

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Computación Paralela y Distribuida

**Paralelización del modelo
hidrodinámico y de transporte
de sedimentos Shal2D**



Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por la bachiller

Iris C. Espinoza R.
para optar al título de
Licenciada en Computación

Tutor: Prof. Robinson Rivas

Caracas, Mayo 2011

Resumen

El Modelo Shal2D es un modelo hidrodinámico y de transporte de sedimentos bidimensional desarrollado por los ingenieros del Instituto de Mecánica de Fluidos de la facultad de Ingeniería de la Universidad Central de Venezuela. Este modelo se encarga de estudiar la dinámica de fluidos incompresibles y el transporte de sedimentos en esos fluidos. El software permite obtener ciertas características de los ríos como velocidades, profundidad y elevación de varios puntos sobre un caudal. Además de esto, dependiendo de la cantidad de sedimentos y el tipo de material, se estudia como estas son trasladadas, según la velocidad horizontal en ese punto.

Para reducir los tiempos de ejecución de este modelo secuencial en este trabajo se presenta el desarrollo de una aplicación Shal2D paralela que permite dar resultados en un tiempo menor, y un refinamiento de la malla de entrada como preprocesamiento de datos utilizando el software triangle para la generación de más nodos y por lo tanto obtener resultados más precisos. Además se presentan conocimientos básicos en el área de ingeniería sobre los modelos hidrodinámicos, como definiciones de fluidos, hidrodinámica y mecánica de fluidos, hasta transporte de sedimentos, tipos de sedimentos y métodos matemáticos aplicables a los modelos dependiendo del material. También encontrará información base para el desarrollo de una aplicación paralela términos como cluster, librerías MPI y su implementación en este proyecto.

El simulador paralelo hidrodinámico y de transporte de sedimentos Shal2D tiene como finalidad reducir los grandes tiempos de ejecución del modelo en secuencial, esto con el objeto de obtener mejores tiempos de cómputo y resultados más precisos mediante librerías para paralelización como MPI (Message Passing Interface) y el uso del software triangles para la triangulación y refinamiento de la malla.

Palabras Claves:

Modelo Hidrodinámico, Shal2D, Transporte de sedimentos, MPI, Sistemas en paralelo, Paralelización, Malla.

Dedicatoria

Mami por ser la persona que llena mis días de alegría y enseñarme junto con mi papi lo bonito de la vida y que todo es posible con esfuerzo y dedicación, que no existe nada imposible si se cree y quiere de verdad. Solo Dios sabe que mi vida es dedicada a ustedes por ser la parte más importante de mi y haberme dado esos hermanos tan maravillosos. Dios los bendiga. Los amo y los amaré siempre...

Agradecimientos

Primeramente a Dios por haberme ayudado en cada instante de mi vida y de mi trayectoria como estudiante de esta gran casa de estudios. Además de darme esos padres maravillosos y esa familia especial. Así como unos amigos únicos y profesores que guiaron mis pasos.

A la mujer más amada, mi mamá, Judith quien me apoya, soporta, entiende y está conmigo en cada momento, extendiendo esa mano de amiga y madre que sólo ella sabe dar, cubriendome con su amor incondicional y animandome a ser cada día mejor persona.

Al hombre de mi vida, mi papá, Juan quien siempre está conmigo a pesar de que no lo pueda ver. Él me enseñó que se debe seguir luchando a pesar de todo lo que ocurra y que la vida se debe vivir día a día como si fuera el último.

A mis hermanos Juan C., y Anais por ser tan maravillosos y magníficos, que siempre han estado allí cuando los necesito apoyandome y enseñandome lo bonito de tenerlos y cosas nuevas que solo ellos con su amor me pueden enseñar. Gracias mis hermanitos hermosos.

A mis tíos y tías que me apoyan y dan ese amor que cubre mi vida tanto en las buenas y las malas, gracias por estar ahí. A mis primas y primos por esos momentos de alegría. A mis hermanos no de sangre pero si de corazón. Gracias.

A mi abuela por darle la vida a mi madre y ser dulce con todos, por darme maltas a escondidas en la bodega.

A mis profesores, en especial a mi tutor y amigo Robinson Rivas por todo lo enseñado, desde los conocimientos académicos hasta formación personal. Ha sido mi ejemplo a seguir. Gracias.

A mis amigos, grupoCcpd (vane, dans, jorg, kari, chino, ale, niño), Armando, Astrid, Dessire.. Por estar conmigo durante estos cinco años, unos más tiempo, otros menos, pero al fin y al cabo años super especiales. Gracias muchachos.

A la magna ilustre Universidad Central de Venezuela y a todo su personal, por darme la excelente educación.

A todos aquellos que no están aquí pero que forman parte de mi vida, gracias infinitas.

Millones de bendiciones del cielo para todos. Los quiero mucho y siempre estarán en mi corazón.

Índice general

Resumen	III
Dedicatoria	V
Agradecimientos	VII
Introducción	XV
1. Planteamiento del problema	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivos Generales	2
1.2.2. Objetivos Específicos	2
1.3. Justificación de la solución	2
1.4. Alcance del proyecto	3
1.4.1. Metodología de trabajo	3
2. Marco Teórico	5
2.1. Modelos Hidrodinámicos	5
2.1.1. Métodos de análisis	7
2.2. Mecánica de Fluidos	9
2.3. Ecuaciones de Navier-Stokes	10
2.4. Transporte de Sedimentos	13
2.5. Modelos hidrodinámicos	15
2.6. Modelo Shal2D	23
2.6.1. Antecedentes de Shal2D en paralelo	25
2.7. Sistemas en paralelo	33
2.7.1. MPI	36
2.7.2. Cluster	38

3. Marco Metodológico	43
3.1. Plataforma Tecnológica	43
3.2. Tecnologías utilizadas	44
4. Marco Aplicativo	47
4.1. Recolección y análisis de datos	47
4.1.1. Datos de entrada	47
4.1.2. Archivos de salida	48
4.2. Diseño e implementación de la solución	49
4.3. Pruebas y análisis de resultados	51
4.3.1. Refinamiento de datos	51
Conclusiones y Trabajos Futuros	65
Bibliografía	67

Índice de tablas

2.1. Métodos de calculo para el transporte de sedimentos	16
2.2. Distribución del número de elementos y nodos para cada subdominio .	27
2.3. Resultados en tiempos de cómputo del modelo paralelizado	34
4.1. Resultados de partición vertical, 2158 vértices, 3473 triángulos	59
4.2. Resultados de partición vertical, 7791 vértices, 3473 triángulos	62

Índice de figuras

2.1. Definición de carga y modo de transporte de sedimentos (Fuente ISO 4363 (2002))	15
2.2. Representación gráfica de las coordenadas sigmas.	18
2.3. Estructura de TELEMAC	20
2.4. Estructura de FVOM	22
2.5. Distribuciones de dominio topológico	27
2.6. Dominio completo, tramo Norato-Cochino del río Apure	29
2.7. Dominio completo, tramo Norato-Cochino del río Apure con sedimentos	30
2.8. Extensión del río Apure, Venezuela	31
2.9. Parte del río Apure, Venezuela	32
2.10. Diagrama de flujo	33
2.11. Comparación serial-paralela	34
2.12. Sistema de memoria distribuida	36
2.13. Estructura general de un programa MPI	37
2.14. Estructura física de un cluster	39
2.15. Estructura lógica de un cluster	39
2.16. Cluster tipo Beowulf	40
3.1. Cluster del IMF de la UCV	45
3.2. Esquema del Rack del cluster del IMF de la UCV	46
4.1. Archivos de entrada y salida del Modelo Shal2D	49
4.2. Propuesta de división 1	50
4.3. Propuesta de división 2	50
4.4. Propuesta de división 3	51
4.5. Algoritmo Shal2D en secuencial	52
4.6. Algoritmo Shal2D en paralelo	53
4.7. Salida al ejecutar Triangle con la opción -o2	54
4.8. Modificación líneas 14428 y 14432, texto original	55
4.9. Modificación líneas 14428 y 14432, texto nuevo	56

4.10. Modificación líneas 14753 y 14761, texto original	56
4.11. Modificación líneas 14753 y 14761, texto nuevo	57
4.12. Pasos para el refinamiento de la malla	58
4.13. Tiempo resultante de malla con 2158 vértices y 3473 triángulos	59
4.14. Resultados de malla con 2158 vértices y 3473 triángulos	60
4.15. Resultados de malla con 2158 vértices y 3473 triángulos	61
4.16. Resultados de malla con 2158 vértices y 3473 triángulos	61
4.17. Tiempo resultante de malla con 7791 vértices y 3473 triángulos	62
4.18. Resultados de malla con 7791 vértices y 3473 triángulos	63
4.19. Resultados de malla con 7791 vértices y 3473 triángulos	64
4.20. Resultados de malla con 7791 vértices y 3473 triángulos	64

Introducción

Un problema actual que enfrentan los investigadores en el área de la mecánica de fluidos es modelar el comportamiento físico de un caudal y obtener resultados de una manera más rápida. Una forma de solventar esto consiste en desarrollar aplicaciones computacionales que simulen el comportamiento hidrodinámico y de transporte de sedimentos que permiten realizar pruebas de forma automática, que en su defecto tendrían que hacerse manualmente.

Por ejemplo, se presenta el caso de un río cuyo caudal se debe actualizar y dependiendo de la magnitud de la malla (número de nodos y número de elementos), estos modelos pueden proveer resultados en un lapso de tiempo muy largo, por lo que a los ingenieros se les ha dificultado la generación de nuevos modelos debido a la complejidad.

El propósito de este TEG es desarrollar un modelo bidimensional hidrodinámico y de transporte de sedimentos Shal2D en paralelo que les permita a los ingenieros del Instituto de Mecánica de Fluidos (IMF) de la Universidad Central de Venezuela (UCV), reducir los tiempos de espera de respuesta en la ejecución del modelo. Además proveerles un refinamiento de la malla para obtener resultados más precisos.

Esto se lleva a cabo luego de investigar sobre la necesidad de tener un sistema de modelado hidrodinámico que estudia la dinámica de fluidos incompresibles y el transporte de sedimentos en esos fluidos como lo es Shal2D, que fue desarrollado por los ingenieros del IMF.

En este trabajo se provee el modelo Shal2D implementado en paralelo que permite obtener resultados precisos en un tiempo menor al Shal2D secuencial, basándose en el refinamiento y división de mallas. Se trabajó con el software triangle para el refinamiento de la malla generando más nodos, y se utilizó la librería MPI para la paralelización. Además vale destacar que se mantuvo el lenguaje de programación de origen del modelo, Fortran, y fue desarrollado sobre un cluster con el Sistema Operativo Scientific Linux de 64bits.

Los resultados del programa paralelo se comparan con los del secuencial y se evidencia la reducción en los tiempos de ejecución del modelo Shal2D.

A continuación se muestra un resumen del contenido de este trabajo clasificado por los capítulos que lo contienen:

El capítulo 1 trata sobre el planteamiento del problema. En este capítulo se describe el problema planteado y se establecen un conjunto de objetivos específicos y generales. Además se incluye la justificación del tema de este Trabajo Especial de Grado y se presenta un resumen de la metodología y alcance que delimita el proyecto.

El capítulo 2 es el marco teórico, donde se presenta la fundamentación teórica del TEG como conceptos y teorías utilizadas para la formulación de argumentos que sirven como base de este trabajo.

La plataforma tecnológica que describen las tecnologías a utilizar en el desarrollo de este Trabajo Especial de Grado y el enfoque dado al mismo se presentan en el capítulo 3 con el marco metodológico.

Además de esto se mostrará en el capítulo 4 el marco aplicativo donde se presenta la solución del problema con el sistema Shal2D paralelizado y la descripción de los datos con los que se realizaron las pruebas del modelo, así como el análisis de los resultados obtenidos.

Para finalizar se ofrecen las conclusiones y trabajos a futuro sobre este proyecto evaluando el cumplimiento de los objetivos planteados, así como las limitaciones encontradas, sugerencias y recomendaciones.

Capítulo 1

Planteamiento del problema

En este capítulo se presentan características generales sobre el Trabajo Especial de Grado como descripción del problema, los objetivos a cumplir, la justificación del tema de investigación y el alcance de la solución planteada.

1.1. Planteamiento del problema

Shal2D es un modelo bidimensional hidrodinámico y de transporte de sedimentos desarrollado por los ingenieros del Instituto de Mecánica de Fluidos (IMF) de la Universidad Central de Venezuela para el estudio de la mecánica del agua y el transporte de materiales en dichos caudales, sea por suspensión o arrastre, y con esto determinar velocidades y comportamientos finales de un fluido dado.

Luego de investigaciones con los ingenieros del Instituto se determinó que el modelo presentaba grandes inconvenientes referentes al tiempo de ejecución y obtención de resultados, ya que no obtenía respuesta sino hasta varios días después de ejecutado cada modelo. Esto no les era eficiente por lo que el sistema no ha sido aprovechado plenamente.

Esta diferencia de tiempo en cuanto a la obtención de resultados también dependían del tipo de hardware utilizado, pero además de esto, se observó que el preprocesamiento de los datos (la división de la malla) de la versión previa que fue paralelizada, se realizó de forma manual y con datos no contiguos, lo que generaba más tiempo de comunicación y por lo tanto más tiempo de espera en la producción de la salida.

1.2. Objetivos

1.2.1. Objetivos Generales

Para el presente Trabajo Especial de Grado se estableció el siguiente objetivo general, que consiste en paralelizar el sistema para modelado hidrodinámico y transporte de sedimentos Shal2D, mediante la optimización del código y refinamiento de la malla, además de el montaje en el cluster del Instituto de Mecánica de Fluidos de la Universidad Central de Venezuela.

1.2.2. Objetivos Específicos

Como objetivos específicos para este TEG se definieron los mostrados a continuación:

- Preprocesamiento de datos: Optimizar la malla triangular, de forma tal que los cálculos sean confiables y precisos, para ello se utilizará software triangle que me permite generar mallas bidimensionales y con triangulación de Delaunay.
- Automatización del modelo: Dividir la malla automáticamente por el sistema (paralelizando los datos) y no de forma manual, como fue desarrollado en la versión anterior.
- Adaptar la solución paralela del modelo Shal2D al Cluster del IMF, el cual consta de 8 nodos de dos procesadores cada uno bajo el sistema operativo Scientific Linux de 64bits, y utiliza el Network File System (Sistema de archivos de red NFS), Network Information Service (Sistema de Información de Red NIS), librerías MPICH2 y compiladores para Fortran, C entre otros, que permiten un buen funcionamiento.
- Evaluar la eficiencia de la solución paralela sobre el cluster.

1.3. Justificación de la solución

El propósito de este Trabajo es proveer a los ingenieros del Instituto de Mecánica de Fluidos de la UCV, una versión paralelizada del modelo bidimensional hidrodinámico y de transporte de sedimentos Shal2D que les permita aprovechar mejor el cluster del instituto para obtener resultados en un tiempo menor al Shal2D secuencial, de esta manera podrían obtener resultados en un tiempo más corto. Además de proveerles un código fuente que pueda ser modificado por ellos de una manera más fácil, bajo un

lenguaje de programación que es usado por ellos. Y poseer una malla refinada que les permita obtener resultados más precisos.

1.4. Alcance del proyecto

Shal2D será una aplicación de escritorio que podrá ser ejecutado sobre el cluster del Instituto para el análisis de datos de ríos y transporte de sedimentos, utilizado por ingenieros en investigaciones referentes al área de fluidos dentro del mismo instituto, por lo que no será un sistemas Web o dispondrá de una interfaz gráfica.

No se contempla la manipulación de imágenes ni el postprocesamiento de la información, además de aprovechar el paralelismo con el paradigma de memoria distribuida y no compartida.

1.4.1. Metodología de trabajo

La metodología de trabajo es el conjunto de pasos a seguir para el cumplimiento de los objetivos planteados previamente que permiten la ejecución del presente Trabajo Especial de Grado. En la siguiente lista encontrarán en resumen las actividades:

- Recolección de datos de entrada para el modelo Shal2D.
 - Uso del sistema Triangles para el refinamiento de la malla.
 - Prueba de la malla refinada con el programa Shal2D en secuencial.
 - Paralelización del código e implementación en MPI con el lenguaje de programación Fortran.
 - Pruebas en el cluster del Instituto y comparación de la versión secuencial vs paralela del modelo Shal2D.
 - Documentación del trabajo realizado.
-

Capítulo 2

Marco Teórico

Es este capítulo se describen definiciones de características importantes en el tema de los modelos hidrodinámicos y transporte de sedimentos, además de como las librerías para la programación paralela ayudan a estos sistemas, y las diferentes estructuras físicas en cuanto a la programación paralela se refiere. También veremos las ecuaciones que permiten realizar estos cálculos utilizados en el modelo Shal2D.

2.1. Modelos Hidrodinámicos

Un Modelo en ingeniería y cualquier otra ciencia, es una representación teórica muchas veces simplificada del prototipo del diseño de un sistema que se utiliza para predecir ciertos aspectos del comportamiento o naturaleza que pueden ser muy abstractos para la mente humana. Estos modelos pueden ser matemáticos, físicos, esquemáticos, gráficos y de simulación, dependiendo de lo que sea necesario modelar y lo que se requiera describir con el sistema. [1]

La validez de un modelo depende de si éste, independientemente de sus inexactitudes, puede dar una predicción confiable del funcionamiento de un sistema.

Las formas sinuosas en que estos fluidos cambian bruscamente y se retuercen sobre si mismas, presentan un problema matemático complejo. El avance de L. Euler quien en 1755 formalizó su descripción escribiendo por primera vez las ecuaciones diferenciales que rigen el movimiento de un fluido no viscoso, permitió un gran paso en esta área. Años después, en 1825 C. Navier y G. Stokes introdujo el término de viscosidad en las ecuaciones que hoy llamamos Navier-Stokes. Estas ecuaciones describen y cuantifican el comportamiento de fluidos, aunque en muchos casos su tratamiento exacto no es factible, por lo que corresponde evaluarlo desde la perspectiva computacional. [2]

Esta visión y las potencias de cálculo de los computadores actuales han permitido enormes avances, y se ha constituido en una rama de la Mecánica de Fluidos llamada Mecánica de Fluidos Computacional (CDF).

En particular, la hidrodinámica de fluidos se encarga de estudiar el movimiento de los líquidos, en particular el agua [3], aunque también se incluye el estudio de la dinámica de otros fluidos. Para ello se consideran entre otras cosas la velocidad, presión, flujo y gasto del fluido.

Un modelo hidrodinámico es una herramienta utilizada en sistemas de aguas superficiales complejas, para representar detalles en los patrones de transporte. Estos modelos son capaces de describir y representar de alguna manera el movimiento del agua y predecir el transporte de sedimentos en el mismo. También tiene numerosas aplicaciones industriales como diseño de canales, construcción de puertos y presas, fabricación de barcos y turbinas, entre otros.

Con el desarrollo tecnológico de los modelos numéricos (resolución de ecuaciones diferenciales), junto con sistemas de cálculo avanzados, la modelación hidrodinámica se ha convertido en el campo más amplio de la dinámica de fluidos computacional. Por lo que, los modelos hidrodinámicos utilizados para aplicaciones oceánicas costeras están relacionados con los modelos desarrollados para la meteorología, la industria aeroespacial, entre otras. La base común de estas actividades de modelación es la solución numérica de las ecuaciones de cantidad de movimiento y masa de un fluido.

