltiplos de los pasos de tiempo de los modelos hijos.

Al utilizar la estrategia de modelos encajados es posible la ejecución en paralelo de dichos modelos. El paradigma de paralelismo utilizado por el framework MOHID es el de pipeline, y además, permite ejecutar diferentes modelos en diferentes procesadores comunicados a través del paradigma de pasaje de mensajes utilizando el estándar MPI [17]. Es importante hacer notar que la opción de emplear la versión paralela, obliga a que los pasos de tiempos de los diferentes modelos sean iguales (entre el padre y el hijo) para que no ocurran incoherencias en los valores que estén simulando los diferentes procesos.

En cuanto a la organización del Framework MOHID, se pueden identificar tres componentes principales:

- **Herramientas numéricas,** permiten el cálculo de los diferentes modelos y otras herramientas auxiliares.
- **Traspaso y almacenamiento de datos,** se encarga de salvar la información producida por el modelo.
- **Interfaces de usuario (GUI's),** permite el manejo de datos de entrada, control de ejecución del programa y análisis de resultados.

En las siguientes secciones se explica con más detalle las características de la arquitectura así como de los componentes principales del Framework MOHID.

3.1 Arquitectura de MOHID

Los módulos de MOHID están estructurados según una jerarquía, la cual se presenta en la Figura 1.

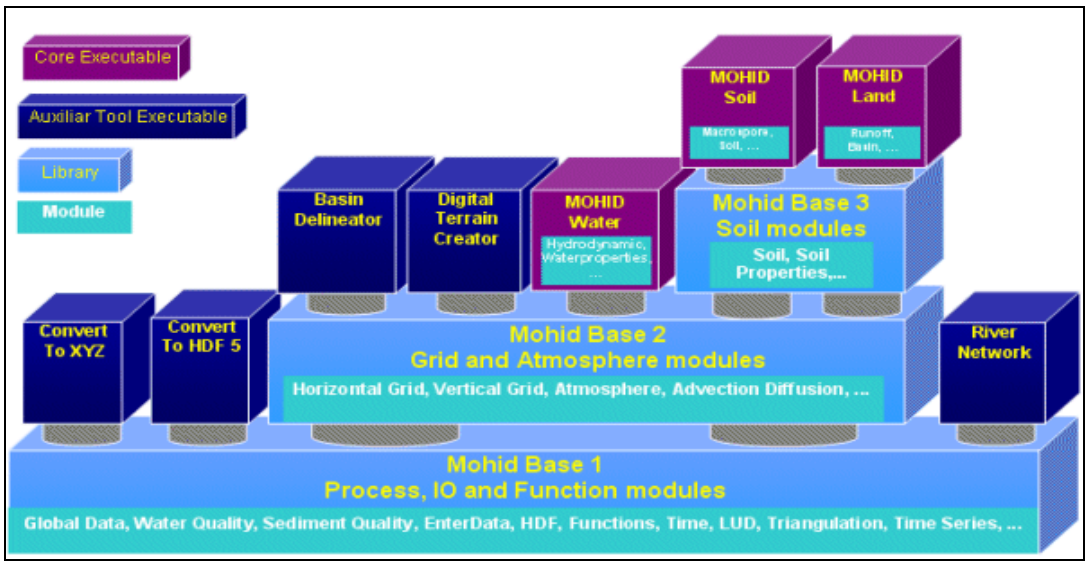

Figura 1 - Arquitectura del framework de MOHID.

Como fue mencionado anteriormente, la versión de diseño actual de MOHID está basada en el paradigma de orientación a objetos implementado a través del lenguaje Fortran 95. Si bien el lenguaje no está especializado en este paradigma puede ser utilizado para implementarlo tomando ciertas precauciones.

El diseño global de MOHID se divide en módulos de Fortran95 que hacen a su vez de clases del sistema. Esto permite a MOHID utilizar algunas de las principales características de la orientación a objetos como polimorfismo, encapsulación, herencia y sobrecarga de funciones. Además, permite que cada módulo sea responsable de manejar ciertos datos asociados a cada uno, así como el manejo de la memoria asociada a estos datos, la cual se reserva dinámicamente. Todos los módulos (objetos) tienen cuatro familias de funciones estándar como lo son el constructor, los métodos selectores, los métodos modificadores y el destructor.

Por otro lado, el intercambio de la información entre módulos se realiza bajo el paradigma cliente-servidor. Esto significa que cuando una clase necesita algún valor que no pertenece a él, se lo solicita a la clase encargada de manejar ese atributo. Esta estrategia permite asegurar que la información sea encapsulada por cada módulo.

Como se ve en la figura 1, en la base de la arquitectura se encuentran dos bibliotecas de suma importancia, denominadas Mohid Base 1 y Mohid Base 2. Estas bibliotecas resuelven problemas específicos, ya sea, numéricos, procesamiento de datos de entrada y salida, etc.

3.1.1 Biblioteca Mohid Base 1

La biblioteca Mohid Base 1 se encuentra en la base de la estructura del framework MOHID. La biblioteca agrupa las funciones que realizan el procesamiento de entrada y salida de datos, o sea, lee los archivos que sirven como datos de entrada a la simulación en curso de MOHID y escribe en archivos los resultados (intermedios o finales), para ser analizados una vez finalizada la simulación o para un post-procesamiento de datos por parte de otras herramientas. A su vez, se encarga de realizar funciones de cálculo típicas como puede ser resolver sistemas lineales. También tiene módulos que se encargan del manejo del tiempo en un formato propietario que sirve entre otras cosas para establecer los tiempos de simulación.

En la figura 2 se presenta las relaciones de dependencia que existen entre los distintos módulos pertenecientes a la biblioteca Mohid Base 1.

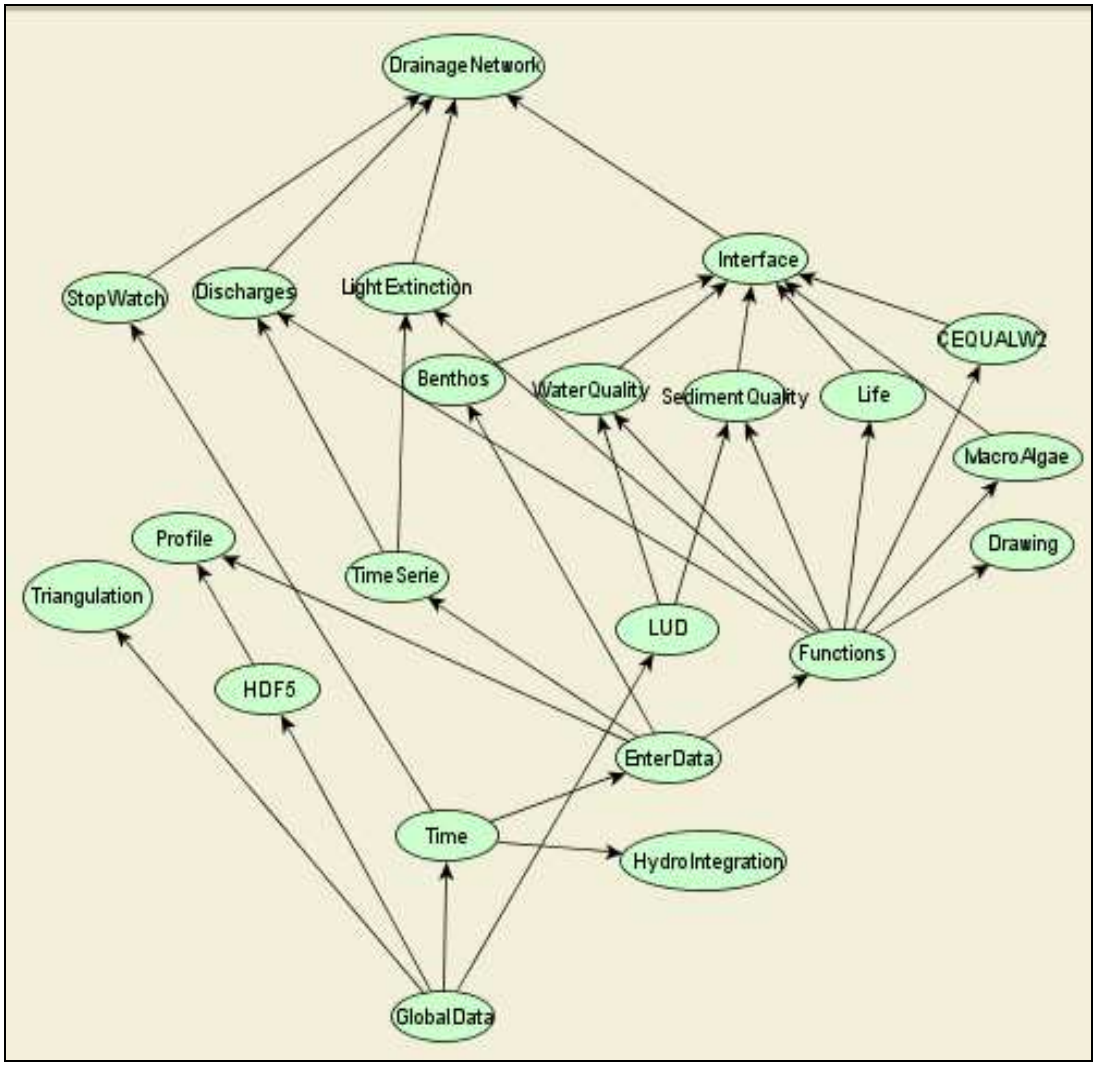

Figura 2 – Relaciones de dependencia entre módulos Mohid Base 1.

3.1.2 Biblioteca Mohid Base 2

La biblioteca Mohid Base 2, a diferencia de Mohid Base 1 que utiliza únicamente módulos dentro de la misma biblioteca, utiliza o se basa en módulos existentes en Mohid Base 1 para poder realizar cálculos más complejos que esta última no brinda y que sirven para otros módulos que se encuentran más arriba en la arquitectura del Framework MOHID.

Entre las funcionalidades más importantes la biblioteca Mohid Base 2 posee módulos que modelan la discretización del dominio en grillas, módulos para mejorar la interacción del cuerpo de agua con la atmósfera, y módulos que realizan cálculos para las propiedades biogeoquímicas.

En la figura 3 se muestra los módulos presentes en la biblioteca Mohid Base 2 y sus relaciones de dependencia dentro de la misma.

