UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DE UNA APLICACIÓN PARA VISUALIZAR SALIDAS DE MODELOS NUMÉRICOS DE CIRCULACIÓN OCEÁNICA

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

MATEOS JASSO, ADRIANA

ASESOR: ZAVALA HIDALGO, JORGE

MÉXICO, D. F. 2006

ÍNDICE

Resumen

Se desarrolló una herramienta para visualizar, mediante animaciones, los datos generados por modelos de circulación oceánica.

En este trabajo se utilizaron las salidas del modelo *Navy Coastal Ocean Model* (NCOM) para visualizar procesos físicos en el Golfo de México. Los datos que se utilizan son de nivel del mar, salinidad, temperatura y las componentes horizontales de la velocidad. Estas variables se grafican sobre una matriz que contiene la batimetría del Golfo de México, representando la corriente mediante flechas con una magnitud proporcional a la velocidad y cuyo desplazamiento en el tiempo corresponde a la distancia real recorrida por las masas de agua. Las flechas curvas son graficadas mediante segmentos de recta los cuales se construyen con una función basada en el método de Runge-Kutta. Las variables del nivel del mar, salinidad y temperatura son representadas en mapa de falso color.

Para la generación de los videos, la aplicación tiene la opción de interpolar en tiempo los datos de los modelos para lograr que los videos tengan una mejor continuidad y no se vean saltos en las secuencias. La aplicación también permite reescalar el tamaño y la densidad de las flechas para lograr una representación óptima. Se buscó que el formato de compresión de los videos generados fuera el más adecuado para que no hubiera pérdida en la calidad de las imágenes y a la vez el tamaño no fuera muy grande y pudiera ser bajado desde Internet.

INTRODUCCIÓN

Esta tesis tiene como objetivo desarrollar una aplicación que permita visualizar, mediante animaciones, los datos de las salidas de los modelos numéricos de circulación oceánica, con el propósito de apoyar la investigación y la divulgación de los procesos físicos.

La visualización es una herramienta para la divulgación del conocimiento científico, que se ha convertido en un área de gran importancia en la computación. Desde un punto de vista técnico, es un proceso demandante de recursos de cómputo (algo propio de la graficación por computadora, y aumentado por el procesamiento matemático de los datos) de manera que la visualización científica, sólo estaba al alcance con grandes equipos de cómputo, equipo especializado o supercomputadoras. Sin embargo al igual que en otras áreas, gracias al avance del software y al abaratamiento del hardware, recientemente se han hecho grandes avances en la visualización científica.

La simulación de los procesos físicos de los océanos y la atmósfera por medio de computadoras, ha sido la rama del conocimiento de mayor demanda de súper cómputo, de uso civil. Los resultados de estos modelos numéricos arrojan volúmenes inmensos de datos, del orden de Gigabytes o Terabytes, que requieren ser analizados. Una de las herramientas más importantes para el estudio de estos resultados es su visualización mediante gráficas, imágenes y animaciones, que permiten estudiar y observar procesos físicos que de otra manera son muy difíciles de identificar.

El presente trabajo consta de 5 capítulos cuyo contenido se resume a continuación.

En el capítulo 1 se da una breve explicación de los conceptos de visualización, de los métodos para visualizar los diferentes tipos de sistemas y variables que existen en

nuestro medio ambiente, así como modelos de objetos en 2D y 3D. También se introduce el concepto de visualización científica, que es el tema de este trabajo.

En el capítulo 2, se describe en forma muy general qué son los modelos de circulación oceánica, y se hace una breve explicación de cómo está diseñado el modelo ***N****avy* ***C****oastal* ***O****cean* ***M****odel* (NCOM), que es el modelo de circulación oceánica utilizado para demostrar la aplicación.

En el capítulo 3, se describen los procesos físicos que se encuentran en el océano, que son los procesos que se simulan en los modelos de circulación oceánica.

En el capítulo 4, se muestra el desarrollo que se siguió para la representación gráfica de las corrientes del Golfo de México, mediante la construcción de las flechas curvas.

En el capítulo 5, se presenta la aplicación que se desarrolló para la representación gráfica de las variables escalares, la generación de las imágenes y de las animaciones. Asimismo se da una explicación del modo de uso de la interfaz gráfica para la realización de estas imágenes.

Finalmente se presentan las conclusiones, se describen los resultados obtenidos en el presente trabajo y se proponen trabajos que se realizarán a futuro dentro del campo de trabajo de la visualización de datos obtenidos a partir de simulaciones de modelos numéricos de circulación oceánica.

CAPÍTULO 1

VISUALIZACIÓN

1.1 Introducción

En los últimos años, los avances en la tecnología de la información han facilitado la obtención de grandes cantidades de información, tanto para la gente común como para los centros de investigación. Sin embargo, estos datos en formato crudo no son útiles, y por eso es necesario que el usuario tenga una forma fácil y eficiente de interpretarlos y analizarlos.

La visualización por computadora se ha convertido en una importante herramienta para analizar estos conjuntos de información y estudiar el comportamiento de ciertos procesos en las diferentes áreas, como análisis comercial, ingeniería, ciencia, etc. De tal modo que las simulaciones numéricas que se efectúan en supercomputadoras, así como la información obtenida de sensores y satélites, acumulan grandes cantidades de datos más rápido de lo que pueden ser interpretados. Se ha visto que el proceso para determinar tendencias y relaciones de grandes conjuntos de números es tedioso e ineficaz. Pero si se convierten los datos a una forma visual, es fácil que se perciban de manera casi inmediata las tendencias y los patrones [*Hearn, 1997*]. El resultado gráfico que se espera de la visualización de datos no es meramente cuantitativo, pues no necesariamente se busca la representación fiel de valores, sino una representación cualitativa, se busca un entendimiento global de determinadas propiedades de los datos.

1.2 Definición

La visualización "*es el mapeo de datos en representaciones que pueden ser percibidas"* [*Foley, 1990*]. La visualización no es un fenómeno nuevo. El hombre ha utilizado estas técnicas desde hace miles de años para entender mejor su medio ambiente. Con el

paso de los años, la computadora se ha integrado a la visualización y actualmente constituye una herramienta fundamental en ella.

La visualización abarca la compresión y síntesis de la imagen. Es decir, la visualización es una herramienta útil para la interpretación de la información [*McComick,* 1987].

Existen muchas clases de conjuntos de datos y los esquemas de visualización efectivos dependen de las características de los datos. Un conjunto de datos contiene cantidades escalares, vectoriales, tensoriales o cualquier combinación de éstas. Los conjuntos de datos pueden ser bidimensionales o tridimensionales. La codificación de colores es una manera de visualizar un conjunto de datos que se complementa con técnicas adicionales que incluyen trazos, gráficas y diagramas de contorno, presentaciones de superficie y visualización de interiores en volúmenes.

1.2.1 Métodos para la visualización del movimiento

1.2.1.1 Sistemas de partículas

Un método para modelar objetos naturales u otros objetos deformables, que presentan propiedades *tipo fluido,* es el de los sistemas de partículas. Este método es de especial utilidad para describir los objetos que cambian de posición con el paso del tiempo.

En este sistema se utiliza un proceso aleatorio para generar objetos en alguna región definida del espacio y variar sus parámetros con el paso del tiempo. En algún momento, al azar, se suprime cada objeto. Durante la vida de una partícula, las características de su trayectoria y superficie pueden tener códigos de colores y desplegarse.

En este sistema la forma de las partículas pueden ser esferas, elipsoides, recuadros pequeños u otras formas. Del mismo modo, otras propiedades como la transparencia,

el color y el movimiento de una partícula pueden variar en forma aleatoria. En algunas aplicaciones, el movimiento de la partícula puede determinarse por medio de fuerzas, por ejemplo un campo de gravedad. Conforme la partícula se mueve, su trayectoria se traza y despliega en un color específico.

1.2.1.2 Modelo con base en las características físicas

Un objeto no rígido se puede representar con métodos de modelado con base en las características físicas que describen el comportamiento del objeto en términos de la interacción de las fuerzas externas e internas que actúan sobre él, por ejemplo, una tela sobre el respaldo de una silla.

Un método para modelar un objeto no rígido, es aproximar el objeto con una red de nodos de punto con conexiones flexibles entre los nodos. Un tipo sencillo de conexión es un resorte. Una red bidimensional de resortes se podría utilizar para aproximar el comportamiento de una lámina de hule. Cuando se aplican fuerzas externas a una red de resortes, la cantidad de estiramiento o compresión de los resortes individuales depende del conjunto de valores para la constante del resorte (k).

Si los objetos son completamente flexibles, regresan a su forma original cuando se dejan de aplicar las fuerzas externas; pero cuando el objeto es deformable, es necesario modificar las características del resorte de modo que los resortes no regresan a su forma original al quitar las fuerzas externas, de tal forma que otro conjunto de fuerzas podría deformar el objeto de alguna otra manera.

También se pueden modelar las conexiones entre nodos con materiales elásticos, con lo cual se reducirían al mínimo las funciones de tensión para determinar la forma del objeto bajo la influencia de fuerzas externas.

Para modelar un objeto no rígido, primero se establecen las fuerzas externas que actúan sobre él. Después se considera la propagación de las fuerzas a través de toda la red que representa el objeto.

Este método también se aplica en animaciones para describir de manera exacta las trayectorias de movimiento. Anteriormente se utilizaban trayectorias de *spline (*curva definida a trozos mediante polinomios*)* y cinemática, donde los parámetros de movimiento se basan sólo en la posición y la velocidad. En este modelo se describe el movimiento utilizando ecuaciones dinámicas, que comprenden las fuerzas y aceleraciones.

1.2.2 Visualización de diferentes conjuntos de datos

Como se mencionó anteriormente, los conjuntos de datos pueden ser muy distintos y se clasifican de acuerdo a su distribución espacial y al tipo de datos. Los conjuntos de datos bidimensionales tienen valores que se distribuyen en una superficie y los conjuntos de datos tridimensionales tienen valores que se distribuyen en el interior de un cubo, una esfera o en alguna otra región del espacio. Los tipos de datos incluyen escalares, vectores, tensores y datos con variables múltiples.

Existen diferentes formas de representar estos tipos de datos y cada una cuenta con una técnica diferente, pero no queda excluida la posibilidad de combinarlas.

1.2.2.1 Representación de datos escalares

Para visualizar los datos escalares el método más común es el de utilizar gráficas o tablas que muestran la distribución de los valores de datos como una función de otros parámetros, como lo son la posición y el tiempo. Cuando encontramos los datos distribuidos en una superficie, se pueden trazar los valores de datos con barras verticales que se elevan de la superficie.

Los métodos de falso-color también se utilizan para distinguir diferentes valores en un conjunto de datos escalares. Las técnicas de codificación de colores se pueden combinar con los métodos de gráficas y tablas. Para codificar con colores un conjunto de datos escalares, elegimos un rango de colores y asociamos el rango de los valores de datos al rango de colores.

Los trazos de contornos se utilizan para desplegar *isolíneas* (líneas de valor escalar constante) de un conjunto de datos que se distribuyen en una superficie. Las isolíneas se separan en algún intervalo conveniente para mostrar el rango y la variación de los valores de datos en la región del espacio. Una aplicación típica es un trazo de contornos para representar las elevaciones de un terreno. Por lo general, los métodos de contornos se aplican a un conjunto de valores de datos que se distribuye en una malla regular.

Por lo general los resultados de las simulaciones numéricas se establecen en una malla regular, pero los conjuntos de datos observados casi siempre tienen una distribución irregular. Los métodos de contornos se diseñaron para diferentes tipos de mallas no regulares, pero en la mayoría de los casos las distribuciones no regulares se convierten a mallas regulares. Un algoritmo de contorno bidimensional, traza las isolíneas de celda a celda adentro de la malla al verificar las cuatro esquinas de las celdas de la malla determinando qué aristas de la celda están cruzadas por una isolínea en particular. Los paquetes de contornos pueden permitir un ajuste interactivo de las isolíneas por parte del investigador para corregir cualquier inconsistencia.

En el caso de los campos de datos escalares tridimensionales, se pueden tomar secciones de corte transversal y desplegar las distribuciones de datos bidimensionales en dichas secciones. También se pueden trazar una o más isosuperficies, que sólo son trazos de contorno tridimensional. Cuando se despliegan dos isosuperficies que se superponen, la superficie exterior se hace transparente, de modo que podemos ver la forma de ambas. La creación de isosuperficies es similar al trazo de isolíneas, con la diferencia que ahora tenemos mallas tridimensionales y se necesita verificar los

valores de ocho esquinas de una celda para localizar las secciones de una isosuperficie.

1.2.2.2 *Representación de datos vectoriales*

Para visualizar un campo vectorial se trazan pequeñas flechas que indican la magnitud y dirección del vector. En campos tridimensionales este método se utiliza con mayor frecuencia en secciones de corte transversal, ya que cuando hay acumulación de datos puede ser difícil ver las tendencias con flechas superpuestas. Las magnitudes para los valores vectoriales se pueden mostrar variando la longitud de las flechas, o podemos hacerlas todas del mismo tamaño pero de distintos colores de acuerdo con el código de color seleccionado para las magnitudes vectoriales.

Las cantidades vectoriales también se pueden representar trazando *líneas de campo* o *líneas de flujo de campo.* La magnitud de los valores vectoriales se indica por medio del espaciado entre las líneas de campo y la dirección es la tangente de las líneas de flujo, las cuales se pueden desplegar con flechas anchas, en particular cuando está presente un remolino o vórtice.

1.2.2.3 Representación de datos tensoriales

La visualización de un campo tensorial de segundo orden, se basa en el diseño de formas que tienen seis parámetros. Los tres elementos diagonales del tensor se utilizan para construir la magnitud y dirección de la flecha y tres términos fuera de la diagonal se emplean para establecer la forma y el color del disco elíptico (Figura 1.1).

En lugar de intentar visualizar los seis componentes de una cantidad tensorial, podemos reducir el tensor a un vector o a un escalar. Al utilizar una representación vectorial, podemos simplemente desplegar una representación vectorial para los elementos diagonales del tensor, y al aplicar las operaciones de contracción del tensor, podemos obtener una representación escalar.

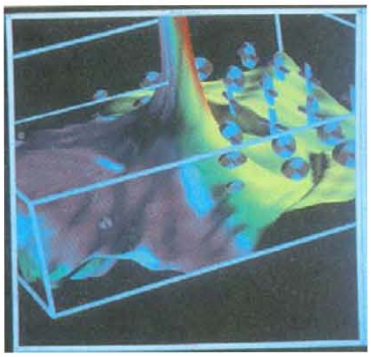


Figura 1.1 Representación de tensores de presión y tensión con un disco elíptico y una flecha sobre la superficie de material que se tensa (*Nacional Center for Supercomputing Applications, University of Illinois at Urbana-Champaign*).

1.2.2.4 Representación de datos con variables múltiples

Para visualizar los campos de datos con variables múltiples es necesario construir los objetos gráficos, que en ocasiones se conocen como *glifos*, con partes múltiples. Cada parte de un *glifo* representa una cantidad física. El tamaño y color de cada parte se puede utilizar para desplegar información acerca de las magnitudes escalares. Para ofrecer información direccional para un campo vectorial podemos utilizar una cuña, un cono o alguna otra forma indicadora para la parte del glifo que representa el vector.

Podemos tener valores múltiples de datos, por ejemplo en algunas aplicaciones en cada posición de una cuadrícula en alguna región del espacio, que puede ser una combinación de valores escalares, vectoriales e inclusive tensoriales (Figura 1.2).

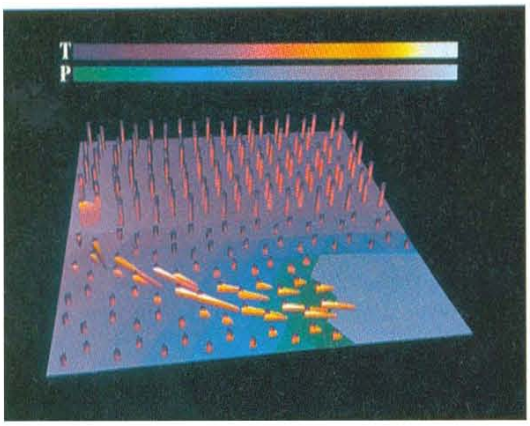


Figura 1.2 Marco de una visualización animada de un campo de datos con variables múltiples utilizando glifos. La parte del glifo con forma de cuña indica la dirección de una cantidad vectorial en cada punto (*Nacional Center for Supercomputing Applications, University of Illinois at Urbana-Champaign*).

1.3 Modelos 2D y 3D

Los trabajos desarrollados en el ***M**assachusetts **I**nstitute of **T**echnology* (MIT) por el equipo dirigido por David Marr durante los años setenta sobre el sistema de visión humano [*Marr, 1982*], marcan uno de los hitos más importantes en el desarrollo de una metodología para abordar soluciones a los complejos problemas que presentan tanto la visión humana como la visión a través de mecanismos artificiales. En su aproximación, Marr estableció una metodología modular de tipo "*bottom/up*" (procesos que necesitan computarizarse conforme vayan apareciendo) sobre el procesamiento de la información subyacente en la imagen percibida. Los tres grandes bloques propuestos por Marr para el procesamiento de la información son:

Nivel 2D: Cálculo del esbozo primitivo como estructura informacional que caracteriza los distintos rasgos básicos (bordes, líneas, arcos, ceros, manchas, etc.) presentes en la imagen de intensidades.

Nivel 2.5D: Construcción de las superficies presentes en la imagen desde el punto de vista del observador, a partir de la información suministrada por el esbozo primitivo y las informaciones tridimensionales proporcionadas por la estereoscopía, sombreado, reflectancia, iluminación, etc.

Nivel 3D: Construcción del modelo que representa a los objetos en el espacio y permite catalogarlos y compararlos con información previamente almacenada.

Estas ideas sobre la descomposición de la información en niveles **2D**, **2.5D** y **3D** así como sus esquemas de representación de la información en cada uno de estos niveles, han marcado una profunda influencia no solo en el campo del estudio de los mecanismos de la visión humana, sino también en el estudio y el análisis de las imágenes digitales.

1.3.1 Modelo 2D

El modelado de gráficos 2D trabaja en el espacio de la imagen, recorriendo por línea de barrido. Por dicha razón se le denomina método *scan-line*. Son los más eficientes en tiempo, aunque la precisión depende del dispositivo gráfico de salida. Además trabajan bien en relación con los algoritmos de iluminación y sombreado.

Para la representación de gráficos en dos dimensiones se tienen lo que se llaman primitivas gráficas, que son:

Puntos: Se especifican a partir de su localización y color. Su discretización es directa.

Segmentos de recta: Son esenciales para la mayor parte de las entidades. Se especifican a partir de un par de puntos que representan sus extremos.

Circunferencias: En algunos casos representar entidades curvadas con segmentos poligonales puede ser inadecuado o costoso, por lo que en la práctica las circunferencias o círculos se adoptan como primitivas. Se especifican con la posición de su centro y su radio.

Polígonos: Son indispensables para representar entidades sólidas. Se representan a partir de la secuencia de puntos que determina la poligonal de su perímetro.

Otro tipo de primitivas gráficas son las primitivas poligonales, las cuales son grupos de polígonos que se describen de forma conjunta para ahorrar espacio de almacenamiento y costo de visualización en tiempo real, razón por la cual son ampliamente utilizadas tanto para la representación en 2D como en 3D.

Algunas de las primitivas poligonales más utilizadas son:

Tira de cuadriláteros: Los primeros cuatro vértices definen el primer cuadrilátero y cada nuevo par de vértices define otro cuadrilátero formado por este par y el anterior. Podemos ver que este tipo de primitiva ahorra, respecto de la especificación de polígonos aislados, casi la mitad de espacio. Sin embargo, tiene el inconveniente de que no se garantiza que en cada cuadrilátero los vértices sean coplanares (Figura 1.3).

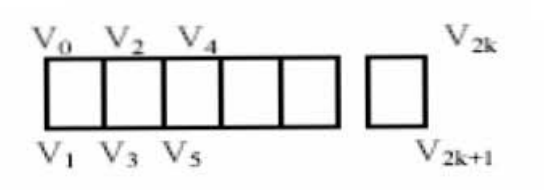


Figura 1.3 Tira de cuadriláteros.

Tira de triángulos: Con los tres primeros puntos se construye un triángulo y los demás se forman añadiendo puntos sucesivos. Cada nuevo triángulo se forma por los tres últimos vértices añadidos, de tal forma que con N puntos se obtienen N-2 triángulos (Figura 1.4).

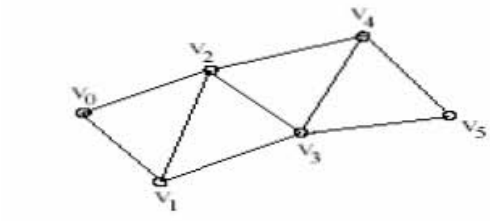


Figura 1.4 Tira de triángulos.

Abanico (Fan): Se da un primer punto y el resto se obtiene siguiendo un abanico. Aparecen N-2 triángulos formados por el primer vértice y los vértices i, i+1 (i diferente de 1). Se emplea para conseguir formas imposibles de lograr con la tira de triángulos. Esta primitiva, al igual que la anterior, tiene la ventaja de que no es necesario comprobar la coplanariedad (Figura 1.5).

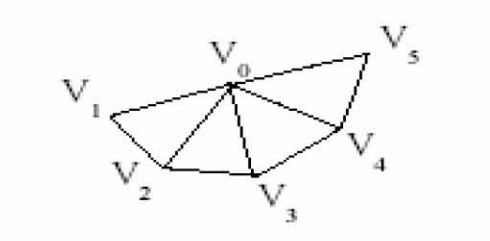


Figura 1.5 Abanico.

Malla rectangular: Se utiliza directamente una matriz de n por m vértices (Figura 1.6).

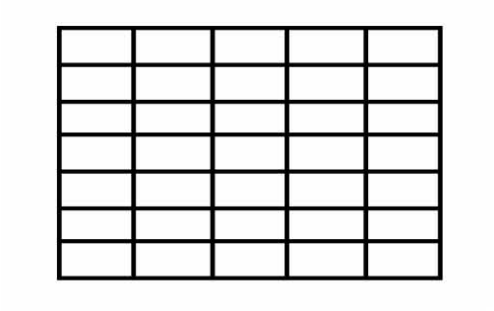


Figura 1.6 Malla rectangular.

1.3.2 Modelo 3D

El término “gráficos 3D por computadora” (3D Computer Graphics) se refiere a trabajos de arte gráfico que fueron creados con ayuda de computadoras y software especial 3D (tridimensional). En general, el término puede referirse también al proceso de crear dichos gráficos o al campo de estudio de técnicas y tecnologías relacionadas con los gráficos 3D.

El arte de los gráficos 3D es similar a la escultura o la fotografía, mientras que el arte de los gráficos 2D es análogo a la pintura. En los programas de gráficos por computadora esta distinción es a veces difusa: algunas aplicaciones 2D utilizan técnicas 3D para alcanzar ciertos efectos como iluminación, mientras que algunas aplicaciones 3D primarias hacen uso de técnicas 2D.

Para formar un gráfico en una computadora existen varias técnicas. El modelado en 3D es dependiente de la aplicación. Sin embargo, en la mayoría de los casos se utilizan estructuras de datos basadas en representaciones poligonales. Por ejemplo, la poliedrización de un objeto puede dar como resultado un arreglo de triángulos.

1.3.2.1 Tecnología

OpenGL y Direct3D son dos **I**nterfaces de **P**rogramación de **A**plicaciones (APIs, **A**pplication **P**rogramming **I**nterface) muy populares para la generación de imágenes 3D en tiempo real. Muchas tarjetas de vídeo modernas proveen cierto grado de aceleración por hardware basado en estas APIs, frecuentemente habilitando el despliegue de complejos gráficos tridimensionales en tiempo real. Sin embargo, no es necesario emplear alguna de estas interfaces para crear imágenes 3D.

1.3.2.2 Creación de gráficos 3D

El proceso de creación de gráficos 3D por computadora puede ser dividido en tres fases básicas:

1.3.2.2.1 Modelado

La etapa de modelado consta de ir dando forma a objetos individuales que luego serán usados en la escena. Existen diversas técnicas de modelado: Constructive Solid Geometry, modelado con NURBS y modelado poligonal son algunos ejemplos. Los procesos de modelado pueden incluir la edición de la superficie del objeto o las propiedades del material (por ejemplo color, luminosidad, difusión, especularidad, características de reflexión, transparencia u opacidad, o el índice de refracción), agregar texturas, mapas de relieve (bump-maps) y otras características.

El proceso de modelado puede incluir algunas actividades relacionadas con la preparación del modelo 3D para su posterior animación. A los objetos se les puede

asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. Esto ayuda al proceso de animación, en el cual el movimiento del esqueleto automáticamente afectará las porciones correspondientes del modelo.

El modelado puede ser realizado por programas dedicados (como Lightwave 3D, Rhinoceros 3D o Moray), un componente de una aplicación (Shaper, Lofter en 3D Studio) o por un lenguaje de descripción de escenas (como en POV-Ray). En algunos casos, no hay una distinción estricta entre estas fases; en dichos casos, el modelado es sólo una parte del proceso de creación de escenas (por ejemplo, con Caligari trueSpace).

1.3.2.2.2 Composición de la escena

Esta etapa involucra la distribución de objetos, luces, cámaras y otras entidades en una escena que será utilizada para producir una imagen estática o una animación. Si se utiliza para animación, en esta fase, en general, se hace uso de una técnica llamada *Keyframing*, que facilita la creación de movimientos complicados en la escena. Con la ayuda de la técnica de keyframing, en lugar de tener que corregir la posición de un objeto, su rotación o tamaño en cada cuadro de la animación, solo se necesita marcar algunos cuadros clave (keyframes). Los cuadros entre keyframes son generados automáticamente, lo que se conoce como 'Interpolación'.