Lo que diferencia la Modelación hidrodinámica de otras especialidades de la dinámica de fluidos computacional es su enfoque en el movimiento del agua. Estos modelos representan el medio ambiente costero por medio de las leyes de la física, resueltos por las técnicas de cálculo numérico, pero como en muchos casos realizan un gigantesco número de operaciones se requieren de sistemas de computación con alto desempeño.

Los Modelos de dinámica de fluidos computacional están limitados a las propiedades descritas por los sistemas de ecuaciones, la capacidad del algoritmo en la resolución de estas ecuaciones numéricas, y por la capacidad del sistema computacional.

La base de cálculo de los Modelos Hidrodinámicos está en el conjunto de ecuaciones que describen el movimiento de los fluidos, como lo hacen las ecuaciones de Navier-Stokes. Estas ecuaciones se derivan de la segunda ley de Newton de movimiento y

permiten describir la acción de la fuerza aplicada al fluido, es decir, los cambios resultantes en el flujo bajo un enfoque de medio continuo. La ecuación de Navier-Stokes define la propiedad de conservación del momento donde la aceleración depende de la fuerza ejercida y es inversamente proporcional a su masa.

La Hidrodinámica computacional también impone el principio de continuidad, en donde la masa y la energía se conservan siempre que se mantengan dentro del dominio. Para el modelado hidrodinámico, son usadas las ecuaciones para el flujo comprensible las cuales permiten modelar el comportamiento de un fluido Newtoniano de Navier-Stokes que son simplificadas por las propiedades específicas de los mares costeros. Las ecuaciones resultantes son las ecuaciones de aguas poco profundas como Shallow Water, llamado así por la escala de las características en la horizontal, la cual es mucho mayor que en la vertical, es decir, son usadas en aguas someras (superficiales) o ecuaciones de Saint Venant donde la longitud de onda es mucho mayor que la profundidad del fluido, por lo tanto la velocidad y la aceleración verticales son despreciables y el fluido se torna casi horizontal.

Las ecuaciones de agua superficial o poco profundas permiten la solución numérica más eficiente del flujo del agua en estos entornos. Es por ello que los modelos hidrodinámicos pueden ser usados para flujos de pequeña o gran escala. [4]

La complejidad del sistema de interés (río, estuario o el mar) impide la solución que rigen las ecuaciones analíticamente. Los científicos de computación han permitido a los investigadores abordar estos problemas complejos a través de métodos numéricos. Sin embargo, los computadores solamente pueden realizar cálculos discretos y por lo tanto las ecuaciones continuas se dividen en pequeños problemas individuales que pueden resolverse rápidamente en un computador.

Las ecuaciones de un modelo forman un sistema de ecuaciones diferenciales parciales que representan las propiedades de almacenamiento y transporte. El procedimiento de solución de dichas ecuaciones puede basarse en soluciones analíticas, o en métodos de solución numérica aproximadas tales como el de diferencias finitas o el de elementos finitos:

2.1.1. Métodos de análisis

Método de Diferencias Finitas:

Este método se ha utilizado para resolver ecuaciones de flujo y calidad del agua.

Posee gran simplicidad conceptual y facilidad de programación en un computador, ya que en este método las ecuaciones diferenciales se transforman directamente en ecuaciones aproximadas de diferencias finitas.

En términos generales los pasos a seguir para obtener la solución numérica son:

1. Seleccionar un método de diferencias finitas adecuado para este tipo de ecuaciones.
2. Discretizar las ecuaciones diferenciales.
3. Resolver el sistema de ecuaciones algebraicas obtenido.

En el proceso de discretización de las ecuaciones diferenciales por medio de diferencias finitas, las variables dependientes que describen el estado del sistema se consideran definidas en un número finito de puntos o nodos de la red computacional dentro del dominio a considerar, cuyas dimensiones están dadas por el número de variables independientes.

Al aplicar el esquema de diferencias finitas sobre los puntos de la malla, con un conjunto dado de condiciones de contorno expresadas de forma aproximada, se obtiene un sistema de ecuaciones lineales para cuya solución puede utilizarse un método iterativo como el de Gauss-Seidel.

La aplicación clásica del método de diferencias finitas a la solución de un problema de derivadas parciales en dos dimensiones involucra el cálculo de la discretización del dominio mediante una malla de celdas cuadradas o rectangulares de dimensiones fijas.

Su principal desventaja es que presenta limitaciones en su aplicación a dominios de geometría irregular.

Método de Elementos Finitos:

En este método el dominio se subdivide en segmentos o subdominios, y todos los balances son efectuados directamente sobre el elemento computacional, el cual puede tener una forma arbitraria.

Para construir el modelo de una geometría compleja, se utiliza una representación basada en formas geométricas simples, siendo usado frecuentemente elementos cuadrados, rectangulares y triangulares, los cuales pueden tener lados curvos.

Una ventaja de relevancia en este método es que permite una discretización muy flexible, de forma que puede reducirse el número de elementos utilizando elementos de mayor tamaño en las zonas del dominio numérico en las cuales espera observarse mayor uniformidad en los valores de las variables del problema, tales como las velocidades de flujo y concentraciones de las sustancias. Además la forma y orientación de los elementos puede adaptarse mejor a un contorno irregular, de donde resulta un mejor tratamiento de las condiciones de contorno.

Aunque la aplicación del método de elementos finitos no pueda efectuarse en forma tan directa como el método de diferencias finitas, este ha llegado a un alto desarrollo en las técnicas de solución de las ecuaciones, y tratamiento de geometrías irregulares. [6]

Ambos tipos de métodos numéricos se han aplicado a los sistemas de computación de alto rendimiento, que permite la simulación de complejas regiones costeras en alta resolución. Estos modelos hidrodinámicos costeros se aplican a muchos problemas ambientales de diferentes tipos, mediante configuraciones del modelo. Por ejemplo, estos modelos pueden ser utilizados en una profundidad promedio eficiente en aplicaciones de dos dimensiones (2D) o en un formulario completo de tres dimensiones (3D).

2.2. Mecánica de Fluidos

Un Fluido es una sustancia que se deforma continuamente, es decir, se escurre cuando esta sometido a un esfuerzo de corte o tangencial.

Las propiedades físicas de los fluidos, como todos los materiales, permiten caracterizar y cuantificar su comportamiento así como distinguirlos de otros. Algunas de estas propiedades son exclusivas de los fluidos y otras son típicas de todas las sustancias.

Características:

- Masa específica: Es la cantidad de materia por unidad de volumen de una sustancia.
 - Peso específico: Este valor corresponde a la fuerza con que la tierra atrae a una unidad de volumen.
 - Densidad: Es la relación que existe entre la masa específica de una sustancia
-

cualquiera y una sustancia de referencia. Para los líquidos se utiliza la masa específica del agua a 4°C como referencia.

- Viscosidad: Esta ligada a la resistencia que opone un fluido a deformarse continuamente cuando se le somete a un esfuerzo de corte. Los fluidos pueden ser en general clasificados de acuerdo a la relación que exista entre el esfuerzo de corte aplicado y la velocidad de deformación.
- Compresibilidad: Representa la relación entre los cambios de volumen y los cambios de presión a que esta sometido un fluido. Las variaciones de volumen pueden relacionarse directamente con variaciones de la masa específica si la cantidad de masa permanece constante. En general se sabe que en los fluidos la masa específica depende tanto de la presión como de la temperatura de acuerdo a la ecuación de estado.
- Presión de vapor: Cuando un líquido se le disminuye la presión a la que está sometido hasta llegar a un nivel en el que comienza a bullir, se dice que ha alcanzado la presión de vapor. Esta presión depende de la temperatura.
- Tensión superficial: Entre dos fluidos que no se mezclan ocurre un efecto que funciona como si fuera una membrana tensa. La tensión superficial es la fuerza que se requiere para mantener en equilibrio una longitud unitaria. El valor de ella dependerá de los fluidos en contacto y de la temperatura.

Características como la viscosidad, tensión superficial y presión de vapor solo se pueden definir en los líquidos y gases. Sin embargo la masa específica, el peso específico y la densidad son atributos de cualquier materia.

Un fluido dependiendo de las características que presente, puede ser agrupado para uno de los siguientes rangos de clasificación.

- Un fluido newtoniano: El cual es representado por aquel fluido cuya viscosidad puede considerarse constante en el tiempo.
- Un fluido no newtoniano: Es el fluido cuya viscosidad varía con la temperatura y presión, o que depende del tiempo.

2.3. Ecuaciones de Navier-Stokes

Estas ecuaciones reciben su nombre gracias a Claude-Louis Navier(1758-1836) y George Gabriel Stokes(1819-1903). Se trata de un conjunto de ecuaciones en derivadas

parciales no lineales de segundo orden que describe el movimiento de un fluido. Estas ecuaciones gobiernan la atmósfera terrestre, las corrientes oceánicas y el flujo alrededor de sedimentos, partículas, en general, cualquier fenómeno en el que se involucren fluidos newtonianos.

Estas ecuaciones se obtienen aplicando los principios de conservación en la mecánica y la termodinámica a un volumen fluido. Haciendo esto se obtiene la llamada formulación integral de las ecuaciones. Para llegar a su formulación diferencial se manipulan aplicando ciertas consideraciones, principalmente aquella en la que los esfuerzos tangenciales guardan una relación lineal con el gradiente de velocidad (ley de viscosidad de Newton), obteniendo de esta manera la formulación diferencial que generalmente es más útil para la resolución de los problemas que se plantean en la mecánica de fluidos.

Las ecuaciones diferenciales de movimiento para un fluido son aplicables a cualquier sólido o fluido en movimiento o reposo. Sin embargo antes de poder usar estas ecuaciones para resolver problemas específicos, es necesario obtener alguna información adicional sobre los esfuerzos.

Las ecuaciones 2.1, 2.2 y 2.3 correspondiente a las coordenadas x , y , z respectivamente. La ecuación de continuidad para flujos incompresibles cuando la densidad de flujo ρ es constante en todo el campo del flujo, obtenemos la ecuación 2.4:

$$\rho g_x + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} = \rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) \quad (2.1)$$

$$\rho g_y + \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} = \rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) \quad (2.2)$$

$$\rho g_z + \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = \rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) \quad (2.3)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (2.4)$$

La ecuación 2.4 es válida para flujos estables e inestables de fluidos incompresibles, se basa en la conservación de la masa y se obtuvo a partir de un método para volumen

de control.

Para fluidos newtonianos incompresibles se sabe que los esfuerzos están linealmente relacionados con las razones de deformación y se pueden expresar en coordenadas cartesianas:

Para esfuerzos normales:

$$\sigma_{xx} = -p + 2\mu \frac{\partial u}{\partial x} \quad (2.5)$$

$$\sigma_{yy} = -p + 2\mu \frac{\partial u}{\partial y} \quad (2.6)$$

$$\sigma_{zz} = -p + 2\mu \frac{\partial u}{\partial z} \quad (2.7)$$

Para esfuerzos cortantes:

$$\tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (2.8)$$

$$\tau_{yz} = \tau_{zy} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial y} \right) \quad (2.9)$$

$$\tau_{zx} = \tau_{xz} = \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \quad (2.10)$$

donde p es la presión, el negativo de la media de los tres esfuerzos normales; es decir, como se muestra en la ecuación 2.11

$$-p = (1/3)(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \quad (2.11)$$

Para fluidos viscosos en movimiento los esfuerzos normales no necesariamente son los mismos en direcciones diferentes; de ahí la necesidad de definir la presión como el promedio de los tres esfuerzos normales. Para fluidos en reposo, o fluidos sin fricción, los esfuerzos normales son iguales en todas las direcciones.

Una cuestión importante a observar es que mientras para sólidos elásticos los esfuerzos están linealmente relacionados con la deformación, para fluidos newtonianos los esfuerzos están linealmente relacionados con la razón de deformación.

Estos esfuerzos se pueden sustituir en las ecuaciones diferenciales de movimiento y simplificarse usando la ecuación de continuidad para obtener las direcciones x, y, z.

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial p}{\partial x} + \rho g_x + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \quad (2.12)$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = -\frac{\partial p}{\partial y} + \rho g_y + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \quad (2.13)$$

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial p}{\partial z} + \rho g_z + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \quad (2.14)$$

Donde las ecuaciones se han reordenado de modo que los términos de la aceleración estén en el miembro izquierdo y los términos de la fuerza estén en el derecho. Estas ecuaciones cuando se combinan con las ecuaciones de conservación de la masa, constituyen una descripción matemática compleja del flujo de fluidos newtonianos incompresibles. [7]

Las ecuaciones de Navier Stokes son consideradas las ecuaciones diferenciales de movimiento que rigen el movimiento de fluidos newtonianos incompresibles.

2.4. Transporte de Sedimentos

Los sedimentos se pueden definir como material fragmentado, formado principalmente por la desintegración física y química del suelo. Este material es transportado a los cuerpos de agua, principalmente por la acción de los cuerpos y el drenaje superficial. Estos sedimentos pueden ser clasificados de acuerdo a su diámetro como limos finos (sedimentos finos no cohesivos), entre otros.

La sedimentación es el proceso por el cual los materiales sólidos (sedimentos), son transportados por una corriente de agua se depositan o acumulan en el fondo del río, embalse, canal artificial, o dispositivo construido especialmente para tal fin. [8]

El mecanismo del movimiento de sedimentos es un fenómeno tridimensional, inestable y muy complejo, gobernado principalmente por el campo de corrientes, la temperatura y la calidad del agua y de los sedimentos. Dependiendo de las características del sedimento, fuerzas mecánicas y físico-químicas, actúan simultáneamente

creando un patrón de sedimentación. Sin embargo, haciendo las simplificaciones validas, utilizando la ecuación de dispersión-difusión con los términos adecuados al problema planteado, se pueden obtener buenos resultados a pesar de su complejidad.

El mecanismo de transporte, erosión y deposición de sedimentos cohesivos es de diferente naturaleza al que rige a los sedimentos no cohesivos, los cuales se basan en partículas discretas con dimensión y densidad constante. También el sedimento arrastrado por las mareas en algunos estuarios, puede llegar a ser tan fino que su modo de transporte es la suspensión.

De acuerdo con el origen de los sedimentos y según su comportamiento al ser transportados, se distinguen dos tipos de transporte o carga: a) Transporte de material de fondo, correspondiente al material del lecho del cauce que puede ser transportado sobre el fondo del río. b) Transporte de lavado, que corresponde al material más fino (generalmente arcillas y limos) con origen en la cuenca, o proveniente de la erosión que el mismo río produce, y es transportado en suspensión la mayor parte del tiempo, a excepción en zonas de aguas tranquilas donde el material muy fino puede decantarse (por ejemplo en los embalses).

La diferencia principal entre estos dos tipos de transporte radica en que el transporte de fondo depende de las características hidráulicas de la corriente y de las características físicas del material; lo anterior no ocurre con el transporte de lavado, pues en forma general un río puede transportar tanto material de lavado como llegue a él, casi independiente de las características hidráulicas de la corriente [39].

A continuación se presenta en la Figura 2.1, la clasificación del transporte de sedimentos según la norma ISO 4363 (2002) [9], de acuerdo al origen de los sedimentos y el modo de transporte de los mismos.

También, tenemos que de acuerdo con la clasificación de la Norma ISO-4363 las siguientes relaciones:

$$St = Sb + Sl$$

$$Sb = Sbb + Sbs$$

$$St = Sbb + Sbs + Sl$$

$$Ss = Sbs + Sl$$

$$St = Sbb + Ss$$

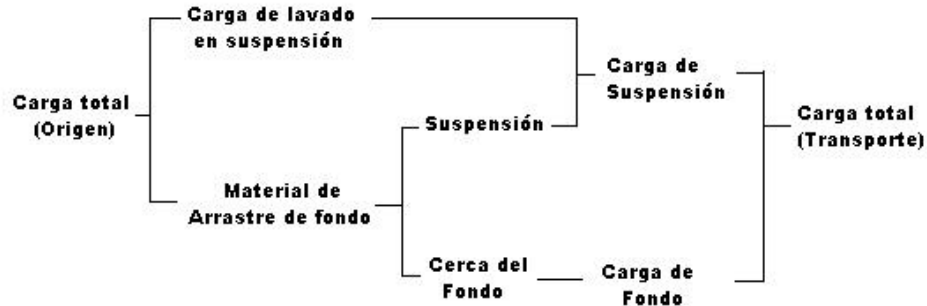


Figura 2.1: Definición de carga y modo de transporte de sedimentos (Fuente ISO 4363 (2002))

Donde:

S_t = Carga Total de Sedimentos.

S_b = Carga de Material de Fondo.

S_l = Carga de Lavado.

S_{bb} = Carga de Fondo.

S_{bs} = Carga en Suspensión.

S_s = Carga Total en Suspensión.

Además de esto, existen métodos para realizar cálculos de transporte de sedimentos. Alguno de ellos son los presentados en la tabla 2.1:

El método Meyer-Peter-Muller no es aplicable si hay muchos sedimentos suspendidos. [10]

Estos métodos permiten calcular el movimiento de acarreo de los sedimentos de un curso del agua, y están limitadas por condiciones definidas, como el caudal específico, la pendiente, el diámetro característico de los sedimentos, las relaciones geométricas y la densidad de los sedimentos. En general, la determinación del transporte de sedimentos depende tanto de las propiedades de los sedimentos como de las condiciones de flujo.

2.5. Modelos hidrodinámicos

Los Modelos hidrodinámicos son capaces de describir el movimiento del agua en una amplia gama de ambientes costeros. La salida puede incluir un historial de tiempo

Métodos para el cálculo de transporte de sedimentos		
Transporte de fondo (S _{bb})	Transporte en suspensión (S _{bs})	Transporte de material de fondo (S _b)
Meyer-Peter-Muller	Van Rijn	Engelund-Hansen
Van Rijn	Einstein	Ackers-White
Schoklitsch		Bagnold
Bagnold		Laursen
Einstein-Brown		Colby
Einstein Modificado		Toffaletti
Van Rijn		Inglish-Lacey
		Van Rijn

Tabla 2.1: Métodos de calculo para el transporte de sedimentos

de elevación de la superficie del agua, la velocidad de la corriente, la temperatura y salinidad, así como el transporte y el destino de los componentes.

Para la comprobación de precisión de los resultados que han sido demostrados en los modelos, se realiza la comparación de estos resultados con las observaciones de los experimentos físicos realizados por los investigadores.

Algunos modelos hidrodinámicos:

- DHM

El Modelo Hidrodinámico de Difusión DHM desarrollado en California aproxima los efectos hidráulicos de los canales, las superficies de la tierra y la interconexión de estos dos sistemas hidráulicos para representar desbordamiento de los canales y el flujo de retorno.[40]

El código computacional del Modelo Hidrodinámico de Difusión puede ser fácilmente manipulado por la mayoría de computadores actuales que soporten un compilador FORTRAN. En las típicas aplicaciones envolventes de problemas a gran escala. El pre y post procesador deberían ser desarrollados para facilitar las demandas de datos de entrada, y representar gráficamente la gran cantidad de resultados de los modelos generados por los modelos informáticos.