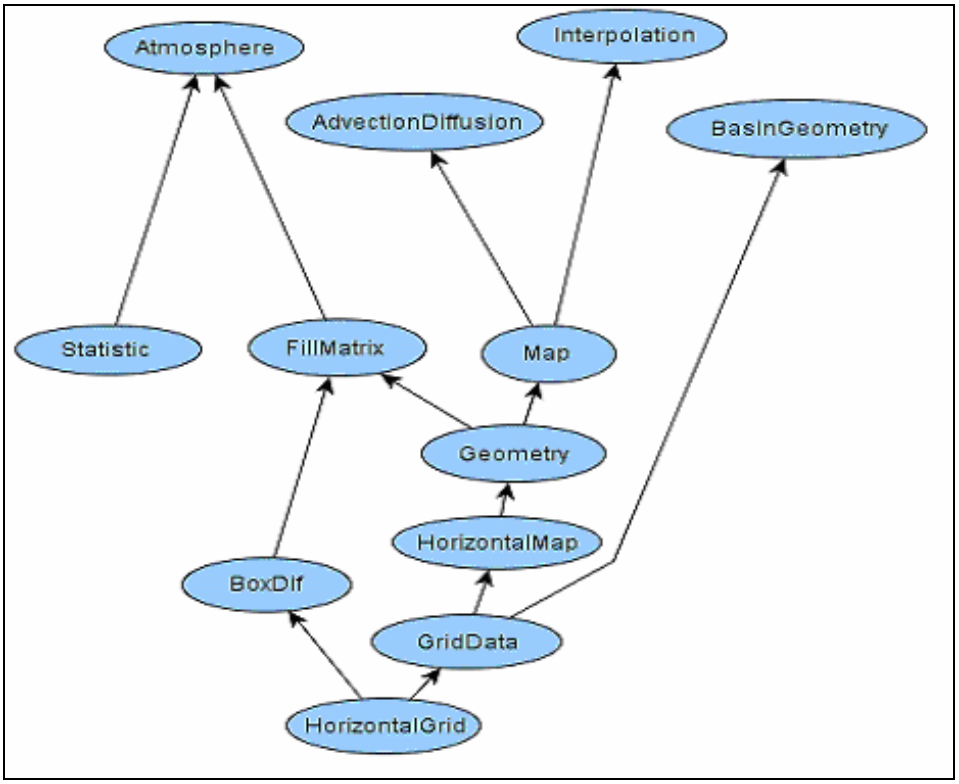

Figura 3 – Dependencias entre módulos de la biblioteca Mohid Base 2.

3.2 Núcleos ejecutables

El Framework MOHID presenta 3 ejecutables principales, estos son:

- MOHID Water: programa para la simulación tridimensional de cuerpos de agua (océanos, estuarios, etc).
- MOHID Land: programa para realizar simulaciones en cuencas hídricas y acuíferos.
- MOHID Soil: resuelve la ecuación de Richards para medios porosos saturados y no saturados.

3.3 Estrategias de Entrada/Salida

MOHID usa 2 componentes o módulos para intercambiar información entre el modelo y la interfaz gráfica (GUI).

- Basado en el módulo EnterData:
  Permite leer y escribir archivos de datos ASCII, estructurados en forma similar a un archivo en formato XML. La razón por la cual no se optó por leer y escribir con el formato XML es porque se implementó tiempo antes de que este formato se popularizara.
  Durante la simulación, casi todos los módulos que manejan información del proceso crean una instancia en particular del módulo EnterData, para leer datos de diferentes archivos.
- Utilizando el módulo HDF:
  El formato HDF [17] es usado por MOHID para almacenar los resultados intermedios o finales producidos por el modelo.
  Este formato fue desarrollado por la National Center for Super Computation Application (NCSA) [18]. Algunas de las ventajas del formato HDF5, con respecto a otros formatos, son que los archivos pueden almacenar mucha más información, el acceso al archivo es más rápido y los archivos pueden ser leídos incluso si la ejecución no llegó a su fin.

3.4 Interfaz de usuario

A medida que las herramientas numéricas del framework fueron creciendo en complejidad, la necesidad de una buena interfaz gráfica para el usuario aumentó. La interfaz gráfica de usuario en MOHID está compuesta por 2 programas principales, ambos desarrollados en Microsoft .NET. Esta interfaz se encuentra disponible solamente en la versión de MOHID para el sistema operativo Windows.

- MOHID GUI: maneja la estructura de directorios y los archivos de datos necesarios para establecer una simulación en MOHID, maneja y genera los archivos de salida de la ejecución.

- MOHID GIS: es una herramienta de tipo sistema de información geográfica que maneja variables temporales y espaciales requeridas o producidas por las aplicaciones de MOHID.

Ambas interfaces incorporan otras herramientas auxiliares, de post-procesamiento para el manejo de los datos de salida (o entrada) de MOHID, como lo son:

- Mohid PostProcessor: es una interfaz gráfica que despliega en pantalla, en forma de animación, datos almacenados en el formato HDF.
- ConvertToHDF5: es una herramienta que permite la aplicación de diferentes operaciones, que se conocen como acciones, que involucran archivos en el formato HDF5. Estas acciones pueden ser la conversión de datos a HDF5, interpolación de grillas y concatenación de muchos archivos.
- Digital Terrain Creator: es un programa que se utiliza principalmente para la creación de una grilla de datos y para generar el archivo de entrada de batimetría del modelo.
- Basin Delineator: realiza la delimitación de cuencas hidrológicas para utilizar el ejecutable MOHID Land.
- River Network: es un programa numérico que permite simular un sistema de ríos.
- Convert to XYZ: es una pequeña herramienta numérica que extrae datos tridimensionales y los transforma en coordenadas XYZ. Es útil para extraer datos para crear batimetrías.

4 MOHID Water

MOHID Water es un programa numérico en tres dimensiones que simula diversos procesos que ocurren en cuerpos de agua como ríos, reservas, estuarios, áreas costeras u océanos. Es uno de los programas principales dentro del sistema de modelación MOHID. El programa se compone de una serie de módulos que están construidos sobre las bibliotecas Mohid Base 1 y Mohid Base 2 y que primordialmente se encargan de calcular los procesos físicos y biogeoquímicos y además se encarga del manejo de la lectura y escritura de los datos requeridos por el modelo.

Este programa numérico fue diseñado para simular sistemas acuáticos considerando además los procesos de intercambio con otros medios, como ser la interacción con la atmósfera y con el fondo.

La figura 4 muestra la interacción entre distintos módulos del sistema, mostrando las relaciones y las interfaces con los diferentes medios.

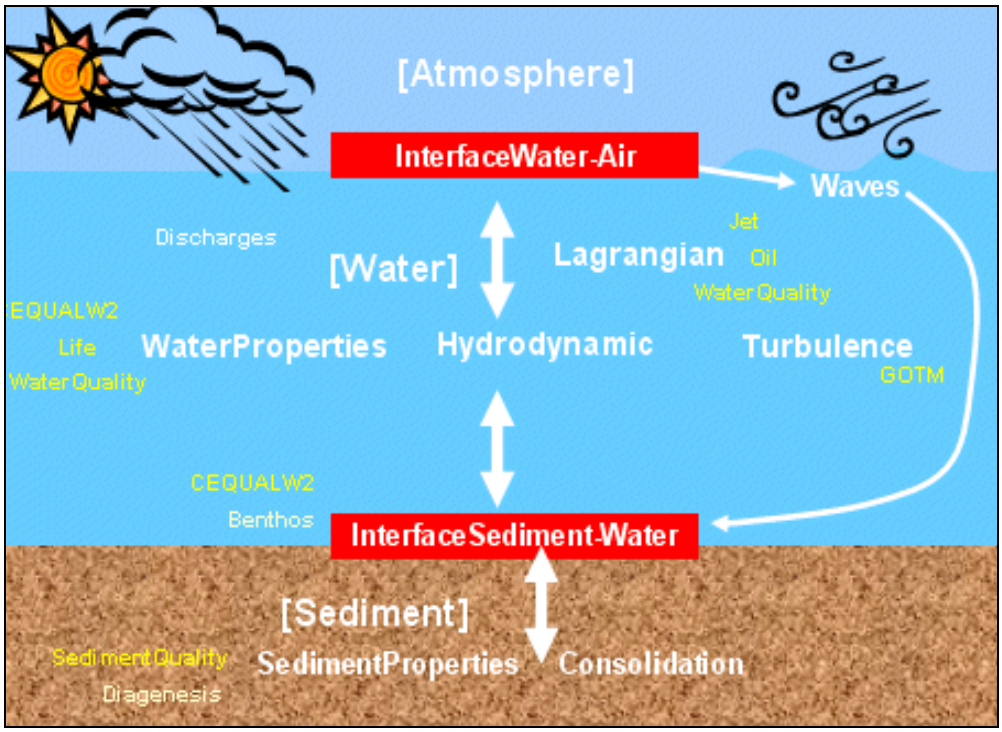

Figura 4 - Esquema de MOHID Water: principales módulos involucrados.

4.1 Principales módulos de MOHID Water

La estructura de módulos de MOHID se organiza en seis grandes grupos funcionales, cada uno de ellos compuesto por diversos módulos:

- **Módulos de parámetros globales:** se encargan de mantener parámetros globales como el tiempo de simulación, funciones o variables generales:
  - Module GlobalData posee principalmente información estática como los identificadores de dominios (generalmente numéricos), identificadores de tipos de error, constantes, parámetros y algunos tipos de datos derivados, usados frecuentemente en otras clases.
  - Module Time posee funciones para el manejo del tiempo en MOHID.
- **Módulos de funciones independientes:** manejan operaciones específicas que no se pueden representar en otros módulos o que dada la complejidad de lo que resuelve deben estar en un módulo aparte:
  - Module Functions calcula diferentes funciones matemáticas o científicas. Este módulo puede ser visto como una biblioteca matemática disponible a todas las clases.
  - Module Triangulation construye una triangulación de Delaunay con el objetivo de proveer la interpolación de niveles (“Gauge”) en la frontera abierta del dominio.
  - Module LUD resuelve sistemas de ecuaciones lineales (se encuentra dentro de la biblioteca Mohid Base 1).