La iluminación es un aspecto importante de la composición de la escena. Como en la realidad, la iluminación contribuye al resultado estético y a la calidad visual del trabajo terminado. Por eso, puede ser un arte difícil de dominar. Los efectos de iluminación pueden contribuir en gran medida al humor y la respuesta emocional generados por la escena.

1.3.2.2.3 Generación de la escena

Se llama rénder al proceso final de generar la imagen 2D o animación a partir de la escena creada. Esto puede ser comparado con tomar una foto o filmar la escena en la vida real después que se terminó de armar. Generalmente se buscan imágenes de calidad foto realista, y para este fin se han desarrollado muchos métodos especiales. Las técnicas van desde el rénder de alambre (wireframe rendering), pasando por el rénder basado en polígonos, hasta las técnicas más modernas como Scanline Rendering, Raytracing o radiosidad.

El software de rénder puede simular efectos cinematográficos como *lens flare*, profundidad de campo, o *motion blur* (desenfoque de movimiento). Estos artefactos son, en realidad, un producto de las imperfecciones mecánicas de la fotografía física, pero como el ojo humano está acostumbrado a su presencia, la simulación de dichos efectos aporta un elemento de realismo a la escena. Se han desarrollado técnicas con el propósito de simular otros efectos de origen natural, como la interacción de la luz con la atmósfera o el humo. Ejemplos de estas técnicas incluyen sistemas de partículas que pueden simular lluvia, humo o fuego, muestreo volumétrico para simular niebla, polvo y otros efectos atmosféricos, y cáusticas para simular el efecto de la luz al atravesar superficies refractantes.

El proceso de rénder necesita una gran capacidad de cálculo, pues requiere simular gran cantidad de procesos físicos complejos. La capacidad de cálculo se ha incrementado rápidamente a través de los años, permitiendo un grado superior de realismo en los rénders. Estudios de cine que producen animaciones generadas por una computadora hacen uso, en general, de lo que se conoce como *rénder farm* (granja de rénder) para generar imágenes de manera más rápida.

1.4 Visualización científica

La manera de visualizar los datos no está regida por técnicas o recetas, sino principalmente por la abstracción que el científico tiene de sus datos, estableciéndose como un proceso analítico, creativo y de exploración.

El científico debe determinar los parámetros para ver sus datos: dimensiones, espacios, tiempos, colores, rangos, escalamientos, etc. Los elementos básicos con que cuenta son: puntos, líneas, superficies y colores, y algunos más complejos, particulares de cada visualización. Existen algunas técnicas y nomenclaturas propuestas para la visualización científica, sin embargo, nada impide al científico ver las cosas de la manera en que las quiere ver.

Desde el punto de vista técnico la visualización científica es un proceso demandante de recursos de cómputo (algo propio de la graficación por computadora, y aumentado por el procesamiento matemático de los datos), de manera que la visualización científica *flexible* solo está al alcance cuando se usan grandes equipos de cómputo, equipo de cómputo especializado o supercomputadoras.

La visualización científica se ha enriquecido enormemente con los avances logrados por la graficación por computadora y las técnicas de interacción, tales como: animación, dispositivos tridimensionales, *rendering* y actualmente la Realidad Virtual. No se puede dejar de reconocer el impulso proveniente de las aplicaciones comerciales (principalmente el cine) y militares.

1.4.1 Conceptos de visualización científica

a) Es una rama de la Graficación por Computadora y su objetivo es mostrar datos científicos mediante gráficas generadas por computadora, aprovechando la versatilidad y poderío de cálculo de las computadoras [*Gallagher, 1995*].

b) Es un proceso computacional que transforma datos intangibles de una computadora en imágenes sobre un dispositivo de despliegue; estos datos regularmente provienen de la observación, la experimentación o la simulación.

c) Es la generación de imágenes a partir de datos, para transformarlos en información y mejorar el entendimiento así como para comunicarlo a otros.

d) Es considerada una herramienta para la investigación, útil y creativa, que permite a los científicos estudiar y observar fenómenos que a simple vista no sería posible.

La visualización científica se ha convertido en un área de gran importancia en la computación. Al igual que en otras áreas, gracias al avance del software y al abaratamiento del hardware, se han hecho grandes avances en visualización. Fue en 1987, con la edición especial de *Computer Graphics* en visualización científica, cuando comenzó a crecer el interés por este tema. Desde entonces, tanto *The **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers, INC* (IEEE) como la ***A**ssociation for **C**omputing **M**achinery* (ACM SIOGRAPH) han organizado conferencias y grupos de trabajo dedicados tanto al tema general de visualización científica como a áreas específicas, como la visualización de volúmenes.

Según R.A. Earnshaw [1992], la meta de la visualización científica es promover un nivel más profundo de los datos que se están investigando y proporcionar mayor profundidad a los procesos, confiando en el sistema visual humano y en su habilidad para interpretar datos visuales. Las herramientas y técnicas de visualización han sido utilizadas para analizar grandes volúmenes de datos multidimensionales de forma que permitan al usuario extraer resultados significativos de forma rápida y fácil.

Un ejemplo de visualización científica es el estudio del clima. El *Scientific Visualization Studio* del ***N**ational **A**eronautics and **S**pace **A**dministration* (NASA) en el *Goddard Space Flight Center* trabaja ayudando a los científicos para que su trabajo sea

accesible para todo el mundo. En este centro se producen aproximadamente cuarenta horas de video al año.

Algunos proyectos de este centro son los satélites SeaWiFS y el ***T****ropical* ***R****ainfall* ***M****easuring* ***M****ision* (TRMM). El satélite TRMM es una colaboración entre la NASA y la ***N****ational* ***S****pace* ***D****evelopment* ***A****gency* (NASDA) de Japón, diseñado para monitorear y estudiar las tormentas tropicales y los flujos de energía asociados que generan la circulación atmosférica global, dando forma al clima y a la distribución de temperatura.

Un video clip de cinco segundos es obtenido de un video de cinco horas producido por 500 horas de trabajo con datos. El estudio de visualización científica utiliza muchas computadoras con múltiples procesadores por las enormes cantidades de datos involucradas en la creación de las imágenes.

La visualización científica tiene dos áreas específicas:

1.4.2 Visualización de volúmenes

La visualización de volúmenes se refiere generalmente a campos escalares. Se extiende desde el examen de datos científicos, la reconstrucción de datos dispersos y la representación de objetos geométricos sin la descripción matemática de su superficie [*Gallagher, 1995*].

1.4.3 Visualización de flujos

Se utiliza para la visualización en general de sistemas dinámicos, es decir, aquellos sistemas en los que están involucradas variables que evolucionan en el tiempo. El comportamiento cualitativo de dichos sistemas puede comprenderse adecuadamente a partir de la estructura de la evolución temporal de sus trayectorias. Estos sistemas contienen implícitamente una gran cantidad de datos que no es directa, ni fácilmente, observable [*Gallagher, 1995*].

1.5 Sistemas de visualización

Hay tres partes importantes en un sistema de visualización:

1- La construcción de un modelo empírico de los datos: Este modelo puede tener consideraciones sobre la teoría del muestreo, como el *teorema de Nyquist* [A5] y esquemas de interpolación matemática. También debe tomarse en cuenta la posibilidad de que haya errores en los datos.

2- La selección de esquemas: Significa tomar como modelo un objeto de visualización abstracta (un mapa, por ejemplo).

3- La representación de la imagen en un ambiente gráfico.

El propósito de la visualización es analizar, comprender y comunicar la *información* que viene contenida en datos. A lo largo de los años han sido desarrollados un enorme número de formatos de gráficos e imágenes por computadora, por lo que es importante mencionar que no existe ningún paquete o herramienta única que se use para visualizar y pueda hacer todo lo que se requiere.

Por lo anterior es importante contar con programas de visualización que sean flexibles, compatibles con otras aplicaciones y se puedan adaptar al constante cambio de tecnologías, tanto de software como de hardware. Estos programas deben ser capaces de proveer métodos que acepten una gran variedad de tipos de datos y formatos de archivos así como de servicios que contengan convertidores eficientes. Muchas de las técnicas de visualización y sus procesos utilizan varios pasos en forma simultánea, por lo que la automatización de la conversión y lectura transparente de datos es más que deseable.

En el presente trabajo se utiliza el paquete MATLAB versión 6.5 que es un programa que realiza cálculos numéricos con vectores y matrices. Como caso particular, puede

trabajar con números escalares, tanto reales como complejos, con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones.

MATLAB es un gran programa de cálculo técnico y científico. Para ciertas operaciones es muy rápido, cuando puede ejecutar sus funciones en código nativo con los tamaños más adecuados para aprovechar sus capacidades de vectorización. En otras aplicaciones resulta bastante más lento que el código equivalente desarrollado en C/C++ o Fortran. En la versión 6.5, MATLAB ha incorporado un acelerador JIT (*Just In Time*) que mejora significativamente la velocidad de ejecución de los ficheros *.m en ciertas circunstancias, por ejemplo cuando no se hacen llamadas a otros ficheros *.m, no se utilizan estructuras, clases, etc.

Además de contar con la característica de que los ficheros desarrollados en la aplicación y ejecutados sobre un sistema operativo Windows, también pueden ser ejecutados sobre un sistema operativo distinto como Unix o Linux.

Se decidió desarrollar la aplicación en MATLAB porque puede ser modificada o mejorada por los propios investigadores, ya que MATLAB es un paquete manejado por la comunidad oceanográfica y muchos científicos de otras áreas en el mundo.

CAPÍTULO 2

PROCESOS FÍSICOS DEL OCÉANO

2.1 Introducción

Los océanos intercambian momento, calor, agua (a través de la evaporación y la precipitación) y CO_2 con la atmósfera. Debido a su enorme masa y su gran capacidad de almacenamiento de calor, los océanos tornan más lento el cambio climático e influyen en la escala temporal de la variabilidad del sistema océano-atmósfera. Se han hecho progresos considerables en la comprensión de los procesos oceánicos que guardan relación con el cambio climático. Se han logrado importantes avances en la modelación de los procesos oceánicos, en particular del transporte de calor y procesos de mesoescala. Estos avances, unidos a un aumento en la resolución espacial, han sido importantes para producir simulaciones realistas. Asimismo el aumento de la resolución, así como una mejor representación (parametrización) de procesos importantes a escala subreticular (por ejemplo vórtices de mesoescala), han aumentado el realismo de las simulaciones. Sin embargo sigue habiendo grandes incertidumbres en torno a la representación de los procesos de pequeña escala, como los flujos por canales estrechos (por ejemplo entre Groenlandia e Islandia), las corrientes occidentales de frontera (es decir, corrientes angostas de gran escala a lo largo de la línea del talud de la plataforma continental) y los fenómenos de convección y mezcla.

La circulación oceánica superficial es el resultado de varios procesos de interacción con la atmósfera (*Figura 2.1*), especialmente del esfuerzo del viento que actúa sobre la superficie del agua, y de las diferencias de densidad. Si se asume que el sistema de corrientes observado es simplemente el resultado del esfuerzo del viento, la circulación sería muy similar a los principales cinturones de vientos en la Tierra y, efectivamente, así ocurre. Sin embargo, en el Hemisferio Norte los vientos se desvían hacia la derecha y hacia la izquierda en el Hemisferio Sur. Esto se debe a la rotación de la Tierra y se explica por el efecto de Coriolis [A8]. En la categoría de corrientes marinas

superficiales se incluyen las corrientes permanentes de los océanos como la corriente de Humboldt, la del Golfo y las corrientes Ecuatoriales, las cuales forman parte de la circulación general de los océanos.

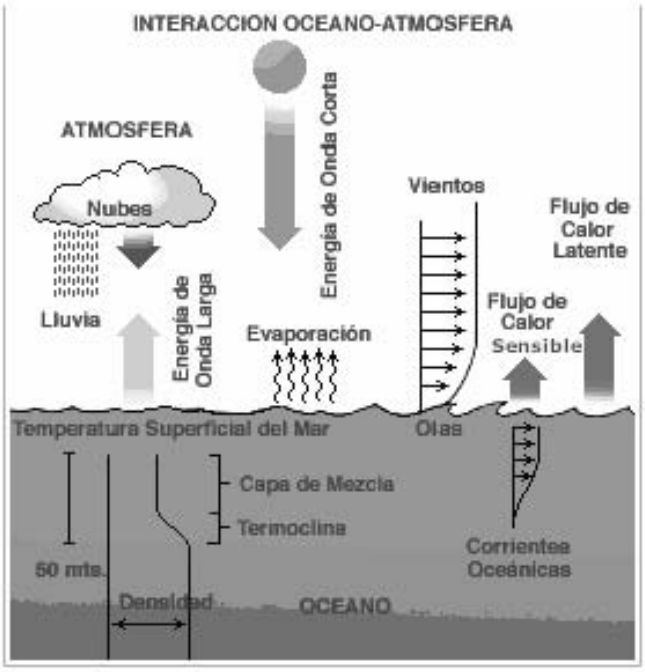


Figura 2.1 Esquema de los procesos de interacción entre el océano y la atmósfera.

2.2 Forzamiento y corrientes marinas

Las principales causas de las corrientes marinas son:

- Vientos permanentes soplando sobre la superficie del mar (que producen fricción y arrastre de las moléculas superficiales del agua).
- Diferencias de densidad entre masas de agua.
- Influencia de la disposición de los continentes y los litorales.

El oceanógrafo Munk propone una explicación general para la formación de las corrientes a partir de la consideración de un océano teórico. Considera 3 casos de complejidad creciente.

- **Caso 1**: Un océano rectangular, cuyo eje principal es Norte-Sur, con límites continentales al Oeste y Este. El océano está rotando y sobre él soplan vientos zonales uniformes del Oeste, con una dirección e intensidad constante (Figura 2.2).

En este caso hay una tendencia a la acumulación de agua en el lado Este del océano, con una pendiente uniformemente equilibrada por el empuje constante del viento.

En el plano vertical se establece una circulación con convergencia en lado Este y hundimiento del agua hasta una profundidad que es función de la fuerza del viento y de la distribución vertical de la densidad.

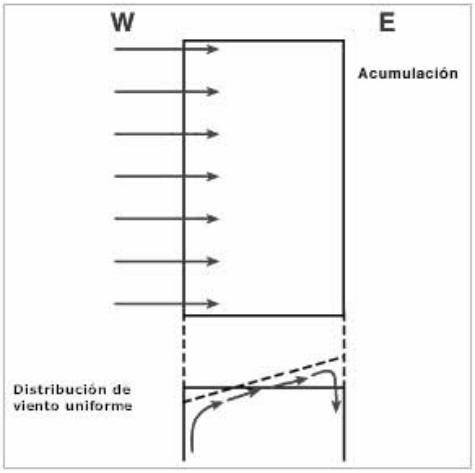


Figura 2.2 Respuesta de la superficie del océano a un viento uniforme.

- **Caso 2**: Si en este mismo océano los vientos tienen una dirección constante, como el caso anterior, pero hay una variación de la intensidad con la latitud, habrá asimetría de las fuerzas de arrastre debidas al viento (Figura 2.3).

En este caso la pendiente será más abrupta hacia el Sur que hacia el Norte, generando así en el Este un movimiento horizontal llevando agua de la zona de fuertes vientos a la de vientos débiles. Por la existencia de límites continentales y por la continuidad de los vientos, el movimiento será giratatorio.

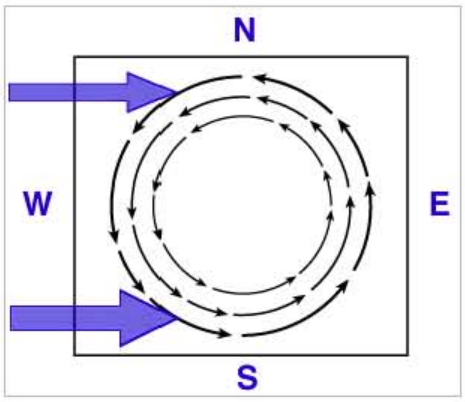


Figura 2.3 Circulación resultante con un viento con la misma dirección pero distinta magnitud.

- **Caso 3**: Si los vientos no sólo tienen variaciones latitudinales de intensidad sino también de dirección, se generará, bajo la influencia de estas fuerzas diversas y en presencia de los límites Este y Oeste de los océanos, una circulación activa de células horizontales con movimientos giratorios. Este caso es muy próximo a la realidad (Figura 2.4).

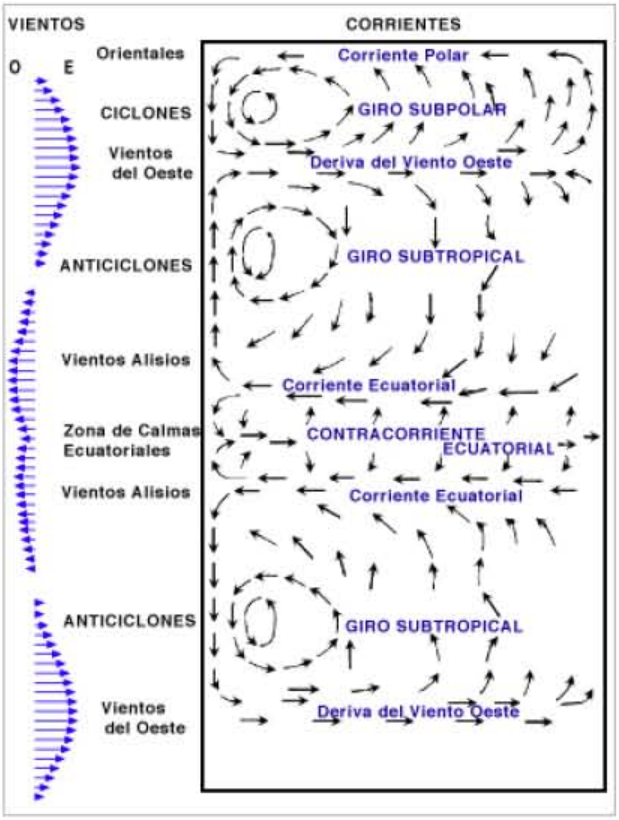


Figura 2.4 Diagrama de la circulación en un océano idealizado sometido sólo a la acción de los vientos.

De esta forma los vientos alisios y los del Oeste determinan en cada hemisferio la circulación en giros del agua superficial del océano. Cada uno de los circuitos tiene una rama ecuatorial en dirección Este-Oeste, y una rama sub-polar en dirección Oeste-Este; las dos están unidas por circulaciones con una componente meridional más o menos paralela a los continentes.

Entre las dos corrientes ecuatoriales (separadas por una zona de calmas), se forma una contracorriente ecuatorial de dirección inversa (con una pendiente de 4 cm por cada 1000 Km). Es una corriente de retorno que lleva, a través de esta zona de calmas, una parte de las aguas acumuladas en el lado Oeste por los alisios.

En el Hemisferio Norte, entre los vientos del Oeste y la circulación polar establecida por vientos del Norte y Noreste relacionados con el anticiclón polar, aparece una circulación ciclónica sub-polar. Debido a la rotación de la Tierra hacia el Este y también por la variación de la intensidad del efecto de Coriolis con la latitud, el centro de los giros está desplazado hacia el lado Oeste y las corrientes son más fuertes en el lado Oeste de los océanos que en el lado Este. De esta manera, en cada giro hay una corriente fuerte y persistente en el lado Oeste y una corriente de compensación en el sector central y Este.

2.3 Transportes en los océanos

Las corrientes marinas transportan masas de agua de enorme volumen. La unidad de medida del flujo es el Sverdrup, en donde 1 Sv=10^6 [m^3/s] (indica el flujo de 1 millón de [m^3/s]).

Las corrientes limítrofes fluyen en los bordes continentales transportando aguas cálidas hacia los polos en el lado occidental de los océanos y aguas frías hacia el ecuador.

El flujo de las corrientes marinas se debe a los siguientes mecanismos:

- Vientos superficiales que por fricción mueven el agua de la superficie oceánica.

- La forma de las cuencas oceánicas y los continentes que las rodean rigen las corrientes en un movimiento circular.

- La fuerza de Coriolis actúa desviando las corrientes y el viento.

- Existen diferencias de altura en la superficie del mar.

2.4 Altura en la superficie del mar

La superficie del mar no es plana, tiene diferencias de altura de uno a dos metros, que se reflejan en las corrientes superficiales. Esto está ligado al concepto de topografía dinámica.

El agua que se acumula en el centro de los giros de las corrientes produce elevaciones que pueden sobrepasar el metro. Estos apilamientos afectan los primeros cien metros de la columna de agua y reposan sobre las capas de agua más profundas y frías. Las diferencias de altura de la superficie del mar también dependen de las variaciones de la densidad del agua, así el agua más cálida y poco salina es menos densa y la superficie más elevada. La elevación del agua es más importante en el Pacífico Norte que en el Atlántico Norte.

Por otra parte, si el océano fuera homogéneo y se encontrara en reposo, el nivel del mar estaría sometido a una fuerza de gravedad constante. La superficie donde la fuerza de gravedad es de intensidad constante se conoce como geoide. Esta superficie es una esfera aplanada en los polos. Las anomalías del geoide son abultamientos y depresiones que constituyen la topografía dinámica. Así, la diferencia entre la superficie oceánica real y un geoide es la topografía dinámica.

La importancia de lo anterior en la circulación oceánica consiste en que los abultamientos de agua crean fuerzas horizontales (gradientes de presión) en la

dirección de la pendiente del agua, que tiende a moverse de los lugares más altos a los más bajos. Sin embargo, como ya se señaló, el efecto de Coriolis desvía el sentido del flujo a la izquierda en el Hemisferio Sur y a la derecha en el Hemisferio Norte.

Los modelos de circulación oceánica representan y consideran las fuerzas y procesos físicos descritos arriba.

CAPÍTULO 3

MODELOS DE CIRCULACIÓN OCEÁNICA

3.1 Introducción

Los modelos de circulación oceánica permiten efectuar pronósticos de las condiciones oceanográficas. Estos modelos, conocidos como modelos numéricos, utilizan complejos programas de cómputo que, por lo general se ejecutan en supercomputadoras.

Un modelo numérico es un conjunto de ecuaciones matemáticas cuya solución requiere de métodos numéricos. Los métodos numéricos más comúnmente usados para resolver el sistema de ecuaciones diferenciales en derivadas parciales son: diferencias finitas, métodos espectrales y elementos finitos.

3.2 Clasificación de los Modelos Oceánicos

En las dos últimas décadas han surgido diferentes criterios para modelar el océano, por lo cual la comunidad oceanográfica ha decidido clasificar los modelos tomando en cuenta los siguientes criterios (Figura 3.1):

- El geográfico, que depende del tamaño y las características particulares de la cuenca oceánica que se estudie, como el Atlántico, las Cuencas del Pacífico, el Mediterráneo, el Golfo de México, etc.

- Los diferentes procesos físicos. Los modelos pueden ser hidrodinámicos y/o termodinámicos.

- La condición de la superficie del océano.

- El número de grados de libertad vertical que maneja (inclusive la descomposición modal en la vertical).

- Las variaciones de la densidad.

Figura 3.1 Criterios para la clasificación de los modelos oceánicos.

3.2.1 Inicio de la modelación numérica

En 1950, el estadounidense Jule Charney (1917-1981), el noruego Ragnar Fjörtoft y Von Neumann realizaron la primera predicción numérica del tiempo. Para ello consideraron que lo mejor era utilizar un modelo simplificado que tuviera validez meteorológica, y eligieron el modelo barotrópico propuesto por Rossby. Los cálculos numéricos los realizaron en el ENIAC (Electronic Numerical Integrator and Computer) instalado en Aberdeen (Maryland, USA). Hay que resaltar que necesitaron 33 días con sus noches para programar y ejecutar tres predicciones para un plazo de 24 horas. Los resultados obtenidos para la previsión del movimiento medio de la troposfera fueron muy alentadores y esta experiencia histórica marca el punto de partida de la predicción numérica moderna.

En 1951 el meteorólogo Norman Phillips (1924-), que pertenecía al mismo equipo que los tres anteriores, intentó incluir la estructura vertical de la atmósfera e introdujo un modelo baroclínico formado por dos niveles.

En general, un flujo se dice barotrópico si la presión es constante sobre las superficies de densidad constante. Por tanto, en esos casos, la temperatura no depende de la presión o de los cambios de densidad.

Hay que resaltar que el modelo barotrópico inicial sufrió diversas modificaciones, algunas de las cuales resultaron interesantes. Así, el principio del balance de la vorticidad vertical absoluta fue reemplazado por el de la conservación de la vorticidad geostrófica absoluta, y a los modelos que utilizaron el viento geostrófico se les denominó modelos geostróficos. Posteriormente, a los modelos obtenidos aproximando el viento y la vorticidad (de forma selectiva, solo en algunos términos) por unos valores geostróficos se les llamó cuasigeostróficos.

Como contraposición a los modelos barotrópicos están los modelos baroclínicos, en los que las superficies de igual densidad están inclinadas con respecto a las superficies de igual presión, o lo que es lo mismo, la presión no es constante sobre las superficies con densidad constante sino que varía con las variaciones de la temperatura.

Mientras que los modelos barotrópicos predicen el movimiento de la troposfera media, los modelos baroclínicos también incluyen la estructura vertical de la atmósfera.

3.2.1.1 Modelos oceánicos según el número de grados de libertad vertical

Los modelos de Bryan [1969], de Madala y Piacsek [*Levitus, 1984*] y de Killworth et al., [1991], todos han utilizado niveles fijos en la vertical (en la dirección **z**) con un espaciamiento variable en los niveles de profundidad para resolver los cambios rápidos en la capa superficial. Por otro lado, Blumberg y Mellor [1987] y Haidvogel et al., [1991] introdujeron una coordenada llamada sigma, definida como $\sigma = z/D$, donde D es la profundidad. Esto lleva a que los valores de σ en las fronteras sean σ=0 en la superficie y σ=-1 en el fondo con una distribución automática para los contornos de profundidad. Haidvogel et al., [1991] ha introducido una representación semi-

espectral de la dimensión vertical (la disposición de sigma) en términos del polinomio de Chebyshev (el método de colocación). Ambos se han empleado para la formulación de la superficie libre (Figuras 3.2 y 3.3).