- COHERENS

Entre 1990 y 1998 un grupo de la multinacional Europea desarrolló el modelo hidrodinámico tridimensional (toma en cuenta la longitud, anchura y profundidad), multipropósito COHERENS (COupled Hydrodynamical Ecological model for RegioNal Shelf seas) para costas y plataformas marinas, las cuales se acoplan a la biología, modelos de re suspensión, físicos, transporte de sedimentos y contaminantes, además de resolver los procesos de mesoescala a escala estacional.

Importantes ventajas de este modelo son su transparencia debido a su estructura modular y su flexibilidad debido a la posibilidad de seleccionar diferentes procesos o sistemas de una aplicación particular. Esto permite su uso para estudios de procesos, así como para fines de predicción operativa sin previo conocimiento de su estructura detallada. Su código fuente se encuentra en FORTRAN y la evolución futura se puede implementar sin afectar el núcleo del programa. [11]

- OPTOS

OPTOS es un modelo hidrodinámico basado en el modelo COHERENS. El modelo permite el pronóstico de corrientes y elevación de la superficie marina inducida por la marea, la presión atmosférica y del viento. También puede ser utilizado para predecir la temperatura y salinidad.

Para ejecutar este modelo es necesario que se provea como entrada la batimetría (es la ciencia que mide las profundidades marinas para determinar la topografía del fondo del mar) en cada nodo de la malla, un conjunto de componentes de las mareas a lo largo de las fronteras abiertas, la presión atmosférica, los componentes de la velocidad del viento a intervalos de tiempo regulares, y otras variables meteorológicas necesarias para los módulos de temperatura y salinidad. [12]

En la figura 2.2 las líneas negras representan las superficies en donde la coordenada vertical sigma es constante, y la superficie más baja coincide con la superficie terrestre. [13]

- TRANSMER

El modelo hidrodinámico TRANSMER, es un modelo rápido y preciso para simulaciones de dispersión de Radionucleídos en el canal y el Sur del Mar del Norte de Europa, a medio y a largo plazo, como consecuencia de las emisiones de contaminantes solubles en el ecosistema marino en condiciones normales o

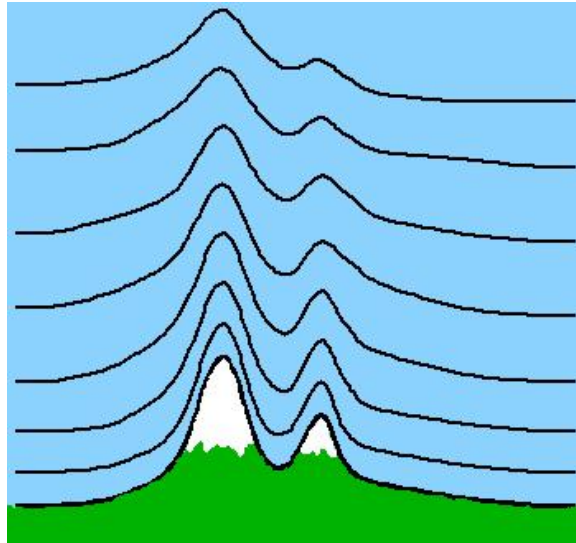


Figura 2.2: Representación gráfica de las coordenadas sigmas.

accidentales, con un costo computacional muy pequeño.

Este modelo utiliza las corrientes residuales de las mareas en coordenadas baricéntricas (centro de masa) calculadas con un modelo de corrientes instantáneas, pero aplicando el método desarrollado por Salomón y Orbi entre 1985 y 1993.

Las corrientes de marea instantáneas no se tienen en cuenta, el modelo es útil sólo cuando la corriente residual inclinada respecto al momento actual, es alrededor de una semana después de la aparición. Este cálculo permite simulaciones realistas de la dispersión de radionucleídos durante varios años. [14]

- POM

El modelo Ocean Princeton (POM) desarrollado por Blumberg y Mellor, 1980, 1987. Se utiliza para realizar cálculos en tres dimensiones. El POM simula patrones de circulación en largas y medianas escalas, la estratificación vertical, la velocidad de distribución y ondas de superficie. Este modelo también se utiliza para simular un balance térmico para lagos, y genera tensiones tangenciales turbulentas para el modelo de transporte de sedimentos.

El POM ha sido demostrado para poder simular la física predominante de grandes cuerpos de agua. El modelo incorpora el sistema de turbulencia Mellor-Yamada desarrollado en la década de 1970 por George Mellor y Ted Yamada, este sub-modelo de turbulencia es ampliamente utilizado por los modelos oceánicos y atmosféricos. [15]

- RMA2

RMA2 es un modelo numérico hidrodinámico en dos dimensiones con un promedio de profundidad de elementos finitos. Este modelo calcula elevaciones de la superficie de agua, los componentes de la velocidad horizontal y la libre circulación de la superficie del agua en los campos del flujo en dos dimensiones.

RMA2 calcula una solución de elementos finitos de Reynolds de la ecuación de Navier-Stokes para flujos turbulentos. La fricción es calculada con la ecuación de Manning's o de Chezy, y los coeficientes de viscosidad son usados para definir características de las turbulencias.

El RMA2 fue desarrollado por Norton, King y Orlob (1973). Opera bajo la hidrostática, donde las aceleraciones en la dirección vertical son despreciables. Se trata de dos dimensiones en el plano horizontal, pero para flujos más complejos donde las variaciones de los flujos verticales de las variables son importantes, deben ser evaluados utilizando un modelo en tres dimensiones como RMA10. [16]

- MIKE

En la versión clásica del modelo MIKE 3 se aplica un método de compresibilidad artificial. La base matemática es la ecuación de conservación de la masa para fluidos compresibles, las ecuaciones del promedio de Reynolds de Navier-Stokes en x, y, z direcciones, incluidos los efectos de turbulencia y densidad variable, junto con las ecuaciones de salinidad y la temperatura. [17]

MIKE FLOOD es un modelo hidrodinámico acoplado en una y dos dimensiones. Fue utilizado para simular el alcance y la profundidad de inundación en la región del delta de la cuenca del río Mahanadi en la India. [18]

El modelo MIKE 21 fue vinculado dinámicamente al modelo MIKE 11 en un paquete único llamado MIKE FLOOD desarrollado en el instituto Danés Hidráulico

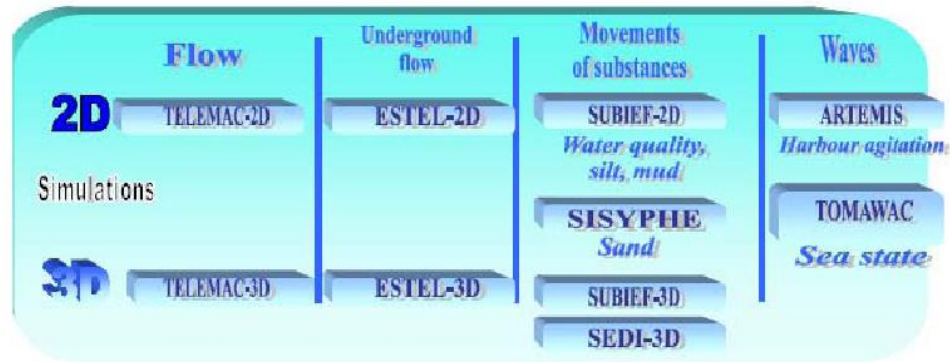


Figura 2.3: Estructura de TELEMAC

(Kjelds y Rungo, 2002; Rungo y Olesen, 2003). [19]

El modelo hidrodinámico MIKE 11 es implícito en diferencias finitas para un cálculo tridimensional de flujos inestables en ríos y estuarios, y se puede aplicar para las redes de bucles y simulaciones casi bidimensional en las llanuras de inundación (DHI, 1992) [20]. MIKE 11 es capaz de utilizar cinemática, difusa o completa dinámica, integrada verticalmente en las ecuaciones de masa y movimiento (ecuaciones de Saint-Venant). [21]

■ TELEMAC

TELEMAC 2D fue desarrollado por LNHE (Laboratoire National d'Hydraulique et Environnement) - Departamento de investigaciones (EDF) en el año 1987 [22]. Resuelve las ecuaciones de Saint-Venant que se derivan de las ecuaciones de Navier Stokes, mediante el método de elementos finitos o volumen finito, y un cálculo de mallas de elementos triangulares. Con esto se pueden realizar simulaciones de condiciones transitorias y permanentes. Muestra de esto en la imagen 2.3

TELEMAC-2D brinda al usuario un conjunto de subrutinas FORTRAN que pueden ser modificadas para ajustarse a los requisitos específicos de cada modelo, brindando la especificación de condiciones iniciales o condiciones de contornos complejos, vínculos con otros sistemas de modelado y la introducción de nuevas

funciones. También hay disponibles versiones paralelizadas, usando librerías PVM (Parallel Virtual Machine). [23]

- FVCOM

Grid Finite Volume Coastal Ocean Model (FVCOM) fue desarrollado por Chen et al (2003). Una de sus ventajas es que este modelo usa una malla no-estructurada para superficies libres, que se adapta al contorno de la costa y del fondo (3D) permitiendo mayor eficiencia en los cálculos. El modelo se basa en las ecuaciones de momento, continuidad, temperatura, salinidad y densidad, además de utilizar submodelos de cierre de turbulencia vertical. Estas ecuaciones se resuelven a través de volúmenes finitos donde se utiliza la idea de los métodos de elementos finitos (flexibilidad de red) y diferencias finitas (eficiencia numérica y la simplicidad de código). [24]

La estructura del FVCOM se muestran en la figura 2.4.

FVCOM se codificó originalmente para coordenadas sigmas en la vertical y ahora se ha actualizado una topografía generalizada siguiendo un sistema de coordenadas terrestres diversos topográficos siguientes coordenadas. FVCOM se escribe con Fortran 90 con paralelización MPI, y se ejecuta eficientemente en máquinas multi-procesadores. [25]

- ROMS

El modelo ROMS consiste en un conjunto de ecuaciones primitivas del océano ampliamente utilizado por la comunidad científica en una amplia variedad de aplicaciones (por ejemplo, Haidvogel et al., 2000; Marchesiello et al., 2003; Peliz et al., 2003 ; Di Lorenzo, 2003; Dinniman et al., 2003; Budgell, 2005; Warner et al., 2005a, b; Wilkin et al., 2005), para seguimiento de terrenos en superficies libres. [26]

El núcleo computacional de ROMS se compone con algoritmos no lineales. Estos se describen en detalle en Shchepetkin y McWilliams (2003, 2005), y los granos lineal tangente y adjunto y las plataformas se describen en Moore et al. (2004).

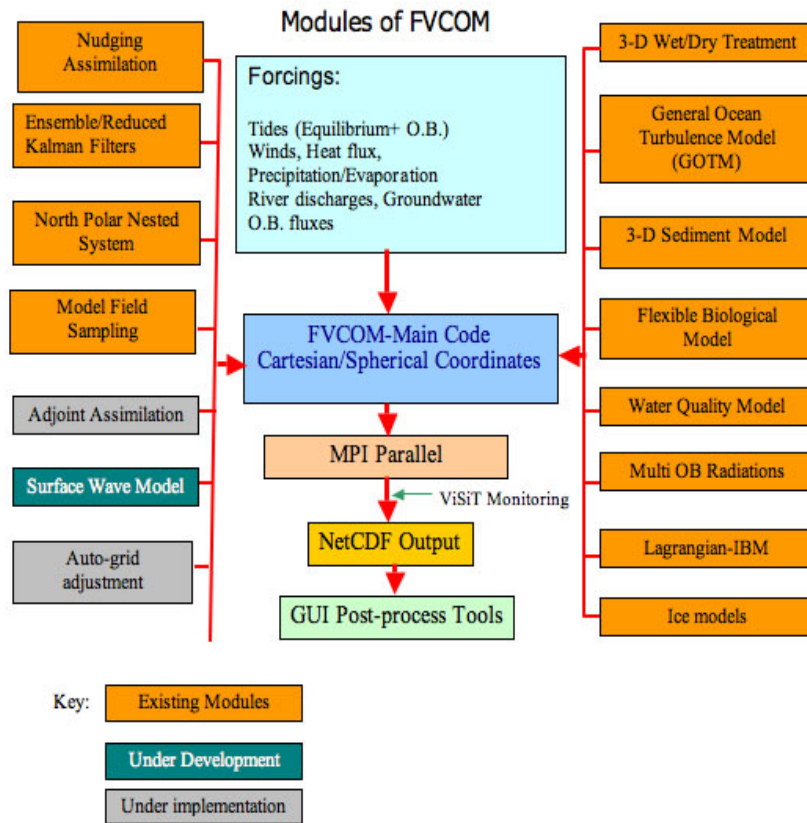


Figura 2.4: Estructura de FVOM

El modelo de hielo marino incluye varios sistemas de mezcla vertical, los múltiples niveles de anidamiento de las redes y compuesto. El algoritmo numérico en ROM está diseñado para reducir los errores de este tipo (Shchepetkin y McWilliams, 2003).

- SSIIM

Sediment Simulation In Intakes whit Multiblock option (SSIIM), es un programa que está diseñado para ser usado en la enseñanza e investigación de la ingeniería hidráulica, sedimentación, o estudio de ríos. Resuelve las ecuaciones de Navier-Stokes utilizando el método de control de volumen con un algoritmo SIMPLE y el modelo de turbulencia de k-epsilon, para simular flujos subcriticos. Además resuelve las ecuaciones de convención y difusión del transporte de sedimentos, usando la formula de van Rijn's para controlar el limite. También está incluido un modulo para calidad de agua.[27]

Se han creado versiones SSIIM para sistemas operativos de 64 bits. Estas versiones son necesarias para el modelado de las redes más grandes que alrededor de 4 millones de células. Se cuenta con una versión anterior de SSIIM para OS/2.[28]

2.6. Modelo Shal2D

Este modelo utiliza una variante del algoritmo conocido como "Selective Lumping" desarrollado por Kawahara (1981), que conduce a una discretización explícita en el parámetro tiempo.

El algoritmo SHAL2D, está basado en las ecuaciones de aguas poco profundas que describen un flujo con superficie libre, con la velocidad horizontal verticalmente integrada, es decir, la velocidad vertical del fluido en la superficie y el fondo, y se genera integrando la ecuación de conservación de la masa a lo largo de una columna de agua. El modelo se basa en la discretización de las ecuaciones de continuidad y de momento en las direcciones X e Y y de continuidad de los sedimentos. Se fundamenta en el algoritmo selective lumping en un proceso de cuatro pasos, y tiempo explícito para resolver en cada etapa el sistema de ecuaciones diferenciales ordinarias, que resulta de la aplicación del método de los residuos ponderados de Galerkin.

En el capítulo anterior se mostraron las ecuaciones de Navier-Stokes en tres dimensiones, pero vale destacar que Shal2D es un modelo bidimensional, por lo que esas ecuaciones varían, y se describen como se puede observar en las ecuaciones: 2.15, 2.16, 2.17 y 2.18.

$$\frac{\partial n}{\partial t} + \frac{\partial UH}{\partial x} + \frac{\partial VH}{\partial y} = 0 \quad (2.15)$$

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} + g \frac{\partial n}{\partial x} + \frac{T_{bx}}{\rho H} = 0 \quad (2.16)$$

$$\frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} + g \frac{\partial n}{\partial y} + \frac{T_{by}}{\rho H} = 0 \quad (2.17)$$

$$(1 - \lambda) \frac{\partial z_b}{\partial t} + \frac{\partial Q_{sx}}{\partial x} + \frac{\partial Q_{sy}}{\partial y} = 0 \quad (2.18)$$

donde x, y son las coordenadas horizontales, t el tiempo, n la elevación de la superficie del agua, H es la profundidad del agua, U y V son los promedios de velocidad en las coordenadas x , y respectivamente, ρ es la densidad del agua, g es la aceleración gravitacional z_b es la elevación del lecho, Q_{sx} y Q_{sy} indican el caudal de sedimentos en las direcciones x , y respectivamente, y λ indica la porosidad de los elementos. [29].

La discretización espacial lineal utiliza elementos triangulares. El tiempo de régimen de refuerzo, mejora el original método de Kawahara en dos fases, lo que permite mejorar su capacidad para simular el flujo complejo de superficie libre y transporte de sedimentos en los ríos.

Los esquemas numéricos explícitos son computacionalmente más rápido por etapa que esquemas implícitos, facilitando la paralelización e incorporación de métodos de adaptación, y son adecuados para simulaciones en corto plazo. Sin embargo, para simulaciones largas, los esquemas explícitos no pueden ser tan rentables como los sistemas implícitos.

El modelo explícito de dos dimensiones SHAL2D, está dedicado para resolver el promedio vertical de aguas poco profundas y las ecuaciones de transporte de sedimentos para las simulaciones en corto y largo plazo. El modelo implementa un tiempo

de cuatro pasos lo que permite intervalos de tiempo más largos que los anteriores modelos explícitos de elementos finitos. Además, se incorpora un algoritmo que reduce considerablemente los tiempos de cómputo a largo plazo, mientras que las simulaciones mantienen las ventajas de los esquemas explícitos. Los resultados del modelo se comparan muy bien con las soluciones analíticas para una simplificación y los problemas de dos dimensiones. Una simulación de cinco años para el Río Apure en Venezuela, demuestran la capacidad del modelo para predecir la evolución a largo plazo para los grandes ríos de una manera eficiente de cómputo.[29]

A menudo, las carreteras, puentes y canales de navegación se ven afectados por la evolución dinámica de los ríos aluviales. El uso de modelos matemáticos se ha convertido en una herramienta esencial para ayudar en el diseño de las obras de corrección, mejora de estructuras de navegación, y ayudar a predecir la evolución del lecho del río bajo la influencia de las estructuras de concreto. Estas aplicaciones incluyen la predicción de flujo complejo y los patrones de sedimentos con la presencia de bancos irregulares y batimetría.

Muchos esquemas de integración en el tiempo en el contexto de los métodos de elementos finitos para resolver las ecuaciones no estacionario de aguas poco profundas han aparecido en la literatura. Kawahara et al. (1978) presentaron un modelo de aguas someras bidimensionales basado en un esquema de dos pasos para la propagación de ondas del tsunami. Sin embargo, este esquema mostró excesivo cálculo numérico debido al uso de la matriz globalizada de masa. Más tarde, Kawahara et al. (1982) mejoraron el esquema original mediante la aplicación del algoritmo selective lumping (agrupamiento selectivo) que reduce sustancialmente el efecto de amortiguación, manteniendo las ventajas del uso de cómputo usando matrices de masa globalizadas. Kashiya et al. (1995) propusieron un método de tres pasos para las ecuaciones de aguas poco profundas basadas en el esquema original, selective lumping Kawahara que demostró reducir aún más el efecto de amortiguamiento al tiempo que permite mayores intervalos de tiempo.

2.6.1. Antecedentes de Shal2D en paralelo

Los avances en los equipos de cómputo y el aumento en la velocidad de las CPU's, han incrementado el ancho de banda en las comunicaciones, tanto en la reducción de los cambios de latencias como en el rendimiento bruto de los sistemas de hardware disponibles y su estructura. El software paralelo es la clave para alcanzar el potencial de rendimiento y generalización.