- **Módulos estructurales:** son los encargados de la discretización de la geometría del dominio, transformaciones geométricas y referencias para moverse en la jerarquía de los dominios. Se nutre de variables como distancias, áreas, volúmenes, etc:
    - Module HorizontalGrid maneja la discretización horizontal.
    - Module HorizontalMap realiza un mapeo horizontal de la grilla en 2D.
    - Module Map realiza el mapeo 3D de las celdas de la grilla.
    - Module Geometry maneja diferentes formas de discretización vertical.
- **Módulos de manejo de datos:** realizan operaciones de entrada/salida:
    - Module EnterData lee y escribe archivos en formato ASCII, realiza la lectura de los archivos de entrada y genera los archivos de salida de la simulación.
    - Module HDF5 lee y escribe los datos de diferentes matrices en formato HDF5.
    - Module TimeSerie lee y escribe archivos con información de la simulación a medida que avanza la ejecución.
- **Módulos de funciones específicas:** implementan operaciones específicas que pueden ser utilizadas solamente por MOHID:
    - Module BoxDif se encarga principalmente de inicializar las propiedades definiendo cajas. Una caja o box file define un área en 2D o 3D. Estas áreas pueden ser usadas para inicializar diferentes propiedades o integrar valores entre las diferentes cajas, esto se aplica o se puede aplicar para todas las variables modeladas. Otra característica del módulo BoxDif es que integra los valores de las variables que modela en espacio y tiempo dentro de una caja y entre diferentes cajas. Las operaciones que realiza este módulo se pueden aplicar en un dominio de dos o tres dimensiones.
    - Module Statistics computa operaciones estadísticas básicas (en el espacio y tiempo) y va almacenando resultados de interés durante la simulación.
    - Module Interface transfiere información (condiciones de fuerza y variables de estado) entre las estructuras de grillas de una, dos y tres dimensiones hacia arreglos unidimensionales y para llamar a los procedimientos de los módulos bioquímicos de dimensión cero.
    - Module WaterQuality es un módulo que calcula la dinámica de diversos organismos o microorganismos que viven en cuerpos de agua.
- **Módulos de procesos,** son módulos que corresponden a los diferentes procesos que tienen lugar en los diferentes compartimentos del medio.

En la figura 5 se muestran los módulos que componen MOHID Water y la relación de dependencia entre estos.

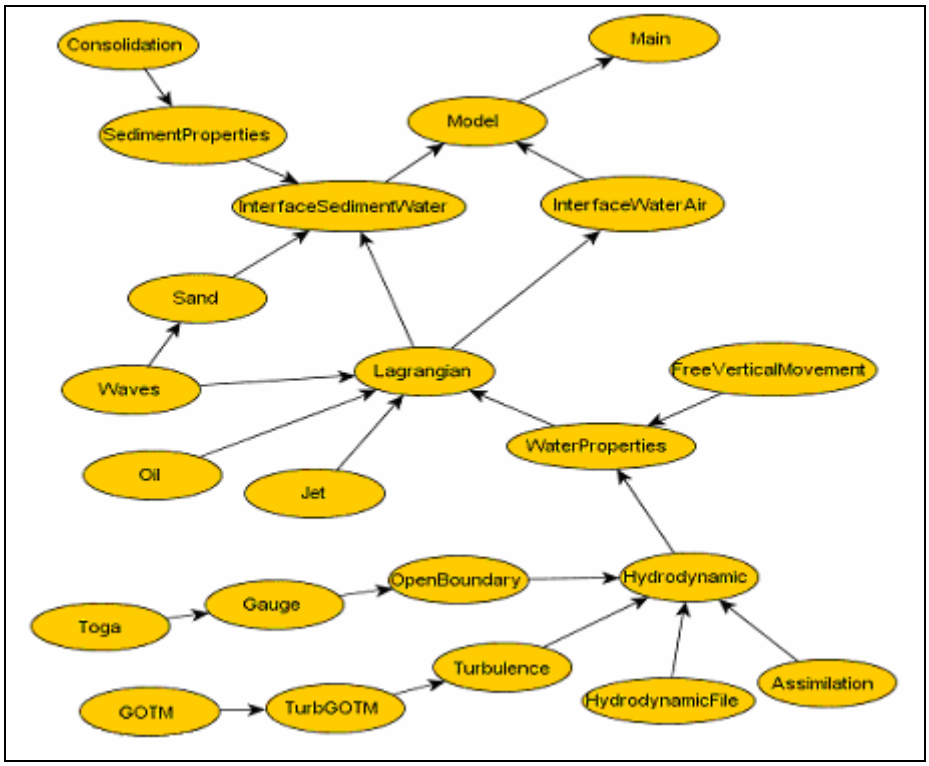

Figura 5 – Relación de dependencia entre módulos de MOHID Water.

A continuación se enumerarán los principales módulos del sistema y una breve descripción de ellos.

4.1.1 Model

Es el módulo principal en la arquitectura del sistema. El módulo Model maneja toda la información de un solo modelo. Es responsable de construir, modificar y destruir cada modelo. También se encarga de controlar el flujo de información entre diferentes modelos y la evolución del tiempo de simulación. Coordina la ejecución del módulo Hydrodynamic y el módulo Transport; y depende de los módulos Hydrodynamic, WaterProperties y Lagrangian.

La coordinación de ejecución del modelo consiste en la actualización del tiempo global del modelo y la actualización de los módulos Transport e Hydrodynamic en un solo modelo.

Cuando se trabaja con modelos encajados las comunicaciones entre modelos se hacen en una sola vía y de forma recursiva, es decir, las condiciones iniciales y otros datos que se calculan durante la simulación se envían desde el padre al hijo.

En la figura 6 se puede observar un diagrama con la interacción del módulo Model con otros módulos.

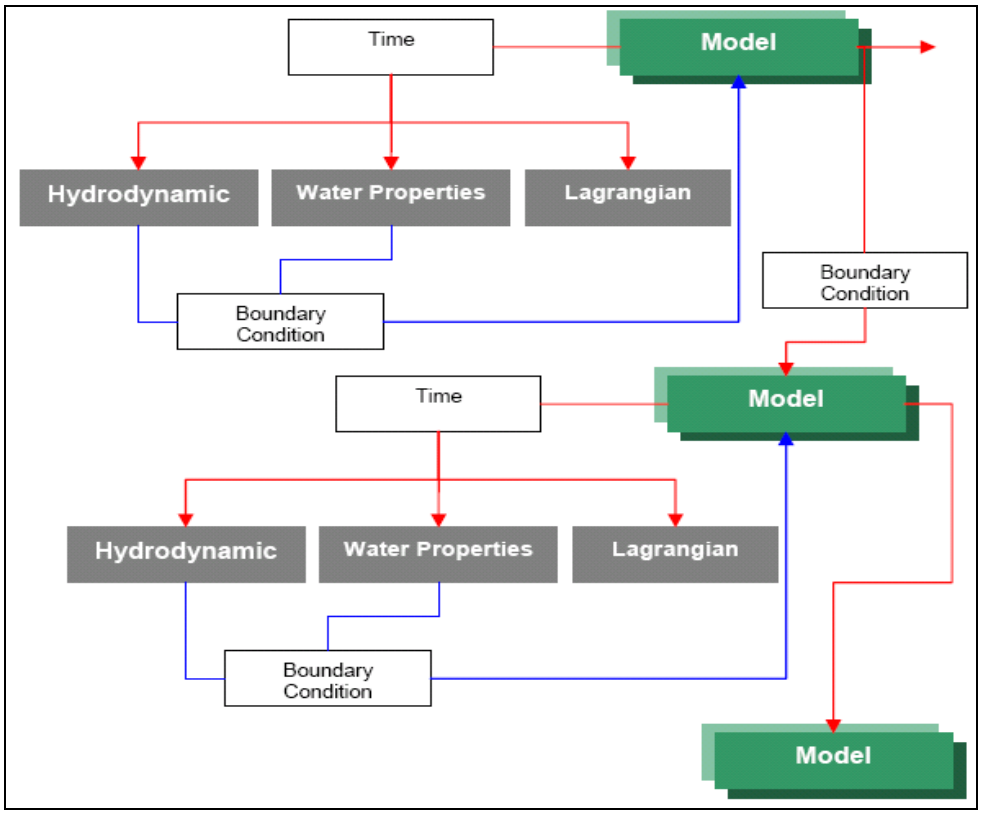

Figura 6 – Esquema de coordinación entre módulos de la ejecución**.**

4.1.2 Geometry

El módulo Geometry almacena y actualiza la información sobre los volúmenes finitos, calculando las áreas laterales y los volúmenes de la discretización según la demanda. Para realizar estos cálculos se basa en la elevación de la superficie libre calculada por el módulo hidrodinámico y en los datos batimétricos del dominio ingresados.

El enfoque de volúmenes finitos usado en MOHID (ver anexo A), permite que la resolución de las ecuaciones de conservación sea independiente de la geometría de cada celda o volumen de control, debido a que solo se requiere el flujo a través de las caras de la celda. De esta manera se logra una separación total entre las variables físicas y la geometría. Como los volúmenes pueden variar a lo largo de la simulación, la geometría es calculada y/o actualizada en cada paso de simulación después de calcular las variables físicas. Además, las coordenadas espaciales son independientes entre sí y se puede elegir cualquier geometría en cualquier dimensión. En la dirección horizontal se puede usar tanto coordenadas cartesianas como curvilíneas y en la dirección vertical pueden utilizarse coordenadas verticales genéricas con diferentes subdominios. De hecho, el módulo Geometry pude dividir la columna del cuerpo de agua en diferentes tipos de coordenadas como Sigma (coordenadas diseñadas para simular variaciones importantes del fondo dado que siguen la forma del terreno), Cartesianas, Lagrangianas, de Espaciamiento Fijo (Fixed Spacing), Armónicas, etc.

El módulo Geometry proporciona datos a los módulos lagrangian, turbulence y water properties.

4.1.3 Hydrodynamic

El módulo Hydrodynamic calcula el nivel, la velocidad y el flujo de agua en cada paso de tiempo. La discretización espacial es realizada por una aproximación de volúmenes finitos y la discretización temporal se realiza con un algoritmo semi-implícito ADI (Alternating Direction Implicit); por más información ver Anexo B. Este algoritmo computa alternativamente una componente de la velocidad horizontal implícitamente mientras la otra es calculada explícitamente. El sistema de ecuaciones resultantes es tridiagonal y por lo tanto puede ser resuelto de manera eficiente por el algoritmo de Thomas; este algoritmo se presenta en el Anexo C.

En la figura 7 se muestran las dependencias del módulo Hydrodynamic con el resto del sistema.