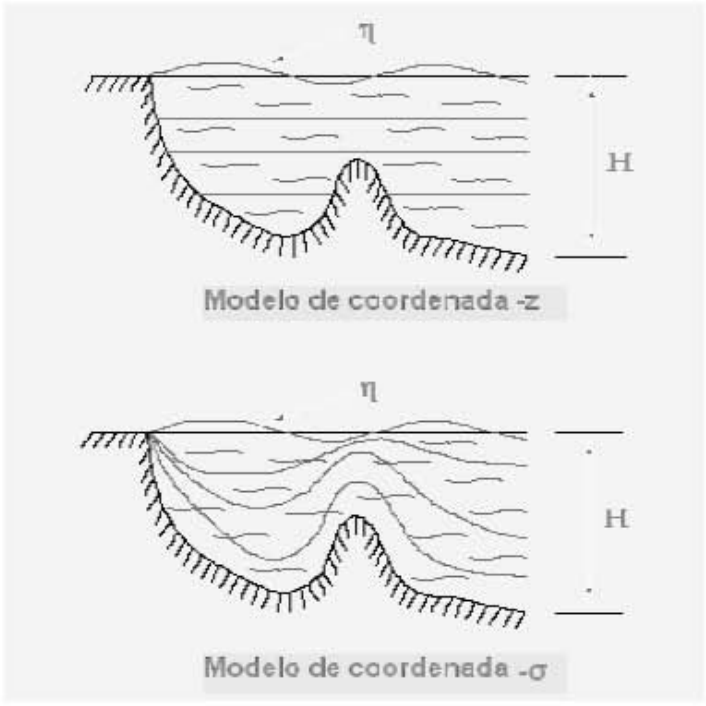


Figura 3.2 Discretización vertical en modelos oceánicos, H es la profundidad y η es la desviación de la superficie del mar.

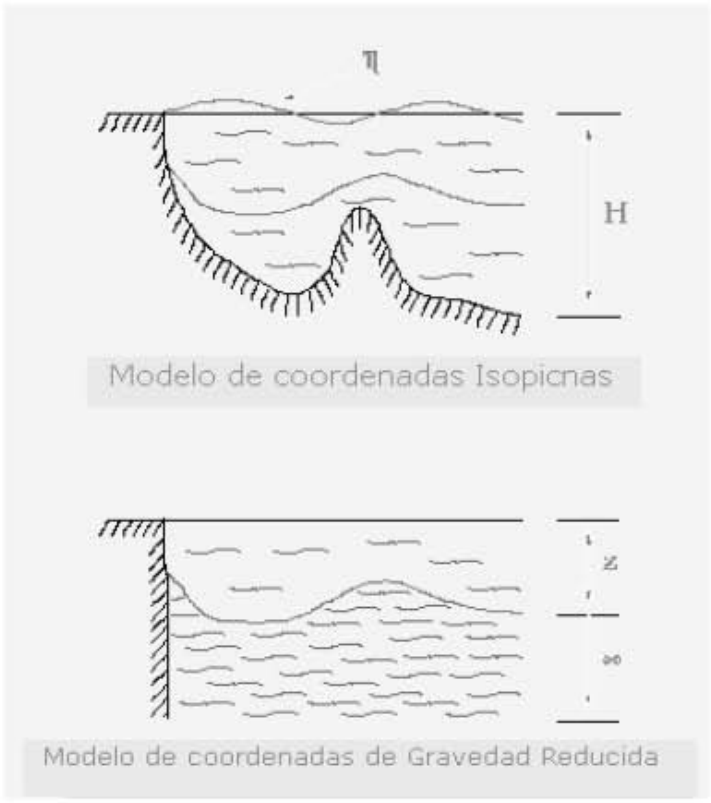


Figura 3.3 Discretización vertical en modelos oceánicos, H es la profundidad, z es el espesor de la capa superior, y η es la desviación de la superficie del mar.

Casi al mismo tiempo que se desarrollaron los modelos en el plano *z*, varios investigadores avanzaron en el estudio del océano poco profundo e hidrostático, encontrando que su estructura vertical tiene una tendencia semi-permanente. Una descomposición modal por funciones empíricas ortogonales o por modos dinámicos lineales ha mostrado que los primeros tres o cuatro modos pueden capturar generalmente el 95 por ciento de la energía. Así, integrando verticalmente las ecuaciones de movimiento entre superficies isopícnas (empleando la relación hidrostática [*Backhaus, 1983*]), han derivado un sistema donde varían las velocidades de la capa-promedio dependiendo del espesor de la capa. Puesto que las superficies isopícnas se mueven con el agua, esta representación es de hecho cuasi-Lagrangiana [A7]. El primer modelo que utiliza este enfoque en su formulación es el de superficie libre y fue desarrollado por O'Brien [*1985*], y la formulación del modelo de tapa rígida por Holland y Lin [*1975*]. Las formulaciones más recientes han sido hechas por Bleck y Boudra [*1986*] y por Oberhuber [*1993*].

3.3 Construcción y aplicación de un modelo

Los modelos generales se basan en leyes de la física representadas por ecuaciones matemáticas que se resuelven utilizando una malla tridimensional sobre el globo terráqueo. A fin de simular el clima, los principales componentes del sistema climático deben representarse en submodelos (la atmósfera, los océanos, la superficie terrestre, la criosfera y la biósfera), junto con los procesos que ocurren entre ellos y dentro de cada uno de ellos. Los modelos climáticos mundiales en los que se han acoplado los componentes atmosféricos y oceánicos se conocen también con el nombre de Modelos de Circulación General Atmósfera–Océano (MCGAO).

En el módulo atmosférico, por ejemplo, se resuelven ecuaciones que describen la evolución a gran escala del momento, el calor y la humedad. Se resuelven ecuaciones similares con respecto a los océanos. Actualmente, la resolución de la parte atmosférica de un modelo típico es de aproximadamente 250 km en la horizontal y de alrededor de 1 km en la vertical, por encima de la capa límite. La resolución de un modelo oceánico típico oscila aproximadamente entre 200 y 400 m en la vertical, con una resolución horizontal de entre 125 y 250 km. Aunque recientemente se han

logrado experimentos con 5 o 10 veces mayor resolución, lo que implica un poder de súper computo de 500 a 1000 veces más. Las ecuaciones se resuelven generalmente para cada período de media hora de un modelo integrado. Muchos procesos físicos, como los que están relacionados con las nubes o la convección oceánica, ocurren en escalas espaciales mucho más pequeñas que la malla de los modelos y en consecuencia no pueden modelarse y resolverse en forma explícita. Sus efectos medios se incluyen en forma aproximada con un método simple, aprovechando sus relaciones (basadas en la física) con las variables de mayor escala. Esta técnica se conoce con el nombre de parametrización.

Los modelos climáticos se han perfeccionado en las últimas décadas gracias al desarrollo de las computadoras. Durante ese período se crearon modelos separados de cada uno de los componentes principales, la atmósfera, la superficie terrestre, los océanos y el hielo marino, que se fueron integrando gradualmente. El acoplamiento de los distintos componentes es un proceso difícil. En la figura 3.4 puede verse la evolución de los modelos climáticos en el pasado y el presente, y su posible evolución en el futuro.

Algunos modelos corrigen los errores y los desequilibrios en los flujos en la superficie mediante *ajustes de flujo*, que son ajustes sistemáticos determinados empíricamente en la interfaz océano-atmósfera que se mantienen fijos en el tiempo para aproximar el clima simulado al estado observado. Se ha diseñado una estrategia para realizar experimentos climáticos que elimina gran parte de los efectos que algunos errores de los modelos tienen en los resultados. A menudo se hace, en primer lugar, una *pasada de control* de la simulación climática con el modelo. Después se ejecuta la simulación del experimento de cambio climático, por ejemplo con un aumento del CO_2 en la atmósfera del modelo. Por último, se calcula la diferencia para obtener una estimación del cambio sufrido por el clima a causa de la perturbación. La técnica de diferenciación elimina la mayor parte de los efectos de cualquier ajuste artificial en el modelo, así como los errores sistemáticos que son comúnes a ambas formas de ejecución del *modelo*. Sin embargo, la comparación de los diferentes resultados de los modelos, demuestra que hay cierta clase de errores que siguen influyendo en los ellos.

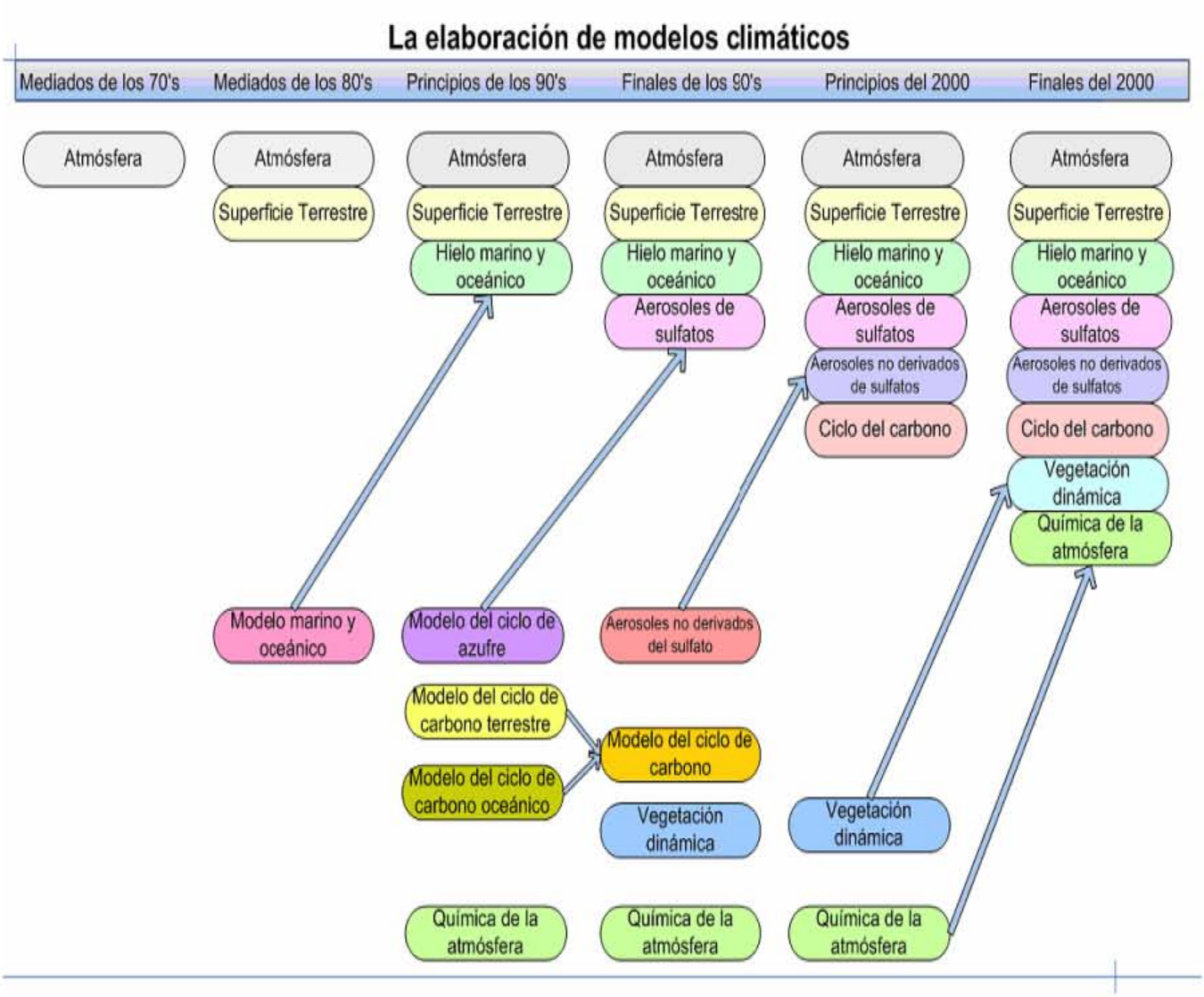


Figura 3.4 La elaboración de modelos climáticos en los últimos 30 años se caracterizó en un principio por el desarrollo separado de los distintos componentes, que luego se fueron acoplando en modelos climáticos integrales.

3.4 Modelo NCOM

El modelo utilizado para las representaciones realizadas en este trabajo es el ***N**avy **C**oastal **O**cean **M**odel* (NCOM). El NCOM se ha desarrollado en el ***N**aval **R**esearch **L**aboratory* (NRL) de los E. U., para simular procesos de interacción océano-atmósfera de mesoescala, asi como estudiar procesos en las regiones costeras [*Hodur et al., 2002, Zavala-Hidalgo et. al.2003*].

El NCOM utiliza coordenadas verticales híbridas, las cuales se habían utilizado en otro modelo, el Sigma/Z-Level Model [Martin et al., 1998]. Este sistema de coordenadas utiliza coordenadas sigma σ cerca de la superficie y coordenadas geopotenciales (z) por debajo de una profundidad especificada. Esto proporciona una flexibilidad para establecer la malla vertical. El modelo se puede correr únicamente con coordenadas sigma, con coordenadas z (aunque por lo menos una capa de coordenadas sigma se requiere para acomodar la superficie libre), o con una combinación de ambas.

El NCOM es un modelo oceánico tridimensional de ecuaciones primitivas, hidrostático, incompresible, y con las aproximaciones de Boussinesq. Las ecuaciones, en coordenadas cartesianas, son:

$$\frac{\partial u}{\partial t} = -\nabla \bullet (\nu u) + Qu + fv - \frac{1}{\rho_0}\frac{\partial p}{\partial x} + F_u + \frac{\partial}{\partial z}\left(K_M \frac{\partial u}{\partial z} \right), \tag{3.1}$$

$$\frac{\partial v}{\partial t} = -\nabla \bullet (\nu w) + Qv + fu - \frac{1}{\rho_0}\frac{\partial p}{\partial y} + F_v + \frac{\partial}{\partial z}\left(K_M \frac{\partial v}{\partial z} \right), \tag{3.2}$$

$$\frac{\partial p}{\partial z} = -\rho g \ , \tag{3.3}$$

$$\nabla \bullet \nu = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial t} + \frac{\partial w}{\partial z} = Q \, , \tag{3.4}$$

$$\frac{\partial T}{\partial t} = -\nabla \bullet (\nu T) + QT + \nabla_h (A_H \nabla_h T) + \frac{\partial}{\partial z}\left(K_H \frac{\partial T}{\partial z} \right) + Q_r \frac{\partial \gamma}{\partial z}, \tag{3.5}$$

$$\frac{\partial S}{\partial t} = -\nabla \bullet (\nu S) + QS + \nabla_h (A_H \nabla_h S) + \frac{\partial}{\partial z}\left(K_H \frac{\partial S}{\partial z} \right) \ , \tag{3.6}$$

$$\rho = \rho(T, S, z) \ , \tag{3.7}$$

Notación

Q	Término inicial de flujo de volumen.
x, y, z	Son las coordenadas de dirección.
t	El tiempo.
$v = (u, v, w)$	Vector de velocidad en tres dimensiones.
T	Temperatura potencial.
S	Salinidad.
∇_h	Gradiente del operador horizontal.
f	Parámetro de Coriolis.
p	Presión.
ρ	Densidad del agua.
ρ_0	Densidad de referencia del agua.
g	Aceleración de la gravedad.
F_u, F_v	Términos de mezcla horizontal para las ecuaciones de momento.
A_h	Términos de mezcla horizontal para las variables escalares (T, S).
K_M	Coeficiente de viscosidad vertical para las ecuaciones de momento.
K_H	Coeficiente de viscosidad vertical para las ecuaciones escalares.
Q_r	Radiación solar.
γ	Función que describe extinción de la radiación solar con la profundidad.

La forma de estas ecuaciones en coordenadas sigma está dada por Blumberg y Mellor [1987].

Las condiciones de frontera de la superficie están dadas por las ecuaciones de momento (3.1) y (3.2), el flujo de calor en la superficie está dado por la ecuación (3.5), el flujo de salinidad está dado por la ecuación (3.6). La ecuación (3.7) es calculada usando las fórmulas de Friedrich y Levitus [1972] o Mellor [1991].

El sistema de ecuaciones se resuelve en una malla C Arakawa. La malla horizontal es ortogonal y curvilínea.

3.5 Simulación del Golfo de México del NCOM

Los datos utilizados en este trabajo provienen de una simulación del Golfo de México utilizando el NCOM con una resolución de 0.05º en latitud y longitud de las variables deseadas. En la horizontal la malla es de 352x320 puntos, abarcando de 98.15º W a 80.60º W, y de 15.55º N a 31.50º N (Figura 3.5). Las fronteras abiertas se encuentran por la orilla oriental del dominio, en el Caribe y el Estrecho de Florida. En la vertical esta simulación tiene 20 puntos de espaciado en los niveles sigma, hasta una profundidad de 100 m, y 20 en los niveles z por debajo de los 100 m, con una profundidad máxima de 4000 m.

Los campos iniciales de temperatura y salinidad para esta simulación se derivaron del World Ocean Atlas 1994 [*National Oceanic and Atmospheric Administration, 1994*]. Los campos que forzan la superficie, como el viento, calor latente, calor sensible, y los flujos de calor radiativos se tomaron de DaSilva et al. [1994], los cuales tienen una resolución de 0.5º x 0.5º, para analizar la climatología mensual. Treinta ríos son incluidos en la simulación utilizando el término fuente de flujo de volumen en las ecuaciones del modelo. Todos los campos que se usan para forzar el modelo se interpolan linealmente en el tiempo en cada paso de tiempo del modelo.

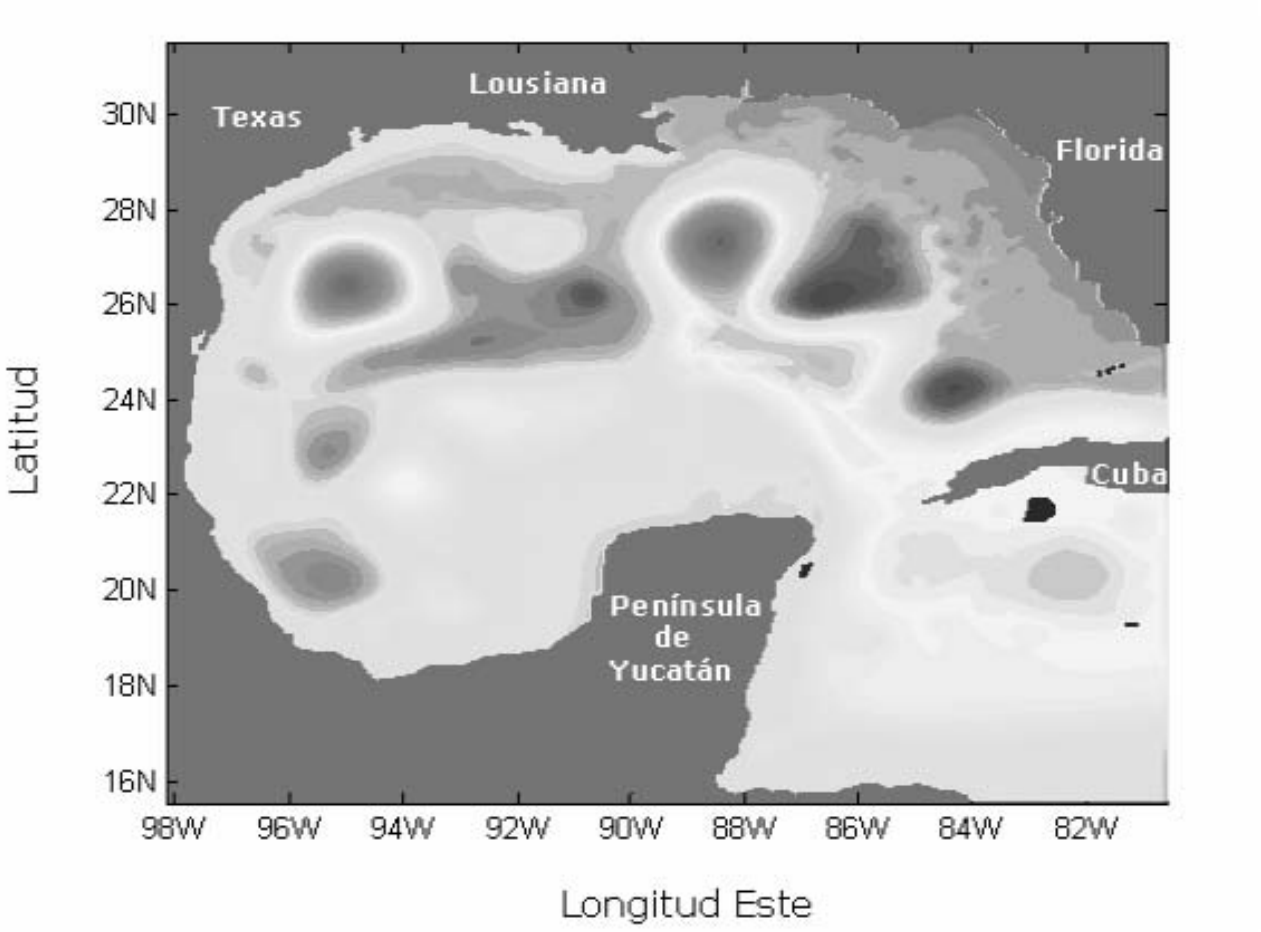


Figura 3.5 Simulación realizada con el NCOM en el Golfo de México. Se muestra el dominio del modelo.

3.6. Características de la solución del modelo

A gran escala, en el Golfo de México se presenta un sistema conocido como Corriente del Lazo, la cual forma parte del sistema de corrientes del Atlántico Norte. Esta corriente atraviesa el Caribe y llega al Golfo de México a través del estrecho de Yucatán, entre Quintana Roo y la isla de Cuba. Ya en el Golfo de México, la corriente se mueve hacia el norte, gira hacia el este y después hacia el sur, para abandonarlo por el Estrecho de Florida. De esta corriente se desprenden remolinos que se mueven hacia el oeste del Golfo.

Más información del NCOM y las simulaciones utilizadas en este trabajo se pueden encontar en el *Description of the Navy Coastal Ocean Model Version 1.0* de Paul J. Martin.

3.7 Ventajas del modelo

Las coordenadas híbridas permiten tener muy alta resolución vertical dentro de la plataforma continental y la capa superficial donde la variabilidad espacial y temporal es mucho mayor.

Las coordenadas híbridas permiten representar de mejor manera el talud continental y, por lo tanto, estudiar mejor los procesos en la plataforma continental y los flujos perpendiculares a ésta.

Cuenta con muchas opciones de parametrización y esquemas numéricos.

CAPÍTULO 4

VISUALIZACIÓN DE LA EVOLUCIÓN EN EL TIEMPO DE LAS CORRIENTES POR MEDIO DE FLECHAS LAGRANGIANAS

4.1 Introducción

Como hemos visto en los capítulos anteriores, existen diferentes métodos para visualizar las variables físicas que se encuentran en el océano.

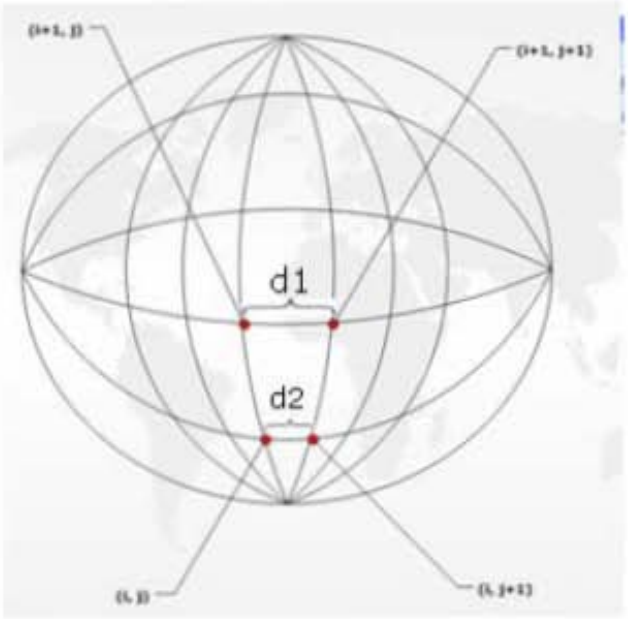
La visualización mediante flechas es una representación que ha sido muy utilizada en el ámbito de la visualización de corrientes oceánicas. Existen muchas instituciones que las utilizan para representar los resultados obtenidos con diferentes modelos.

4.2 Representación de la velocidad de la corriente por flechas

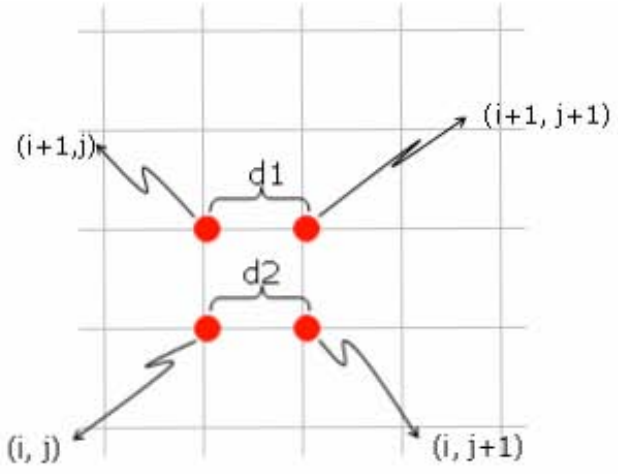
Como se vió en el capítulo 1, para visualizar un campo vectorial frecuentemente se trazan pequeñas flechas, las cuales indican la magnitud y dirección de la corriente sobre la superficie del océano.