El lenguaje de programación Fortran está dedicado a científicos e ingenieros, y fue diseñado para un proceso de memoria compartida. Con la llegada de equipos vectoriales en los procesadores de un núcleo, fue necesario crear en el compilador Fortran instrucciones de vectores fuera de las múltiples instrucciones usadas para recorrer los arreglos.

Hay dos problemas fundamentales; la creación de un trabajo paralelo y la reducción de costos en el procesamiento de los datos. La solución a estos problemas incluye tanto a los usuarios como todos los aspectos del software paralelo; el compilador, el entorno en tiempo de ejecución, el sistema operativo y los entornos de usuario. Sin embargo, el usuario a menudo no quiere volver a trabajar con el código a menos que haya un rendimiento considerable, actualmente se espera un factor de aceleración de diez en la velocidad de procesamiento para que el esfuerzo de revisión valga la pena. Es por esto que el reto del software paralelo es tanto en rendimiento como portabilidad.

Fortran 90 y el nivel de usuario en el modelo computacional presentan dos desarrollos de gran valor en el cálculo numérico intensivo. Los arreglos en el lenguaje Fortran 90 permiten al usuario realizar cálculos en matrices. Un operando en una instrucción puede ser una matriz completa o una subsección de un arreglo, y las operaciones a nivel de elemento como sumas o multiplicación, las cuales pueden ser ejecutadas en paralelo.

Fortran es utilizado para el desarrollo de aplicaciones de investigación en el área de la ingeniería. Un ejemplo de esto es el modelo Shal2D el cual se basa en el método de elementos finitos. Fue desarrollado por los ingenieros del (IMF).

El modelo paralelizado fue validado contra los resultados arrojados por el modelo secuencial, simulando la hidrodinámica del tramo Norato-cochino del Río Apure, el cual tiene una longitud aproximada de 10 km. Este tramo fue discretizado en una malla de cálculo de 3481 elementos con 1987 nodos.[37]. Estos resultados se refieren a previas paralelizaciones del modelo.

Para destacar el aporte de la propuesta (Rodriguez, Freddy 2003) es importante indicar las deficiencias en este intento, ya que no estaban completos los pasos del modelo original y no se realizaron optimizaciones de ninguna clase.

Para este procedimiento se trabajó sobre un cluster compuesto por tres computadores Pentium IV, de 2 GHz en una red de 100 Mbps dual, por esto el dominio (elementos, nodos, condiciones de borde) fue dividido equitativamente en tres proce-

Distribución de dominios		
Subdominio	Nº de elementos	Nº de Nodos
1	1160	659
2	1162	667
3	1159	661
Total	3481	1987

Tabla 2.2: Distribución del número de elementos y nodos para cada subdominio

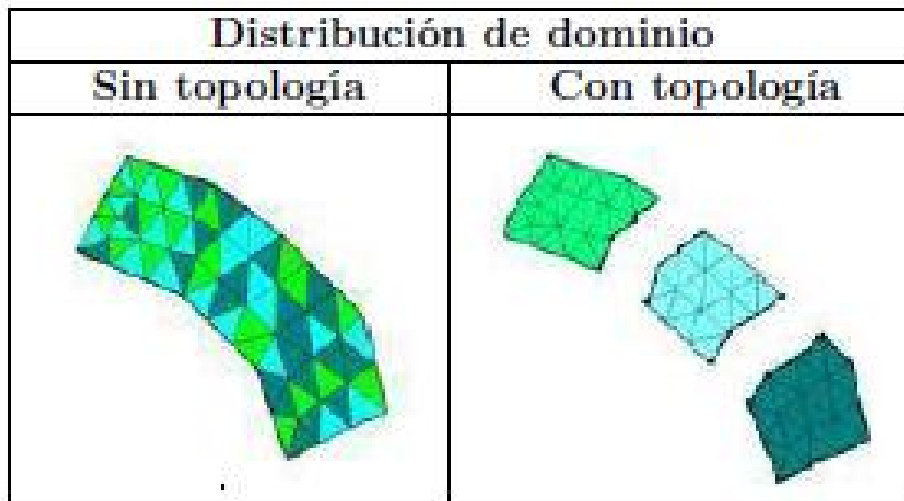


Figura 2.5: Distribuciones de dominio topologico

Los dominios, los cuales quedaron distribuidos tal como se muestra en la tabla 2.2.

Con esta distribución los subdominios 1 y 2 solo comparte ocho 8 nodos y entre el subdominio 2 y 3 comparte nueve 9 nodos.

Los dominios utilizados se pueden observar graficamente en la figura 2.5

Esta distribución de la malla, utiliza datos no contiguos, lo que genera más tiempo de comunicación entre procesos. En las imágenes se puede observar que las divisiones se hicieron por el color que tiene cada triángulo los cuales fueron tomados para cada conjunto de datos asignados a los nodos.

El dominio completo de los datos de entrada se pueden observar en la figura 2.6 y en 2.7, el cual pertenece al tramo Norato-Cochino del río Apure, con una longitud aproximada de 10Km.

El río apure 2.8 tiene una longitud de 820Km aproximadamente y es el mayor afluente venezolano del río Orinoco.

Otra vista de parte del río Apure se puede observar en la figura 2.9, donde se muestra varias de las localidades del río como el tramo Norato, Cochino, la Esperanza y el Encantado.

En este desarrollo se utilizó MPI (Message Passing Interface), que permiten la realización de aplicaciones paralelas basadas en paso de mensajes, para manejar el intercambio de información y la sincronización, bajo el lenguaje de programación C.

Para ese intento previo de paralelización del modelo hidrodinámico Shal2D se usó el esquema del modelo explícito de dos pasos, el cual se divide como lo muestra la figura 2.10.

Los resultados arrojan una reducción del tiempo de cómputo en un 66 por ciento, con el empleo de tres computadores, aunque el valor teórico debería estar alrededor del 75 por ciento. Esto no se logró por dos factores: uno la rapidez de la red empleada; y otro, la ejecución de código extra necesario para su paralelización, que el modelo secuencial no posee. [38]

Además de esto, luego de este intento, se perdió el trabajo en equipo por diferencias en el uso de los lenguajes de programación, y no se logró hacer ejecuciones de programa con un mayor número de procesadores, que son grandes limitantes para la continuidad de este modelo.

La versión paralelizada no fue utilizada ya que el lenguaje con el trabajaron era C, pero este lenguaje está más orientado redes, desarrollo de compiladores y sistemas operativos, no como Fortran que es un lenguaje diseñado al cálculo numérico. Esta es la razón por la cual los ingenieros desarrollaron el programa Shal2D desde el inicio en Fortran.

Otro problema de esta versión resultó al momento de realizar la división de la malla, ya que esta se hizo de forma manual y en tres partes estáticas, como también la toma de datos no contiguos, lo que provoca más tiempo de comunicación entre pro-

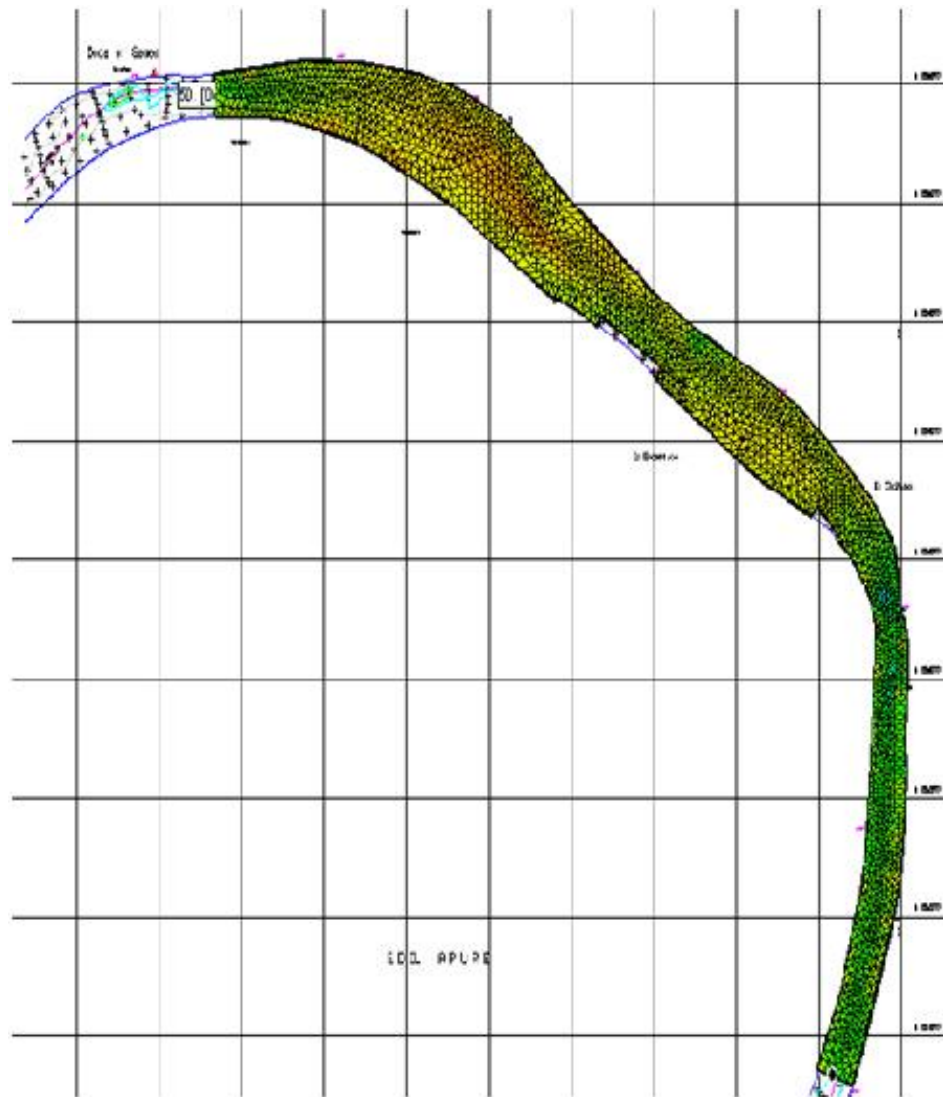


Figura 2.6: Dominio completo, tramo Norato-Cochino del río Apure

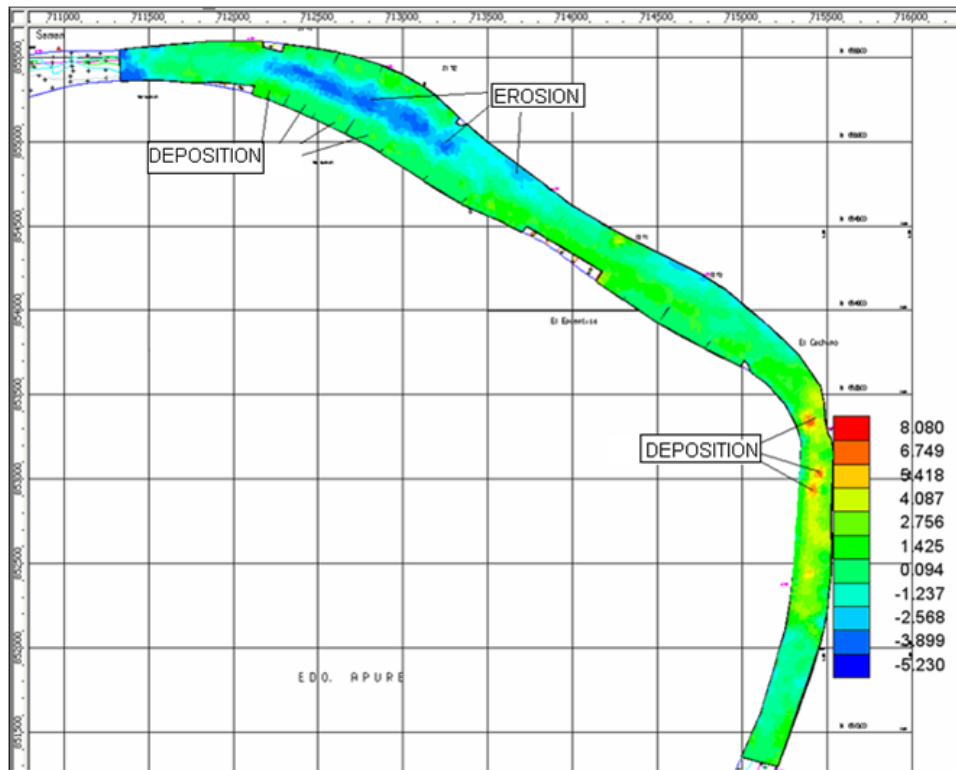


Figura 2.7: Dominio completo, tramo Norato-Cochino del río Apure con sedimentos



Figura 2.8: Extensión del río Apure, Venezuela



Figura 2.9: Parte del río Apure, Venezuela

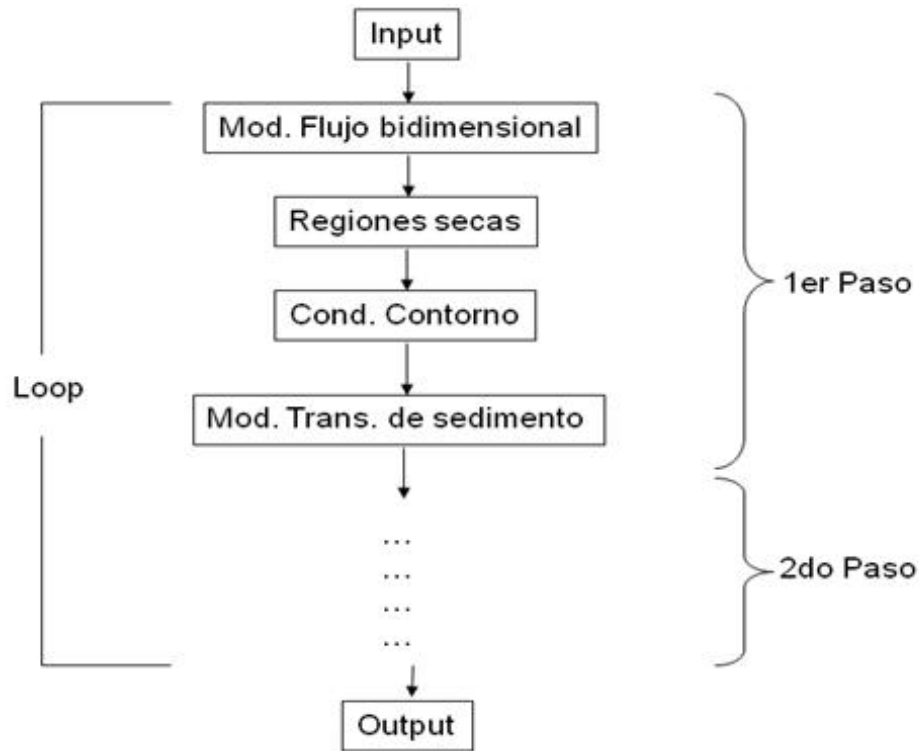


Figura 2.10: Diagrama de flujo

cesos.

Para esta etapa se hicieron varias pruebas, tanto para el modelo paralelizado como el modelo secuencial. Los resultados de los valores medios obtenidos en cuanto al tiempo de cómputo se muestran en la tabla 2.3.

2.7. Sistemas en paralelo

La computación de alto rendimiento se apoya en tecnologías computacionales como clusters, supercomputadores, máquinas SIMD (Single Instruction Multiple Data), paralelos (MIMD, Multiple Instruction stream, multiple Data stream) y programación paralela, y mejora drásticamente problemas de cómputo complejo o cálculos de cómputo intensivo. La mayoría de las ideas actuales de la computación distribuida se han basado en la computación de alto rendimiento.

Tiempos de simulación			
Tiempo de Simulación	Tiempo del Modelo Secuencial	Tiempo del Modelo Paralelo	Relacion Paral/Secuenc
30 días	5.7 horas	2.5 horas	0.44

Tabla 2.3: Resultados en tiempos de cómputo del modelo paralelizado

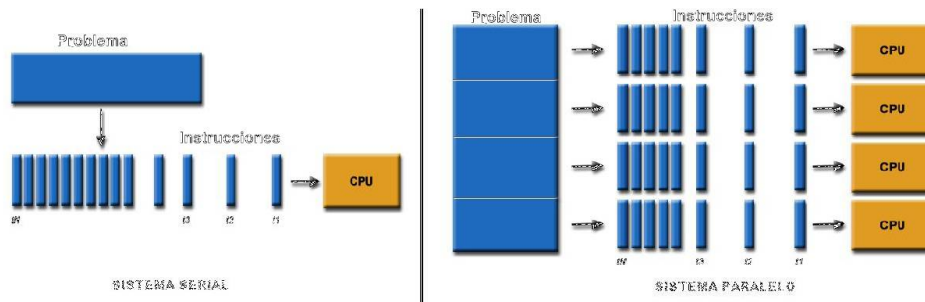


Figura 2.11: Comparación serial-paralela

Los programas en serie se ejecutan en un solo procesador y núcleo y cuentan con una sola unidad de procesamiento (CPU), las instrucciones se ejecutan una tras otra. Mientras que la computación paralela consiste en el uso simultáneo de múltiples recursos físicos de cómputo para resolver un problema computacional y permite ser ejecutado mediante múltiples CPUs, un problema se divide en partes discretas que pueden ser resueltos al mismo tiempo y cada una de estas partes se han subdividido a una serie de instrucciones y que cada parte se ejecute simultáneamente en diferentes CPUs. Figura 2.11

Entre esos recursos de cómputo se puede incluir una computadora con múltiples procesadores, un número de computadores conectados por una red o una combinación de ambos. [30]

La computación paralela se ha utilizado para modelar problemas complejos científicos y de ingeniería en el mundo real, por ejemplo:

- Ambiente, atmosfera y tierra
- Física; aplicada, nuclear, de partículas, materia condensada, alta presión, fusión

y fotónica

- Biociencia, biotecnología, genética
- Química, ciencias molecular
- Geología, sismología
- Ingeniería mecánica; fluidos, a partir de prótesis para naves espaciales
- Ingeniería eléctrica, Diseño de circuitos, Microelectrónica
- Ciencias de la computación, matemáticas, entre otras.

En la actualidad, las aplicaciones comerciales proporcionan igual o mayor fuerza impulsora en el desarrollo de sistemas de cómputo más rápidos, esto se debe a que estas aplicaciones requieren el procesamiento de una gran cantidad de datos en forma sofisticada. Por ejemplo:

- Bases de datos y minería de datos
- Exploración petrolera
- Motores de búsqueda web, y servicios empresariales basados en web
- Proyección de imágenes médicas y diagnóstico
- Diseño farmacéutico
- Gestión corporativa nacional y multinacional
- Modelado económico y financiero
- Gráficos avanzados y realidad virtual, particularmente en las industrias de entretenimientos
- Videos de red y tecnologías multimedia
- Entornos de trabajo colaborativos

El paralelismo es una forma de computación en la cual varios cálculos pueden realizarse simultáneamente, basado en el principio de dividir los problemas grandes para obtener varios problemas pequeños, que son posteriormente solucionados en paralelo. Hay diferentes tipos de paralelismo: a nivel de bit, a nivel de instrucciones, de datos, de tareas, de aplicación y de micro operaciones. El paralelismo ha sido empleado durante muchos años, sobre todo para la Computación de alto rendimiento.