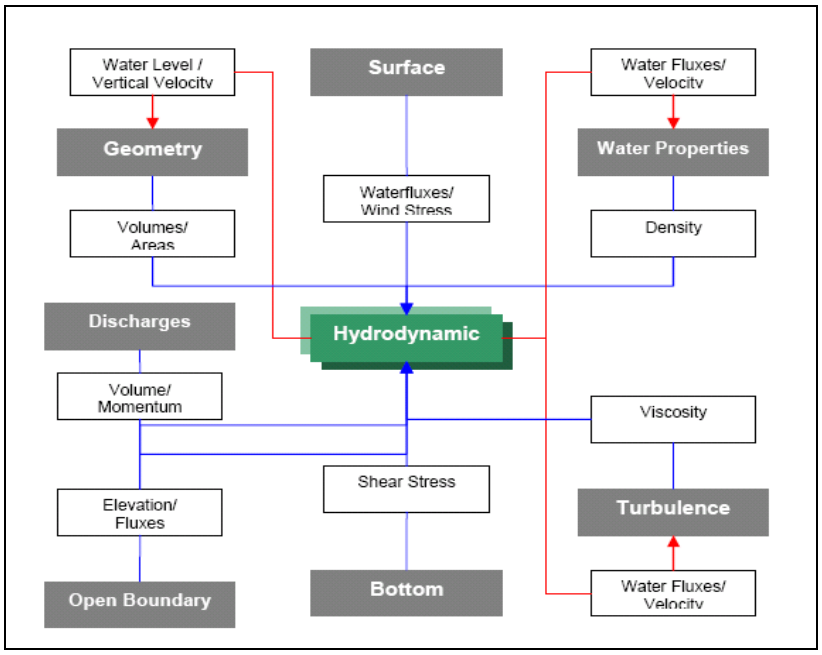

Figura 7 - Muestra la dependencia del módulo *Hydrodynamic* con otros módulos.

4.1.4 Lagrangian

Es un modelo de transporte lagrangiano (gestiona las mismas propiedades que el módulo WaterProperties). Puede ser utilizado, por ejemplo, para la simulación de la dispersión del petróleo. El módulo lagrangiano del MOHID utiliza el concepto de trazador, cuyas propiedades fundamentales son la posición espacial (x, y, z) de las partículas utilizadas como trazadores, su volumen y la concentración de

determinadas propiedades de interés, como pueden ser cualquiera de las evaluadas en el módulo de calidad de aguas (por ejemplo algún contaminante).

En la figura 8 se muestran las dependencias del módulo Lagrangian con el resto del sistema.

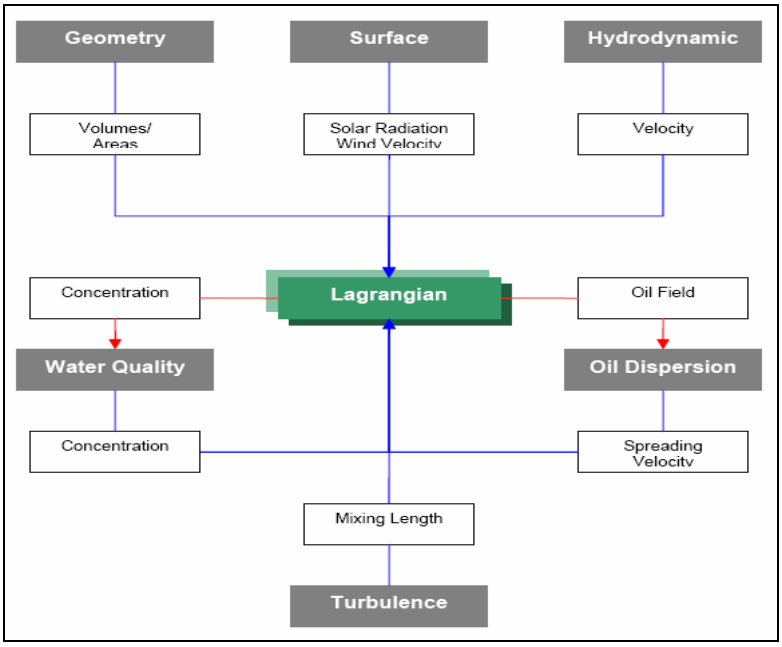

Figura 8 - Muestra la dependencia del módulo *Lagrangian* con otros módulos.

4.1.5 Oil

El módulo Oil se encarga de la simulación del movimiento de derrames de petróleo. Este tipo de herramientas cobran vital importancia a la hora de desarrollar planes de contingencia, permitiendo mitigar alguna catástrofe con este producto en el mar. El módulo también permite la evaluación de algunas características del impacto ambiental asociado al movimiento de petróleo. Por ejemplo, MOHID fue utilizado para calcular el posible impacto y el movimiento de la mancha de petróleo en la catástrofe del buque Prestige en las costas de Galicia en Noviembre de 2002.

El módulo Oil depende de los módulos Surface (presión atmosférica, oleaje, viento), WaterProperties (salinidad, temperatura, sedimentos cohesivos), Lagrangian.

4.1.6 WaterProperties

El módulo WaterProperties coordina y maneja la evolución de las propiedades del agua utilizando un modelo Euleriano de transporte. Para llevar a cabo esta tarea, se apoya o usa otros módulos como el de AdvectionDifussion (Advección-Difusión), encargado de calcular el transporte (por difusión o advección) de las propiedades, o

el módulo WaterQuality (calidad del agua), el cual es uno de los tres módulos encargados de calcular procesos biogeoquímicos.

A través de este módulo, MOHID es capaz de simular diferentes propiedades como la temperatura, salinidad, sedimentos cohesivos, fitoplancton, nutrientes, contaminantes, etc.

En la figura 9 se muestran las dependencias del módulo WaterProperties con el resto del sistema.

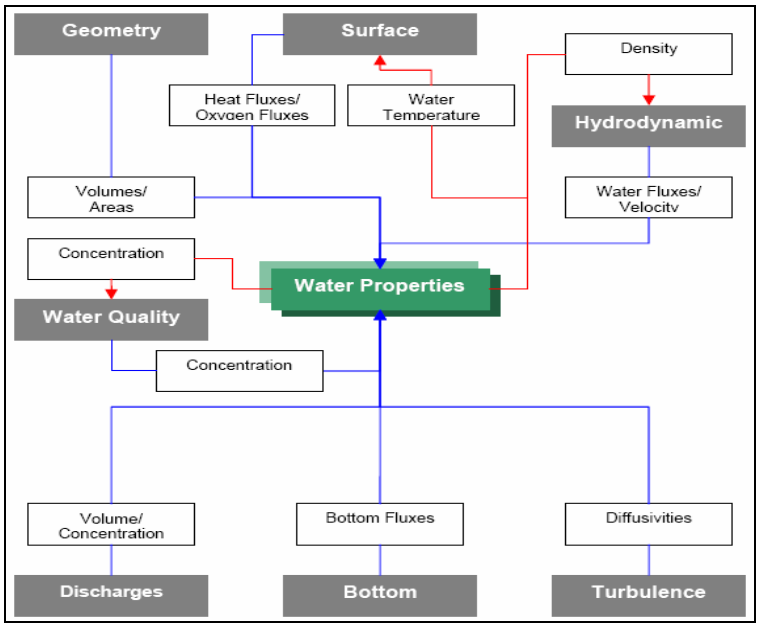

Figura 9 - Muestra la dependencia del módulo *WaterProperties* con otros módulos.

4.1.7 WaterQuality

El módulo WaterQuality simula la producción primaria y secundaria, y el ciclo de los nutrientes de ecosistemas. Los forzantes principales son la temperatura y la luz. El módulo fue desarrollado en términos de fuentes y sumideros de ciertas propiedades, o dicho de otra manera, definiendo flujos de entrada y salidas al sistema, lo cual permite un fácil acoplamiento al módulo de transporte en ambas formulaciones, Euleriana y Lagrangiana. Debido a la interdependencia de las propiedades, un sistema lineal de ecuaciones es calculado para cada volumen de control.

En la figura 10 se muestra la interacción de este módulo con otros.

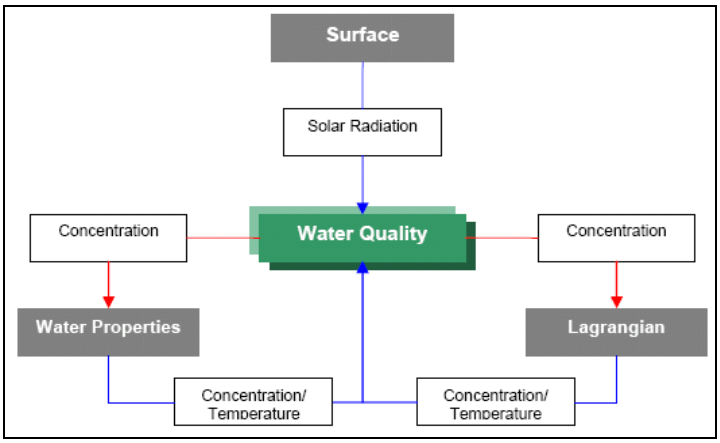

Figura 10 - Muestra la dependencia del módulo *WaterQuality* con otros módulos.

4.1.8 Surface

El módulo Surface plantea las condiciones de frontera en la superficie de la columna de agua y representa la influencia de forzantes externos atmosféricos como el viento y el sol en la superficie del agua.

Hay dos tipos de condiciones. Una dada por el usuario, usualmente datos meteorológicos (velocidad del viento, temperatura del aire, etc), y otra como condiciones de borde calculadas automáticamente por el modelo a partir de las condiciones o datos meteorológicos.

En la figura 11 se refleja la interacción del módulo Surface con otros módulos de MOHID.

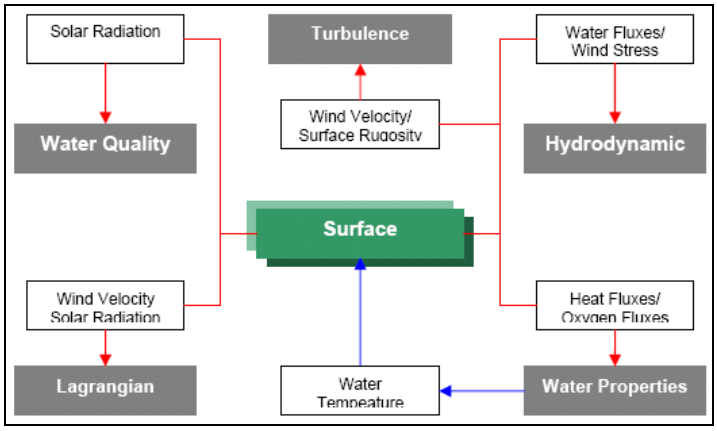

Figura 11 - Muestra la dependencia del módulo *Surface* con otros módulos.

4.1.9 FreeVerticalMovement

El módulo FreeVerticalMovement calcula las propiedades del flujo vertical. Básicamente se usa para determinar el movimiento vertical del flujo (en la dirección vertical). Usualmente se utiliza este módulo para calcular la velocidad de caída al

simular sedimentos cohesivos, o la velocidad de distintas partículas en las simulaciones de transporte.