Con base en lo anterior debemos tomar en cuenta dos cuestiones importantes:

1. En una malla en donde los puntos están a distancias determinadas por incrementos de ángulo constantes, como en el caso de los datos utilizados en este trabajo, la distancia de un punto (i, j) al punto (i, j+1) que se encuentran sobre una cierta latitud, no es la misma que hay de un punto (i+1, j) al punto (i+1, j+1) que se encuentran sobre una latitud diferente a la de las puntos anteriores. Esta distancia varía dependiendo de la latitud en que se encuentren los puntos (Figura 4.1.a), la cual en una superficie plana no varía (Figura 4.1.b).



(a)



(b)

Figura 4.1 En (a) d1 es diferente de d2. En (b) d1 es igual a d2.

2. No todos los puntos de la región son válidos para dibujar las flechas, ya que hay puntos que están en tierra y por lo tanto hay que identificarlos.

Tomando en cuenta estas dos restricciones, las flechas están construidas a base de segmentos rectos, los cuales se calculan usando $d = vt$, donde d es la distancia o tamaño del segmento, v la velocidad en una zona y t el tiempo, y la dirección del segmento es calculada con el método de Runge-Kutta de segundo orden [A9].

4.2.1 Punta de la flecha

Para la construcción de la punta de la flecha, se utiliza el último segmento de la flecha para calcular el ángulo que forma con las dos rectas que conforman la punta (Figura 4.2).

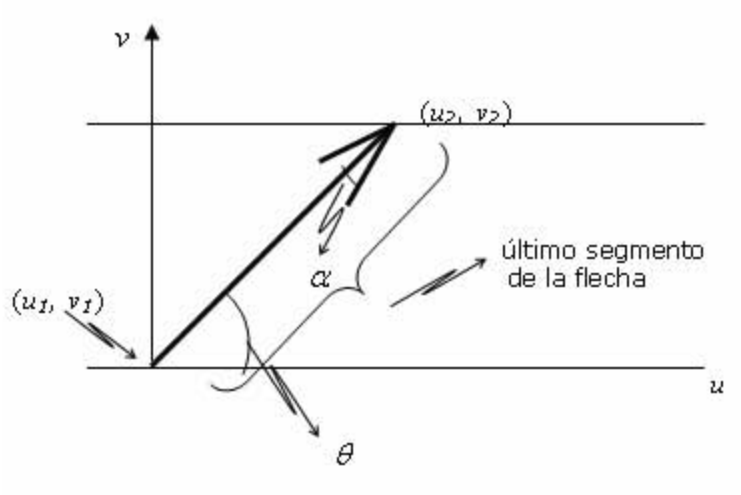


Figura 4.2 Diagrama de la construcción de la flecha.

Donde:

$$\theta = tg^{-1}\frac{\Delta v}{\Delta u} \qquad (4.1)$$

$$\Delta v = v_2 - v_1 \qquad (4.2)$$

$$\Delta u = u_2 - u_1 \qquad (4.3)$$

Donde:

v_2 Coordenada en v del punto donde termina el último segmento de recta.

v_1 Coordenada en v del punto donde empieza el último segmento de recta.

u_2 Coordenada en u del punto donde termina el último segmento de recta.

u_1 Coordenada en u del punto donde empieza el último segmento de recta.

Para facilitar los cálculos de éstas coordenadas se decidió pasarlas a un plano complejo donde es mucho más fácil y rápido el cálculo, utilizando las fórmulas de las ecuaciones (4.4) y (4.5)

$$w_2 = TPunta * e^{(j*(\theta+\alpha))} \quad (4.4)$$

$$w_3 = TPunta * e^{(j*(\theta-\alpha))} \quad (4.5)$$

Donde

- ***TPunta*** es una constante para determinar el tamaño de la punta de la flecha.

4.3 Construcción de la función de graficado

Esta función consiste en graficar un conjunto de flechas con las características descritas en la sección anterior. Esta función dibuja las flechas y calcula su desplazamiento entre imágenes secuenciales y tiene los siguientes parámetros de entrada.

4.3.1 Parámetros de entrada

Los datos utilizados para la creación de las imágenes son tomados, de una simulación numérica del Golfo de México utilizando el NCOM.

La función que se desarrolló para representar gráficamente el campo vectorial de las corrientes oceánicas requiere de los siguientes parámetros de entrada:

(X0, Y0, U, V, X1, Y1, MV, N, DT, C, N2)

Los datos de entrada de X0 y Y0 corresponden a las coordenadas en longitud y latitud de la malla, respectivamente, los cuales tienen los siguientes valores:

X0 =-98.15: 0.05: -80.6.

Y0 = 15.55: 0.05: 31.5.

En total se tienen 352 puntos en longitud y 320 en latitud correspondiente a una malla con resolución de 0.05º x 0.05º en el dominio de estudio.

U y **V** son datos proporcionados por el modelo y corresponden a las componentes zonal y meridional de la velocidad, respectivamente. Son campos bidimensionales de 352 x 320 puntos (Figura 4.3).

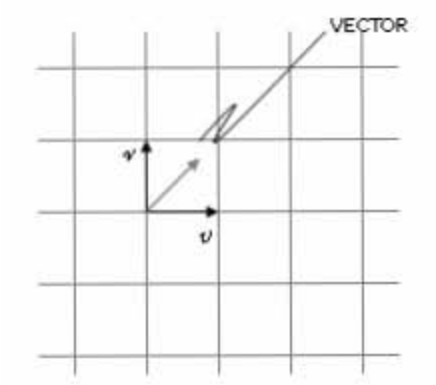


Figura 4.3 Componentes zonal y meridional de la velocidad (u , v).

X1 y **Y1** son vectores o matrices que contienen las coordenadas de inicio de longitud y latitud para los primeros segmentos de las flechas. Estas matrices pueden coincidir con X0 y Y0 o pueden ser diferentes.

El número de elementos que contengan estos vectores está ligado a la densidad de flechas que se quiera tener en la imagen, ya que ésta es una variable que el usuario proporciona desde la una ventana de menú del programa principal el cual se explica en el siguiente capitulo.

MV Es una bandera que indica si los datos de inicio (X1, Y1) son proporcionadas como matrices o vectores. Se utiliza MV=1 si son matrices y MV=0 si son vectores.

N Es el número total de segmentos que compondrán cada flecha, el cual debe tener una relación de correspondencia con N2 (el último parámetro que se proporciona) de la siguiente manera:

$$N > N2 \tag{4.6}$$

DT Es el incremento de tiempo de cada segmento que compondrá la flecha (en horas).

C Es el color que tendrán las flechas.

N2 Es el número de segmentos que avanza un punto entre dos imágenes secuenciales y esta determinado por el tiempo de simulación entre dos imágenes (en horas):

$$N2 = \frac{\left(incT\,[dia] \,/\, NI\,ma\right) * 24\left[hrs. \,/\, dia\right]}{DT\,[hrs.]} \tag{4.7}$$

Donde

- **incT** es el incremento de tiempo en que es leído cada archivo.
- **NIma** es el número de imágenes que se generarán por cada archivo de datos leído.
- **DT** Es el incremento de tiempo de cada segmento que compondrá la flecha.

Por ejemplo, si el paso de tiempo entre cada imagen es de 12 horas, el DT es de 4 y N es de 6, cada segmento de la flecha nos representa 4 horas y como cada flecha está

compuesto por 6 segmentos entonces en una imagen una flecha representa 24 horas. Para determinar cada siguiente inicio de la flecha, debemos tomar el último segmento que representa el total de horas que hay en el paso de tiempo de la imagen, o sea, que si en este caso 3 segmentos representan 12 horas se debe tomar el punto final del tercer segmento de la flecha para comenzar la flecha en la siguiente imagen. Gráficamente es lo siguiente:

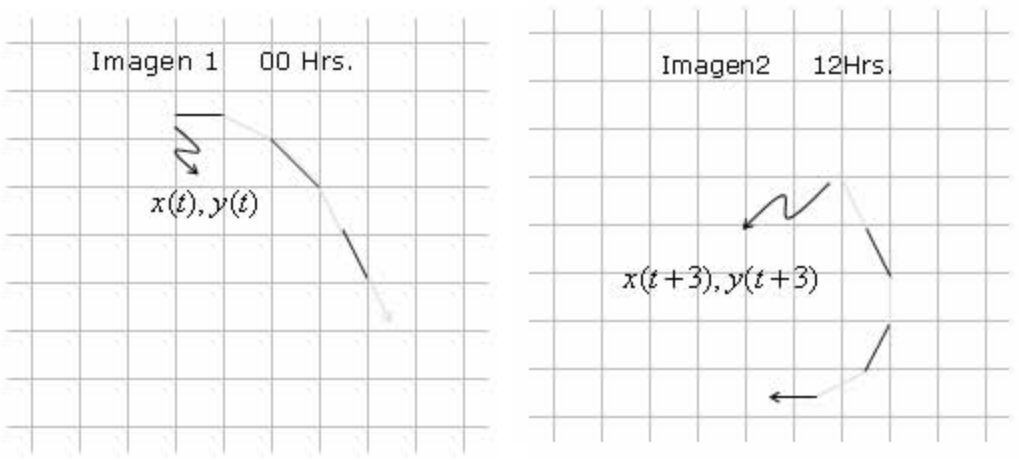


Figura 4.4 Tenemos una imagen inicial a las 00 hrs. y la siguiente imagen a las 12 hrs.

Donde cada flecha está compuesto por 6 segmentos como sigue:

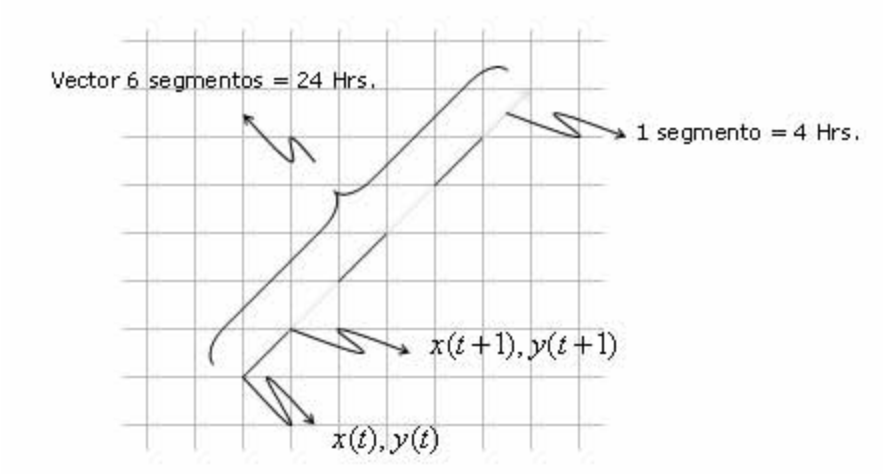


Figura 4.5 Composición gráfica de la flecha y su avance en el tiempo.

Para la imagen 1 la flecha inicia en el punto $(x(t), y(t))$ en el tiempo t, para la imagen 2 inicia en el punto $(x(t+3), y(t+3))$ en un tiempo $t+3$ ya que, siguiendo la ecuación (4.7), N2=3.

En la función los datos de *N*, *DT*, *NIma* e *IncT* son proporcionados por el usuario, para lo cual tiene que tomar en cuenta las restricciones mencionadas anteriormente.

4.3.2 Salidas de la función

La función anterior proporciona los arreglos NXN y NYN que contiene las coordenadas de inicio de las flechas para el siguiente paso de tiempo en la imagen.

4.4 Error de la función

Para hacer el cálculo del error de la función, se analiza una circunferencia construida con la función.

En la figura 4.6 se puede observar que el círculo se cierra, en la figura 4.7, haciendo un aumento en la región donde empieza y termina la función, se nota un pequeño corrimiento.

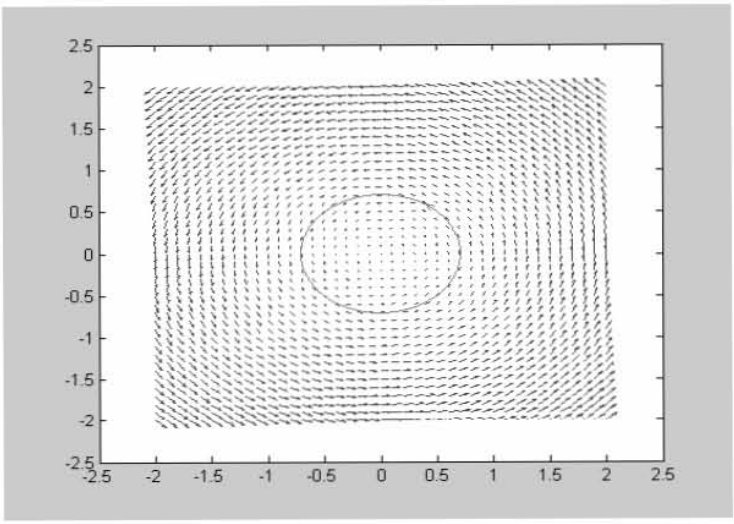


Figura 4.6 Construcción de una circunferencia con la función desarrollada.

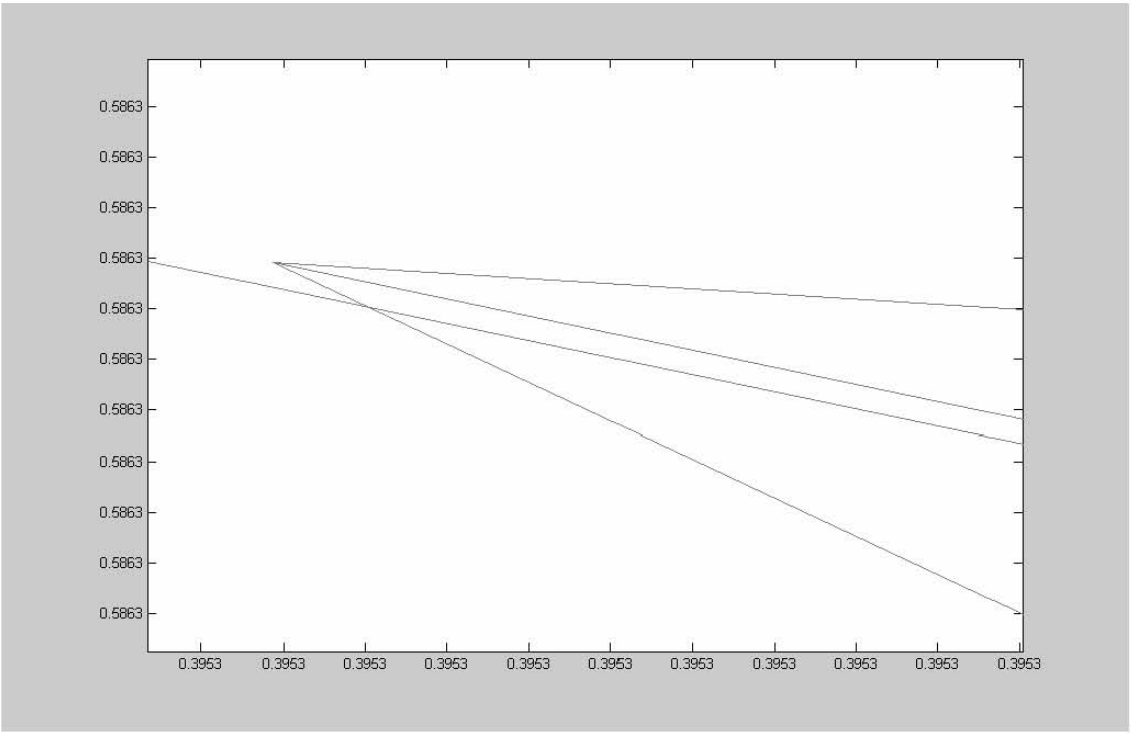


Figura 4.7 Aumento de la región donde empieza y termina la circunferencia.

Se puede notar que el error es muy pequeño. En este caso el círculo tiene un diámetro un poco menor de 1 y el error es menor de 1/10000, lo cual muestra la buena precisión del método utilizado.

CAPÍTULO 5

VISUALIZACIÓN DE CORRIENTES Y VARIABLES ESCALARES DEL OCÉANO

5.1 Introducción

Como se vió en el capítulo 3, existen diferentes procesos físicos en el océano donde intervienen diversas variables que pueden ser representadas en una imagen. Para esto contamos con diferentes formas en las que estas variables pueden ser mostradas y estudiadas de una manera rápida y eficaz, ya que, como hemos mencionado, es mayor la cantidad de información generada por las diferentes fuentes de información de lo que ésta puede ser estudiada y una forma muy eficiente para hacerlo es a través de imágenes.

5.2 Variables Físicas

Las variables físicas que se muestran en este trabajo son nivel del mar, temperatura, salinidad y corrientes superficiales.

Las variables de temperatura, salinidad y nivel del mar son variables escalares y pueden ser representadas de diferentes formas. El método usado en este trabajo para generar las imágenes es el de falso color.

En las siguientes figuras se muestra la distribución del nivel del mar, la salinidad y la temperatura con este método. Por ejemplo, en el caso de las variaciones del nivel del mar en el espacio y el tiempo, se usa una gama de colores que van desde un azul intenso que representa las zonas que están 0.5 m por abajo del nivel medio del mar, hasta un rojo intenso que representa las zonas que están 0.5 m por arriba del nivel medio del mar, (Figura 5.1). En forma análoga, en las figuras 5.2 y 5.3 se muestra la variación de la salinidad y de la temperatura respectivamente.

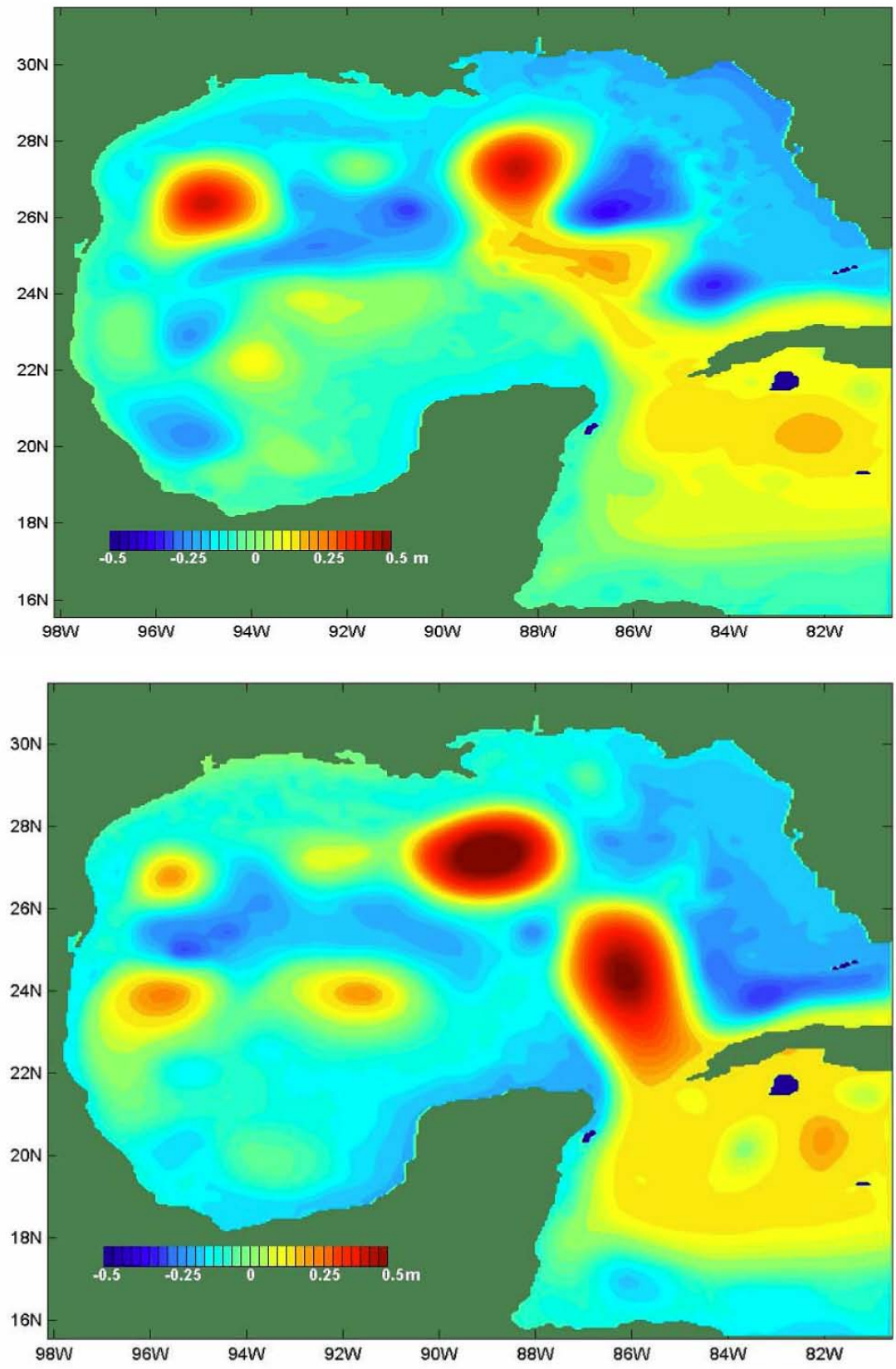


Figura 5.1 Nivel del Mar. Mes de Enero (arriba) Julio (abajo).

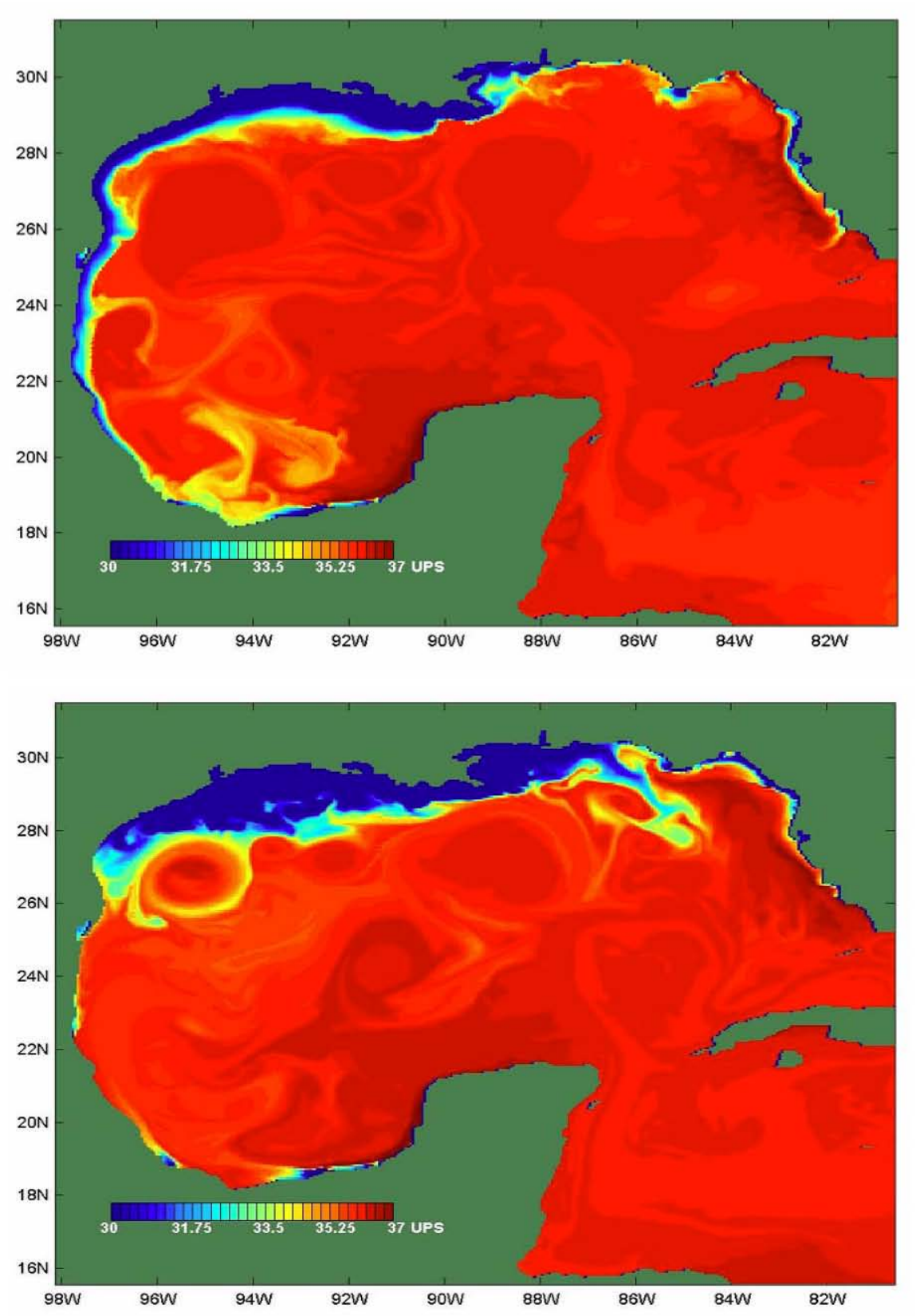


Figura 5.2 Salinidad. Mes de Enero (arriba), Julio (abajo).

