Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación
Servicio Meteorológico Nacional - Servicio de Hidrografía Naval

Asimilación de datos en un modelo de olas con análisis enriquecido del viento

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Pablo Honorio Echevarria
133/00

Director: Dr. Pablo Miguel Jacovkis
Codirectora: Dra. María Paula Etala
Buenos Aires, 22 de mayo de 2014

ASIMILACIÓN DE DATOS EN UN MODELO DE OLAS CON ANÁLISIS ENRIQUECIDO DEL VIENTO

El modelado de olas tiene particular interés para la navegación y para la previsión de la erosión costera; en este contexto la *asimilación de datos* se plantea como una alternativa válida para incorporar observaciones a un modelo numérico y así mejorar la determinación del campo de olas.

En este trabajo desarrollamos un sistema completo de asimilación de datos, completando ciclos de asimilación de 6 hs. Al modelo de olas NOAA WAVEWATCH III se le asimilan observaciones de dos altímetros en los satélites Jason 1, Jason 2 y el escaterómetro del satélite MetOp-A.

En cuanto a las observaciones, se desarrolló un sistema orientado a objetos de adquisición, decodificación, control de calidad y preparación de las mismas, de modo tal de que queden listas para ser asimiladas por el sistema, que además es extensible a nuevas observaciones.

Para la asimilación utilizamos: 4D-LETKF(Local Ensemble Transform Kalman Filter), la cual fue modificada para poder adecuarse al problema de olas.

Para la ejecución del modelo de olas se desarrolló una rutina de asimilación interna, como así también el preparado del entorno de ejecución de cada ciclo.

Salvo el modelo, el filtro y algún otro pequeño software de control de calidad, el resto está implementado en Python. Tanto el modelo como el filtro están programados para trabajar en forma paralela.

El resultado fue la implementación de un sistema de asimilación de datos de olas que es eficiente y factible en tiempo de cómputo.

Palabras claves: Asimilación de datos, 4D-LETKF, modelado oceánico de olas, MPI, computación de alta performance, Jason1, Jason2, altímetro,ASCAT, dispersómetro, NOAA WAVEWATCH III.

DATA ASSIMILATION IN A WAVE MODEL ENRICHED WITH THE WIND ANALYSIS

The modeling of waves is of particular interest for navigation and forecasting coastal erosion. In this context the data assimilation is proposed as a valid alternative to incorporate observations to a numerical model and thus improve the determination of the wave field.

In this work we developed a complete 6-hours assimilation cycle, assimilating observations from two altimeters on satellites Jason 1 and Jason 2, and a scatterometer on satellite MetOp-A in the wave model WAVEWATCH III NOAA/NCEP.

We also developed an object oriented system of acquisition, decoding, quality control and preparation of observations, so that they are ready to be assimilated. Besides, the system may easily accept new sources of data. Following current trends and its versatility, we selected Python as programing language.

For assimilation 4D-LETKF (Local Ensemble Transform Kalman Filter) was used, which was modified to accommodate the ocean waves problem.

A routine of internal assimilation, and the preparation of the execution environment for each cycle, were also developed.

The model and the filter are parallelized. We succeeded in implementing an efficient and feasible in computing time, wave data assimilation system.

Keywords: Data Asimilation, 4D-LETKF, Wave model, MPI, HPC, Jason 1, Jason 2, altimetry, ASCAT, Scaterometer, metop, NOAA WAVEWATCH III.

AGRADECIMIENTOS

"Somos como enanos a los hombros de gigantes. Podemos ver más, y más lejos que ellos, no por alguna distinción física nuestra, sino porque somos levantados por su gran altura."