4.1.10 HydrodynamicFile

El módulo HydrodynamicFile es un módulo auxiliar que le permite al usuario de MOHID integrar espacial y temporalmente la solución obtenida con el modelo en un archivo.

La integración espacial consiste en unir varias celdas para tratarlas como una sola. Por otro lado, la integración temporal consiste en juntar varios pasos discretos de tiempo de la solución hidrodinámica y puede ser directamente conectada con la integración espacial.

4.1.11 Turbulence

El módulo Turbulence calcula los coeficientes de viscosidad horizontal y vertical y las difusividades para diferentes métodos de cierre de turbulencia. Estas propiedades pueden ser calculadas de una manera simplificada utilizando coeficientes de difusión constantes. Por otro lado el usuario puede calcular la evolución de las propiedades de flujo turbulento de una manera más realista mediante el modelo GOTM (Global Ocean Turbulence Model) [19]. Una interfase de este módulo calcula los coeficientes utilizando un cierre de turbulencia de una o dos ecuaciones con la subrutina tomada del modelo general de turbulencia oceánica GOTM.

4.2 Características de la ejecución

La ejecución de un programa numérico de MOHID (water) tiene ciertas características que se mencionan en esta sección.

4.2.1 Descripción de los archivos de datos

El archivo nomfich.dat contiene las referencias (y las ubicaciones) de los distintos archivos de entrada o salida que maneja MOHID. En este archivo están definidas una serie de palabras clave (Keywords) que hacen referencia a cada uno de los archivos de entrada o de salida. Para que la lectura de datos que realice el programa sea correcta, nomfich.dat debe ubicarse en el directorio en donde se ejecuta el modelo.

La siguiente información es una pequeña muestra de las entradas más importantes. El resumen se basa en el manual de usuario de MOHID Water [20], donde se puede encontrar más información.

- IN_BATIM: archivo de entrada que contiene la discretización horizontal y la batimetría de entrada (cota del fondo). Allí se define el tamaño del dominio y las coordenadas de los puntos de la grilla. Para generar la grilla y

la batimetría, se utiliza la herramienta Digital Terrain Creator, ayudándose con la interfaz de usuario MOHID GIS.

- DOMAIN: archivo de entrada donde se define la geometría vertical. Esta geometría puede ser dividida en diferentes tipos de coordenadas y también es posible subdividir el dominio y definir distintos tipos de coordenadas en cada uno.
- IN_MODEL: archivo de entrada donde se define el período a simular y la discretización temporal de la simulación, es decir, el paso de tiempo con que se resuelven las ecuaciones. También se puede especificar que no se utilizará el módulo hidrodinámico (debido a que es una solución estacionaria o porque se cargan los valores de un archivo).

  Parámetros del archivo:
  - start: fecha de comienzo de la ejecución.
  - end: fecha de finalización de la ejecución.
  - dt: tiempo transcurrido entre dos instantes de cálculo (delta t).
  - splitting: el double_splitting significa que el modelo debe resolver las ecuaciones primitivas. En cambio, si se desea una solución estacionaria o que las propiedades hidrodinámicas se lean de un archivo, se pone No_Splitting.
  - variabledt: chequea si el usuario desea un paso de tiempo variable durante la simulación (0 significa que no).

- IN_DAD3D: es el archivo de entrada donde se definen las condiciones de resolución del módulo hidrodinámico. Se especifican distintas opciones relacionadas con los forzantes de la simulación, como por ejemplo la definición de los valores iniciales para cada celda de volumen finito, o la definición del tipo de condiciones de borde abiertas en la dirección horizontal y en la dirección vertical o la definición de la frecuencia con que se almacenan resultados, entre otros.

- BOT_DAT: es el archivo de entrada donde se definen las condiciones de borde en el fondo para el módulo correspondiente.

- IN_TURB: es el archivo de entrada donde se definen las condiciones que se utilizarán en el módulo Turbulence.

4.2.2 Construcción del dominio

Para la construcción del dominio en MOHID, primero se inicializan las variables que almacenan los datos asociados a una grilla o modelo. Luego cada grilla definida se agrega a una lista (lista encadenada simple) que contiene todos los modelos (padre y todos los modelos encajados, anidados, también llamados hijos o subdominios) cada uno con un identificador (para ser usado en caso que corresponda con MPI). Previamente se debe leer el archivo “Tree.dat” para saber cuántos modelos y submodelos posee la simulación y la ubicación de los archivos

de entrada. Luego para cada modelo, comenzando por el padre, se leen los archivos de entrada y la malla horizontal, asignando las diferentes características y propiedades a cada dominio. Luego se proporcionan los valores de la matriz de batimetría para los puntos de frontera y se asignan diferentes propiedades al dominio.

4.2.3 Ejecución del modelo

Después de construido todo el dominio (y todos los subdominios si fuese el caso), se prosigue con la ejecución del programa o dicho de otra forma, con la simulación del modelo. Este modelo posee ciertas características que están indicadas en los archivos que aparecen dentro del archivo nomfich.dat.

La simulación computacional de un paso de tiempo en MOHID se puede dividir en dos partes, una parte de actualización del tiempo y redefinición, si es necesario, de la frontera; y otra parte de ejecución y cálculo de acuerdo a los datos iniciales. El procedimiento que se encarga de la actualización es UpdateTimeAndMapping y el procedimiento que lleva a cabo el cálculo es RunModel.

La primer parte de actualización del tiempo que se está simulando, se basa en el campo DT definido en el archivo de entrada que se indica en la Keyword "IN_MODEL". Además se controla si se llegó al final de la simulación o no tomando en cuenta los campos start y end del archivo anteriormente mencionado. Por otro lado, para actualizar la frontera, se toma en cuenta el nivel del agua y allí se establece qué puntos son de agua y cuales son tierra en el dominio de la simulación.

La segunda parte se dedica a realizar cálculos del modelo en sí mismo. Se resuelven las ecuaciones discretizadas según el método ADI que se implementa (ver AnexoB) y se calcula la evolución en el tiempo de las variables modeladas en cada celda de la grilla, por ejemplo: velocidades, salinidad, temperatura, parámetros de turbulencia, elevación de superficie libre, etc.

Cabe mencionar que cuando se trabaja con dominios encajados se efectúa la comunicación de datos entre el modelo padre y el modelo hijo, en el cual el modelo hijo obtiene los datos de las diferentes variables de cálculo como condición de borde. Esto se realiza entre la actualización del tiempo y los cálculos del modelo. El procedimiento se llama SubModelCommunication.

4.2.4 Información durante la ejecución

Durante la ejecución del modelo y luego de pasada la etapa o fase de construcción aparece un mensaje "Running MOHID, please wait…". A partir de allí y cada 60 segundos de tiempo de CPU aparecen mensajes que brindan diferentes informaciones al usuario.

- **Time instant**: muestra el tiempo de simulación transcurrido desde el inicio hasta ese momento.

- **Elapsed CPU Time**: el tiempo de CPU que ha transcurrido desde que se inició la simulación.

- **Coefficient CPU / Model**: es el cociente entre el tiempo del procesador usado hasta el momento y el lapso de tiempo simulado hasta el momento.

- **Seconds per iteration**: tiempo dedicado a completar un ciclo de trabajo.

- **System time**: fecha y hora actual del sistema.

- **End of the run**: la fecha y hora prevista para la finalización de la simulación.

5 Tiempos de ejecución de MOHID

Las mediciones del tiempo de ejecución son de especial importancia por dos razones. Primero, para identificar las zonas críticas del código en cuanto al tiempo de ejecución para tratar de mejorar el desempeño computacional en esas zonas y así mejorar en forma importante el desempeño global del modelo. Segundo, para tener una referencia de tiempos con el objetivo de poder comparar con otras aplicaciones futuras y así poder evaluar las posibles mejoras que se introduzcan al modelo MOHID.

A continuación, se presentan una serie de tablas con el tiempo insumido (medido en milisegundos) por algunos de los procedimientos más relevantes del ejecutable MOHID Water. El tiempo calculado, en todos los casos, es el promedio de 10 ejecuciones independientes. También se provee la desviación estándar de los promedios calculados.

El dominio utilizado para efectuar las mediciones es una grilla de 10.700 nodos en la dirección horizontal y 10 capas en la dirección vertical. Además, se utilizó el esquema de resolución de 6 ecuaciones de Leendertse. Este esquema se utiliza para resolver las ecuaciones de movimiento del sistema. Para llevar a cabo esto, se divide un paso de tiempo (conocido como incremento) en dos. Por cada medio incremento (en un método de discretización estilo ADI, ver Anexo B), las ecuaciones son discretizadas de modo que en un medio incremento quedan implícitas y en el otro, explícitas, obteniéndose los valores de las variables de flujo (componentes horizontales de la velocidad y elevación de la superficie libre) en las distintas celdas de la grilla de cálculo.

Las simulaciones se realizaron en un equipo Intel Dual Core, donde cada procesador es Pentium 4 de 2.8 GHz con 1Gb de memoria RAM, corriendo bajo el sistema operativo Linux con distribución Debian.

La Tabla 1 muestra los tiempos de ejecución de cada iteración dentro del módulo Main.

Método	Tiempo (ms)	Desviación estándar
Submodel	23	4.83
RunModel	651	5.68

Tabla 1 - Tiempos de ejecución en el módulo Main.

Como se puede apreciar en la Tabla 1 el procedimiento RunModel insume la mayor parte del tiempo por iteración. Debido a esto se decidió medir los tiempos de algunos de los procedimientos más relevantes que implementan RunModel. En la Tabla 2 se puede apreciar la distribución de tiempos para esos procedimientos dentro de RunModel.

Método RunModel	Tiempo (ms)	Desviación estándar
Entre los métodos GetWaterPoints2D, ModifyAtmosphere, ModifyInterfaceWaterAir	14	5.16
Turbulence	31	3.16
ModifyHydrodynamic	450	6.67
WaterPropertiesEvolution	155	5.27

Tabla 2 - Tiempos de ejecución en el método RunModel.

Adicionalmente, se realizó la medición de los tiempos para el procedimiento ModifyHydrodynamic. Si bien el tiempo de ejecución de la rutina WaterPorpertiesEvolution no es despreciable respecto al tiempo de ejecución de ModifyHydrodynamic, parece razonable investigar más a fondo la distribución de tiempos de este último, debido a que es la etapa que requiere más tiempo de cálculo. En la Tabla 3 se puede apreciar la distribución de tiempos dentro del método ModifyHydrodynamic que se encuentra en el módulo Hydrodynamic.