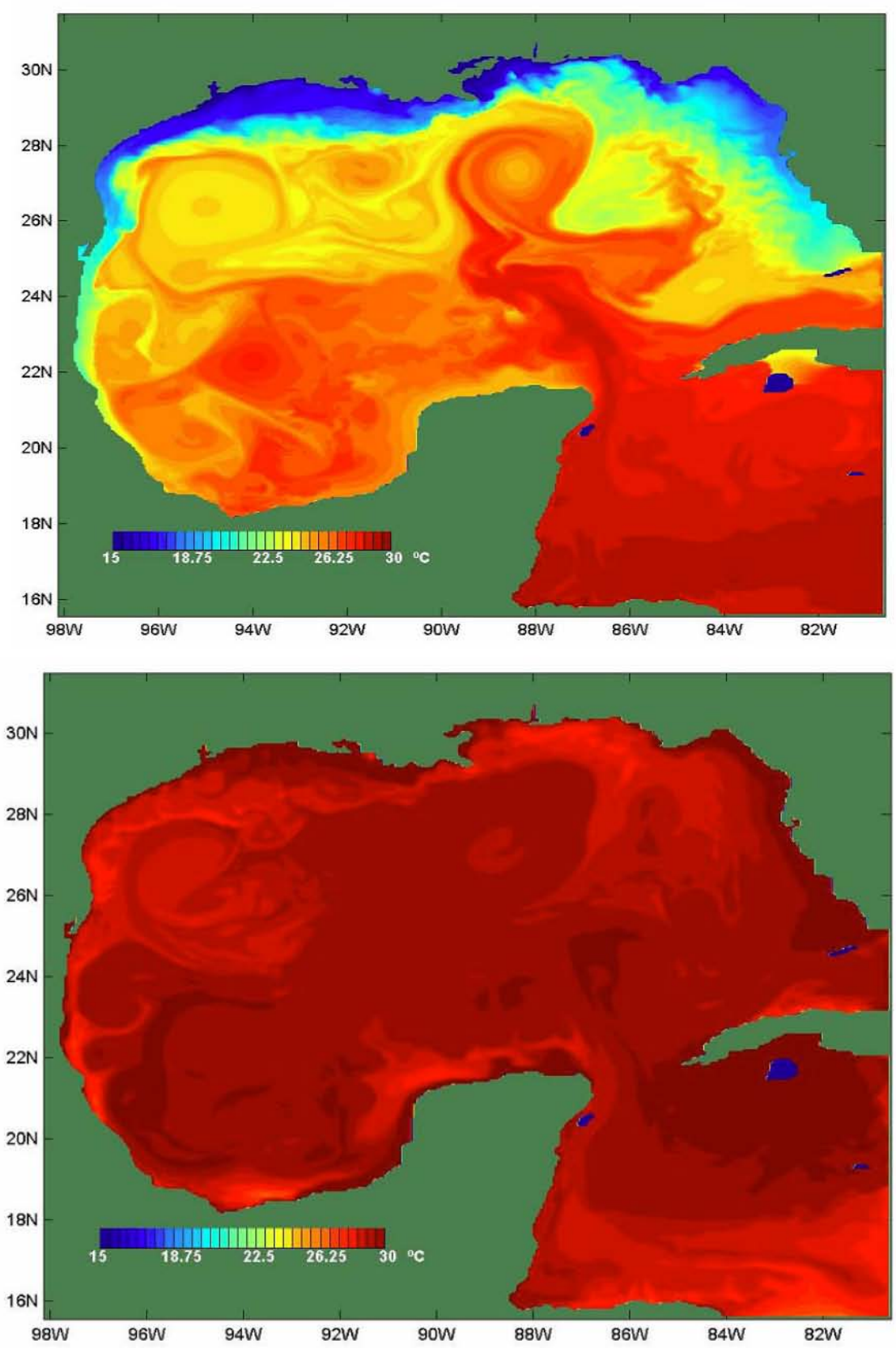
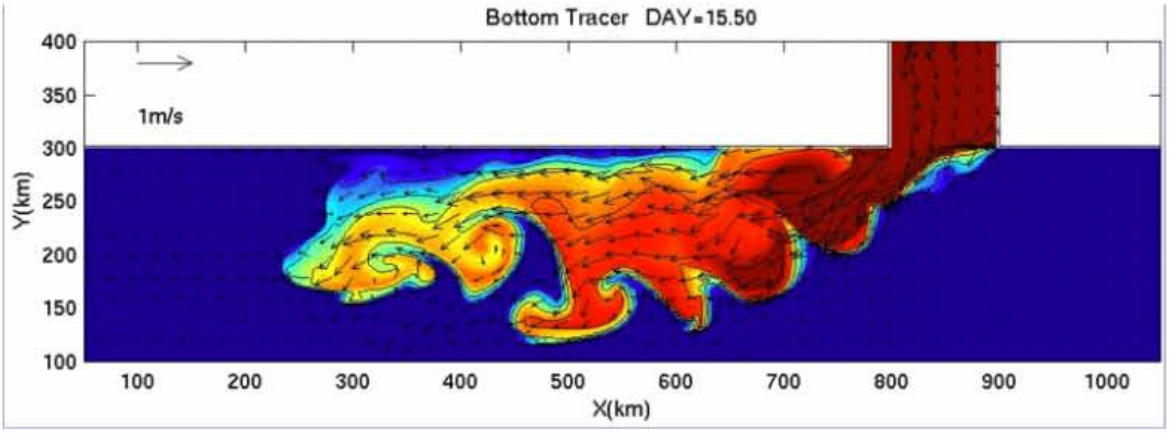


Figura 5.3 Temperatura. Mes de Enero (arriba), Julio (abajo).

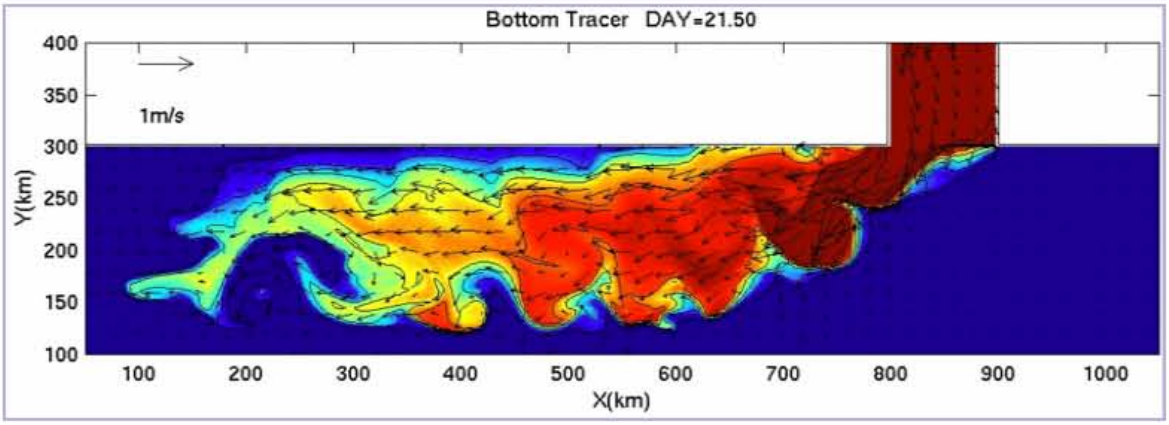
5.3 Visualización de corrientes

Como se menciona en el capítulo anterior, la corriente es un campo vectorial cuya representación se hace mediante flechas, las cuales representan la velocidad y dirección de las corrientes como función del tiempo.

A diferencia de las flechas lagrangianas desarrolladas en este trabajo, las flechas usadas en otras visualizaciones son eulerianas [A8], por ejemplo imágenes generadas con el modelo POM en la Universidad de Princeton (http: //www.aos.princeton.edu /WWWPUBLIC /htdocs.pom/ FTPbackup/ Animations/BBLanim40d.gif) (Figura 5.4.a y b).



(a)



(b)

Figura 5.4 Se muestra en (a) y (b) que las flechas siempre conservan su posición de inicio y lo que cambia es el flujo. En este caso se representa una pluma de un fluido de diferente densidad que su entorno. [Model: Ezer & Mellor, OM, 2004; Ezer, OM, 2005].

En nuestro caso usamos flechas lagrangianas [A8] y curvilíneas, con ciertas características dinámicas las cuales se describieron con detalle en el capítulo 4. Un ejemplo de las imágenes que se generan con la función descrita en el capitulo anterior es el que se muestra en la figura 5.5 en donde se representan las diferentes corrientes del Golfo de México.

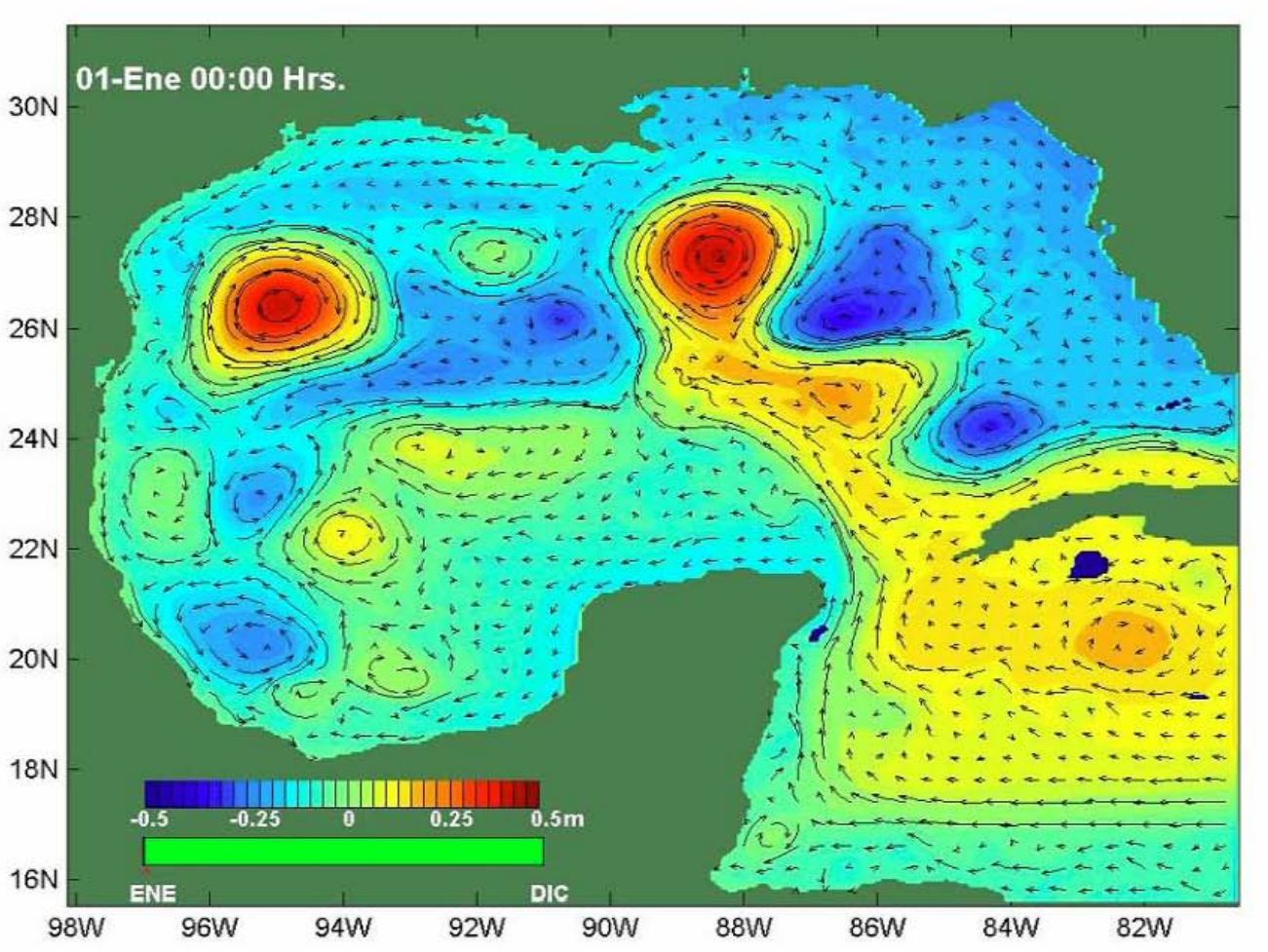


Figura 5.5 Visualización de corrientes mediante flechas lagrangianas. Los colores muestran la variación en el nivel del mar.

Para obtener una imagen como la mostrada en la figura 5.5 se hace una superposición de las flechas con alguna variable escalar generada con la técnica de falso color, en este caso la temperatura, salinidad o nivel del mar, lo que permite mostrar la mayor

cantidad de variables posibles en una imagen y hacer un estudio de los diferentes procesos físicos que se presentan.

5.4 Generación de imágenes y animaciones

5.4.1 Calidad de la visualización

Las imágenes se generaron con MATLAB en formato JPEG ya que una de sus ventajas es comprimir las imágenes y por ello se generan archivos de menor tamaño. Comprime las imágenes eliminando información que no es visible para el ojo. Soporta un gran número de colores (16 millones). Esto es de gran utilidad para el manejo de estas imágenes en Internet.

El video es la reproducción en forma secuencial de las imágenes, que al verse con una determinada velocidad y continuidad dan la sensación al ojo humano de apreciar el movimiento natural. Para esto se utilizó el programa Adobe Premiere 6.0 con el cual podemos generar videos en diferentes formatos, tipos de compresión y variar el tipo de resolución de acuerdo al formato. Los formatos que se utilizan son FLC, AVI y Windows Media. Por ejemplo, en el caso del formato FLC la ventaja que se tiene para hacer un mejor estudio es que, después de haberse generado la animación, se puede variar el número de cuadros por segundo en la animación y la intensidad de los colores, lo cual no es posible con los otros formatos.

5.4.2. Percepción adecuada de las variables físicas

La percepción de las variables físicas en cada una de las imágenes es muy importante. Una de las variables representadas es la corriente, para lo cual, se construyó la función cuyas características se explicaron en el capitulo anterior.

Otras variables son las de nivel del mar, salinidad y temperatura, datos proporcionados por el modelo cada dos días, para lo cual la generación de imágenes, las cuales varían en diferentes lapsos de tiempo menores a la de los datos

proporcionados. Buscando que no se vean saltos bruscos en la animación, se hace una interpolación lineal de estas variables.

5.5 Descripción del Programa

Para hacer más fácil la generación de estas imágenes se desarrolló un programa en Matlab 6.5 con una interfaz gráfica (Figura 5.6) para el mejor manejo de todas las variables, las cuales pueden ser modificadas por el usuario de acuerdo a su percepción.

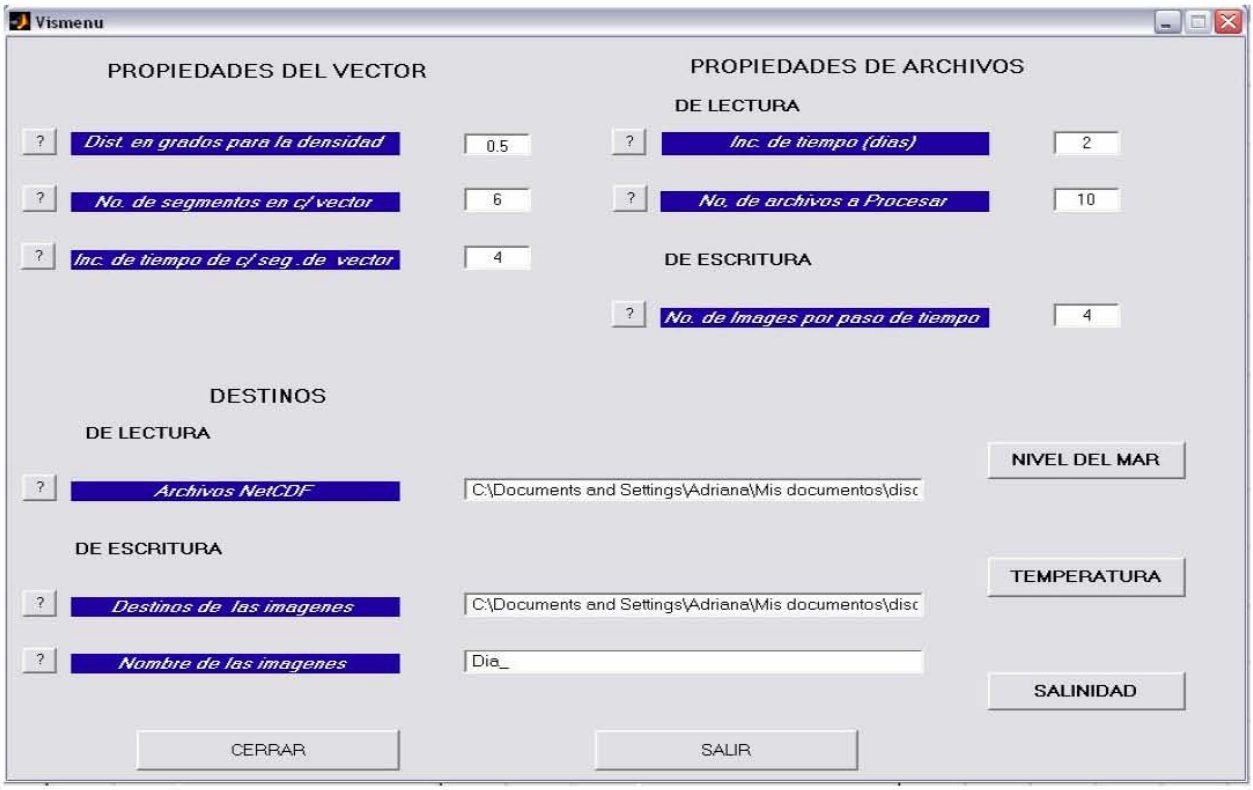


Figura 5.6 Menú de la interfaz de usuario.

5.5.1 Parámetros que proporciona el usuario

Este menú se puede dividir en cuatro partes que se describen a continuación:

1. *Propiedades del Vector*. Se encuentra en la parte superior izquierda del menú.

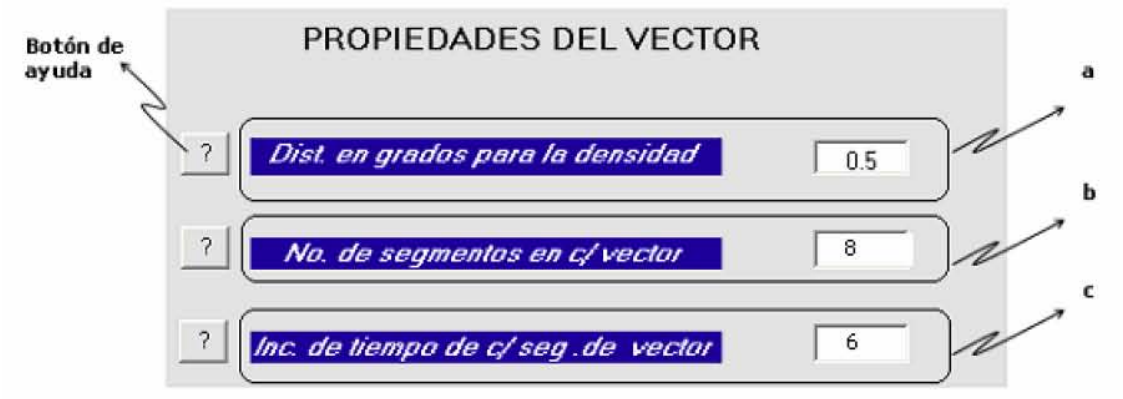


Figura 5.7 Área donde se proporcionan los datos de las propiedades del vector.

a) ***Distancia en grados entre vectores*** (Figura 5.7.a). Por ejemplo en la imagen 5.8.a se generan vectores cada medio grado (0.5º) y en la imagen 5.8.b se generan imágenes cada cuarto de grado (0.25º), lo que determina la densidad de vectores que tendrá la imagen.

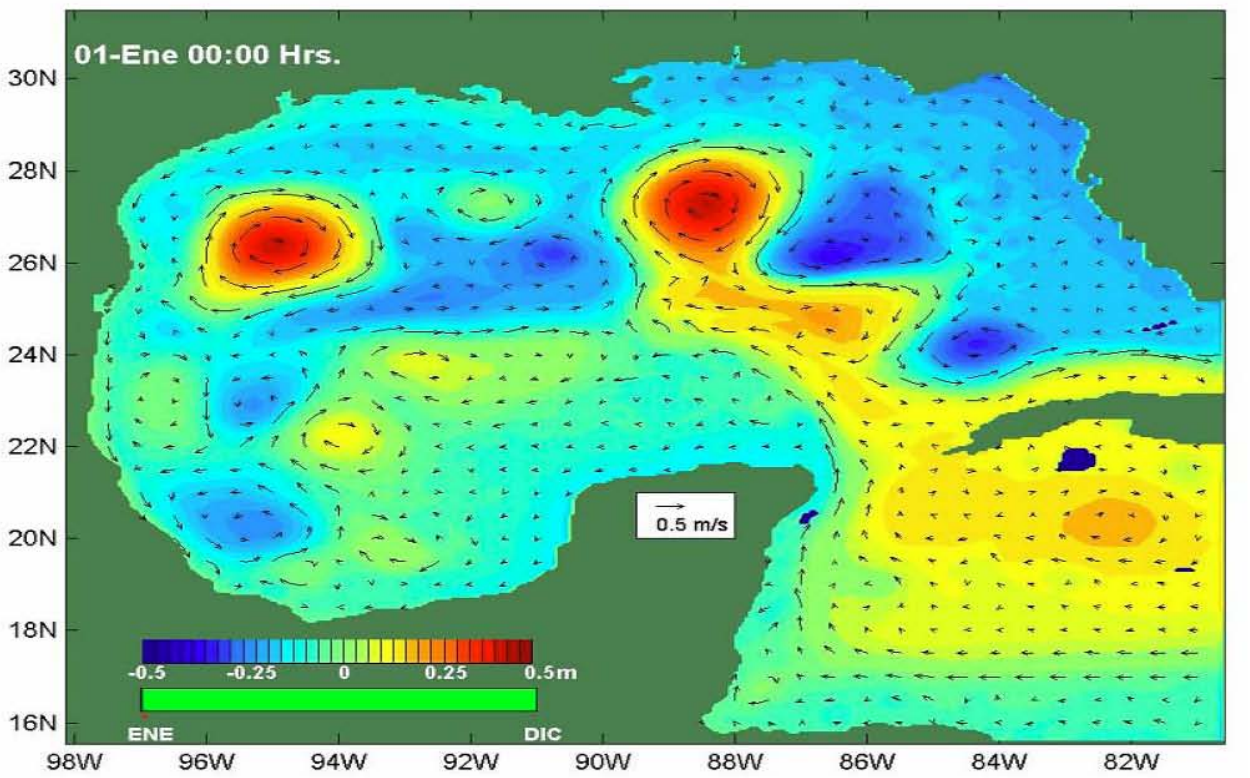


Figura 5.8.a Imagen del Golfo de México generando vectores cada 0.5 de grado.

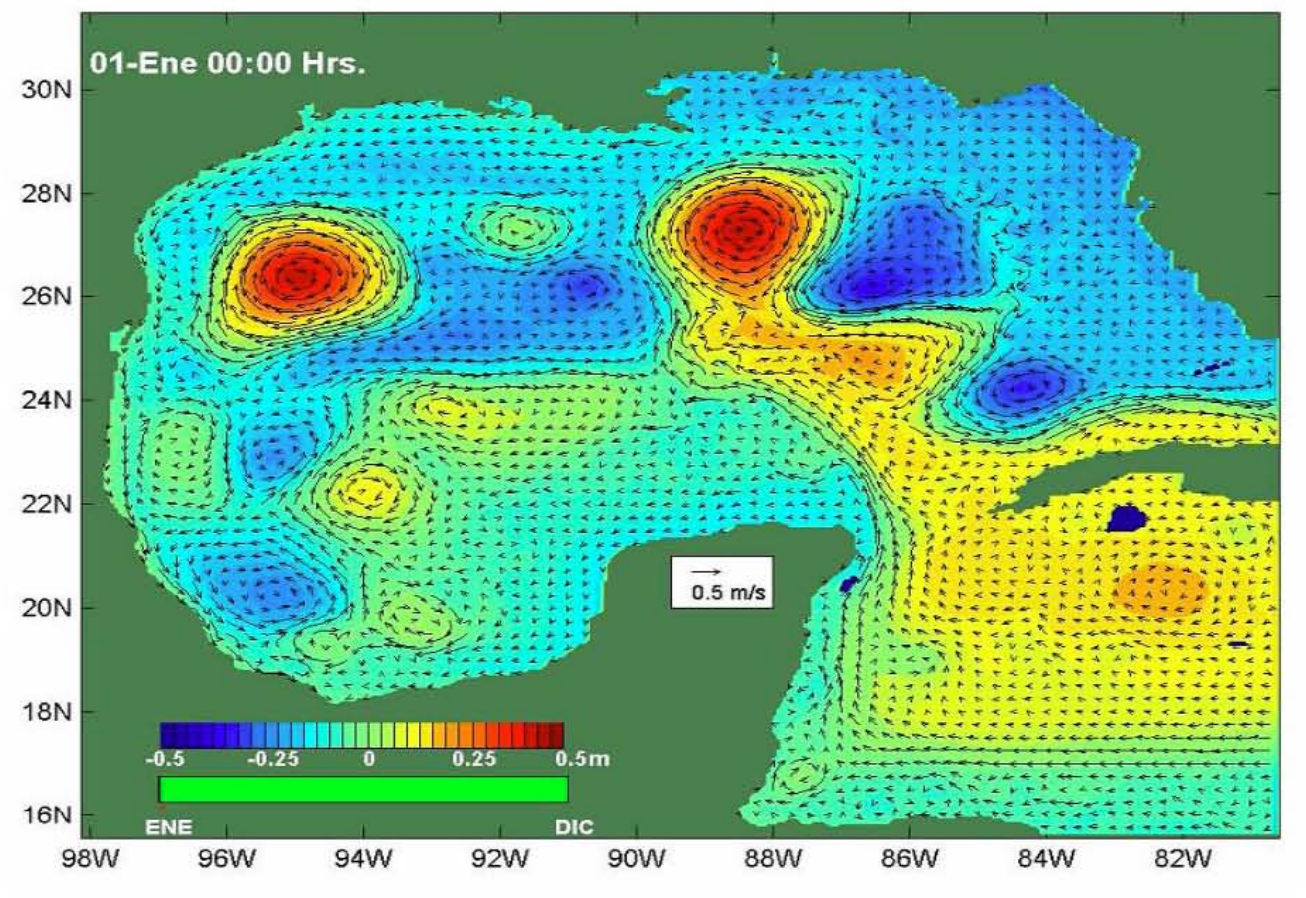


Figura 5.8.b Imagen del Golfo de México generando vectores cada 0.25 de grado.

b) ***Número de segmentos en cada vector***. Este número especifica el total de segmentos que forman el vector para representar el movimiento del flujo en un determinado tiempo (Figura 5.7.b). La formación de cada segmento se explicó en el capítulo 4.

c) ***Incremento de tiempo de cada segmento de vector***. Especifica el tiempo de recorrido que representa cada segmento de vector en la imagen. Esta propiedad está dada en horas (Figura 5.7.c).