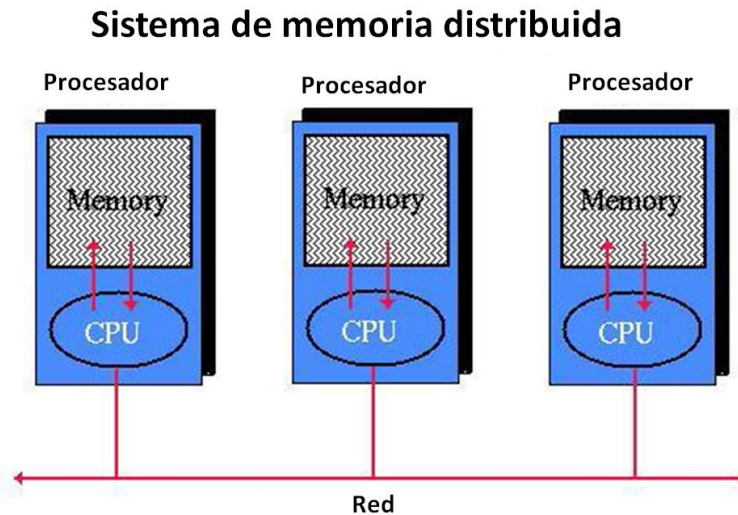


Figura 2.12: Sistema de memoria distribuida

2.7.1. MPI

Interfaz de pase de mensajes o MPI (Message Passing Interface) es una librería estándar para el pase de mensajes basada en el consenso del Foro MPI, el cual cuenta con más de 40 organizadores participantes, incluidos los vendedores, investigadores, desarrolladores de software de biblioteca, usuarios y clientes. El objetivo de MPI es establecer un estándar portable, eficiente y flexible para el paso de mensajes. [31]

MPI se presta a cualquier modelo de programación paralela con memoria distribuida (figura 2.12). Por otra parte, MPI se utiliza comúnmente para aplicar algunos modelos de memoria compartida, como de datos en paralelo en las arquitecturas de memoria distribuida. En la figura 2.13 se presenta como la estructura general de un programa MPI:

La Plataforma de MPI es un sistema de memoria distribuida incluyendo máquinas en paralelo, grupos de SMP, clusters de estaciones de trabajo y redes heterogéneas. Todo paralelismo es explícito: el programador es responsable de la correcta identificación de paralelismo y la aplicación del algoritmo resultante usando MPI construcciones (como las de la imagen anterior).[32]

El número de tareas dedicada a un programa paralelo es estática. Las nuevas tareas no pueden ser dinámicamente generados durante el tiempo de ejecución.

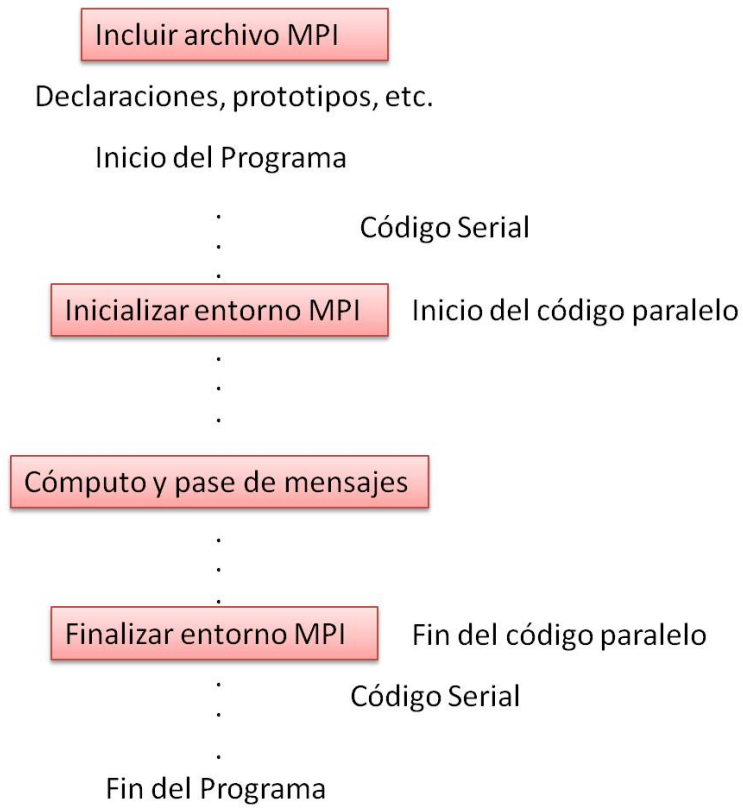


Figura 2.13: Estructura general de un programa MPI

Puede ser usado con C y Fortran. Los enlaces de lenguaje C++ y Fortran 90, son objeto de MPI-2.

También existe otra librería de multiprocesamiento denominada OpenMP (Open Multi-Processing) que permite el desarrollo de aplicaciones con el uso de memoria compartida. La API coporta C, C++ y Fortran (77, 90 y 95) en múltiples arquitecturas, incluyendo UNIX y Windows NT [33]. El entorno de ejecución asigna a los procesadores las actividades en función del uso, carga de la máquina y otros factores. El número de hilos pueden ser asignados por el entorno de ejecución basado en variables de entorno o en el código utilizando funciones.

2.7.2. Cluster

Un cluster es un conjunto de computadoras que utilizan componentes comunes y actúan como si fuera un solo sistema u ordenador, es decir, aquellos conglomerados informáticos que se construyen a partir de piezas de hardware compartidas que impulsan el comportamiento de todas las unidades tal como si fuesen una sola.

Este tipo de conjuntos se utilizan en áreas científicas, de la ingeniería y de diverso tipo de investigación para operaciones a gran escala. También se utilizan clusters en la configuración de servidores y bases de datos de alto rendimiento. [34]

El funcionamiento de los clusters es muy simple: se trata de un conjunto de piezas, por ejemplo, de varios microprocesadores a través de una red de alta velocidad que los vincula de forma tal que funcionan como un único ordenador pero de una potencia mayor al convencional.

Esto es posible porque existen piezas de hardware y de software económicas que hacen factible la configuración de unidades a gran escala con un bajo costo pero de alto rendimiento. Además, de presentar alta disponibilidad, equilibrio de carga y escalabilidad.

En la figura 2.14 se muestra un cluster físico y en la figura 2.15 se muestra la estructura lógica de un cluster.

Típicamente, además de conectar a los computadores entre sí para conformar el cluster, se incluye un sistema que facilita el manejo del conjunto, permitiendo que el usuario disponga de una experiencia optimizada en los procesos que corren en el cluster.



Figura 2.14: Estructura física de un cluster

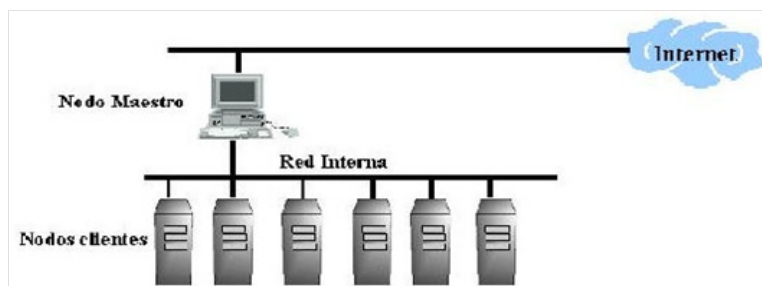


Figura 2.15: Estructura lógica de un cluster



Figura 2.16: Cluster tipo Beowulf

Existen distintos tipos de clusters. Por ejemplo, el homogéneo se vale de la misma configuración de hardware y sistema operativo en todos sus componentes. Por otro lado, el semi-homogéneo posee un rendimiento disímil pero la arquitectura general es similar. El cluster heterogéneo, en cambio, tiene distinto hardware y sistema operativo, esto los hace más económicos y rendidores.

Beowulf fue construido por Donald Becker y Thomas Sterling en 1994, con 16 computadores personales con procesadores Intel DX4 de 200 MHz, que estaban conectados a través de un switch Ethernet. El rendimiento teórico era de 3.2 GFlops.

La figura 2.16 es un ejemplo del cluster Beowulf, el cual utiliza como sistema op-

erativo Unix.

Además de este tipo de infraestructura como los es el cluster, existen otras como cloud computing y grid que permiten la integración colectiva de computadores de alto rendimiento, a través de redes mucho más rápidas y con acceso a más componentes de hardware, para el desarrollo de aplicaciones de cómputo avanzado[35].

Esta tecnología ha sido aplicada a los problemas científicos, matemáticos, y académico de cómputo intensivo de computación a través de voluntarios, y se utiliza en las empresas comerciales para aplicaciones diversas, como el descubrimiento de fármacos, previsiones económicas, análisis sísmicos y de procesamiento de datos en apoyo al comercio electrónico y servicios Web. [36]

Capítulo 3

Marco Metodológico

En el capítulo presentado a continuación se expone información relacionada con los componentes tecnológicos utilizados para este Trabajo Especial de Grado, como los elementos de hardware necesarios para la implementación de la aplicación.

3.1. Plataforma Tecnológica

En cuanto a la plataforma se especifica el contenido del elemento computacional primario tanto a nivel de software como de hardware.

Software:

Cluster de 9 nodos con el sistema operativo Scientific Linux del Instituto de Mecánica de Fluidos de la Universidad Central de Venezuela. Este tiene en su configuración; NFS (Network File System), protocolo a nivel de aplicación que permite en un entorno de distintos sistemas conectados, el manejo de archivos remotos como si se trataran de archivos locales., NIS (Network Information Service), protocolo de servicio de directorios cliente-servidor que permite el envío de datos de configuración en sistemas distribuidos como nombres de usuarios y hosts; y MPICH2, librería que posee una gran implementación del estándar MPI para el paso de mensajes en aplicaciones de memoria distribuida.

Hardware:

El cluster del IMF es del tipo 1350 IBM, contiene un nodo de control y otros 8 nodos para computo. Los nodos cumplen con las siguientes características;

- 2 Unidades Centrales de Procesamiento (CPU) AMD Opteron(tm) 246 (2GHz,

1MB L2, 64 bits)

- Memoria de 12 GB (DDR, PC2700, SDRAM)
- Myrinet 225 MHz Fiber/PCI-X (2MB)
- Broadcom 10/100/1000 Gigabit Ethernet NIC

Además de las especificaciones anteriores las cuales son comunes entre ellos, el nodo de control posee 3 Broadcom del tipo especificado anteriormente y 2 discos ATA/100 120GB, mientras que los nodos de computo tienen 2 Broadcom discos SCSI 36 GB Ultra320 15K RPM. La figura 3.1 muestra la vista externa (consola y rack) del cluster del insituto, y la 3.2 el esquema interno del rack del cluster.

3.2. Tecnologías utilizadas

En la siguiente lista se muestran las tecnologías utilizadas para el desarrollo de este TEG.

- Sistema Operativo Scientific Linux 64bits.
- Software triangle para la división de la malla y software showme para la visualización.
- Lenguaje de programación Fortran 90.
- Librerías MPI en fortran para la paralelización.
- Compiladores gfortran y mpif90.
- Shal2D en secuencial.
- Nodos del cluster del Instituto de Mecánica de Fluidos de la UCV

En el siguiente capítulo se podrá observar una nueva paralelización del modelo Shald2D, que es la propuesta de este trabajo, y los pasos llevados a cabo para cumplir con los obojivos específicos previamente nombrados.



Figura 3.1: Cluster del IMF de la UCV

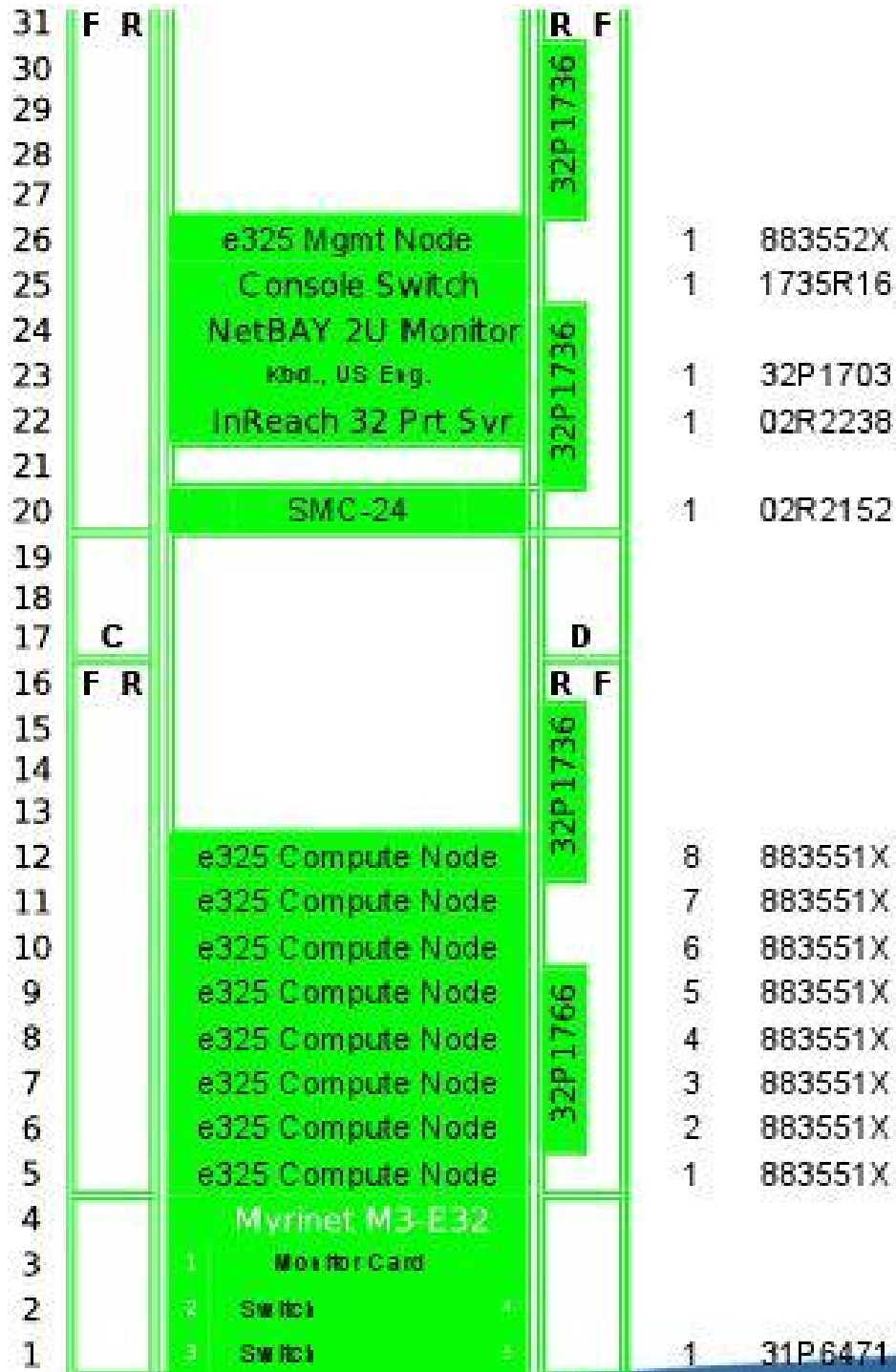


Figura 3.2: Esquema del Rack del cluster del IMF de la UCV

Capítulo 4

Marco Aplicativo

En esta sección se mostrarán los pasos y los métodos utilizados para llegar a la solución planteada y como los datos de la malla fueron refinados para conseguir resultados más aproximados a la realidad.

4.1. Recolección y análisis de datos

Se presentan los archivos de entrada y salida con su respectivo detalle, ya que son necesarios para el programa Shal2D.

4.1.1. Datos de entrada

Estos archivos son requeridos para la ejecución del algoritmo, es decir, los datos de entrada para el sistema de simulación.

- Archivo *.NO: Este tipo de archivo contiene la información general de las simulaciones, como pasos de tiempo de cálculo, pasos de tiempo en la generación de resultados parciales, datum elevation (datos de elevación), y un grupo de números de nodos para los cuales se desean resultados particulares, entre otros. El datum elevation es un dato que representa el nivel de donde se mide la profundidad del flujo de agua, es decir, la elevación desde donde se mide la profundidad para cada sector del flujo de agua. «Imagen del datumElevation»
- Archivo *.FED: (.NODO Y .ELE) Este archivo contiene los datos de la malla de elementos finitos como número de elementos, número de nodos, coordenadas y elevación del fondo para cada nodo. La elevación del fondo de cada nodo es llamado ZF. El ZF representa la elevación del sedimento bajo el agua y

la profundidad es la distancia entre el datum elevation y el sedimento que se encuentra bajo el agua. «ejemplo de profundidad»

- Archivo *.TBA: Este archivo incluye un listado de nodos de contorno para islas y tierra.
- Archivo *.IFL: Contiene la información relacionada con los nodos condicionados y el nombre y ubicación de los archivos con los datos de las condiciones de borde. En este archivo deben indicarse las condiciones de contorno en cada uno de los nodos condicionados mediante otros archivos con extensiones *.QVT, para hidrogramas de Caudal vs. Tiempo y *.HVT para hidrogramas de elevación del agua vs tiempo.
- Archivo *.SED: Posee información sobre los sedimentos.

4.1.2. Archivos de salida

Estos archivos representan la salida del modelo Shal2D luego de su ejecución

- Archivo *.HHH: contiene datos de velocidades, profundidad, elevación inicial y elevación final, para cada tiempo de impresión (el intervalo para el tiempo de impresión es indicado como parámetro de entrada del algoritmo) para un grupo de nodos indicados en el archivo *.NO.
 - Archivo *.OUT: Contiene información sobre los nodos, como ubicación y elevación de los elementos y nodos que lo conforma, además de las coordenadas de los nodos de borde.
 - Archivo *.UVH: Este archivo incluye los valores de velocidades, nivel de la superficie del agua, profundidad, elevación inicial, elevación final y el cambio de la elevación del fondo para cada tiempo de impresión de resultados y para cada nodo.
 - Archivo *.VTP: Archivo en forma binario que incluye toda la información de los nodos, para el último paso de tiempo de cálculo, como velocidades, elevación del fondo, entre otros. La existencia de este archivo le indica al modelo que debe reiniciar los cálculos a partir del tiempo para el cual se grabó el archivo *.VTP. Este archivo es un respaldo si por alguna razón no se puede terminar la ejecución de forma inesperada, aquí estarán guardados los resultados del último instante de tiempo para empezar el programa desde ese punto.
-

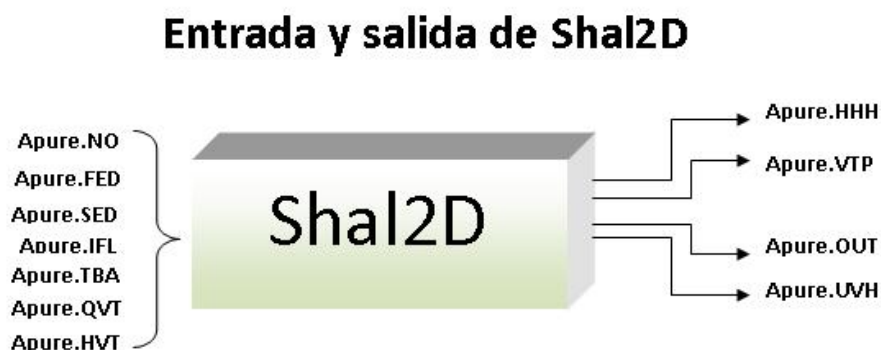


Figura 4.1: Archivos de entrada y salida del Modelo Shal2D

Los archivos de datos y resultados tienen el mismo nombre y solo difieren de la extensión, y los archivos de condiciones de contorno tienen un nombre libre, pero con la extensión *.QVT para la condición del caudal y con extensión *.HVT para las condiciones de nivel, pero deben estar especificados en el archivo *.IFL. Ver figura 4.1

4.2. Diseño e implementación de la solución

El método utilizado para la paralelización está basado en la segmentación de datos conocida como descomposición del dominio de datos (la malla) y donde todos los procesadores (tanto el de control o master, como los demás) realizan funciones de computo, con la diferencia de que al principio solo el maestro tiene acceso a los datos de entrada y es este quien los lee para transmitirlos al resto de los procesadores.