Etapas del Método ModifyHydrodynamic	Tiempo (ms)	Desviación estándar
One_Iteration	395	5.27
ComputeResidualFlowProperties	19	3.16
Hydrodynamic_OutPut	34	5.16

Tabla 3 - Tiempos de ejecución en el método ModifyHydrodynamic.

De acuerdo a los tiempos de ejecución relevados para el ejemplo presentado, se puede inferir que no hay una función que sea la que insuma la mayor parte del tiempo en cada iteración ya que los tiempos se encuentran distribuidos en varias funciones. Sin embargo, la función One_Iteration, que se encuentra dentro de la función ModifyHydrodynamic del Módulo Hidrodinámico, implica un porcentaje importante del tiempo de ejecución.

A continuación se muestra otro estudio realizado con el objetivo de tener mediciones para un modelo que utiliza la estrategia de dominios encajados. En particular se utilizó un caso con dos modelos, un modelo padre y un modelo hijo.

El dominio utilizado del modelo padre es el mismo que en las pruebas anteriores, o sea, una grilla con 10.700 nodos en la dirección horizontal, 10 capas en la dirección vertical y un dt de 180 segundos. Se utilizó el mismo esquema (Lendertsee) para resolver las ecuaciones. Además, se hizo la misma evaluación de tiempos para los mismos procedimientos tanto en el modelo padre como en el modelo hijo.

El modelo hijo consiste en 31.652 nodos en la dirección horizontal, nuevamente 10 capas en la dirección vertical y un dt de 90 segundos. Las Tablas 4 a la 9 muestran los tiempos de ejecución para las dos iteraciones que se ejecutan en el modelo hijo. Se analizaron los tiempos de ejecución para los mismos procedimientos que el modelo padre. Cabe acotar que el modelo hijo resuelve el mismo sistema de ecuaciones que el padre, o sea, calcula las mismas variables pero con distintas condiciones de borde.

A continuación se presentan los tiempos de ejecución para un primer paso de tiempo en el modelo hijo una vez finalizado un paso de simulación en el modelo padre.

Método	Tiempo (ms)	Desviación estándar
Submodel	386	17.13
RunModel	2544	129.55

Tabla 4 - Tiempos de ejecución en el módulo Main.

Método RunModel	Tiempo (ms)	Desviación estándar
Entre los métodos GetWaterPoints2D, ModifyAtmosphere, ModifyInterfaceWaterAir	49	3.16
Turbulence	124	8.43
ModifyHydrodynamic	1839	109.08
WaterPropertiesEvolution	528	23.00

Tabla 5 - Tiempos de ejecución en el método RunModel.

Etapas del Método ModifyHydrodynamic	Tiempo (ms)	Desviación estándar
One_Iteration	1641	104.09

ComputeResidualFlowProperties	71	3.16
Hydrodynamic_OutPut	122	7.89

Tabla 6 - Tiempos de ejecución en el método ModifyHydrodynamic.

Las siguientes tres tablas (7, 8 y 9), presentan los tiempos de ejecución para un segundo paso de tiempo del modelo hijo. De esta manera, debido al dt utilizado, con 2 pasos en el modelo hijo, se iguala 1 paso en el modelo padre.

Método	Tiempo (ms)	Desviación estándar
Submodel	147	8.23
RunModel	2455	151.68

Tabla 7 - Tiempos de ejecución en el módulo Main.

Método RunModel	Tiempo (ms)	Desviación estándar
Entre los métodos GetWaterPoints2D, ModifyAtmosphere, ModifyInterfaceWaterAir	53	10.59
Turbulence	124	8.43
ModifyHydrodynamic	1812	63.91
WaterPropertiesEvolution	519	11.97

Tabla 8 - Tiempos de ejecución en el método RunModel.

Etapas del Método ModifyHydrodynamic	Tiempo (ms)	Desviación estándar
One_Iteration	1592	68.93
ComputeResidualFlowProperties	72	3.16
Hydrodynamic_OutPut	120	0

Tabla 9 - Tiempos de ejecución en el método ModifyHydrodynamic.

A partir de los datos obtenidos se puede concluir que el tiempo de ejecución de un submodelo es mayor que el tiempo de procesamiento de un paso del modelo padre, debido a que este dominio considera una discretización más refinada, o sea con más celdas de cálculo, o en el caso de MOHID, con más volúmenes de control.

De acuerdo a los resultados obtenidos los tiempos de ejecución del procedimiento RunModel, considerando la desviación estándar, en las dos iteraciones del modelo hijo son cercanos entre sí. Pero la diferencia que existe entre estas dos iteraciones se expresa en el tiempo de ejecución del procedimiento SubModelCommunication. En la primera iteración del modelo hijo, el tiempo de ejecución de este procedimiento es mayor que en la segunda iteración, debido a la inicialización y obtención de las condiciones de frontera del modelo padre. Los tiempos de ejecución del procedimiento RunModel que se muestran en las tablas 4 y 7 son cercanos entre sí; suponemos que si se utiliza un paso de tiempo menor a 90 segundos, o sea por cada iteración que ocurre en el modelo padre, se realizan más iteraciones en el modelo

hijo, los tiempos de ejecución del procedimiento RunModel correspondiente a cada iteración en el submodelo, serán cercanos entre sí.

Para comprobar la suposición del párrafo anterior se realizaron pruebas para la medición de los tiempos de ejecución para el mismo modelo padre y el mismo modelo hijo, pero considerando como paso de tiempo de simulación en el submodelo 60 segundos. Por lo tanto se realizan tres iteraciones del modelo hijo por cada iteración del modelo padre. Los datos obtenidos en las pruebas realizadas, muestran que los tiempos de la primera y segunda iteración del submodelo no cambian respecto a los tiempos mostrados en las Tablas 4 a la 9 (recordar que son los tiempos de la primera y segunda iteración con un paso de tiempo en el modelo hijo de 90 segundos). Además, muestran que los tiempos de ejecución de la tercera iteración, son similares a la segunda, por lo que si se usara un paso de tiempo menor a 60, o sea, si se realizaran más iteraciones del modelo hijo por cada iteración del modelo padre, se puede inferir que los tiempos de ejecución de las nuevas iteraciones no cambian, salvo al tiempo de ejecución del procedimiento SubModelCommunication. Esto se debe a una penalización de tiempos que ocurre al obtener y calcular los datos iniciales para ejecutar en el modelo hijo.

6 Conclusiones y trabajo futuro

Este documento es una versión preliminar del estudio del modelo de simulación hidrodinámica MOHID. Se prevé que el resultado de este estudio sirva para adquirir el conocimiento necesario para sustentar la siguiente etapa de trabajo, que consiste en mejorar el desempeño computacional con líneas de trabajo que incluyen la incorporación de estrategias de programación paralela.

En particular se desea incluir estrategias de memoria compartida y memoria distribuida. Dentro de las estrategias de memoria compartida se evaluará la utilización de OpenMP [21][22]. En cuanto a memoria distribuida se estudiará la aplicación de esquemas de descomposición de dominio utilizando el estándar MPI [23][24] para el pasaje de mensajes.

Anexo A – Volúmenes finitos

Leyes de conservación

Como etapa previa a explicar el método de los volúmenes finitos, se desarrollarán someramente los conceptos teóricos que este método intenta modelar de la realidad.

En la mayoría de las aplicaciones en Ingeniería, los sistemas físicos son gobernados por un conjunto de leyes de conservación. Por ejemplo, para la dinámica de fluidos, se trabaja con la conservación de la masa, de cantidad de movimiento y de energía. Estas leyes de conservación se escriben en forma de integrales asociadas a un volumen físico determinado. Supongamos que tenemos un dominio bidimensional Ω, con la frontera del dominio $\delta\Omega$. Entonces la ecuación de conservación es:

$$\frac{\partial}{\partial t}\int_{\Omega} U\,\partial A + \int_{\delta\Omega} [F(U)\,\vec{i} + G(U)\,\vec{j}] \cdot \vec{n}\,\partial s = \int_{\Omega} S(U,t)\,\partial A \qquad \text{(A1)}$$

donde U es la propiedad que se conserva, F y G son los flujos de la propiedad respecto a la dirección x e y, n es la normal en la frontera del dominio y S es la fuente o el origen de la propiedad.

Por ejemplo, en un problema de conservación de la masa en un fluido compresible, la ecuación quedaría de la siguiente manera:

$$\frac{\partial}{\partial t}\int_{\Omega} \rho\,\partial A + \int_{\delta\Omega} [\rho u\,\vec{i} + \rho v\,\vec{j}] \cdot \vec{n}\,\partial s = 0 \qquad \text{(A2)}$$

donde los términos correspondientes en la ecuación (A1) son $U = \rho$, $F = \rho u$, $G = \rho v$, $S = 0$. La densidad del fluido es $\rho(x, y, t)$ y las componentes x e y de la velocidad son expresadas a través de $u(x, y, t)$ y $v(x, y, t)$.

Los distintos términos pueden englobar diferentes características o diversas leyes de conservación, si este es el caso, simplemente se consideran como vectores y cada posición del vector corresponde a una propiedad según sea la ley de conservación que se esté considerando. Por ejemplo, las ecuaciones de Euler para un fluido compresible se escriben de la forma:

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u H \end{pmatrix} \quad G = \begin{pmatrix} \rho v \\ \rho u v \\ \rho u^2 + p \\ \rho v H \end{pmatrix} \quad S = 0 \qquad \text{(A3)}$$

Donde *p* es la presión estática, *E* es la energía total y *H* es la entalpía total. La primera fila de estos vectores, representa la conservación de la masa. La segunda y tercera fila indican la conservación de la cantidad de movimiento tanto en la dirección *x* como en la dirección *y*, la cuarta fila representa la conservación de la energía.

Método de los volúmenes finitos

El método de volúmenes finitos es un procedimiento de discretización utilizado para resolver ecuaciones diferenciales, basándose en la forma integral de las leyes de conservación. La estrategia de trabajo de los volúmenes finitos recorre el camino inverso al utilizado en la obtención de las ecuaciones diferenciales de transporte. A partir del volumen que ocupa el fluido (finito), se obtienen las relaciones para una partícula del fluido, reduciendo el volumen de integración hasta alcanzar el volumen correspondiente a una partícula (infinitesimal). Para ello se integra la ecuación diferencial original sobre un volumen finito, cuya forma concreta depende de la malla y del sistema de coordenadas que se esté empleando.