Bernardo de Chartres, Metalogicon de 1159 (III, 4)

- a Paula Etala, por su guía, paciencia, consejos y dedicación.
- a Pablo Jacovkis, por sus consejos, guía y por transmitirme ya hace tiempo que la interdisciplina es un camino muy fértil y apasionante. También que no cumplir con el programa de estudios no es algo necesariamente negativo.
- a Esteban Mocskos^n, por las n veces que fue molestado, la paciencia y dedicación.
- a Juan Ruiz, por sus consejos y absoluta generosidad.
- a Fernando Lores, por su amistad y ayuda con Bash.
- a Estela Collini, por sus consejos y ayuda con Fortran.
- a Carolina Tauro^n, por su ayuda con los satélites.
- a mis compañeras y compañeros de oficina.
- a Maxi, Valeria, David por al veloz lectura de último minuto.
- a todas personas que conocí durante estos años de cursada. Gracias por los momentos compartidos.

¡GRACIAS!

A Magalí, gracias a sus consejos y ayuda pude cerrar esta etapa en mi formación. A mi Madre, quien tuvo claro y supo transmitir que estudiar era algo positivo. A Anibal Kacero y a Liliana Ferreyra, dos maestros quienes compartieron su amor por el conocimiento y su pasión por la ciencia. Y a tres entrañables amigos que ya no están con nosotros, Andrés Perdomo, Gabriel Perdomo con quienes inicié mis primeros pasos en esta disciplina y Daniel Barboza.

Índice general

Glosario . . . 1
0.1. Lista de acrónimos . . . 1

1.. Introducción . . . 5

2.. El Problema del modelado numérico de olas oceánicas . . . 11
2.1. Descripción estadística del campo de olas . . . 11
2.1.1. Representación espectral . . . 12

3.. Conjunto de campos de viento del GFS (GEFS) . . . 13

4.. Modelo de olas . . . 15

5.. Asimilación de datos . . . 17
5.1. Breve introducción a la teoría de asimilación de datos . . . 17
5.2. Ecuaciones de Kalman Filter extendido . . . 18
5.3. Métodos de asimilación . . . 23
5.4. 4D-LETKF . . . 23

6.. Observaciones . . . 25
6.1. Altímetros . . . 25
6.1.1. Satélite JASON 1 . . . 25
6.1.2. Satélite JASON 2 . . . 27
6.2. advanced scaterometer (ASCAT) . . . 27
6.2.1. Satélite METOP-A . . . 27
6.3. Formato de archivos meteorológicos . . . 29
6.3.1. Binary Universal Form for the Representation of Meteorological data (BUFR) . . . 29
6.3.2. Network Common Data Form (NetCDF) . . . 29
6.4. Tiempo de procesamiento de las observaciones . . . 29

7.. Cluster . . . 31

8.. Desarrollo . . . 35
8.0.1. Asimilación . . . 37
8.0.2. Hora de análisis . . . 38
8.0.3. Organización del código . . . 38
8.1. Desarrollar un método de adquisición sistemática de la información del conjunto de pronósticos de NOAA/NCEP . . . 39
8.1.1. TIGGE . . . 39
8.2. Adaptar el sistema existente de pronóstico de olas para ejecuciones de conjuntos de modelos basado en el conjunto de NOAA/NCEP . . . 40
8.3. Estudiar la dimensión del problema . . . 41
8.4. Instalar y adaptar el 4D-LETKF para la asimilación de altura de olas y viento 46
8.4.1. Radio de localización . . . 46

8.4.2. Descripción de Módulos/Rutinas modificadas 46
8.4.3. Variables "harcodeadas" en el 4D-LETKF 48
8.4.4. Ciclicidad de Dominio Horizontal 49
8.5. Adaptar la interfase de entrada de datos para recibir la altura de olas altura significativa de ola (hs) obtenida por los satélites Jason-1, Jason-2 y vector viento (la componente u del vector viento (wu) y la componente v del vector viento (wv)) MetOp-A (ASCAT) 51
8.6. Adquisición y preparación de datos de observaciones de satélite 52
8.6.1. Obtención y decodificación 53
8.6.2. Control de Calidad de las Observaciones de Jason 1 y Jason 2 54
8.6.3. Preparación para ser asimiladas 55
8.7. Incorporar el análisis como estado inicial del modelo de olas, modificando la interfase prevista para asimilación 57
8.7.1. Etapa 1 en el proceso de desarrollo: Escalar HS 57
8.7.2. Implementación 58
8.7.3. Etapa 2 en el proceso de desarrollo: Corrección del espectro 61
8.8. Cerrar el ciclo, utilizando el estado nuevo inicial para correr el conjuntos de pronósticos 65
8.9. Visualización de resultados 66
8.10. Experimentos y Resultados 69
8.10.1. Medida del error 71
8.11. Performance 74