El mecanismo utilizado fue el envío de todos los datos y cada procesador realiza particiones locales tomando los datos que le corresponde, esto permite que los procesadores tengan a pesar del trabajo repetido, menos tiempo de ocio y más tiempo de trabajo simultaneo.

La partición de la malla se realizó de tres formas diferentes; vertical, horizontal y combinada (vertical y luego horizontal). La figura 4.2 muestra la división en fragmentos verticales para el caudal. La figura 4.3 se basa en divisiones en fragmentos horizontales. La figura 4.4 presenta la división en fragmentos verticales y horizontales para el caudal.

Las figuras 4.2, 4.3 y 4.4 muestran a la izquierda una porción de la malla completa del río como se trabaja en el modelo Shal2D en secuencial, y a la derecha la malla

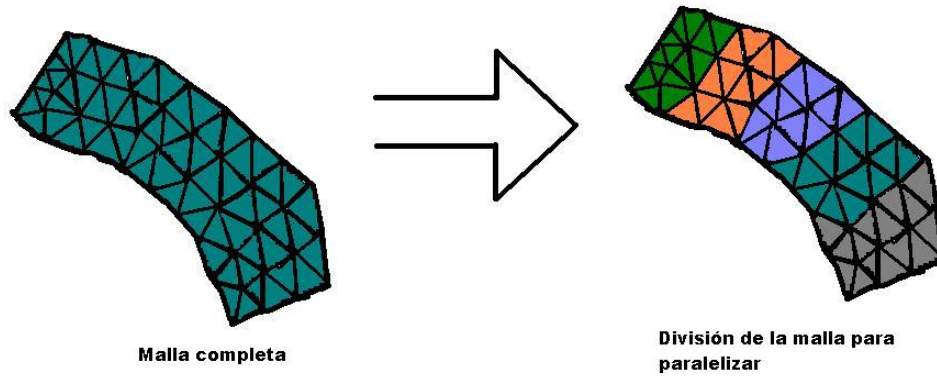
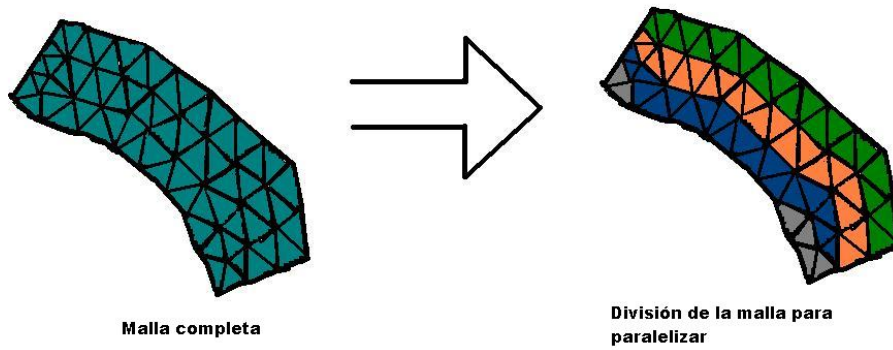


Figura 4.2: Propuesta de división 1

Figura 4.3: Propuesta de división 2

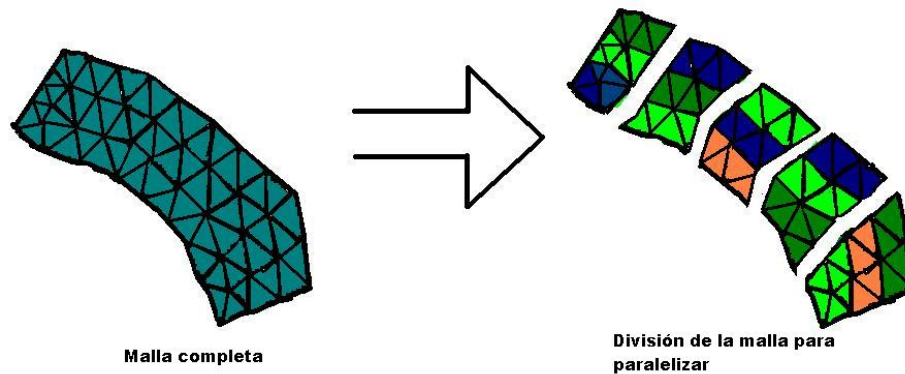


Figura 4.4: Propuesta de división 3

dividida por color, donde cada color indica que todos esos triángulos consecutivos (de igual color), son enviados a un procesador.

En la figura ?? el algoritmo en secuencial y en la figura 4.6 se presenta el algoritmo del modelo Shal2D paralelo. Donde se incluyen las comunicaciones entre procesadores en el algoritmo paralelizado para mantener los datos actualizados, y se puede observar que ahora más de un procesador está realizando la tarea de forma simultánea y no uno solo como en el caso del secuencial.

4.3. Pruebas y análisis de resultados

4.3.1. Refinamiento de datos

Se muestran los datos que se usaron para la interpolación. En un principio habían 3473 triángulos y 2158 vértices, y se generaron 0 triángulos y 5633 vértices nuevos. La explicación a esta información es que se crean más puntos pero no se modifican los triángulos, es decir, los vértices siguen siendo los mismos que en un inicio.

Esto se obtuvo con el sistema triangle y el comando -o2 el cual nos permite generar elementos subparamétricos de segundo orden con 6 nodos cada uno. Obteniendo 7791 vértices y 3473 triángulos, que como dijimos anteriormente es la misma cantidad de triángulos iniciales pero ahora con 6 nodos en cada triángulo. La salida de triangle es mostrada en la figura 4.7

Para realizar el refinamiento de datos con este sistema fue necesario tener los

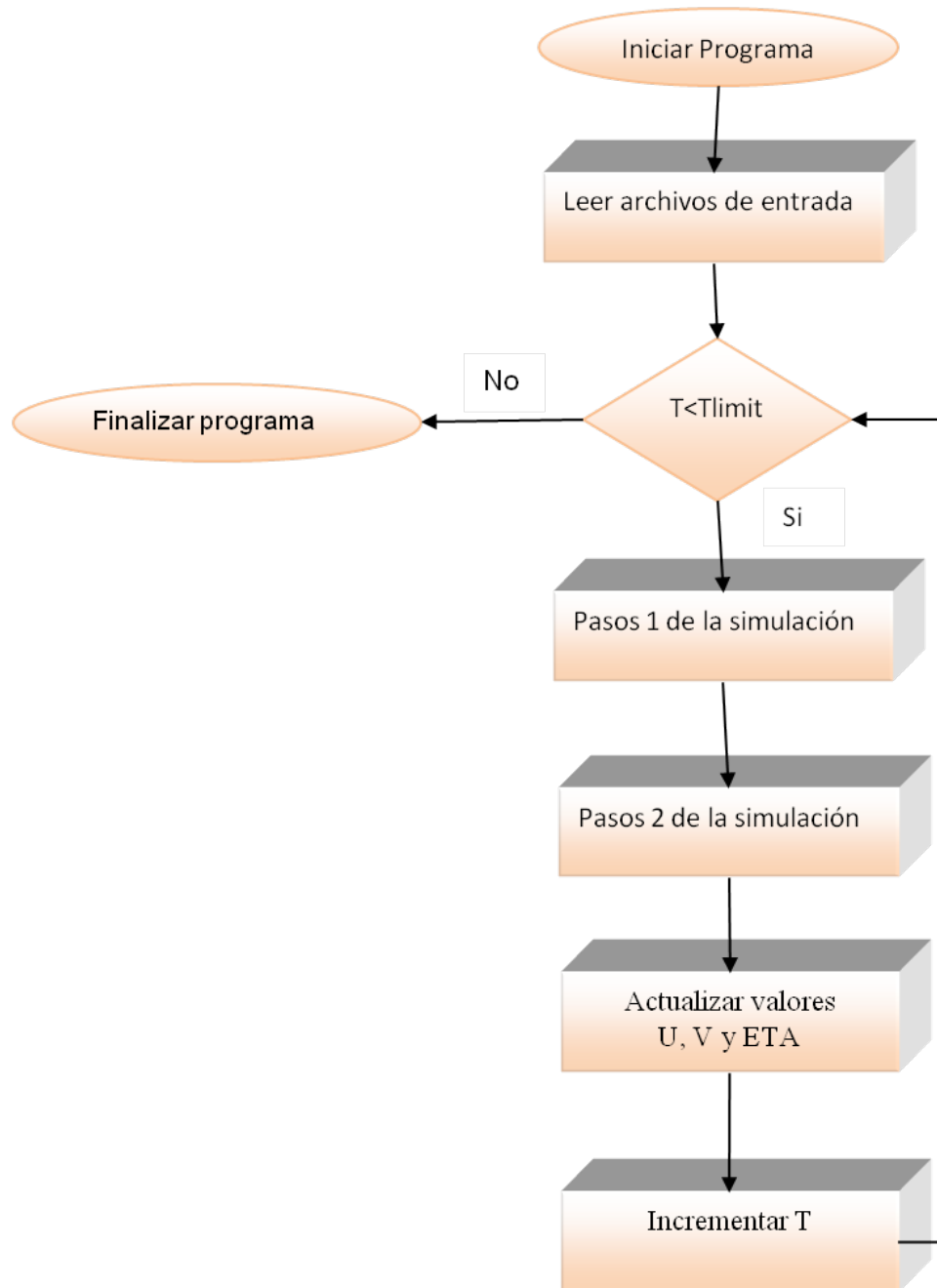


Figura 4.5: Algoritmo Shal2D en secuencial

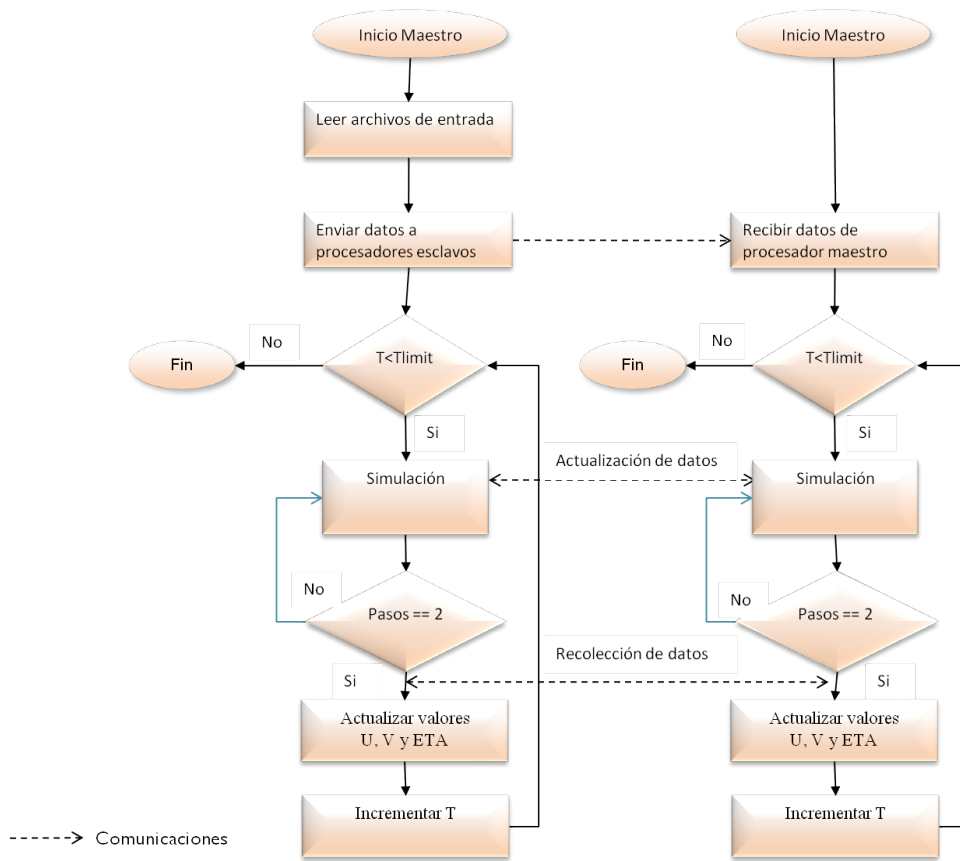
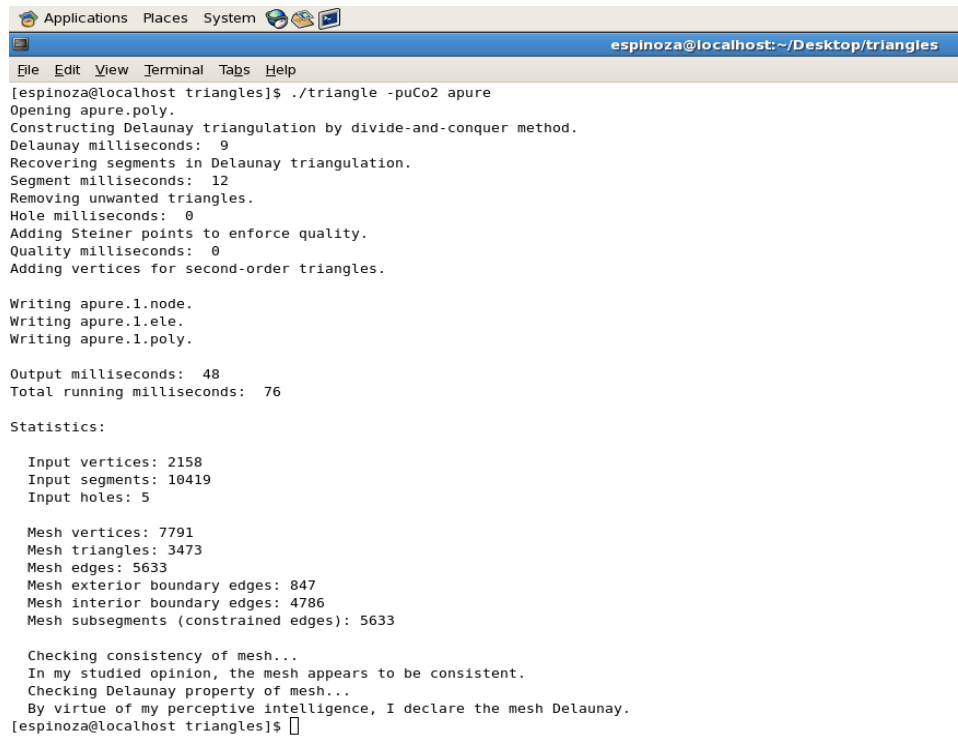


Figura 4.6: Algoritmo Shal2D en paralelo



```
Applications Places System [Icons] [System Tray]
espinoza@localhost:~/Desktop/triangles
File Edit View Terminal Tabs Help
[espinoza@localhost triangles]$ ./triangle -puCo2 apure
Opening apure.poly.
Constructing Delaunay triangulation by divide-and-conquer method.
Delaunay milliseconds: 9
Recovering segments in Delaunay triangulation.
Segment milliseconds: 12
Removing unwanted triangles.
Hole milliseconds: 0
Adding Steiner points to enforce quality.
Quality milliseconds: 0
Adding vertices for second-order triangles.

Writing apure.1.node.
Writing apure.1.ele.
Writing apure.1.poly.

Output milliseconds: 48
Total running milliseconds: 76

Statistics:

Input vertices: 2158
Input segments: 10419
Input holes: 5

Mesh vertices: 7791
Mesh triangles: 3473
Mesh edges: 5633
Mesh exterior boundary edges: 847
Mesh interior boundary edges: 4786
Mesh subsegments (constrained edges): 5633

Checking consistency of mesh...
In my studied opinion, the mesh appears to be consistent.
Checking Delaunay property of mesh...
By virtue of my perceptive intelligence, I declare the mesh Delaunay.
[espinoza@localhost triangles]$
```

Figura 4.7: Salida al ejecutar Triangle con la opción -o2

```
4424     pmlist[vertexnumber - b->firstnumber] = vertexmark(vertexloop);
4425     }
4426 #else /* not TRILIBRARY */
4427 /* Vertex number, x and y coordinates. */
4428     fprintf(outfile, "%4d  %.17g  %.17g", vertexnumber, vertexloop[0],
4429             vertexloop[1]);
4430     for (i = 0; i < m->nextras; i++) {
4431         /* Write an attribute. */
4432         fprintf(outfile, "  %.17g", vertexloop[i + 2]);
4433     }
4434     if (b->nobound) {
4435         fprintf(outfile, "\n");
4436     } else {
4437         /* Write the boundary marker. */
4438         fprintf(outfile, "  %d\n", vertexmark(vertexloop));
4439     }
```

Figura 4.8: Modificación líneas 14428 y 14432, texto original

siguientes archivos;

- Archivo *.ele: Contiene los elementos del tramo de Apure, representa los nodos que genera cada uno de los triángulos.
- Archivo *.node: Contiene las coordenadas X y Y para cada uno de los nodos del tramo.

Se creo el archivo generpoly.f90. Este archivo permite generar el archivo *.poly, el cual contiene los segmentos (unión de dos vertices de un triángulo) y nodos del tramo del caudal, además de los huecos que contiene la malla. Este archivo .f90 se diseñó para un mejor refinamiento de la malla ya que se pueden especificar cuales partes que no pertenecen al río y si se quiere refinar un tramo en específico. Este programa toma como entrada los archivos *.ele y *.node, y como salida el archivo *.poly el cual contiene los segmentos y nodos del tramo apure.

Para el refinamiento se modificó el codigo fuente de Triangle en la línea 14428 y la línea 14432 4.8 por 4.9

Estos cambios fueron necesarios para que el archivo de salida *.1.node tuviera el formato de datos adecuado para la interpolación. Además se cambió la línea 14753 y la línea 14761 4.10 por 4.11.

Estos cambios se realizaron para que el archivo de salida *.1.poly tuviera el formato adecuado para la interpolación.