El enfoque de volúmenes finitos es ligeramente diferente al método de diferencias finitas y posee como gran ventaja la facilidad para representar condiciones de frontera. Este método considera la discretización espacial, como un conjunto finito de regiones contiguas que se conocen como volúmenes de control, en vez de considerar un conjunto de puntos como en otros métodos. En cada volumen de control se aplican las ecuaciones de conservación en donde las variables son calculadas en el centro de ese volumen de control. Los valores de esas variables en la superficie del volumen de control se calculan mediante la interpolación del valor de la variable en el centro.

Partiendo de la ecuación (A1), presentada en la subsección anterior, la idea es discretizar cada integral, siendo esta la gran diferencia respecto a muchos otros métodos que llevan el problema a una formulación diferencial. La ecuación característica (A1) de este método, tiene como principal ventaja la de trabajar con el término de los flujos sobre la frontera del dominio. Por lo que, si el costo computacional es dominado por la operación de calcular los flujos en la frontera (segundo término del lado izquierdo de la ecuación), la reducción del mismo puede ser considerablemente bueno. A partir de la ecuación (A1) se necesita discretizar las integrales de alguna forma y lograr el sistema discreto final a resolver. Este planteamiento sobre la forma integral de las leyes de conservación se debe satisfacer en cada subdominio, lo cual implica satisfacerlas sobre el dominio global.

La forma discretizada de las leyes de conservación aplicadas a un volumen discreto Ωj es:

$$\frac{\partial}{\partial t}(U_j \Omega_j) + \sum_{lados} (F \cdot S) = Q_j \Omega_j \qquad \text{(A4)}$$

La ecuación (A4) es la formulación general del método de los volúmenes finitos. El usuario tiene que definir para una celda seleccionada cómo estimará el volumen, las caras de la celda y cómo se aproximarán los flujos sobre estas caras. Esto tiene cierta equivalencia con elegir el esquema en diferencias para las derivadas, en el método de diferencias finitas.

A continuación, se plantean algunas diferencias entre el método de volúmenes finitos y el método de diferencias finitas:

1- En volúmenes finitos, a partir de un dominio Ω discretizado, si se tiene en cuenta una celda Ωj y la variable asociada a esta, Uj. Las coordenadas del nodo j, que es la ubicación precisa de la variable U dentro de la celda Ωj, no aparece explícitamente. Consecuentemente Uj no está asociada a ningún punto fijo del dominio y puede considerarse como un valor promedio de la variable de flujo U sobre la celda.

2- En volúmenes finitos las coordenadas de la malla aparecen solamente para definir el volumen y las áreas de las caras de la celda.

3- Al utilizar volúmenes finitos para problemas estacionarios sin fuentes, el único término que permanece es el de la suma de los flujos sobre las caras. Con lo cual el método puede programarse para recorrer estas caras, e ir descargando los flujos sobre las dos celdas al mismo tiempo considerando la diferencia en signo.

4- El método de los volúmenes finitos permite introducir fácilmente condiciones de contorno, especialmente aquellas que vienen expresadas en término de los flujos que pueden ser directamente impuestos en el respectivo término.

La malla o grilla es la representación discreta de la geometría del dominio en donde el problema va a ser resuelto. Divide la solución del dominio en una cantidad limitada de subdominios (pueden ser elementos, volúmenes de control, etc). En estos subdominios se calculan las diferentes variables incluidas en una ecuación de conservación.

En cuanto a los diferentes tipos de mallas que se pueden utilizar, el método de los volúmenes finitos cuenta con la misma flexibilidad que el método de los elementos finitos, pero restringido a elementos con lados rectos o caras planas. Entre las mallas posibles podemos hacer una división entre mallas estructuradas y no estructuradas:

1- Mallas estructuradas consiste en una familia de líneas con la propiedad de que los miembros de una familia no se cruzan entre ellos y se cruzan como máximo una vez, con los miembros de otra familia. Esto permite que las líneas de un conjunto dado, se puedan numerar en forma consecutiva y que cualquier nodo o volumen de control, se pueda ubicar en término de dos índices (i, j) en 2D o tres índices (i, j, k) en 3D.

Esta estructura de grilla es la más simple y es similar a una grilla cartesiana. Cada celda tiene cuatro vecinos adyacentes en 2 dimensiones y seis en 3 dimensiones y los índices de los vecinos difieren en ±1 respecto a la coordenada actual por ejemplo (*i*,*j*) en 2D. En la figura 12 se muestra un ejemplo de una grilla estructurada en dos dimensiones. Esta imagen fue extraída del libro de Ferziger [16].

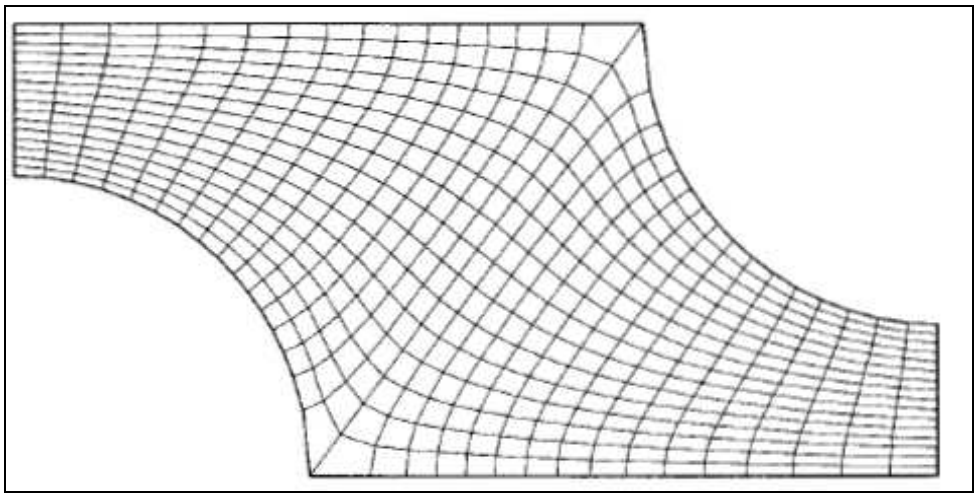

Figura 12 – Ejemplo de grilla estructurada en 2D.

Esta forma de conectividad simplifica la programación y la matriz asociada al sistema de ecuaciones lineales tiene una estructura regular, con lo cual se pueden aplicar técnicas eficientes para resolver este tipo de sistemas lineales. La principal desventaja de las grillas estructuradas es que pueden ser aplicadas solamente en la resolución de problemas que tengan dominios simples.

2- Mallas no estructuradas son aquellas donde no es posible encontrar una expresión de 2 o 3 índices que permita ubicar un nodo. Es aplicable a problemas en donde el dominio donde se halla la solución es irregular y no es posible utilizar una grilla estructurada. Son comúnmente empleadas por el método de los elementos finitos.

Una vez que la malla se ha construido el usuario debe decidir entre 2 opciones propias del método de los volúmenes finitos:

1- Centrado en las celdas. Las variables de flujo representan un promedio de los valores de la misma sobre la celda y están asociadas a la celda.

2- Centrado en los vértices. En este caso las variables de flujo representan los valores en los vértices o los puntos de la malla.

Para mantener las ecuaciones de conservación, es importante que cada uno de los volúmenes de control no se solapen entre sí, sino que estén uno al lado de otro. Las

figuras 13 y 14 muestran los volúmenes de control típicos en una malla cartesiana 2D y 3D con su notación respectivamente. El centro del volumen de control se denota como *P*. Estas figuras fueron extraídas del libro de Ferziger [16].

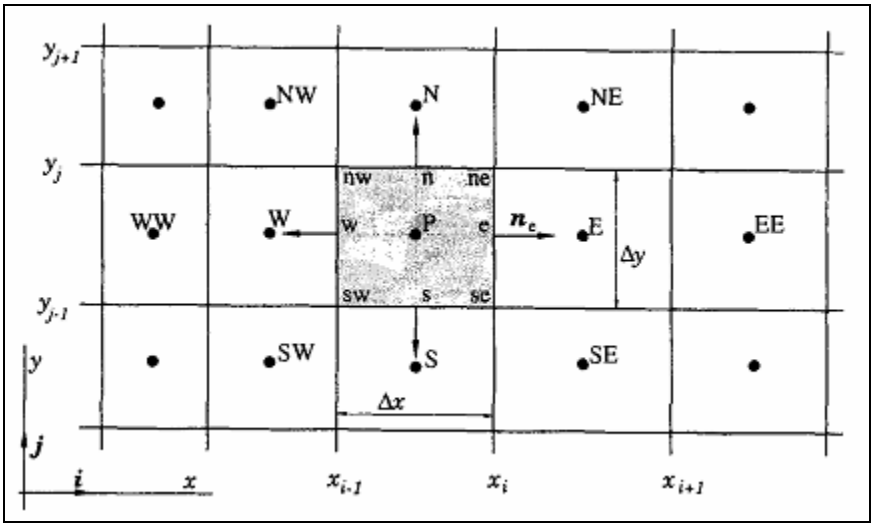

Figura 13 – Un volumen de control con su notación en una malla cartesiana 2D.

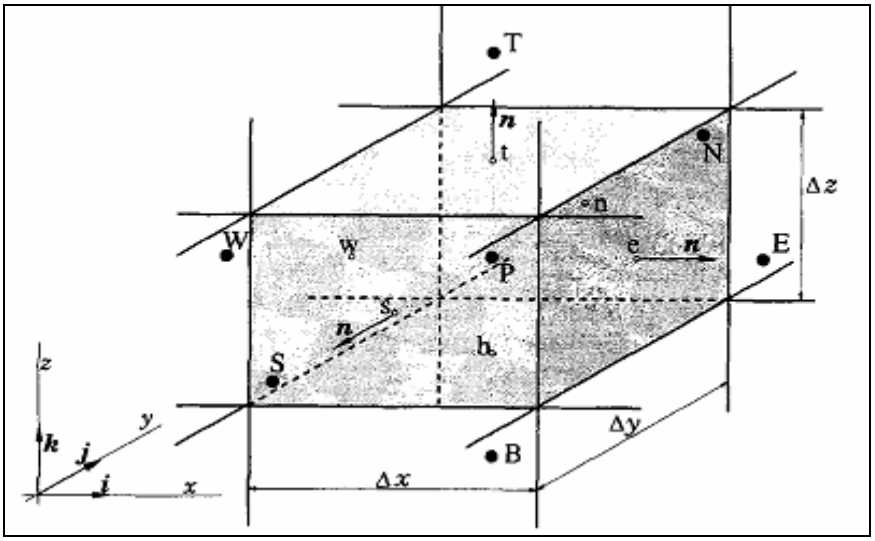

Figura 14 – Un volumen de control y su notación en una grilla cartesiana 3D.