Las propiedades (b) y (c) se pueden ejemplificar de la siguiente forma. Cada vector estará conformado por 8 segmentos (propiedad (b)) y el tamaño de cada segmento será el que represente el recorrido en horas (incremento en el tiempo). El tamaño del segmento se calcula usando la relación $d = \upsilon t$, donde d es la distancia o tamaño del

segmento, υ la velocidad en una zona y t el incrementado de tiempo, que en este ejemplo es de 6 horas (propiedad (c)), por lo que un vector completo mostrará la trayectoria recorrida durante 48 horas. (8 segmentos de 6 horas cada segmento) (Figura 5.9).

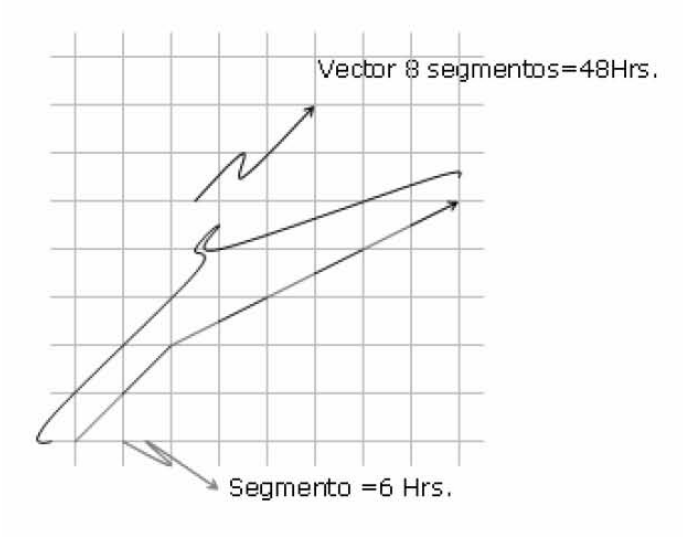


Figura 5.9 Diagrama de un vector de 8 seg., donde cada segmento representa 6 hrs.

2. ***Propiedades de los archivos***. Se encuentra en la parte superior derecha del menú. Tiene opciones para especificar las propiedades de lectura y las de escritura.

Imágenes y animaciones

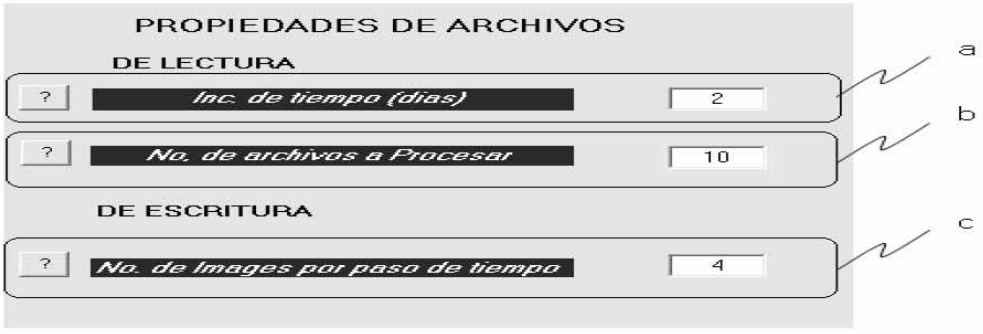


Figura 5.10 Área donde se proporcionan las propiedades de los archivos tanto de lectura como los de escritura.

Lectura

a) ***Incremento de tiempo***. Los datos generados por las simulaciones de los modelos están dados cada cierto tiempo, por ejemplo cada 6, 24, 48 hrs., por lo cual es necesario poner el incremento de tiempo de los archivos de lectura. Esta variable está dada en días (Figura 5.10.a).

b) ***Número de archivos a procesar***. Es el número de archivos que se leerán para generar un cierto número de imágenes de un período en particular (Figura 5.10.b).

Escritura

c) ***Número de imágenes por paso de tiempo***. Es la cantidad de imágenes que se generarán por cada archivo de datos que se lea. Esta propiedad tiene una relación directa con el incremento de tiempo y el número de segmentos de cada vector (Figura 5.10.c).

3. *Destinos*. Se encuentra en la parte inferior izquierda del menú y está dividido en destinos lectura y de escritura.

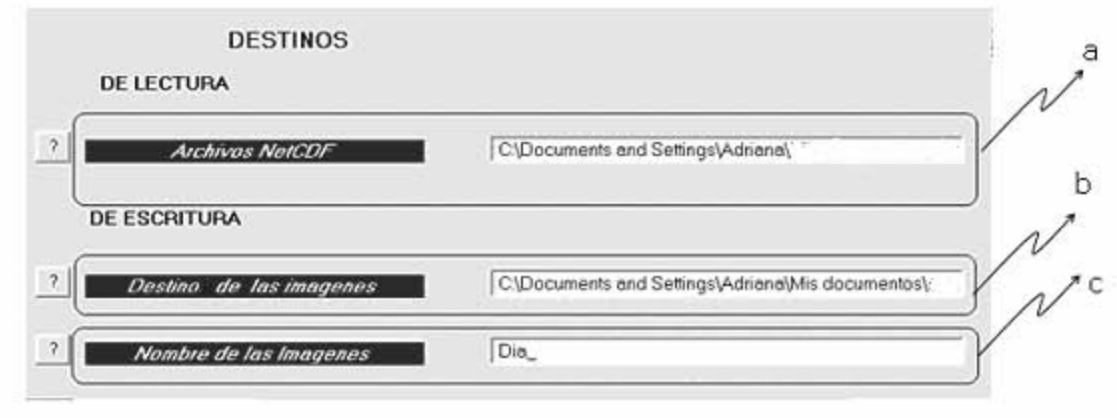


Figura 5.11 Área donde se proporcionas las rutas de los archivos que contienen la información y donde se colocaran las imágenes generadas.

Lectura

a) Se escribe la ruta donde se encuentran los archivos de datos de entrada. Por ahora acepta archivos en el formato NetCDF, que son los que contienen los datos con los cuales se generan las imágenes (Figura 5.11.a).

Escritura

b) ***Destino de las imágenes***. Se escribe el destino en donde se guardarán las imágenes generadas por el sistema (Figura 5.11.b).

c) ***Nombre de las imágenes***. Se añade un prefijo a las imágenes para que puedan ser distinguidas. El nombre esta dado por el sistema y esta formado por una letra que puede ser H, T, S seguida de la fecha y hora a la que corresponde la imagen. (p. ej. nMMDDHH, donde la n del inicio pertenece al tipo de datos, en este caso nivel del mar, MM al mes, DD al día, HH hora) (Figura 5.11.c).

4. En esta sección del menú se especifica el tipo de variable escalar que se va a graficar junto con el conjunto de circulación: nivel del mar, temperatura o salinidad, y ésta se encuentra en la parte inferior derecha (Figura 5.12).

Figura 5.12 Área de selección donde se especifica qué variable escalar se desea graficar.

Conclusiones

En este trabajo se desarrolló una aplicación que permite visualizar, mediante animaciones, las salidas de los modelos numéricos de circulación oceánica, con el objeto de apoyar la investigación y la divulgación de las simulaciones numéricas de los procesos físicos de los océanos.

La aplicación implementada permite visualizar las corrientes superficiales con las siguientes características:

El movimiento representado mediante flechas lagrangeanas corresponde a los desplazamientos reales de las corrientes, y su tamaño es directamente proporcional a la velocidad. Las características de las flechas nos permiten aparentar trayectorias curvas y son ajustables en tamaño, número de segmentos, densidad, grueso, color, tamaño de punta y ángulo de la punta, lo que permite lograr una visualización óptima de acuerdo con el tamaño del dominio, la escala de tiempo y el rango de velocidad.

Se crearon animaciones interpolando en tiempo los campos escalares y vectoriales para lograr una visualización continua y sin saltos en las secuencias, y para generar el número de cuadros adecuado para las escalas de tiempo del modelo y los procesos que se desea representar.

El número de cuadros por segundo en la animación se determina por medio del programa Adobe Premiere, dependiendo del formato deseado. Dentro de las animaciones se combinan gráficamente las variables escalares (temperatura, salinidad, nivel del mar) con las variables vectoriales que representa la corriente.

La aplicación aquí desarrollada apoya el estudio de procesos oceánicos, en particular en el Golfo de México, como son: la corriente del Lazo, los flujos a lo largo y perpendiculares a la plataforma continental, los frentes y remolinos oceánicos, entre otros.

Los trabajos en los cuales se está trabajando actualmente buscan desarrollar este tipo de visualización de 3D y que estas sean de manera interactiva y no solo en video. Esto se esta trabajando dentro del proyecto *Visualización avanzada para la enseñanza de los procesos de interacción océano atmósfera*, para la sala del Ixtli de la UNAM en el cual se generara material didáctico para la descripción de procesos de interacción océano atmósfera y de circulación oceánica. En particular se visualizará el fenómeno de El Niño mostrando la evolución de las capas superficiales del océano, la circulación atmosférica y la circulación tridimensional en el Golfo de México destacando la respuesta a eventos atmosféricos extremos.

APÉNDICE A

[A1] Cantidad Escalar

Una cantidad escalar es aquella que se puede expresar mediante un número. Los conjuntos de datos escalares contienen valores que se pueden distribuir en el tiempo y el espacio. Tales valores también pueden ser función de otros parámetros escalares. Ejemplos de cantidades físicas escalares: energía, densidad, masa, temperatura, presión, carga eléctrica, resistencia, reflectividad, frecuencia y volumen.

[A2] Cantidad Vectorial

Una cantidad vectorial (V) es aquella que se puede expresar mediante una magnitud y una dirección en el espacio. En un espacio tridimensional tiene tres valores escalares (V_x, V_y, V_z), uno para cada dirección de coordenada y un vector bidimensional tiene dos componentes (V_x, V_y). Las cantidades vectoriales pueden ser funciones de la posición, el tiempo y otros parámetros. Ejemplos de cantidades vectoriales son: velocidad, aceleración, fuerza, campos eléctricos, campos magnéticos, campos gravitacionales y corriente eléctrica.

[A3] Cantidad Tensorial

Una cantidad tensorial en un espacio tridimensional tiene nueve componentes y se puede representar con una matriz de 3 por 3. Esta representación se usa para un tensor de segundo orden y los tensores de orden más alto ocurren en algunas aplicaciones, en particular en la relatividad general. Ejemplos de cantidades tensoriales son: la tensión en un material que está sujeto a fuerzas externas y el tensor métrico, que ofrece las propiedades de un espacio de coordenadas en particular.

[A5] Teorema de muestreo de Nyquist-Shannon

El teorema de muestreo de Nyquist-Shannon es un teorema fundamental de la teoría de la información, especialmente útil en las telecomunicaciones. También se le conoce como teorema de muestreo de Whittaker-Nyquist-Kotelnikov-Shannon, o simplemente criterio de Nyquist.

Este teorema fue formulado como conjetura por primera vez por Harry Nyquist en 1928 ("Certain topics in telegraph transmission theory"), y fue probado formalmente por Claude E. Shannon en 1949 ("Communication in the presence of noise").

El teorema afirma que cuando se muestrea una señal, para poder reconstruir la señal original a partir de las muestras, la frecuencia de muestreo debe ser mayor que dos veces el ancho de banda de la señal de entrada. Si B es el ancho de banda de la señal y Fm es la frecuencia de muestreo, el teorema puede expresarse del siguiente modo:

$$2B < Fm$$

Hay que notar que el concepto de ancho de banda no necesariamente es sinónimo del valor de la frecuencia más alta en la señal de interés. Las señales para las cuales esto sí es cierto se les llama señales de banda base, y no todas las señales comparten tal característica (por ejemplo, las ondas de radio en frecuencia modulada). Si el criterio no es satisfecho, existirán frecuencias cuyo muestreo coincide con otras (el llamado aliasing).

[A6] Método de Runge-Kutta

Es un método de integración numérica para aproximar las soluciones de las ecuaciones diferenciales ordinarias sobre un intervalo dado. Este método utiliza un paso de prueba en el punto medio del intervalo para calcular los términos de error de orden bajo.

Descripción

Este método resuelve el problema de valores iniciales utilizando una función de paso que evalúa la derivada y0 = f (t; y) en distintos puntos (t; y).

En el método de Euler, que puede considerarse como el Runge –Kutta de primer orden, cada paso se mueve a lo largo de la tangente de una cierta curva que está "cerca" de la curva desconocida o buscada. El método de Runge-Kutta extiende esta idea geométrica al utilizar varias derivadas o tangentes intermedias, en lugar de solo una, para aproximar la función desconocida. El método de Runge-Kutta más simple se obtiene usando dos de estas derivadas intermedias.

El método de Euler consiste, gráficamente, en ir de un valor Y_n conocido de la solución de la ecuación diferencial (4.1) en un punto al siguiente por medio de la tangente *T1* a la curva integral $Y = f(X)$ en el mismo punto de la solución conocida, como se muestra en la Figura 4.2.

$$Y' = f(X,Y) \qquad (4.1)$$

De este planteamiento gráfico puede verse que una mejor aproximación a la solución de la ecuación diferencial se obtendría si en vez de ir por la tangente *T1* para determinar la solución en el siguiente *Punto Pivote*, se utiliza una secante con pendiente igual al promedio de pendientes de la curva integral en los puntos coordenados (X_n,Y_n), (X_{n+1},Y_{n+1}) en donde X_{n+1} y Y_{n+1} pueden estimarse con el procedimiento normal de Euler (Fig. 4.3).

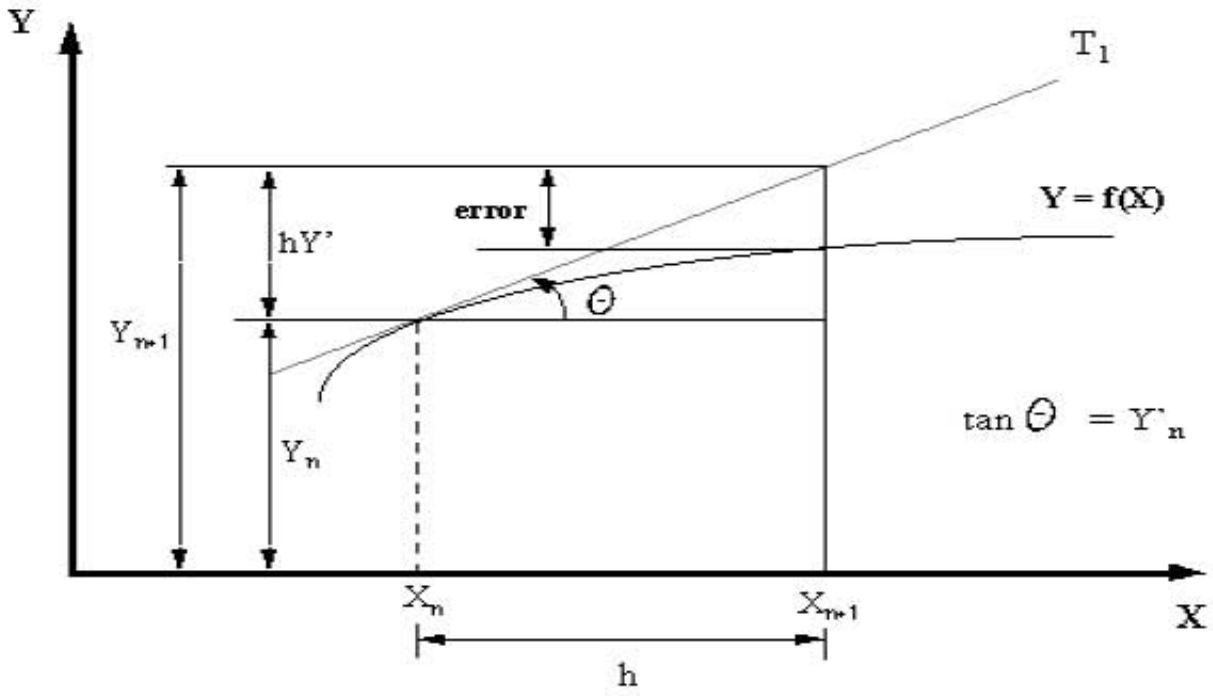


Figura 4.2 Método de Euler

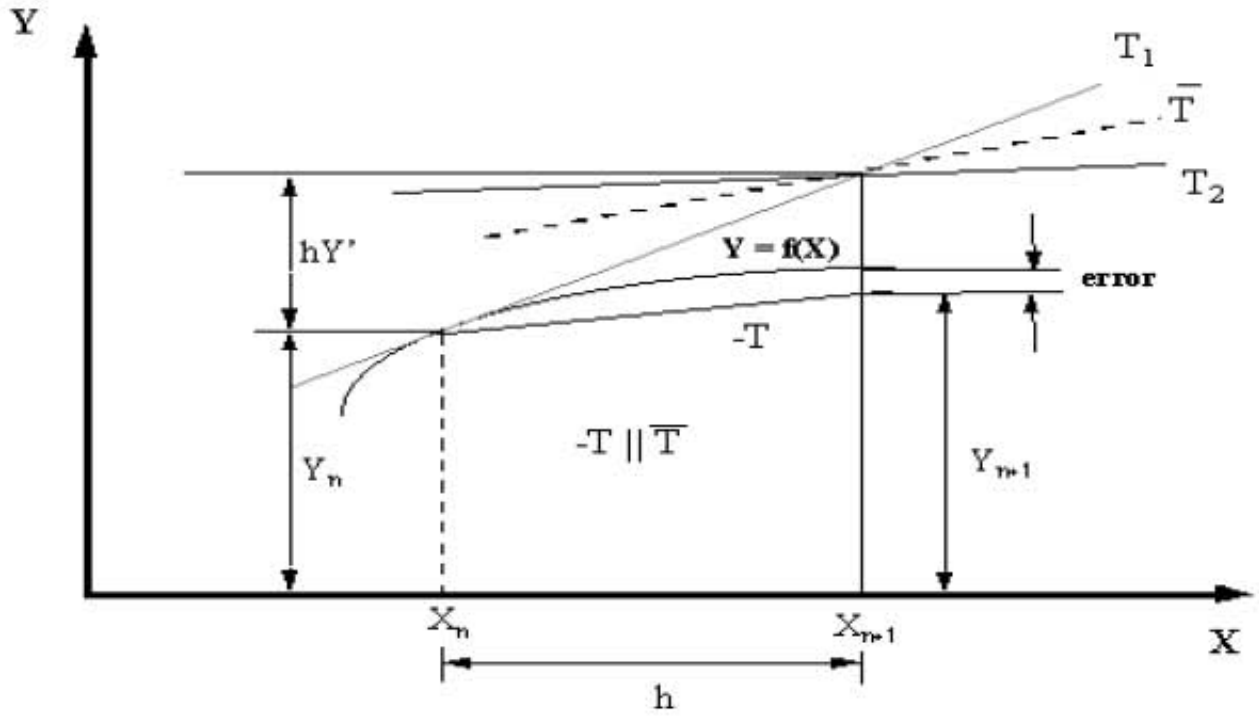


Figura 4.3 Método de Euler mejorado o Método de Runge-Kutta

Con lo anterior se obtiene un método mejorado de Euler o método de Runge-Kutta con error del orden de h^3 definido por la expresión

$$Y_{n+1} = Y_n + \tfrac{h}{2}(f(X_n, Y_n) + f(X_{n+1}, Y_{n+1})) \tag{4.2}$$

en donde $f(X_{n+1}, Y_{n+1})$ es el valor de la función $f(X, Y)$ para:

$$X = X_{n+1}$$

$$Y = Y_n + h\, f(X_n, Y_n)$$

Observando las expresiones para resolver la ecuación diferencial, puede decirse que ambas consisten en aplicar la fórmula de recurrencia

$$Y_{n+1} = Y_n + h\phi(X_n, Y_n) \tag{4.3}$$

en donde en el *método de Euler*

$$\phi(X, Y) = f(X, Y) \tag{4.4}$$

y en el *método de Runge-Kutta*

$$\phi(X, Y) = \tfrac{1}{2}(f(X, Y) + f(X + h, Y + h)) \tag{4.5}$$

en donde

$$Y' = f(X, Y) \tag{4.6}$$

Estas características dan origen a una gran variedad de métodos conocidos como métodos de *Runge-Kutta*. La diferencia entre ellos consiste en la forma como se define la función $\phi(X, Y)$ que aparece en la expresión (4.3).

[A7] Método de Euler

La solución de un problema de ecuaciones diferenciales con valores iniciales se obtiene generalmente paso a paso por métodos de integración hacia adelante, lo que permite valuar Y_{i+1} tan pronto se conozcan los valores Y_i, Y_{i-1} de *Y* en uno o más pivotes anteriores. El más simple de estos métodos, debido a *Euler*, es aplicable a ecuaciones de primer orden y no requiere conocer la solución en los pivotes anteriores.

Dado el problema de valores iniciales

$$\frac{dY}{dX} = f(X,Y), \qquad Y(X_0) = Y_0$$

se debe integrar la ecuación diferencial en el intervalo $X_i \le X \le X_{i+1} = X_i + h$ y evaluar la integral aplicando la fórmula de integración numérica:

$$\int_{X_i}^{X_{i+1}} \frac{dY}{dX} dX = \int_{X_i}^{X_{i+1}} f(X,Y) dX$$

$$Y\Big|_{X_i}^{X_{i+1}} = hf(X_i,Y_i) + O(h^2)$$

entonces

$$Y_{i+1} - Y_i = hf(X_i,Y_i) + O(h^2)$$

de donde se obtiene la siguiente expresión aproximada llamada *fórmula de Euler*

$$Y_{i+1} = Y_i + hf(X_i,Y_i) + O(h^2)$$

[A8] Descripción Lagrangiana y Euleriana

Existen básicamente dos formas de describir el movimiento de un fluido. La primera manera, llamada Lagrangiana, consiste en fijar la atención sobre una porción muy pequeña del fluido en movimiento. Por ejemplo, en el instante t=0 consideramos la

partícula que ocupa la posición 0. Nos interesa seguir esta partícula con movimiento constante, la cual ocupa un lugar en un tiempo t. El vector de posición depende de qué partícula se haya elegido y qué tiempo haya transcurrido.

En la descripción llamada Euleriana fijamos la atención en un punto (x, y, z) en el espacio. Nos interesa conocer las características del flujo, como velocidad, densidad, temperatura, etc., de las partículas que pasen por este punto como función del tiempo. (Nótese que no se está siguiendo una partícula como en la descripción Lagrangiana). Si se hace lo mismo para todos los puntos del espacio que ocupa el flujo, se tiene una descripción completa del flujo [*Sámano*, 2002].

[A9] Efecto de Coriolis

La rotación de la Tierra (hacia el Este), provoca que los cuerpos en movimiento (masas de aire, aguas oceánicas), se desvíen hacia la derecha en el hemisferio norte y hacia la izquierda en el hemisferio sur. La magnitud de la desviación depende de la velocidad del objeto y de su latitud. La desviación es cero en el Ecuador y es máxima en los polos. Los objetos que se mueven rápido se desvían más que los que se mueven lentamente. El efecto de Coriolis no tiene influencia en la energía del movimiento y modifica sólo la dirección.

[A10] Satélites meteorológicos

Los satélites meteorológicos pueden clasificarse en dos grandes grupos, de órbita polar o heliosincrónicos (significa que están sincronizados con el Sol) que como su nombre lo indica orbitan la Tierra de polo a polo y lo constituyen principalmente la serie TIROS de la agencia NOAA (National Oceanic and Atmospheric Administration) de origen norteamericano y los METEOR de origen ruso. El segundo grupo se compone de los satélites Geoestacionarios o Geosincrónicos (significa que están sincronizados con el movimiento de rotación de la Tierra), que orbitan a mayor altura y se encuentran sobre la línea del Ecuador.

La utilidad de los satélites meteorológicos es la de poder visualizar el conjunto Tierra-océano-atmósfera, y extraer la máxima información posible a través de distintas técnicas y procesos para obtener los productos cuyo objetivo se basa en el análisis cualitativo y cuantitativo de las imágenes obtenidas. Las imágenes de los satélites meteorológicos se utilizan principalmente para la visualización de nubes, clasificación, observación del vapor de agua existente en la alta y media atmósfera, temperaturas de la superficie de tierra y temperatura superficial del mar.