Luego de estos cambios se compiló Triangle con el compilador gcc en Scientific Linux para utilizar el ejecutable y generar la malla refinada con los archivos apure.1.ele

```

14425     }
14426 #else /* not TRILIBRARY */
14427     /* Vertex number, x and y coordinates. */
14428     fprintf(outfile, "%4d  %8.2f  %8.2f", vertexnumber, vertexloop[0],
14429             vertexloop[1]);
14430     for (i = 0; i < m->nextras; i++) {
14431         /* Write an attribute. */
14432         fprintf(outfile, "  %8.2f", vertexloop[i + 2]);
14433     }
14434     if (b->nobound) {
14435         fprintf(outfile, "\n");
14436     } else {
14437         /* Write the boundary marker. */
14438         fprintf(outfile, "  %d\n", vertexmark(vertexloop));
14439     }
14440 #endif /* not TRILIBRARY */

```

Figura 4.9: Modificación líneas 14428 y 14432, texto nuevo

```

14744     subsegnumber++;
14745 }
14746
14747 #ifndef TRILIBRARY
14748 #ifndef CDT_ONLY
14749     fprintf(outfile, "%d\n", holes);
14750     if (holes > 0) {
14751         for (holenumber = 0; holenumber < holes; holenumber++) {
14752             /* Hole number, x and y coordinates. */
14753             fprintf(outfile, "%4ld  %.17g  %.17g\n", b->firstnumber + holenumber,
14754                 holelist[2 * holenumber], holelist[2 * holenumber + 1]);
14755         }
14756     }
14757     if (regions > 0) {
14758         fprintf(outfile, "%d\n", regions);
14759         for (regionnumber = 0; regionnumber < regions; regionnumber++) {
14760             /* Region number, x and y coordinates, attribute, maximum area. */
14761             fprintf(outfile, "%4ld  %.17g  %.17g  %.17g  %.17g\n",
14762                 b->firstnumber + regionnumber,
14763                 regionlist[4 * regionnumber], regionlist[4 * regionnumber + 1],
14764                 regionlist[4 * regionnumber + 2],
14765                 regionlist[4 * regionnumber + 3]);
14766         }
14767     }
14768 #endif /* not CDT_ONLY */
14769

```

Figura 4.10: Modificación líneas 14753 y 14761, texto original


```

14744     subsegnumber++;
14745   }
14746
14747 #ifndef TRILIBRARY
14748 #ifndef CDT_ONLY
14749   fprintf(outfile, "%d\n", holes);
14750   if (holes > 0) {
14751     for (holenumber = 0; holenumber < holes; holenumber++) {
14752       /* Hole number, x and y coordinates. */
14753       fprintf(outfile, "%4ld  %8.2f  %8.2f\n", b->firstnumber + holenumber,
14754         holelist[2 * holenumber], holelist[2 * holenumber + 1]);
14755     }
14756   }
14757   if (regions > 0) {
14758     fprintf(outfile, "%d\n", regions);
14759     for (regionnumber = 0; regionnumber < regions; regionnumber++) {
14760       /* Region number, x and y coordinates, attribute, maximum area. */
14761       fprintf(outfile, "%4ld  %8.2f  %8.2f  %8.2f  %8.2f\n",
14762         b->firstnumber + regionnumber,
14763         regionlist[4 * regionnumber], regionlist[4 * regionnumber + 1],
14764         regionlist[4 * regionnumber + 2],
14765         regionlist[4 * regionnumber + 3]);
14766     }
14767   }
14768 #endif /* not CDT_ONLY */

```

Figura 4.11: Modificación líneas 14753 y 14761, texto nuevo

con los elementos, apure.1.node con los nodos y apure.1.poly con los nodos y segmentos de la malla resultante.

El paso siguiente está compuesto por la interpolación de la malla para detectar los valores de los nuevos nodos generados previamente. Para esta etapa se creó el archivo inter1.f90 que toma los nodos del archivo viejo apure.node y los datum elevation de los nodos desde el archivo apure.FED para interpolarlos con los triángulos anteriores que están en el apure.ele, luego con esto y junto con el nuevo archivo refinado con los nuevos nodos en apure.1.ele genera el nuevo archivo de entrada del Shal2D apure1.FED con los nuevos nodos y sus respectivos valores de elevación para la malla ya refinada. Ver figura 4.12.

Para la división de la malla se trató de utilizar algoritmos ya existentes como Chaco, Zoltan, Metis, entre otros, el más cercano a la división que se esperaba fue Metis pero realizaba las particiones de la malla de diferentes formas sino sólo vertical y no ordenado, además que el código es muy extenso y engorroso para modificarlo por lo que se procedió a hacer los métodos para las particiones verticales, horizontales y en cuadrícula como fue propuesto previamente.

La versión base de trabajo fue la de Febrero del 2004. Y la primera prueba se ejecutó con la malla original sin modificaciones de 2158 vértices y 3473 triángulos, donde se obtuvieron los resultados mostrados en la tabla 4.1 y la figura 4.13. Donde

Pasos para Refinamiento de la malla

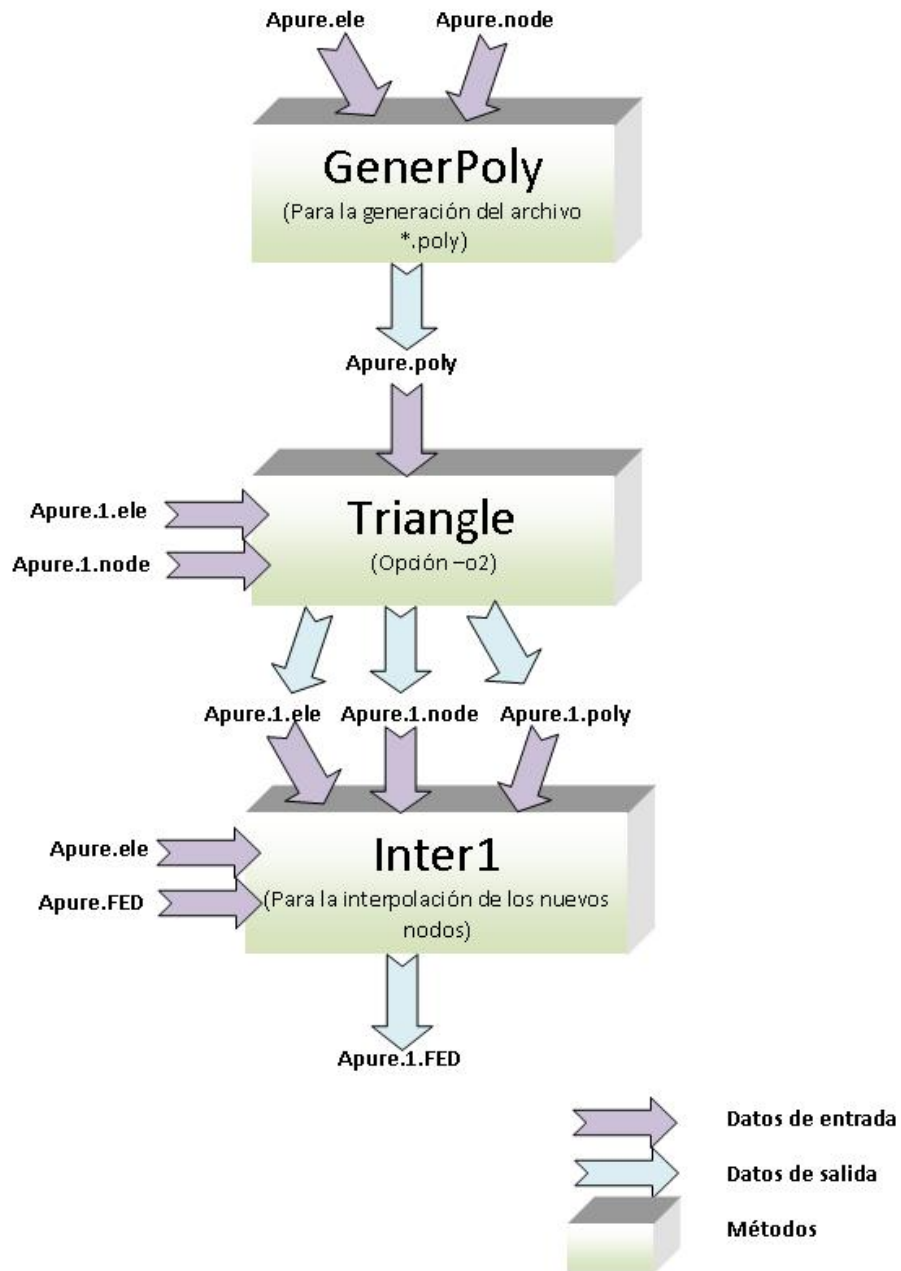


Figura 4.12: Pasos para el refinamiento de la malla

Nro Procesadores	Tiempo de Cómputo
1 (sin partición)	40 minutos
2	27 minutos
4	22 minutos
6	18 minutos

Tabla 4.1: Resultados de partición vertical, 2158 vértices, 3473 triángulos

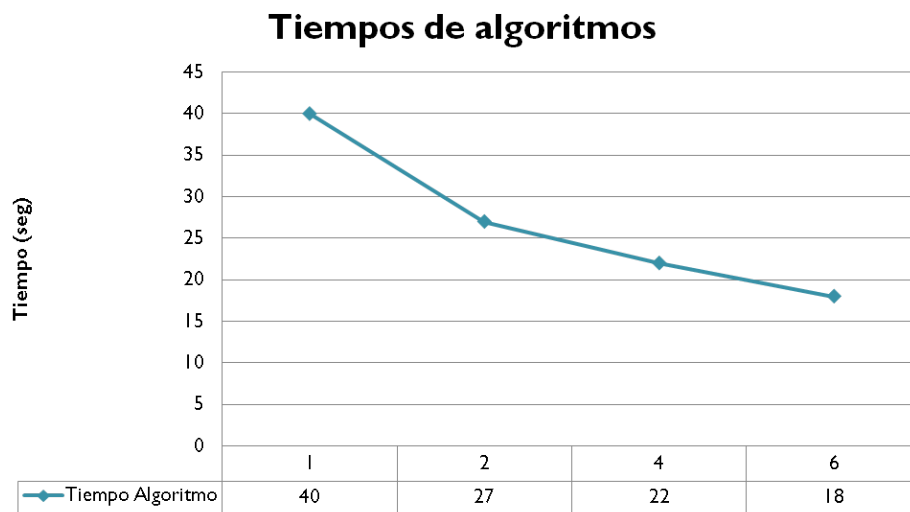


Figura 4.13: Tiempo resultante de malla con 2158 vertices y 3473 triángulos

se muestra una reducción en los tiempos de cómputo de Shal2D en paralelo con 6 procesadores, en comparación con 1, 2 y 4 procesadores. Lo que indica que Shal2D en paralelo reduce los tiempos del algoritmo en secuencial.

En la figura 4.14 se pueden ver los resultados del algoritmo paralelo con particiones estáticas en relación con el tiempo óptimo teórico, y del algoritmo Shal2D secuencial. Este tiempo óptimo teórico es el equivalente al tiempo en secuencial, entre el número de procesadores ver fórmula. Se puede observar que aunque el algoritmo estático no cumple con este último tiempo, si se muestra la mejoría en comparación con el secuencial.

$$T_{\text{OptimoTeorico}} = T_{\text{estatico}} / \text{NroProcesadores}$$

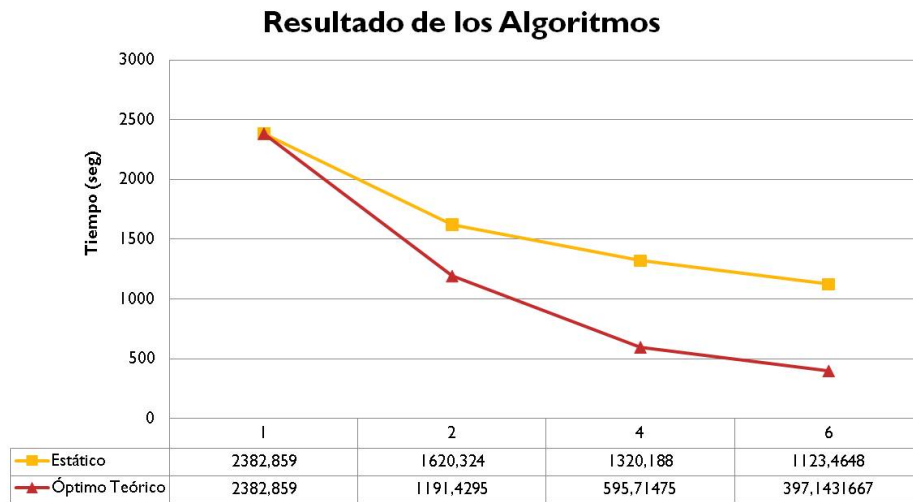


Figura 4.14: Resultados de malla con 2158 vértices y 3473 triángulos

$$\text{Aceleración} = T_{\text{secuencial}} / T_{\text{paralelo}}$$

En la figura 4.15 se puede visualizar como incrementa la aceleración a medida de que aumentamos el número de procesadores. Esto se cumple hasta con seis procesadores, pero hasta cierto punto es válido, ya que en los algoritmos en paralelos a partir de cierto número de elementos la aceleración puede no variar significativamente y hasta disminuir. Con las pruebas realizadas y el número de procesadores utilizados no ocurrió tal situación.

La aceleración es la relación entre el tiempo secuencial y el tiempo en el algoritmo paralelo, ver ecuación, y muestra esa relación entre el aumento o disminución de velocidades de uno con respecto al otro.

La figura 4.16 muestra que en esta prueba el programa en paralelo con dos procesadores es más eficiente que con cuatro y seis, en comparación con el tiempo óptimo teórico para el número de procesadores. Esta eficiencia viene dada por la ecuación siguiente donde indica que la eficiencia es la relación entre el tiempo óptimo teórico y el tiempo del algoritmo en paralelo, con un valor máximo de 1 que ocurre con el algoritmo en secuencial. La eficiencia se puede perder por las comunicaciones entre los procesadores, y es por ello que las gráficas a pesar de tener más procesadores, la eficiencia es menor.

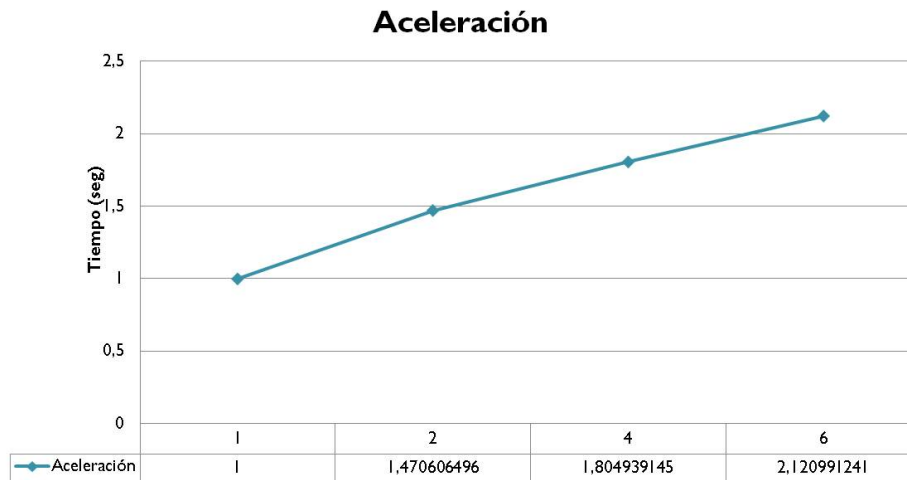


Figura 4.15: Resultados de malla con 2158 vértices y 3473 triángulos

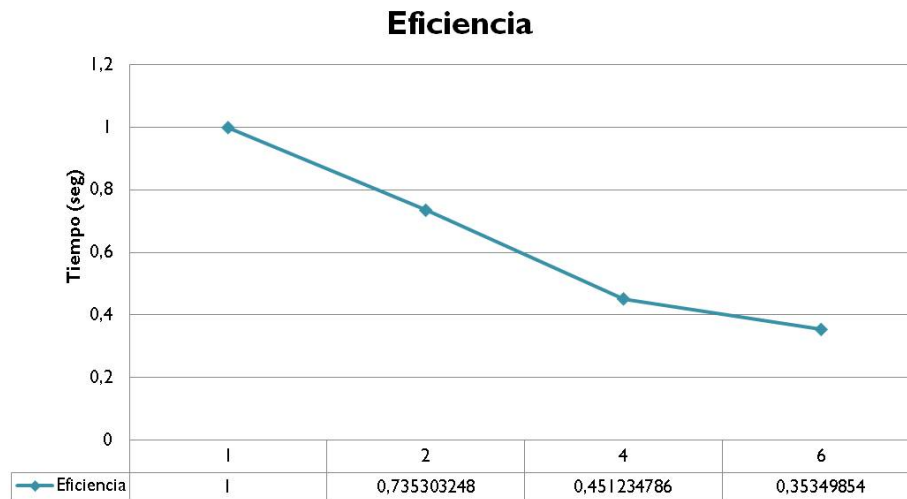


Figura 4.16: Resultados de malla con 2158 vertices y 3473 triángulos

Nro Procesadores	Tiempo de Cómputo
1 (sin partición)	46 minutos
2	28 minutos
6	27 minutos

Tabla 4.2: Resultados de partición vertical, 7791 vértices, 3473 triángulos

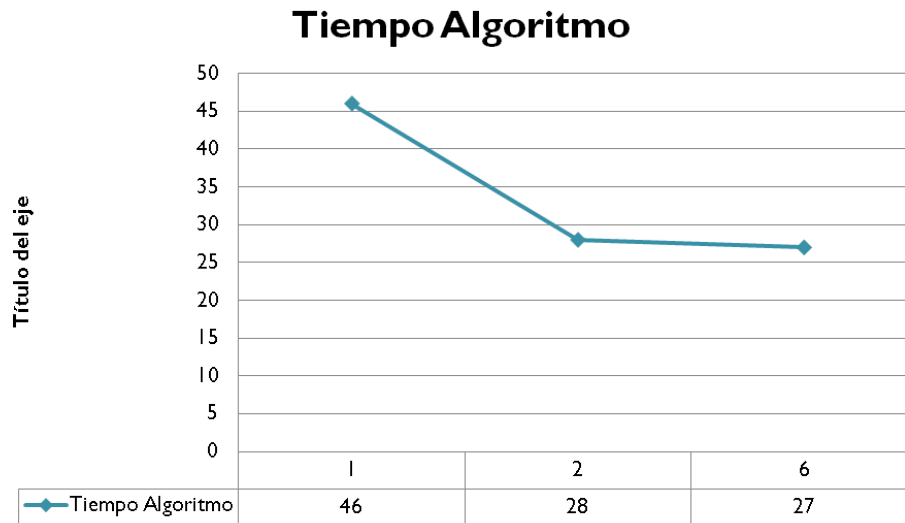


Figura 4.17: Tiempo resultante de malla con 7791 vertices y 3473 triángulos

$$\text{Eficiencia} = \frac{T_{\text{optimo}}}{T_{\text{teorico}}} / T_{\text{paralelo}}$$

Prueba 2: Además de estas pruebas, también se realizaron simulaciones con la malla anterior pero refinada. Esta malla triangular consta de 7791 vértices y 3473 triángulos, obteniéndose los resultados mostrados en la tabla 4.2 y 4.17

se muestra una gráfica comparativa donde se muestra el tiempo total de duración del sistema Shal2D para el algoritmo en secuencial y en paralelo con la partición de datos estática. Se observan que si disminuye el tiempo de cómputo con 2 procesadores, aunque con 6 se no disminuye mucho este tiempo.