Anexo B – Métodos estilo ADI

MOHID utiliza un método semi-implícito para resolver la grilla del plano horizontal. Este método se conoce como el método de dirección implícita alternada (ADI - Alternating Direction Implicit). El método ADI es una estrategia utilizada para resolver principalmente ecuaciones en diferencias, sin embargo, puede ser aplicado también a volúmenes finitos. Es utilizada para resolver ecuaciones diferenciales parciales, en particular elípticas e hiperbólicas.

La esencia del método ADI es dividir el paso de tiempo en dos (pueden ser más), para cada uno de los cuales se resuelven las ecuaciones en una sola de las dos direcciones del espacio. Por ejemplo, en la primera mitad de tiempo un esquema implícito es usado para calcular los flujos en la dirección *x*. Esto significa que la ecuación es reescrita de manera que el tiempo es *n+1/2t* en la variable del flujo en la dirección *x*. Mientras que el tiempo es *n* en la variable del flujo en la dirección *y*. En la segunda mitad de tiempo, se realiza el mismo procedimiento pero en el otro sentido. Usualmente el paso adicional se considera en un nodo que casi siempre es el punto medio entre nodo y nodo en cierta dirección de la malla. Durante la primer mitad del tiempo, un esquema implícito es usado para calcular los flujos en la dirección *x,* la matriz que se obtiene es tri-diagonal, la otra mitad de tiempo se hace lo mismo pero en la otra dirección, como lo muestra la figura 15.

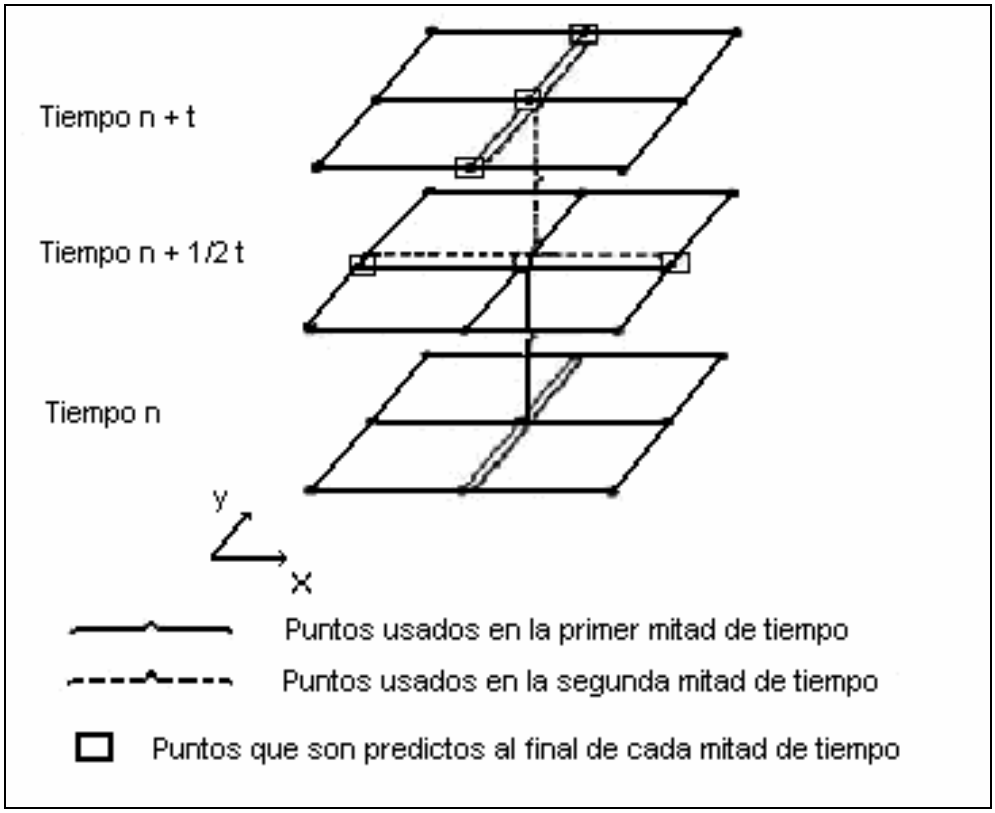

Figura 15 – Esquema indicando la combinación de variables en el método ADI.

Anexo C – Método de Thomas

Una de las ventajas que tiene caracterizar un problema con un sistema Ax = b tri-diagonal, es que su aplicación evita guardar gran cantidad de ceros que no se utilizan en la matriz **A**, y por lo tanto éste requiere menos memoria utilizando estrategias de matrices dispersas. Otra ventaja, es que si se utiliza un algoritmo que evite realizar operaciones innecesarias (por ejemplo operar con ceros) se puede disminuir considerablemente el tiempo de cómputo.

$$\begin{bmatrix} f_1 & g_1 & & & & & \\ e_1 & f_2 & g_2 & & & & \\ & e_2 & f_3 & g_3 & & & \\ & & e_3 & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & g_{n-2} \\ & & & & & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & & e_n & f_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_1 \\ x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ \vdots \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

Un algoritmo que resuelve un sistema lineal tri-diagonal (Ax = b) es el de Thomas, también conocido como algoritmo de matriz tri-diagonal (TDMA). Este método es una variante simplificada del método de eliminación Gaussiana y consiste en reducir el sistema tri-diagonal a un sistema bi-diagonal superior.

Solución por Algoritmo de Thomas:

- Descomposición:

```
DO k=2, n
      ek=ek/fk-1
      fk=fk-ek*gk-1
END DO
```

- Sustitución hacia Adelante:

```
DO k=2, n
      bk=bk-ek*bk-1
END DO
```

- Sustitución hacia Atrás:

```
Xn=bn/fn
DO k=n-1,1,-1
      xk=(bk-gk*xk+1)/fk
END DO
```

Observando el algoritmo de Thomas se puede concluir que reduce la aparición innecesaria de valores cero (problema conocido en la literatura anglosajona como fill-in). En el algoritmo de Thomas, la solución se obtiene en O(n), en vez de O(n^3) como en la eliminación Gaussiana.

Bibliografía

[1] I. Piedra-Cueva, M. Fossati. Residual currents and corridor of flow in the Rio de la Plata, Applied Mathematical Modelling, Volume 31, Issue 3, Pages 564-577, 2007.

[2] I. Piedra-Cueva, M. Fossatti. Informe Final Convenio "Estudios técnicos marítimos para evaluar la alternativa de descarga en Punta del Este" entre Tahal-Fing (IMFIA). 2007.

[3] Sitio web MARETEC (Marine and Environmental Tecnology Center) www.maretec.mohid.com. Consultado en Febrero 2008.

[4] Sitio web de IST (Instituto Técnico Superior). www.ist,utl.pt. Consultado en Febrero 2008

[5] S. Nakamura. Applied numerical methods with software. Prentice-Hall, 1991.

[6] A. J. Santos. Modelo Hidrodinâmico Tridimensional de Circulação Oceânica e Estuarina. Ph. D, Thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, 1995.

[7] M. V. Kind. Desplazamiento del frente de salinidad del Río de la Plata debido al aumento del nivel medio del mar. Tesis de Grado en Ingeniería Civil. Facultad de Ingeniería, Universidad de Buenos Aires, 2004.

[8] F. Martins. Modelação Matemática Tridimensional de Escoamentos Costeiros e Estuarinos usando uma Abordagem de Coordenada Vertical Genérica. Ph. D, Thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, 1999.

[9] P. C. Leitão. Modelo de Dispersão Lagrangeano Tridimensional. Ms. Sc. Thesis, Universidade Técnica de Lisboa, Instituto SuperiorTécnico, 1996.

[10] P. Miranda, F. Braunschweig, P. Leitão, R. Neves, F. Martins y A. Santos. Mohid 2000, a costal integrated object oriented model. Hydraulic Engineering Software VIII, WIT Press, 2000.

[11] J.J. Taboada, R. Prego, M. Ruiz-Villarreal, P. Montero, M. Gómez- Gesteira, A. Santos y V. Pérez-Villar. Evaluation of the seasonal variations in the residual patterns in the Ría de Vigo (NWSpain) by means of a 3D baroclinic model, Estuarine Coastal and Shelf Science 47, pp. 661-670, 1998.

[12] P. Montero. Estudio de la hidrodinámica de la Ría de Vigo mediante un modelo de volúmenes finitos (Study of the hydrodynamics of the Ría de Vigo by means of a finite volume model), Ph.D. Dissertation, Universidad de Santiago de Compostela, 1999.

[13] M.R. Villarreal, P. Montero, R. Prego, J.J. Taboada, P. Leitao, M. Gómez Gesteira, M. de Castro and V. Pérez-Villar. Water Circulation in the Ria de Pontevedra under estuarine conditions using a 3d hydrodynamical model, submitted to Est. Coast. and Shelf Sc, 2000.

[14] L. Cancino y R. Neves. Hydrodynamic and sediment suspension modelling in estuarine systems. Part II: Application to the Western Scheldt and Gironde estuaries, Journal of Marine Systems 22, 117-131, 1999.

[15] F. Braunschweig. Generalização de um modelo de circulação costeira para albufeiras, MSc. Thesis, Instituto Superior Técnico, Technical University of Lisbon, 2001.

[16] J. H. Ferziger y M. Peric. Computational Methods for Fluids Dynamics, 3rd Edition. Springer, 2002.

[17] Sitio web del formato HDF (Hierarchical Data Format) http://hdf.nsca.uiuc.edu. Consultado en Febrero 2008.

[18] Sitio web de la NCSA (Nacional Center for Supercomputing Application at the University of Illinois. www.ncsa.uiuc.edu. Consultado en Mayo 2008.

[19] Sitio web de GOTM (General Ocean Turbulence Model). www.gotm.net. Consultado en Febrero 2008.

[20] Sitio web de MOHID. www.mohid.com. Consultado en Julio 2008.

[21] C. Narus, M. Narus, L. Dagum, D. Kohr, D. Maydan, and J. McDonald. Parallel Programming in OpenMP. Morgan Kaufmann, 2000.

[22] Sitio web de OpenMP. www.openmp.org/drupal/. Consultado en Agosto 2008.

[23] Sitio web MPI. www.mcs.anl.gov/mpi. Consultado en Mayo 2008.

[24] Sitio web foro de MPI. www.mpi-forum.org. Consultado en Mayo 2008.

[25] Sitio web de la wiki de MOHID. www.mohid.com/wiki. Consultado en Agosto 2008.