APÉNDICE B

FUNCION DE GRAFICADO DE LAS FLECHAS

```
function [NXN,NYN] = curquiver(X0,Y0,U,V,X1,Y1,MV,N,DT,C,N2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Versión 2.4
% Con el cambio en el tamaño de las flechas y la interpolación en los
% colores
% CURQUIVER2D FUNCTION para graficar vectores curvos en dos dimensiones
% curquiver2d (X,Y,U,V,X1,Y1,MV,N,DT,C,N2) Versión con cálculos de orden 2
%
% X0,Y0 son matrices o vectores con las coordenadas de los de los campos U y V,
% X1, Y1 son vectores o matrices con las coordenadas del inicio de los vectores,
% MV es 1 si X1, Y1 son matrices y 0 si X1, Y1 son vectores,
% N es el número de segmentos,
% DT es el incremento de tiempo de cada segmento que compone el vector,
% C es el color de los vectores.
% Si el vector se sale del dominio sólo se grafica el la fracción dentro del dominio
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

NN=N2
DeltaT = DT*3600;
alfa=pi/8;      % es el ángulo entre el
                   vector y cada punta de
                   flecha
TPunta=.10;
VecColor=C;
GruesoLinea=.2;
R=6371e+03;   % Radio medio de la
                   tierra (Gill)

DeltaTeta=(2*pi/360);
DelY=R*DeltaTeta;

for ii=1:length(Y0)
   iLat = Y0(ii);
   DelX(ii)=R*DeltaTeta*cos((iLat)*pi/180);
end

%%%%%%
[Sx Sy]=size(X0);
if Sx==1 | Sy==1
   [X,Y] = meshgrid(X0,Y0);
else
   X=X0;
   Y=Y0;
end
[Sx Sy]=size(X);

[Sx1 Sy1]=size(X1);

XX1=X1;
YY1=Y1;
[Sx1 Sy1]=size(XX1);

XN=XX1;
YN=YY1;

DeltaX = X(1,2) - X(1,1);   % supone matriz
                               uniforme
DeltaY = Y(2,1) - Y(1,1);

hold on
for ii=1:Sy1       % realiza los cálculos para
                      cada vector
   MX=[];
   MY=[];
   MX(1) = XX1(ii);
   MY(1) = YY1(ii);
   k=1;
   HayDatos=0;
   for g=1:N       % calcula para cada  segmento
     del vector
      J = 1+floor((MX(k) - X(1,1))/DeltaX);
      I = 1+floor((MY(k) - Y(1,1))/DeltaY);
      if I >= 1 & I < Sx & J >=1 & J < Sy
         %estacion3=1
         if (isnan(U(I,J))~=1 |
isnan(U(I+1,J))~=1) ...
               & (isnan(U(I,J+1))~=1 |
isnan(U(I+1,J+1))~=1)
            %estacion4=1
            if isnan(U(I,J))~=1, Uij=U(I,J);
Vij=V(I,J);
            else, Uij=0; Vij=0;
            end
```

```
            if isnan(U(I,J+1))~=1, Uij1=U(I,J+1);
Vij1=V(I,J+1);
            else, Uij1=0; Vij1=0;
            end
            if isnan(U(I+1,J))~=1, Ui1j=U(I+1,J);
Vi1j=V(I+1,J);
            else, Ui1j=0; Vi1j=0;
            end
            if isnan(U(I+1,J+1))~=1,
Ui1j1=U(I+1,J+1); Vi1j1=V(I+1,J+1);
            else, Ui1j1=0; Vi1j1=0;
            end
            HayDatos=1;
            IncX = (MX(k) - X(I,J))/DeltaX;
            IncY = (MY(k) - Y(I,J))/DeltaY;
            Ua = Uij + (Uij1 - Uij) * IncX;
            Ub = Ui1j + (Ui1j1 - Ui1j) * IncX;
            MU(k) = Ua + (Ub - Ua) * IncY;
            Va = Vij + (Vij1 - Vij) * IncX;
            Vb = Vi1j + (Vi1j1 - Vi1j) * IncX;
            MV(k) = Va + (Vb - Va) * IncY;

            MXInt = MX(k) +
MU(k)*DeltaT/(2*DelX(I));
% se ubica a     delta/2
            MYInt = MY(k) +
MV(k)*DeltaT/(2*DelY);
% se ubica a delta/2
            J = 1+floor((MXInt - X(1,1))/DeltaX);
            I = 1+floor((MYInt - Y(1,1))/DeltaY);
            if I >= 1 & I < Sx & J >=1 & J < Sy
               %estacion5=1
               if (isnan(U(I,J))~=1 |
isnan(U(I+1,J))~=1) ...
                     & (isnan(U(I,J+1))~=1 |
isnan(U(I+1,J+1))~=1);
                  %estacion6=1
                  IncX = (MXInt - X(I,J))/DeltaX;
                  IncY = (MYInt - Y(I,J))/DeltaY;

                  if isnan(U(I,J))~=1, Uij=U(I,J);
Vij=V(I,J);
                  else, Uij=0; Vij=0;
                  end
                  if isnan(U(I,J+1))~=1,
Uij1=U(I,J+1); Vij1=V(I,J+1);
                  else, Uij1=0; Vij1=0;
                  end
                  if isnan(U(I+1,J))~=1,
Ui1j=U(I+1,J); Vi1j=V(I+1,J);
                  else, Ui1j=0; Vi1j=0;
                  end
                     if isnan(U(I+1,J+1))~=1,
                  Ui1j1=U(I+1,J+1);
                  Vi1j1=V(I+1,J+1);
                  else, Ui1j1=0; Vi1j1=0;
                  end
                  Ua = Uij + (Uij1 - Uij) * IncX;
                  Ub = Ui1j + (Ui1j1 - Ui1j) * IncX;
                  MU(k) = Ua + (Ub - Ua) * IncY;
                  Va = Vij + (Vij1 - Vij) * IncX;
                  Vb = Vi1j + (Vi1j1 - Vi1j) * IncX;
                  MV(k) = Va + (Vb - Va) * IncY;

                  MX(k+1) = MX(k) +
MU(k)*DeltaT/(DelX(I));
                  MY(k+1) = MY(k) +
MV(k)*DeltaT/(DelY);
                  k=k+1;
               else
                  MX(k+1) = NaN;
                  MY(k+1) = NaN;
               end
            end
            if k>=N2 & MX ~=0 & MY ~=0
               NXN(ii)=MX(NN);        %(3);
               NYN(ii)=MY(NN);       %(3);
            elseif MX ~= 0 & MY ~=0
               NXN(ii)=MX(k);
               NYN(ii)=MY(k);
            end
         end
      end

      if k>=N
         XN(ii)=MX(NN);         % (3);
         YN(ii)=MY(NN);         %(3);
      else
         XN(ii)=MX(k);
         YN(ii)=MY(k);
      end
   end

plot(MX,MY,'color',VecColor,'LineWidth',GruesoLine
a);
   if k >= N & HayDatos==1
      u(1)=MX(end-1);
      v(1)=MY(end-1);
      u(2)=MX(end);
      v(2)=MY(end);
      w=u+v*j;
      teta2=atan2(v(2)-v(1),u(2)-u(1));
      w2=TPunta*exp(j*(teta2+alfa));
      w3=TPunta*exp(j*(teta2-alfa));
      w2b(1)=w(2);
      w3b(1)=w(2);
      w2b(2)=w(2)-w2;
      w3b(2)=w(2)-w3;

plot(w2b,'color',VecColor,'LineWidth',GruesoLinea);

plot(w3b,'color',VecColor,'LineWidth',GruesoLinea);
      MX(k+1)=MX(k);
      MY(k+1)=MY(k);
      clear w w2 w3 w2b w3b
   end
   end
hold off
```

PROGRAMA DEL MENU PRINCIPAL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Variables que proporcionará el usuario
% DVec  Densidad de los vectores en la imagen
% IncT   Incremento de tiempo de tiempo en los datos de forma decimal 1=24 Hrs.
% NArc  Número de dias que se desean procesar
% NSeg Número de segmentos que forman el vector
% DT    Es el incremento de tiempo de cada segmento que compone el  vector
% NIma  Número de imagenes que se generaran por archivo
% Rarch  Ruta donde se encuentran los archivos NetCDF
% Rimg   Ruta donde se Guardaran las imagenes generadas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function varargout = Vismenu(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @Vismenu_OpeningFcn, ...
   'gui_OutputFcn',  @Vismenu_OutputFcn, ...
   'gui_LayoutFcn',  [] , ...
   'gui_Callback',   []);

if nargin & isstr(varargin{1})
   gui_State.gui_Callback = 
str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = 
gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function Vismenu_OpeningFcn(hObject, eventdata, 
handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = Vismenu_OutputFcn(hObject, 
eventdata, handles)
varargout{1} = handles.output;

function edit1_CreateFcn(hObject, eventdata, 
handles)

if ispc
   set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit1_Callback(hObject, eventdata, 
handles)

function edit2_CreateFcn(hObject, eventdata, 
handles)

if ispc
   set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit2_Callback(hObject, eventdata, 
handles)

function edit3_CreateFcn(hObject, eventdata, 
handles)

if ispc
   set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit3_Callback(hObject, eventdata, 
handles)

function edit4_CreateFcn(hObject, eventdata, 
handles)

if ispc
   set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end
```

```
function edit4_Callback(hObject, eventdata,
handles)

function edit5_CreateFcn(hObject, eventdata,
handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata,
handles)

function edit6_CreateFcn(hObject, eventdata,
handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit6_Callback(hObject, eventdata,
handles)

function popupmenu1_CreateFcn(hObject,
eventdata, handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

% --------------------------------------------
function Untitled_1_Callback(hObject, eventdata,
handles)
%--------------------------------------------
function Untitled_2_Callback(hObject, eventdata,
handles)
% --------------------------------------------
function Untitled_3_Callback(hObject, eventdata,
handles)
% --------------------------------------------
function Untitled_4_Callback(hObject, eventdata,
handles)
% --------------------------------------------
function Untitled_5_Callback(hObject, eventdata,
handles)
% --------------------------------------------
function Untitled_6_Callback(hObject, eventdata,
handles)
% --------------------------------------------
function Untitled_7_Callback(hObject, eventdata,
handles)

function edit7_CreateFcn(hObject, eventdata,
handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit7_Callback(hObject, eventdata,
handles)

function edit8_CreateFcn(hObject, eventdata,
handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit8_Callback(hObject, eventdata,
handles)

function edit9_CreateFcn(hObject, eventdata,
handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit9_Callback(hObject, eventdata,
handles)

function pushbutton2_Callback(hObject,
eventdata, handles)

close MAPA %se cierra la ventana de la figura

function pushbutton3_Callback(hObject,
eventdata, handles)
close all
clear all
clc
```

```
function edit10_CreateFcn(hObject, eventdata,
handles)

if ispc
   set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicont
rolBackgroundColor'));
end

function edit10_Callback(hObject, eventdata,
handles)

function pushbutton4_Callback(hObject,
eventdata, handles)

function pushbutton5_Callback(hObject,
eventdata, handles)

function pushbutton6_Callback(hObject,
eventdata, handles)

function pushbutton7_Callback(hObject,
eventdata, handles)

function pushbutton8_Callback(hObject,
eventdata, handles)

function pushbutton9_Callback(hObject,
eventdata, handles)

function pushbutton10_Callback(hObject,
eventdata, handles)

function pushbutton11_Callback(hObject,
eventdata, handles)

function pushbutton12_Callback(hObject,
eventdata, handles)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% -------------------------------- Grafica del Nivel del Mar---------------------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function pushbutton13_Callback(hObject,
eventdata, handles)

DVec=str2num(get(handles.edit1,'String'));
IncT=str2num(get(handles.edit2,'String'));
NArc=str2num(get(handles.edit3,'String'));
NSeg=str2num(get(handles.edit5,'String'));
DT=str2num(get(handles.edit6,'String'));
NIma=str2num(get(handles.edit7,'String'));
Rarch=get(handles.edit8,'String');
Rimg= get(handles.edit9,'String');
Nimg= get(handles.edit10,'String');

MAPA=figure;

begin=365*3+5;
i1=0;
Time1=0;
Color='k';    %Color de los vectores
MV=0;
% MV es 1 si X1, Y1 son matrices y 0 si X1,
    Y1 son vectores
XD=-98:DVec:-80;
%Puntos para generar las flechas
YD=15+DVec:DVec:31;
%Puntos para generar las flechas

C=-.5:.025:.5;      %Se definen los
                      contornos
colormap(jet(40))  %Se define la barra de
                     color y cuantos
                     elementos va a
                     contener

NumMes = [31 28 31 30 31 30 31 31 30 31 30
31];
NumMes = [31 59 90 120 151 181 212 243 273
304 334 365];

load GoM_Bat  %Lectura de los datos de
                batimetría

%%%%Eliminando los datos de tierra%%%
[I J] = find(Z3>=10);

for i=1:length(I)
   Z3(I(i),J(i)) = NaN;
end

Unit = Z3./Z3;   %Se genera una matriz de
                   unos donde es la región
                   del  mar

XCA0=XD;
YCA0=YD;

[XXD YYD]=meshgrid (XD, YD);

[XS YS]=size (XXD);

kk=1;
for ii=1:XS
   for jj=1:YS
      XXM(kk) =XXD(ii,jj);
      YYM(kk) =YYD(ii,jj);
      kk=kk+1;
```

```
   end
end

clear XS YS;

XD=XXM;
YD=YYM;

for jj=1:IncT:NArc        %iter1 lee cada 2 dias

Nombre=dateNetCDF(jj)

   nc = netcdf([Rarch,num2str(Nombre),'.nc'],
'nowrite');  % Abrir Archivo NetCDF

   variables = var(nc);

   U=nc {'u'}(:);
   V=nc {'v'}(:);
   H=nc{'nivel_mar'}(:);
   X=nc{'longitud'}(:);
   Y=nc{'latitud'}(:);

   nc = close(nc);

% para interpolar los colores
% Apetura del segundo archivo

   Nombre2=dateNetCDF(jj+IncT);
   nc = netcdf([Rarch,num2str(Nombre2),'.nc'],
'nowrite');   % Abrir Archivo NetCDF

   variables = var(nc);

   Uu=nc {'u'}(:);
   Vv=nc {'v'}(:);
   Hh=nc{'nivel_mar'}(:);

   nc = close(nc);

   hold on

   for t=1:NIma

      tiem=IncT/NIma;
       TH=tiem*24;
       N2=TH/DT;

   % Interpolacion de los datos de H , V y U

      for ti=1:320
         for tj=1:352

            DeltaH = Hh(ti,tj) - H(ti,tj);
            NewH(ti,tj) = H(ti,tj) + DeltaH * (((t-
1)*TH)/(IncT*24));

            DeltaU = Uu(ti,tj) - U(ti,tj);
            NewU(ti,tj) = U(ti,tj) + DeltaU * (((t-
1)*TH)/(IncT*24));

            DeltaV = Vv(ti,tj) - V(ti,tj);
            NewV(ti,tj) = V(ti,tj) + DeltaV * (((t-
1)*TH)/(IncT*24));

         end
      end

      [QX QY]=QuitaTierra(U,V,X,Y,XD,YD);

      [XA]=find (QX~=0 );
      [YA]=find (QY~=0 );
      for i=1:length(XA)
         Ii=XA(i);
         NX(i)=QX(Ii);
         NY(i)=QY(Ii);

      end
      XD=NX;
      YD=NY;
      clf

      contourf(X,Y,NewH.*Unit,C), shading flat;

      [NXN
NYN]=curquiver2d(X,Y,NewU(:,:),NewV(:,:),XD,Y,
MV,NSeg,DT,Color,N2);

      XD = [NXN];
      YD = [NYN];

      clear XA YA NY NX;

      DeltaNG=1;
      NumSj=(abs(-98-(-80)))/DeltaNG;
      NumSi=(abs(15-31))/DeltaNG;
      lon0=-98;
      lat0=15;
      Rsol=20*DeltaNG;
      Max=lat0:DVec:lat0+DeltaNG;
      NMax=((length(Max))^2)+1;
      NMin=NMax-3;

      for tti=1:NumSi
         for ttj=1:NumSj
% Punto inicial para las coordenadas en x
            i2 = (tti-1)*Rsol+1;
% Punto final para las coordenadas en x
            i3 = (tti)*Rsol;
% Punto inicial para las coordenadas en y
           j2 = (ttj-1)*Rsol+1;
% Punto final para las coordenadas en y
           j3 = (ttj)*Rsol;

            if i3 > 320
               i3 = 320;
            end
```

```
            if j3 > 352
               j3 = 352;
            end
            [I J]=find (isnan (Unit (i2:i3,
j2:j3))==0 );

            TT(tti, ttj)=length(I)/(Rsol^2);
%Matriz con el porcentaje de mar en cada zona

            I=find (NXN >= lon0+DeltaNG*ttj-
DeltaNG & NXN < lon0+DeltaNG*ttj & ...
               NYN >= lat0+DeltaNG*tti-DeltaNG &
NYN < lat0+DeltaNG*tti);

            II=length(I);
% Número de vectores encontrados por zona

            if TT(tti,ttj) ~= 0

               I2=ceil(TT(tti,ttj)*NMax);
% Número de vectores teórico máximo por zona
               I3=ceil(TT(tti,ttj)*NMin);
% Número de vectores teórico mínimo por zona

               CX=XD;
               CY=YD;

               if TT(tti,ttj) < .1 & II < I3


         %%%AGREGAR VECTORES%%%
               I5=ceil(TT(tti,ttj)*NMax);
%Número de vectores real permitido por zona
               I4=ceil (abs((I5-II)/TT(tti,ttj)));
%Número de vectores que se insertaran
               if I4 > NMax
                  I4 =round( NMax/2);
               end

               if II<I4

                  if ttj < NumSj
                     a = 1;
                  else
                     a = 0.4;
                  end

                  for ka=1:I4

                  Cx=-98+ttj-1+(rand*a);

                  CX(length(CX)+1)=Cx;
                     Cy=15+tti-1+(rand);
                     CY(length(CY)+1)=Cy;
                     XD=CX;
                     YD=CY;
                  end
               end
            else

               I2=ceil(TT(tti,ttj)*NMax);
% Número de vectores teórico máximo por zona
               I3=ceil(TT(tti,ttj)*NMin);
% Número de vectores teórico mínimo por zona

               if II > I2 & II > (2*NMax/3)
         %%%REMOVER VECTORES%%%
                  I6=abs(I2-II);
                  Q=randperm(II);
                  for kq=1:I6
                     Q1=Q(kq);
                     Q2=I(Q1);
                     CX(Q2)=[];
                     CY(Q2)=[];
                     XD=CX;
                     YD=CY;
                  end
               end

               if II < I3

         %%%AGREGAR VECTORES%%%
                  I5=ceil(TT(tti,ttj)*NMin);
%Número de vectores real permitido por zona
                  I4=ceil(abs((I5-II)/TT(tti,ttj)));
%Número de vectores que se insertaran

                  if I4 < NMin

                     if ttj < NumSj
                        a = 1;
                     else
                        a = 0.4;
                     end

                     for ka=1:I4

                     Cx=-98+ttj-1+(rand*a);

                      CX(length(CX)+1)=Cx;
                        Cy=15+tti-1+(rand);
                        CY(length(CY)+1)=Cy;
                        XD=CX;
                        YD=CY;
                     end
                  end
               end
            end
         end
      end

EjeX=[-98 -96 -94 -92 -90 -88 -86 -84 -82];
EjeY=[16 18 20 22 24 26 28 30];
      set(gca,'XTick',EjeX);
```

```
set(gca,'XTicklabel',['98W';'96W';'94W';'92W';'90W
';...

'88W';'86W';'84W';'82W'],'FontSize',10);
      set(gca,'YTick',EjeY);

set(gca,'YTicklabel',['16N';'18N';'20N';'22N';'24N';'
26N';'28N';...
            '30N'],'FontSize',10);

      NumF=num2str(t); %incremento del
                        número de la
                        figura

      dia=jj+t*tiem-tiem;
      dia1=datestrad(dia,0);

      StrTitle = [num2str(dia1),' Hrs.'];
% titulo de la imagen
      text(-98,30.5, StrTitle, 'color', 'w',
'FontName', 'Palatino','FontWeight', 'bold',
'FontSize',12);

%%%%Escala del vector para la velocidad

      u2(1:21,1:21)=.5;
      v2(1:21,1:21)=0;
      x2=-89.5:0.05:-88.5;
      y2=20:0.05:21;
      mx0=-89.2;
      my0=20.7;

      hold on
      xf=[-89.5 -88 -88 -89.5 -89.5]; yf=[20 20 21
21 20];
      fill(xf,yf,'w')

      curquiver2d
(x2,y2,u2,v2,mx0,my0,MV,NSeg,DT,Color,N2);
      text(-89.2,20.3,'0.5 m/s' ,'Color' ,'k',
'FontSize', 8)
      caxis([-.5 .5])
      set(gca,'color',[.3 .5 .3])
      set(gcf,'color','w')
      set(gcf,'InvertHardCopy','off');
      hold on
      xf=[-97 -91 -91 -97 -97]; yf=[16.25 16.25
16.75 16.75 16.25];
      fill(xf,yf,'g')
      x=-97+(dia*0.0164);
      xf=[-97 x x -97 -97]; yf=[16.25 16.25 16.75
16.75 16.25];
      fill(xf,yf,'b')
      text(-
97.2,15.75,'ENE','color','w','FontWeight', 'bold',
'FontSize',9);
      text(-91.2,15.75, 'DIC', 'color', 'w',
'FontWeight', 'bold','FontSize',9);
      text(x,16.1,'^','color','r','FontSize',7);

      text(-97.2,17.1,'-0.5','color','w','FontWeight',
'bold','FontSize',9);
      text(-95.7,17.1,'-
0.25','color','w','FontWeight', 'bold','FontSize',9);
      text(-94.0,17.1,'0','color','w','FontWeight',
'bold ','FontSize',9);
      text(-92.7,17.1,'0.25','color','w','FontWeight',
'bold', 'FontSize',9);
      text(-91.2,17.1,'0.5','color','w','FontWeight',
'bold ','FontSize',9);
      text(-90.7,17.1,'m','color','w','FontWeight',
'bold ','FontSize',9);
      axes('position',[.182  .2  .26  .025])
      pcolor([1:32;1:32])
      axis off

      Fecha=dateNetCDF(dia)
      nom=['H',num2str(Fecha)];
      grafica=[Rimg,Nimg,nom];
      print('-djpeg',grafica);
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ------------------------------ Gráfica de temperatura del mar --------------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton14_Callback(hObject,
eventdata, handles)

DVec=str2num(get(handles.edit1,'String'));
IncT=str2num(get(handles.edit2,'String'));
NArc=str2num(get(handles.edit3,'String'));
NSeg=str2num(get(handles.edit5,'String'));
DT=str2num(get(handles.edit6,'String'));
NIma=str2num(get(handles.edit7,'String'));
Rarch=get(handles.edit8,'String');
Rimg= get(handles.edit9,'String');
Nimg= get(handles.edit10,'String');

MAPA=figure;

begin=365*3+5;
i1=0;
Time1=0;

Color='w';    %Color de los vectores
MV=0;         % MV es 1 si X1, Y1 son matrices
                 y 0 si X1, Y1 son vectores
```

```
XD=-98:DVec:-80;    %Puntos para generar
                      las  flechas
YD=15+DVec:DVec:31;  %Puntos para
                        generar las flechas

C=15:.25:30;        %Se definen los contornos
colormap(jet(40))   %Se define la barra de
                      color y cuantos
                      elementos
                      va a contener
NumMes = [31 28 31 30 31 30 31 31 30 31 30
31];
NumMes = [31 59 90 120 151 181 212 243 273
304 334 365];

load GoM_Bat   %Lectura de los datos de
                  batometría

%%%%Eliminando los datos de tierra%%%
[I J] = find(Z3>=10);

for i=1:length(I)
   Z3(I(i),J(i)) = NaN;
end

Unit = Z3./Z3;
%Se genera una matriz de unos donde es la
región del mar

XCA0=XD;
YCA0=YD;

[XXD YYD]=meshgrid (XD, YD);

[XS YS]=size (XXD);

kk=1;
for ii=1:XS
   for jj=1:YS
      XXM(kk) =XXD(ii,jj);
      YYM(kk) =YYD(ii,jj);
      kk=kk+1;
   end
end

clear XS YS;

XD=XXM;
YD=YYM;

for jj=1:IncT:NArc        %iter1 lee cada 2 días

Nombre=dateNetCDF(jj)

   nc = netcdf([Rarch,num2str(Nombre),'.nc'],
'nowrite');        % Abrir Archivo NetCDF

   variables = var(nc);

   U=nc {'u'}(:);
   V=nc {'v'}(:);
   Tmp=nc{'temperatura'}(:);
   X=nc{'longitud'}(:);
   Y=nc{'latitud'}(:);

   nc = close(nc);

   % para interpolar los colores
   % Apetura del segundo archivo

   Nombre2=dateNetCDF(jj+IncT);
   nc = netcdf([Rarch,num2str(Nombre2),'.nc'],
'nowrite');   % Abrir Archivo NetCDF

   variables = var(nc);

   Uu=nc {'u'}(:);
   Vv=nc {'v'}(:);
   Tmpt=nc{'temperatura'}(:);

   nc = close(nc);

   hold on

   for t=1:NIma

      tiem=IncT/NIma;
       TH=tiem*24;
       N2=TH/DT;

    % Interpolacion de los datos de H , V y U

      for ti=1:320
         for tj=1:352

            DeltaTmp = Tmpt(ti,tj) - Tmp(ti,tj);
            NewTmp(ti,tj) = Tmp(ti,tj) + DeltaTmp
* (((t-1)*TH)/(IncT*24));

            DeltaU = Uu(ti,tj) - U(ti,tj);
            NewU(ti,tj) = U(ti,tj) + DeltaU * (((t-
1)*TH)/(IncT*24));

            DeltaV = Vv(ti,tj) - V(ti,tj);
            NewV(ti,tj) = V(ti,tj) + DeltaV * (((t-
1)*TH)/(IncT*24));

         end
      end

      [QX QY]=QuitaTierra(U,V,X,Y,XD,YD);

      [XA]=find (QX~=0 );
      [YA]=find (QY~=0 );
      for i=1:length(XA)
         Ii=XA(i);
         NX(i)=QX(Ii);
```

```
        NY(i)=QY(Ii);

    end
    XD=NX;
    YD=NY;
    clf

    contourf(X,Y,NewTmp.*Unit,C), shading flat;

    [NXN
NYN]=curquiver2d(X,Y,NewU(:,:),NewV(:,:),XD,Y,
MV,NSeg,DT,Color,N2);

    XD = [NXN];
    YD = [NYN];

    clear XA YA NY NX;

    DeltaNG=1;
    NumSj=(abs(-98-(-80)))/DeltaNG;
    NumSi=(abs(15-31))/DeltaNG;
    lon0=-98;
    lat0=15;
    Rsol=20*DeltaNG;
    Max=lat0:DVec:lat0+DeltaNG;
    NMax=((length(Max))^2)+1;
    NMin=NMax-3;

    for tti=1:NumSi
       for ttj=1:NumSj
          i2 = (tti-1)*Rsol+1;
% Punto inicial para las coordenadas en x
          i3 = (tti)*Rsol;
% Punto final para las coordenadas en x
          j2 = (ttj-1)*Rsol+1;
% Punto inicial para las coordenadas en y
          j3 = (ttj)*Rsol;
 % Punto final para las coordenadas en y

          if i3 > 320
             i3 = 320;
          end
          if j3 > 352
             j3 = 352;
          end
          [I J]=find (isnan (Unit (i2:i3,
j2:j3))==0 );