En la figura 4.18 se observan esos tiempo en comparación con el tiempo óptimo teórico, y aunque el algoritmo estático no cumple con el tiempo óptimo, si muestra

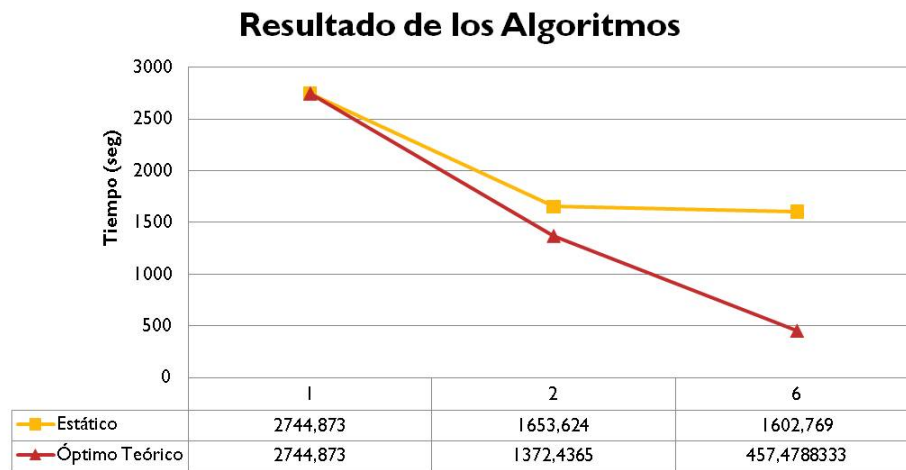


Figura 4.18: Resultados de malla con 7791 vértices y 3473 triángulos

mejoras en comparación con el secuencial, pero también se puede observar que la variación entre dos y seis procesadores no es tan grande. Esto se debe a que las comunicaciones aumentan cuando tenemos más procesadores.

El la figura 4.19 se puede observar cómo se incrementa la aceleración al tener seis procesadores, pero también se puede notar que la variación es muy pequeña, de 0.06seg.

La figura 4.20 muestra que en esta prueba el programa en paralelo con dos procesadores es más eficiente que con seis, en comparación con el tiempo óptimo teórico.

Luego de evaluar los resultados anteriores se puede concluir que el sistema Shal2D en paralelo si permite reducir los tiempos de cómputo y agilizar el proceso en la obtención de resultados, permitiendo realizar más simulaciones en menos tiempo.

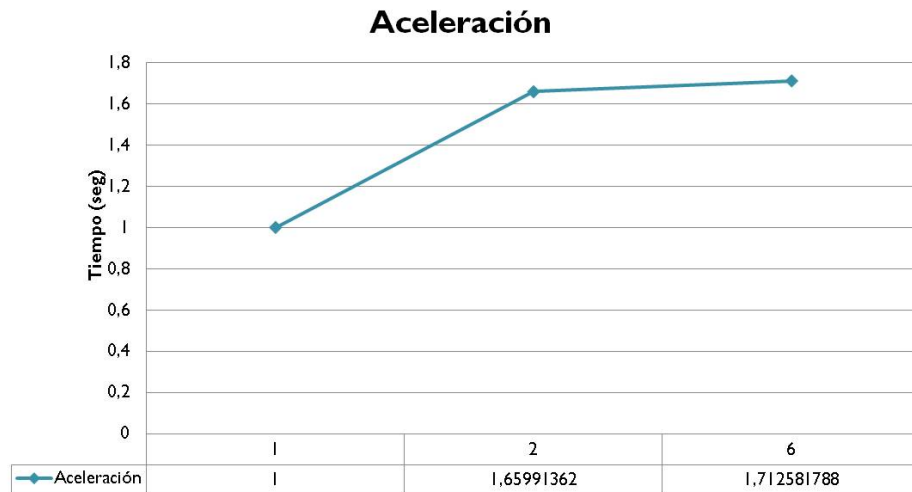


Figura 4.19: Resultados de malla con 7791 vértices y 3473 triángulos

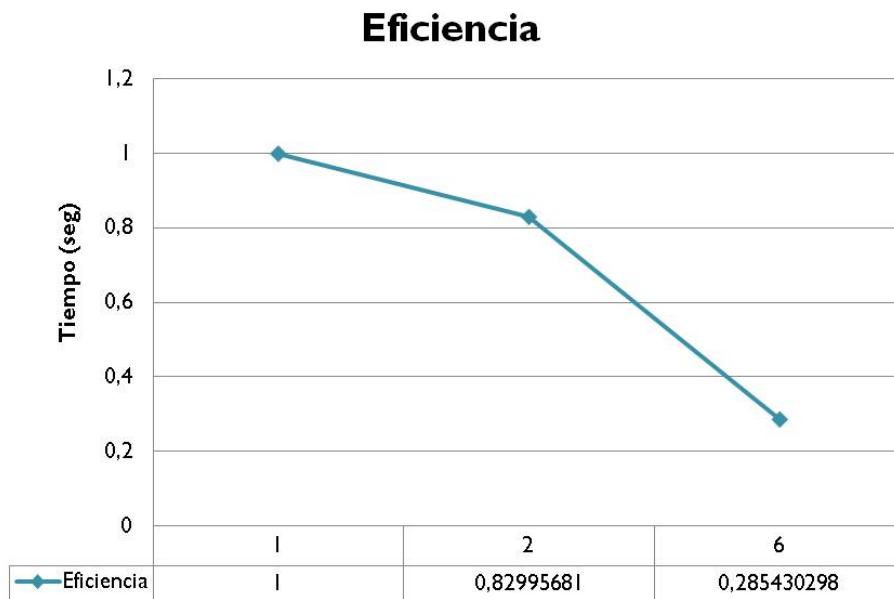


Figura 4.20: Resultados de malla con 7791 vértices y 3473 triángulos

Conclusiones y Trabajos Futuros

Resultados

Se optimizaron los datos de entrada del modelo para la obtención de resultados más precisos.

Se automatizó el proceso de división de la malla, con particiones a disposición del usuario.

Se adaptó la solución paralela al cluster del instituto.

Se observó una reducción en los tiempos de cómputo del modelo shal2D en su versión paralela en comparación con la secuencial.

Conclusiones

El estudio de este Trabajo Especial de Grado permitió conocer más sobre la mecánica de fluidos, el transporte de sedimentos, y como los modelos hidrodinámicos permiten realizar análisis en esos temas, además de observar como los sistemas paralelos pueden proveer herramientas de cómputo necesarias para agilizar el proceso en la obtención de resultados y el manejo de mallas de gran tamaño.

El sistema desarrollado demuestra lo útil y posible de la paralelización de dominios de datos para mallas triangulares en modelos de elementos finitos.

La paralelización del modelo bidimensional Shal2D proporciona al personal de investigación del Instituto de Mecánica de Fluidos de la Universidad Central de Venezuela trabajar con el modelo y realizar pruebas en el cluster del mismo, de forma que la generación de resultados confiables es más rápida que la versión en secuencial.

Tambien es importante resaltar que el refinamiento de la malla permite tener más precisión de datos en la salida del modelo.

Contribuciones

Algunos de los aportes que ofrece este TEG a la Universidad Central de Venezuela se presentan a continuación:

- Tanto el documento como el software presentado sirven como base para trabajos futuros relacionados con la mecánica de fluidos, hidrodinámica, paralelización, entre otras.
- Utilización del cluster como una plataforma de bajo costo y alto rendimiento para el modelado hidrodinámico y transporte de sedimentos con el sistema Shal2D.

Limitaciones

Entre las limitaciones que se hallaron durante el desarrollo de este Trabajo Especial de Grado se encuentran:

- La recopilación de datos de entradas para Shal2D estuvo limitada por los datos que se mantenían en el instituto, además de las salidas que permitieran la comparación con el modelo paralelizado.
- El número de nodos disponibles en el cluster para la realización de las pruebas, ya que no se pudo probar el modelo con un número superior a 7 procesadores.

Trabajos a futuro

Algunas de las mejoras y funcionalidades adicionales que se pudieran implementar en cuanto a la simulación de modelos hidrodinámicos se pudieran incluir:

- Búsqueda de otros sistemas alternativos al triangle para el refinamiento de la malla donde los datos puedan ser manejados a disposición de los científicos, con más libertad en las selecciones de áreas a refinar, y la forma de esta.
- Estudiar la posibilidad de paralelizar por funcionalidad.
- Proveer al modelo de interfaces gráficas para tener un entorno más amigable.

Recomendaciones

Entre las recomendaciones que se pueden hacer referentes a este Trabajo Especial de Grado tenemos:

- Utilizar de forma efectiva y eficiente el cluster del instituto, ya que actualmente en el IMF se tiene una serie de algoritmos secuenciales en donde el tiempo de ejecución es bastante elevado, por lo que sería mejor paralelizarlos y permitir que sean ejecutados en el cluster para disminuir esos tiempos.

 - Utilizar el programa Shal2D en el cluster del instituto y ejecutarlo con datos de entrada de gran tamaño o con compuo intensivo de datos.

 - Promover el uso del sistema en las comunidades científicas.

 - Estudiar el número de pasos de las simulaciones para mejorar la eficiencia del modelo Shal2D.
-

Bibliografía

- [1] Pasos Modelos. Pasos para la construcción de modelos. Recopilado el 02 de Junio de 2010, http://www.investigacion-operaciones.com/Construccion_Modelos.htm
- [2] Mancho, Anna. Mecánica de Fluidos Computacional y Aplicaciones. Madrid: IMDEA matemáticas.
- [3] National Oceanic and Atmospheric Administration (NOAA). How are models hydrodynamics used. Consultado el 28 de Abril de 2010, http://www.nauticalcharts.noaa.gov/csdl/learn_models.html
- [4] Taboada, Raquel. (2007) Modelos de aguas poco profundas obtenidos mediante la técnica de desarrollos asintóticos. Tesis publicada. España: Universidad de Coruña (UDCD).
- [5] Munson, B., Young, D., Okiishi, T. (2002) Fundamentos de mecánica de fluidos. Capítulo 4: Cinemática de Fluidos. México: Editorial Limusa.
- [6] Saavedra, Iván. Modelos de flujo de aguas subterráneas. Método de diferencias finitas y método de elementos finitos. Venezuela.
- [7] Munson, B., Young, D., Okiishi, T. (2002) Fundamentos de mecánica de fluidos. Capítulo 6: Análisis diferencial de los Fluidos. México: Editorial Limusa.
- [8] García, R., Saavedra, I. "Modelaje de corrientes y transporte de sedimentos en la bahía El Tablazo y en el estrecho de Maracaibo". 10mas Jornadas Científico Técnicas de Ingeniería de la Universidad del Zulia (Maracaibo, 26 al 30 de Octubre 1998).
- [9] Val Segura, Rafael. (2005) La medición de sedimentos en México. México: IMTA, UJAT.
- [10] Corporación Autónoma Regional de Valle del Cauca (CVC). (2007) El río Cauca en su valle alto. Colombia: CVC, Universidad del Valle.

-
- [11] Management Unit of the North Sea Mathematical Models (MUMM). (2002-2010) The COHERENS model. Consultado el 28 de Abril de 2010, de <http://www.mumm.ac.be/EN/Models/Coherens/index.php>
- [12] Management Unit of the North Sea Mathematical Models (MUMM). (2002-2010) Hydrodynamic model. Consultado el 28 de Abril de 2010, de <http://www.mumm.ac.be/EN/Models/Operational/hydrodynamics.php>
- [13] EuroMET. (1999). Coordinada Sigma. Consultado el 25 de Abril de 2010, de <http://rammb.cira.colostate.edu/wmovl/VRL/Tutorials/euromet/courses/spanish/nwp/n6300/n6300009.htm>
- [14] Du Bois, Bailly, Dumas Franck. (2004) TRANSMER, un modèle hydrodynamique rapide et précis pour la simulation á moyen et long terme des transferts de radionucléides en Manche et dans le sud de la Mer du Nord. Francia: Jornadas Nacionales de Ingeniería VIII.
- [15] U.S Environmental Protection Agency. (2006) Hydrodynamics Model. Consultado el 29 de Abril de 2010, <http://www.epa.gov/glnpo/lmmb/hydro.html>
- [16] Holtshlag, David., Koschik, John. (2002) A Two-Dimensional Hydrodynamic Model of the St. Clair-Detroit River Waterway in the Great Lakes Basin. U.S.: U.S. Government.
- [17] Passenko, Jelena., Lessin, Gennadi y otros. (2008) Validation of hydrostatic and non-hydrostatic versions of the hydrodynamical model MIKE 3 applied for the Baltic Sea. Estonian Journal of Engineering.
- [18] Patro, S., Singh, R., y otros. (2008) Flood Inundation Modeling using MIKE FLOOD and Remote Sensing Data. India: Indian Society of remote sensing.
- [19] Kofoed, H., Krongborg, P., Cifres, E. Modelling of landslide-generated waves in MIKE 21. España: DHI.
- [20] Tingsanchali, Tawatchai. (2001) Control and mitigation of floods along transbasin diversion channel of Mekong tributaries and Nan river. Thailand: Escuela de Ingeniería Civil, Instituto de tecnología.
- [21] Aziz, Farah., Ole, Mark y otros. Development of flood warning system. Thailand: GISdevelopment.net.
- [22] Zhang, C., Guilbaud, C., Tong, C. 2D Hydrodynamic Modelling for Erhai Lake Applying TELEMAC Modelling System. Edu.cn
-

-
- [23] Telemac. (2009) Telemac-2D. Consultado el 30 de Mayo de 2010, http://www.telemacsystem.com/index.php?option=com_content&view=article&id=17&Itemid=2&lang=en
- [24] Marine Ecosystem Dynamics Modeling (MEDM). (2004) Consultado el 01 de Junio de 2010, <http://fvcom.smast.umassd.edu/FVCOM/index.html>
- [25] Finite Volumen Coastal Ocean Model (FVCOM). (2008) Consultado el 01 de Junio de 2010, <http://www.dgeo.udec.cl/wiki/index.php/FVCOM>
- [26] Ocean Modeling System. ROMS Consultado el 02 de Junio de 2010, <http://www.myroms.org/>
- [27] NTNU Norwegian University of Science and Technology. (2000-2010) SSIIM. Consultado el 01 de Junio de 2010, <http://folk.ntnu.no/nilsol/ssiim/>
- [28] Mantilla, Ricardo. (2008) IIHR- Hydroscience & Engineering. SSIIM. Consultado el 01 de Junio de 2010, http://www.iuhr.uiowa.edu/~ricardo/linux_world_software/linux_world_software-ssiim.htm
- [29] García-Martínez, R., R. Espinoza, E. Valera and M. González. 2006. "An explicit two-dimensional finite element model to simulate short- and long-term bed evolution in alluvial rivers". *Journal of Hydraulic Research*. 44(6):755-766.
- [30] Barney, Blaise., (2010) Introduction to Parallel Computing. Livermore Computing. Consultado el 05 de Junio de 2010, https://computing.llnl.gov/tutorials/parallel_comp/
- [31] Barney, Blaise., (2010) Message Passing Interface (MPI). Livermore Computing. Consultado el 05 de Junio de 2010, <https://computing.llnl.gov/tutorials/mpi/>
- [32] Message Passing Interface (MPI), SP Parallel Programming Workshop. (2002) Maui High Performance Computing Center. Consultado el 05 de Junio de 2010, <http://www.gridbus.org/~raj/csc433/MPITut.pdf>
- [33] OpenMP. Consultado el 04 de Junio de 2010, <http://en.wikipedia.org/wiki/OpenMP>
- [34] Definición ABC. (2009) Cluster. Consultado el 06 de Junio de 2010, <http://www.definicionabc.com/tecnologia/cluster.php>
- [35] Grid Computers. (2004) General Description. Consultado el 06 de Junio de 2010, http://www.thocp.net/hardware/grid_computers.htm
-

- [36] Grid Computing. Consultado el 06 de Junio de 2010, http://en.wikipedia.org/wiki/Grid_computing
- [37] Rodriguez, Freddy. (2003) Paralelización del algoritmo Shal2D. Tesis no publicada. Venezuela
- [38] Artigas, Jacinto., García, Reinaldo. Paralelización de un modelo bidimensional de flujo con superficie libre de fondo móvil basado en elementos finitos. Venezuela: Instituto de Mecánica de Fluidos, Facultad de Ingeniería, Universidad Central de Venezuela.
- [39] García M. (1999) Cap. 6 en Hydraulic Design Handbook Sedimentation and Erosión Hydraulic University of Illinois
- [40] DHM, Consultado el 18 de Noviembre de 2010, del sitio oficial <http://www.diffusionhydrodynamicmodel.com/index2.html>

BIBLIOGRAFÍA NO REFERENCIADA

- [41] Orografía. (2010). Utilidad del concepto. Recopilado el 25 de Abril de 2010, de <http://es.wikipedia.org/wiki/Orograf%C3%ADa>.
- [42] Bergallo, M., Faure, O., Spies, R. (2004) Un método de Galerkin para las ecuaciones de aguas pocos profundas. Argentina: UN, UNT, IMAIL, CONICET.
- [43] Guo, Z. Zhao, T.S., Shi, Y. (2003) Modelo generalizado de flujo de fluidos de nano y macro escala. Universidad de Ciencia y Tecnología de Hong Kong.
- [44] EDF. Telemac. Consultado el 30 de Mayo de 2010, <http://research.edf.com/research-and-the-scientific-community/software/telemac-44341.html>
- [45] AQUAVEO. (2009) SMS:FVCO. Consultado el 30 de Mayo de 2010, <http://www.xmswiki.com/xms/SMS:FVCOM>
- [46] Hidrodinamica. Consultado el 01 de Junio de 2010, <http://es.wikipedia.org/wiki/Hidrodin%C3%Almica>
-

-
- [47] Griborio, A., Meselhe, E., Arceneaux, J. (2006) Model Selection report. University of Louisiana at Lafayette. Consultado el 01 de Junio de 2010, http://sofia.usgs.gov/lox_monitor_model/
- [48] Computación de alto rendimiento. (2010) Consultado el 05 de Junio de 2010, http://es.wikipedia.org/wiki/Computaci%C3%B3n_de_alto_rendimiento
- [49] Paralelismo(Informática). Consultado el 05 de Junio de 2010, [http://es.wikipedia.org/wiki/Paralelismo_\(Inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Paralelismo_(Inform%C3%A1tica))
- [50] OpenMP. (2009) Consultado el 03 de Junio de 2010, <http://openmp.org/wp/>
- [51] Cluster(Informática). Consultado el 04 de Julio de 2010, <http://www.definicionabc.com/tecnologia/cluster.php>
- [52] Computación grid. Consultado el 06 de Junio de 2010, http://es.wikipedia.org/wiki/Computaci%C3%B3n_grid
- [53] Obras fluviales. Consultado el 25 de Febrero de 2011, <http://www.efn.uncor.edu/departamentos/hidraul/Obras>
- [54] Mega PC. (2007) Grid Computer. Consultado el 06 de Junio de 2010, http://mega-pc.blogspot.com/2007/10/grid-computer_6094.html
-