          TT(tti, ttj)=length(I)/(Rsol^2);
%Matriz con el porcentaje de mar en cada zona

          I=find (NXN >= lon0+DeltaNG*ttj-
DeltaNG & NXN < lon0+DeltaNG*ttj & ...
             NYN >= lat0+DeltaNG*tti-DeltaNG &
NYN < lat0+DeltaNG*tti);

          II=length(I);
 % Número de vectores encontrados por zona
                    if TT(tti,ttj) ~= 0

          I2=ceil(TT(tti,ttj)*NMax);    %
Número de vectores teórico máximo por zona
          I3=ceil(TT(tti,ttj)*NMin);    %
Número de vectores teórico mínimo por zona

          CX=XD;
          CY=YD;

          if TT(tti,ttj) < .1 & II < I3

    %%%AGREGAR VECTORES%%%
             I5=ceil(TT(tti,ttj)*NMax);
%Número de vectores real permitido por zona
             I4=ceil (abs((I5-II)/TT(tti,ttj)));
%Número de vectores que se insertaran
             if I4 > NMax
                I4 =round( NMax/2);
             end

             if II<I4

                if ttj < NumSj
                   a = 1;
                else
                   a = 0.4;
                end

                for ka=1:I4

                   Cx=-98+ttj-1+(rand*a);
                   CX(length(CX)+1)=Cx;
                   Cy=15+tti-1+(rand);
                   CY(length(CY)+1)=Cy;
                   XD=CX;
                   YD=CY;
                end
             end

          else

             I2=ceil(TT(tti,ttj)*NMax);
%Número de vectores teórico máximo por zona
             I3=ceil(TT(tti,ttj)*NMin);
% Número de vectores teórico mínimo por zona

             if II > I2 & II > (2*NMax/3)
       %%%REMOVER VECTORES%%%
                I6=abs(I2-II);
                Q=randperm(II);
                for kq=1:I6

                   Q1=Q(kq);
                   Q2=I(Q1);
                   CX(Q2)=[];
                   CY(Q2)=[];
                   XD=CX;
                   YD=CY;
                end
```

```
                end

                if II < I3
      %%%AGREGAR VECTORES%%%
                  I5=ceil(TT(tti,ttj)*NMin);
%Número de vectores real permitido por zona
                  I4=ceil(abs((I5-II)/TT(tti,ttj)));
 %Número de vectores que se insertaran

                  if I4 < NMin

                    if ttj < NumSj
                      a = 1;
                    else
                      a = 0.4;
                    end

                    for ka=1:I4

                      Cx=-98+ttj-1+(rand*a);
                      CX(length(CX)+1)=Cx;
                      Cy=15+tti-1+(rand);
                      CY(length(CY)+1)=Cy;
                      XD=CX;
                      YD=CY;
                    end
                  end
                end
              end
            end
          end

      EjeX=[-98 -96 -94 -92 -90 -88 -86 -84 -82];
      EjeY=[16 18 20 22 24 26 28 30];
      set(gca,'XTick',EjeX);

set(gca,'XTicklabel',['98W';'96W';'94W';'92W';'90W
';...
 '88W';'86W';'84W';'82W'],'FontSize',10);
      set(gca,'YTick',EjeY);

set(gca,'YTicklabel',['16N';'18N';'20N';'22N';'24N';'
26N';'28N';...
            '30N'],'FontSize',10);

      NumF=num2str(t);   % incremento del
                           número de la figura

      dia=jj+t*tiem-tiem;
      dia1=datestrad(dia,0);

      StrTitle = [num2str(dia1),' Hrs.'];      %
titulo de la imagen
      text(-98,30.5,StrTitle,'color','w','FontName',
'Palatino','FontWeight', 'bold ','FontSize',12);

       %%Escala del vector para la velocidad
      u2(1:21,1:21)=.5;
      v2(1:21,1:21)=0;
      x2=-89.5:0.05:-88.5;
      y2=20:0.05:21;
      mx0=-89.2;
      my0=20.7;

      hold on
      xf=[-89.5 -88 -88 -89.5 -89.5]; yf=[20 20 21
21 20];
      fill(xf,yf,'w')

      curquiver2d
(x2,y2,u2,v2,mx0,my0,MV,NSeg,DT,Color,N2);
      text(-89.2,20.3,'0.5
m/s','Color','k','FontSize',8)

      set(gca,'color',[.3 .5 .3])
      set(gcf,'color','w')
      set(gcf,'InvertHardCopy','off');
      hold on
      xf=[-97 -91 -91 -97 -97]; yf=[16.25 16.25
16.75 16.75 16.25];
      fill(xf,yf,'g')
      x=-97+(dia*0.0164);
      xf=[-97 x x -97 -97]; yf=[16.25 16.25 16.75
16.75 16.25];
      fill(xf,yf,'b')
      text(-
97.2,15.75,'ENE','color','w','FontWeight', 'bold
','FontSize',9);
      text(-91.2,15.75,'DIC','color','w','FontWeight',
'bold ','FontSize',9);
      text(x,16.1,'^','color','r','FontSize',7);

      text(-97.2,17.1,'15','color','w','FontWeight',
'bold ','FontSize',9);
      text(-
95.7,17.1,'18.75','color','w','FontWeight', 'bold
','FontSize',8);
      text(-94.0,17.1,'22.5','color','w','FontWeight',
'bold ','FontSize',8);
      text(-
92.7,17.1,'26.25','color','w','FontWeight', 'bold
','FontSize',8);
      text(-91.2,17.1,'30','color','w','FontWeight',
'bold ','FontSize',9);
      text(-90.7,17.1,'ºC','color','w','FontWeight',
'bold ','FontSize',9);
      axes('position',[.182  .2  .26  .025])
      pcolor([1:32;1:32])
      axis off

      Fecha=dateNetCDF(dia)
      nom=['T',num2str(Fecha)];
      grafica=[Rimg,Nimg,nom];
      print('-djpeg',grafica);
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ------------------------------- Grafica de la salinidad del mar ---------------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton15_Callback(hObject,
eventdata, handles)

DVec=str2num(get(handles.edit1,'String'));
IncT=str2num(get(handles.edit2,'String'));
NArc=str2num(get(handles.edit3,'String'));
NSeg=str2num(get(handles.edit5,'String'));
DT=str2num(get(handles.edit6,'String'));
NIma=str2num(get(handles.edit7,'String'));
Rarch=get(handles.edit8,'String');
Rimg= get(handles.edit9,'String');
Nimg= get(handles.edit10,'String');

MAPA=figure;

begin=365*3+5;
i1=0;
Time1=0;

Color='w';    %Color de los vectores
MV=0;         % MV es 1 si X1, Y1 son
                matrices y 0 si X1, Y1 son
                vectores

XD=-98:DVec:-80;    %Puntos para generar
                      las flechas
YD=15+DVec:DVec:31; %Puntos para generar
                      las flechas

C=30:.15:37;        %Se definen los contornos
colormap(jet(40))   %Se define la barra de
                     color y cuantos elementos
                     va a contener

NumMes = [31 28 31 30 31 30 31 31 30 31 30
31];
NumMes = [31 59 90 120 151 181 212 243 273
304 334 365];

load GoM_Bat     % Lectura de los datos de
                   batometría

%%%Eliminando los datos de tierra %%%
[I J] = find(Z3>=10);

for i=1:length(I)
   Z3(I(i),J(i)) = NaN;
end

Unit = Z3./Z3;            %Se genera una matriz de
unos donde es la región del mar

XCA0=XD;
YCA0=YD;

[XXD YYD]=meshgrid (XD, YD);

[XS YS]=size (XXD);

kk=1;
for ii=1:XS
   for jj=1:YS
      XXM(kk) =XXD(ii,jj);
      YYM(kk) =YYD(ii,jj);
      kk=kk+1;
   end
end

clear XS YS;

XD=XXM;
YD=YYM;

for jj=1:IncT:NArc        %iter1 lee cada 2 dias

Nombre=dateNetCDF(jj)

   nc = netcdf([Rarch,num2str(Nombre),'.nc'],
'nowrite');        % Abrir Archivo NetCDF

   variables = var(nc);

   U=nc {'u'}(:);
   V=nc {'v'}(:);
   Sal=nc{'salinidad'}(:);
   X=nc{'longitud'}(:);
   Y=nc{'latitud'}(:);

   nc = close(nc);

   % para interpolar los colores
   % Apertura del segundo archivo

   Nombre2=dateNetCDF(jj+IncT);
   nc = netcdf([Rarch,num2str(Nombre2),'.nc'],
'nowrite');   % Abrir Archivo NetCDF

   variables = var(nc);

   Uu=nc {'u'}(:);
   Vv=nc {'v'}(:);
   Sals=nc{'salinidad'}(:);

   nc = close(nc);

   hold on

   for t=1:NIma
```

```
tiem=IncT/NIma;
 TH=tiem*24;
 N2=TH/DT;

%% Interpolación de los datos de H , V y U

for ti=1:320
   for tj=1:352

      DeltaSal = Sals(ti,tj) - Sal(ti,tj);
      NewS(ti,tj) = Sal(ti,tj) + DeltaSal *
(((t-1)*TH)/(IncT*24));

      DeltaU = Uu(ti,tj) - U(ti,tj);
      NewU(ti,tj) = U(ti,tj) + DeltaU * (((t-
1)*TH)/(IncT*24));

      DeltaV = Vv(ti,tj) - V(ti,tj);
      NewV(ti,tj) = V(ti,tj) + DeltaV * (((t-
1)*TH)/(IncT*24));

   end
end

[QX QY]=QuitaTierra(U,V,X,Y,XD,YD);

[XA]=find (QX~=0 );
[YA]=find (QY~=0 );
for i=1:length(XA)
   Ii=XA(i);
   NX(i)=QX(Ii);
   NY(i)=QY(Ii);

end
XD=NX;
YD=NY;
clf

contourf(X,Y,NewS.*Unit,C), shading flat;

[NXN
NYN]=curquiver2d(X,Y,NewU(:,:),NewV(:,:),XD,YD
,MV,NSeg,DT,Color,N2);

XD = [NXN];
YD = [NYN];

clear XA YA NY NX;

DeltaNG=1;
NumSj=(abs(-98-(-80)))/DeltaNG;
NumSi=(abs(15-31))/DeltaNG;
lon0=-98;
lat0=15;
Rsol=20*DeltaNG;
Max=lat0:DVec:lat0+DeltaNG;
NMax=((length(Max))^2)+1;
NMin=NMax-3;

for tti=1:NumSi
   for ttj=1:NumSj
      i2 = (tti-1)*Rsol+1;
% Punto inicial para las coordenadas en x
      i3 = (tti)*Rsol;
% Punto final para las coordenadas en x
      j2 = (ttj-1)*Rsol+1;
% Punto inicial para las coordenadas en y
      j3 = (ttj)*Rsol;
% Punto final para las coordenadas en y

      if i3 > 320
         i3 = 320;
      end
      if j3 > 352
         j3 = 352;
      end
      [I J]=find (isnan (Unit (i2:i3,
j2:j3))==0 );

      TT(tti, ttj)=length(I)/(Rsol^2);
%Matriz con el porcentaje de mar en cada zona

      I=find (NXN >= lon0+DeltaNG*ttj-
DeltaNG & NXN < lon0+DeltaNG*ttj & ...
         NYN >= lat0+DeltaNG*tti-DeltaNG &
NYN < lat0+DeltaNG*tti);

      II=length(I);
% Número de vectores encontrados por zona

      if TT(tti,ttj) ~= 0

         I2=ceil(TT(tti,ttj)*NMax);
%Número de vectores teórico máximo por zona
         I3=ceil(TT(tti,ttj)*NMin);
%Número de vectores teórico mínimo por zona

         CX=XD;
         CY=YD;

         if TT(tti,ttj) < .1 & II < I3

            %%%AGREGAR VECTORES%%%
            I5=ceil(TT(tti,ttj)*NMax);
%Número de vectores real permitido por zona
            I4=ceil (abs((I5-II)/TT(tti,ttj)));
%Número de vectores que se insertaran
            if I4 > NMax
               I4 =round( NMax/2);
            end

            if II<I4

               if ttj < NumSj
                  a = 1;
               else
                  a = 0.4;
```

```
                end

                for ka=1:I4

                    Cx=-98+ttj-1+(rand*a);
                    CX(length(CX)+1)=Cx;
                    Cy=15+tti-1+(rand);
                    CY(length(CY)+1)=Cy;
                    XD=CX;
                    YD=CY;
                end
            end

        else

            I2=ceil(TT(tti,ttj)*NMax);    %
Número de vectores teórico máximo por zona
            I3=ceil(TT(tti,ttj)*NMin);
 % Número de vectores teórico mínimo por zona

            if II > I2 & II > (2*NMax/3)
        %%%REMOVER VECTORES%%%
                I6=abs(I2-II);
                Q=randperm(II);
                for kq=1:I6

                    Q1=Q(kq);
                    Q2=I(Q1);
                    CX(Q2)=[];
                    CY(Q2)=[];
                    XD=CX;
                    YD=CY;
                end
            end

            if II < I3
        %%%AGREGAR VECTORES%%%
                I5=ceil(TT(tti,ttj)*NMin);
%Número de vectores real permitido por zona
                I4=ceil(abs((I5-II)/TT(tti,ttj)));
%Número de vectores que se insertaran

                if I4 < NMin

                    if ttj < NumSj
                        a = 1;
                    else
                        a = 0.4;
                    end

                    for ka=1:I4

                        Cx=-98+ttj-1+(rand*a);
                        CX(length(CX)+1)=Cx;
                        Cy=15+tti-1+(rand);
                        CY(length(CY)+1)=Cy;
                        XD=CX;
                        YD=CY;
                    end
                end
            end
        end
    end

    EjeX=[-98 -96 -94 -92 -90 -88 -86 -84 -82];
    EjeY=[16 18 20 22 24 26 28 30];
    set(gca,'XTick',EjeX);

set(gca,'XTicklabel',['98W';'96W';'94W';'92W';'90W
';...

'88W';'86W';'84W';'82W'],'FontSize',10);
    set(gca,'YTick',EjeY);

set(gca,'YTicklabel',['16N';'18N';'20N';'22N';'24N';'
26N';'28N';...
            '30N'],'FontSize',10);

    NumF=num2str(t);
 %incremento del Número de la figura

    dia=jj+t*tiem-tiem;
    dia1=datestrad(dia,0);

    StrTitle = [num2str(dia1),' Hrs.'];      %
titulo de la imagen
    text(-98,30.5,StrTitle,'color','w','FontName',
'Palatino','FontWeight', 'bold ','FontSize',12);

     %Escala del vector para la velocidad

    u2(1:21,1:21)=.5;
    v2(1:21,1:21)=0;
    x2=-89.5:0.05:-88.5;
    y2=20:0.05:21;
    mx0=-89.2;
    my0=20.7;

     hold on
    xf=[-89.5 -88 -88 -89.5 -89.5]; yf=[20 20 21
21 20];
    fill(xf,yf,'w')

    curquiver2d
(x2,y2,u2,v2,mx0,my0,MV,NSeg,DT,Color,N2);
    text(-89.2,20.3,'0.5
m/s','Color','k','FontSize',8)

    set(gca,'color',[.3 .5 .3])
    set(gcf,'color','w')
    set(gcf,'InvertHardCopy','off');
    hold on
    xf=[-97 -91 -91 -97 -97]; yf=[16.25 16.25
16.75 16.75 16.25];
    fill(xf,yf,'g')
    x=-97+(dia*0.0164);
```

```
      xf=[-97 x x -97 -97]; yf=[16.25 16.25 16.75
16.75 16.25];
      fill(xf,yf,'b')
      text(-97.2,15.75, 'ENE', 'color', 'w',
'FontWeight', 'bold', 'FontSize',9);
      text(-91.2,15.75, 'DIC', 'color', 'w',
'FontWeight', 'bold','FontSize',9);
      text(x,16.1,'^','color','r','FontSize',7);

      text(-97.2,17.1,'30', 'color', 'w' ,
'FontWeight', 'bold ','FontSize',9);
      text(-
95.7,17.1,'31.75','color','w','FontWeight', 'bold',
'FontSize',8);
      text(-94.0,17.1,'33.5','color','w','FontWeight',
'bold', 'FontSize',8);
      text(-92.7,17.1,'35.25', 'color', 'w',
'FontWeight', 'bold','FontSize',8);
      text(-91.2,17.1,'37', 'color', 'w', 'FontWeight',
'bold ','FontSize',9);
      text(-90.7,17.1, 'SHD', 'color','w',
'FontWeight', 'bold ','FontSize',9);
      axes('position',[.182  .2  .26  .025])
      pcolor([1:32;1:32])
      axis off

      Fecha=dateNetCDF(dia)
      nom=['S',num2str(Fecha)];
      grafica=[Rimg,Nimg,nom];
      print('-djpeg',grafica);
   end
end
```

GLOSARIO

Advección

Transporte de masa o de las propiedades de un fluido debido al movimiento del mismo. Por lo general en la atmósfera este término es referido al transporte horizontal de propiedades como temperatura, presión y humedad. En el océano se refiere tanto a transportes horizontales como vectoriales.
Irrupción en una zona de un fluido con características diferentes a las que allí hay.

Aliasing

En gráficos por computadora, en computación y particularmente en infografía, el aliasing es el artefacto gráfico característico que hace que en una pantalla ciertas curvas y líneas inclinadas presenten un efecto visual tipo "sierra" o "escalón". El aliasing ocurre cuando se intenta representar una imagen con curvas y líneas inclinadas en una pantalla, framebuffer o imagen, pero que debido a la resolución finita del sustrato resulta que éste sea incapaz de representar la curva como tal, y por tanto dichas curvas se muestran en pantallas dentadas al estar compuestas por cuadros pequeños (los pixeles).

Alisios

Vientos constantes del este que soplan suavemente desde las altas presiones subtropicales hacia el ecuador.

Sistema de vientos relativamente constantes en dirección y velocidad que soplan en ambos hemisferios, desde los 30° de latitud hacia el ecuador, del noreste en el hemisferio norte y sudeste en el hemisferio sur.

Baroclínico

Indica un estado de la distribución del campo de masa en un fluido en el que las superficies isobáricas (de igual presión) intersectan superficies isopicnas (de igual densidad). La distribución vertical de la velocidad asociada no es uniforme. Este estado de fluido es el resultado de inhomogeneidades en la distribución horizontal de la

densidad y, obviamente, un fluido baroclínico no puede permanecer en reposo debido a que la intersección de las superficies isobáricas con las isopicnas genera un gradiente de presión que induce el movimiento.

Barotrópico

Estado de la distribución de los campos de masa en un fluido en el cual las superficies isobáricas (de igual presión) coinciden con las superficies isopicnas (de igual densidad). La distribución vertical de la corriente asociada es uniforme. Si el fluido está en reposo, las superficies isobáricas y las isopicnas son paralelas a superficies neopotenciales; en cambio si se induce un movimiento por inclinación de la superficie, el cuerpo de agua se mueve como un todo bajo el gradiente de presión.

Bottom / Up

El diseño ascendente se refiere a la identificación de aquellos procesos que necesitan computarizarse con forme vayan apareciendo, su análisis como sistema y su codificación, o bien, la adquisición de paquetes de software para satisfacer el problema inmediato.

Doldrums

Zona de calmas ecuatoriales.

Euleriana, descripción

Descripción de las características de un fluido en movimiento a partir de lo que ocurre en un punto fijo conforme pasa el tiempo. Para medir un campo de velocidades se realizarían observaciones directas de velocidad y dirección con correntímetros en diferentes puntos y se analizaría la variación de la corriente en cada punto a través del tiempo.

Geoide

La superficie equipotencial gravitacional de la Tierra que mejor se ajusta al nivel medio del mar.

Superficie geopotencial que coincide con la superficie media de los océanos en equilibrio. Suele adoptarse como representación de la figura de la Tierra.

Glifo

Un objeto o símbolo para representar los valores de los datos. Un glifo es generalmente una manera de representar muchos valores de datos y se llama a veces icono. Un glifo común es la flecha, a menudo escogido para representar un vector. La flecha representa tanto la rapidez como la dirección en un punto.

Icono que representa una función. Signos que corresponden a una escritura simbólica o figurada que expresan el significado de las palabras.

Infografía

Generación de imágenes por medio de la computadora. Más específicamente suele hacer referencia a la creación de imágenes que tratan de imitar el mundo tridimensional mediante el cálculo del comportamiento de la luz, los volúmenes, la atmósfera, las sombras, las texturas, la cámara, el movimiento, etc.

Estas técnicas basadas en complejos cálculos matemáticos, pueden tratar de conseguir imágenes reales, en cuyo caso se habla de infografía foto realista.

Isolíneas

Líneas de valor escalar constante.

Isopicna

Línea o superficie que une todos los puntos de densidad constante sobre una gráfica en el espacio o en el tiempo.

Lagrangiana, descripción

La que se hace de las propiedades de un fluido a partir de considerar la historia o trayectoria individual de cada partícula. Por ejemplo, la descripción de corrientes, medidas con cuerpos de deriva.

NetCDF

Network Common Data Form, es una interfase para el acceso de los datos y una colección de librerías de software para la implementación de estas interfaces. Las librerías de NetCDF definen un formato para representar los datos científicos.

Olas

Son movimientos ondulatorios, oscilaciones periódicas de la superficie del mar, formadas por crestas y depresiones que se desplazan horizontalmente.

Plataforma continental

Zona que se extiende desde la línea de inmersión permanente hasta la profundidad cercana a los 200 m.

Spline

En el subcampo matemático del análisis numérico, un spline es una curva definida a trozos mediante polinomios.

En los problemas de interpolación, se utiliza a menudo la interpolación mediante splines porque da lugar a resultados similares requiriendo solamente el uso de polinomios de bajo grado a la vez que se evitan las oscilaciones, que en la mayoría de las aplicaciones resultan indeseables, que aparecen al interpolar mediante polinomios de grado elevado.

Para el ajuste de curvas, los splines se utilizan para aproximar formas complicadas. La simplicidad de la representación y la facilidad de cómputo de las splines les hacen populares para la representación de curvas en informática, particularmente en el terreno de los gráficos por ordenador.

Topografía dinámica

Gráfico formado por isolíneas que muestran las alturas dinámicas de una zona del océano. Este tipo de análisis de los datos oceanográficos sirve para obtener

información de la circulación geostrófica, ya que el fluido tiende a desplazarse por superficies equipotenciales (de igual altura dinámica).

BIBLIOGRAFÍA

BACKHAUS, J.O. and E. Maier-Reimer , "*On Seasonal Circulation in the North Sea"*, in "North Sea Dynamics'', 1983.

BLUMBERG, A. F., and G. l. Mellor, "*A description of Three-dimensional coastal ocean circulation model*", in Three-Dimensional Coastal Ocean Models, Coastal Estuarine Stud., vol. 4 edited by N. Heaps, PP. 1-16, AGU, Washington, D. C., 1987.

FOLEY, Van Dan, Feiner, Hughes. "*Computer Graphics. Principles and Practice*". Addison-Wesley 1990.

GALLAGHER, R. S. "*Computer Visualization. Graphics Techniques for scientific and engineering analysis*". CRC Press 1995.

HAIDVOGEL, D.B., Wilkin, J.L. and Young, R., "*A Semi-spectral Primitive Equation Model Using Vertical Sigma and Orthogonal Curvilinear Horizontal Coordinates"*, J. Comp. Phys., 1991.

HEARN, D. and M. P. Baker, "*Computer Graphics C Version*", Pretince Hall 1997.

HOLLAND, W.R. and Lin, l.B., "*On the generation of mesoscale eddies and their contribution to the oceanic general circulation"*, J. Phys. Ocean, 1975.

MACKINLAY, J., "*Automating the Design of Graphical Presentations of Relational Information*", ACM Trans. on Graphics, Vol. 5, No. 2, April 1986, pp 110-141.

MARR, D., "*Vision - A Computational Investigation into the Human Representation and Processing of Visual Information".* Freeman, 1982.

MARTIN, P. J., "*A description of the Navy Coastal Ocean Model Version 1.0*", NRL/FR/7322-009962, 39 PP. Nav. Res. Lab Stennis Space Center Miss., 2000.

MCCORMICK, B. H., T. A. Defanti, M. D. Brown, "*Visualization in Scientific Computing*", Computer Graphics- Volume 21, Number 6, November 1987, Published by ACM Siggraph, New York.

MOREY, S.L., P. j: Martin, J. J. O'Brien, A.A. Wallcraf, and J. Zavala-Hidalgo, "*Export pathways for river dischsrged fresh water in the northern Gulf of Mexico*", J. Goophys. Res., 108(C10), 3303, doi:10.1029/2002JC001674, 2003.

O'BRIEN, J.J., "*Advanced Physical Oceanographic Numerical Modeling"*, D. Reidel Publ. Co., 1985.

R.A. Earnshaw, K.W. Brodlie, L.A. Carpenter, J.R. Gallop, R.J. Hubbold, A.M. Mumford, C.D. Osland, P. Quarendon, "*Scientific Visualization: techniques an applications*". Springer-Verlag. New York. (1992).

SÁMANO, D. A., (Ecología y Energía, México), Mihir Sen (Universidad de Notre Dame, EE.UU.), Sara Lilia Moya (CENIDET, México), "*Mecánica de fluidos*", 29 de abril de 2002.

ZAVALA Hidalgo J., S. L. Morey and J. J. O'Brien, *"Seasonal circulation on the western shelf of the Gulf of Mexico",* Submitted to J. Geophysical Research, March 2003.