9.. Conclusiones 79

10..Trabajos Futuros 81
10.1. GPU 81
10.2. Sistema Operativo 81
10.3. Cuellos de Botella 81
10.4. Unificar observaciones wu wv 81
10.5. MetOp B y OceanSat-2 81
10.6. Paralelizar lectura de observaciones de 4D-LETKF 81
10.7. Verificación 81
10.8. Sistema de pre y post-condiciones 82
10.9. Ensamble a 80 miembros 82
10.10Radio de localización óptimo 82

Apéndice 85

A.. Publicaciones 87

Glosario

hs	La altura significativa de ola (hs) se define como la altura media del tercio mayor de todas las olas observadas.
Mar de fondo	Es el nombre que se le da a las olas una vez que adquieren más velocidad que el viento, por lo tanto éste deja de tener efecto sobre ellas.
Mar de viento	Son las olas sobre las que el viento está ejerciendo algún efecto; la velocidad del mismo es mayor que las de las olas.
Timestamp	Del Inglés time stamp, marca temporal. Se utiliza para etiquetar temporalmente un objeto o dato.

0.1. Lista de acrónimos

ANL Análisis . . . 71

AMR Advanced Microwave Radiometer . . . 27

ASCAT advanced scaterometer . . . IX

BUFR Binary Universal Form for the Representation of Meteorological data . . . IX

CNES Centre National d'Études Spatiales . . . 27

DORIS Doppler Orbitography and Radio-positioning Integrated by Satellite . . . 27

DAS Data Assimilation . . . 46

ECMWF European Centre for Medium-Range Weather Forecasts . . . 13

EnKF Ensemble Kalman Filter . . . 23

EUMETSAT European Organisation for the Exploitation of Meteorological Satellites . . . 27

ESA European Space Agency 28

GFS Global Forecast System 13

GEFS Global Ensemble Forecast System 13

GPSP Global Positioning System Payload 27

GrADS Grid Analysis and Display System 54

GRIB2 GRIdded Binary or General Regularly-distributed Information in Binary form . 13

GMT Greenwitch Mean Time 37

hs altura significativa de ola x

IO Interpolación Óptima 6

KNMI Koninklijk Nederlands Meteorologisch Instituut 28

lat. latitud 47

LRA Laser Retroreflector Array 27

lon. longitud 47

MetOp Meteorological Operational Satellite Programme 27

MPI Message Passing Interface 15

NASA National Aeronautics and Space Administration 27

NCEP National Centers for Environmental Prediction 13

NetCDF Network Common Data Form ix

NOAA National Oceanic and Atmospheric Administration 13

NWP Numerical Weather Prediction 13

NRT Near Real Time 54

OSTM Ocean Surface Topography Mission 27

obs Observación 55

RMS error cuadrático medio 25

SMN Servicio Meteorológico Nacional 15

SHN Servicio de Hidrografía Naval 15

SSHA Sea Surface Height Anomalies 54

TIGGE THORPEX Interactive Grand Global Ensemble 39

UCAR University Corporation for Atmospheric Research 29

WMO World Meteorological Organization 29

wu la componente u del vector viento x

wv la componente v del vector viento x

1. INTRODUCCIÓN

El pronóstico de las olas en alta mar tiene suma importancia para la seguridad en la navegación de los barcos, por ejemplo cuando se proyecta una ruta. También hay razones económicas para el estudio del pronóstico de olas en relación con la construcción de muelles y defensas costeras, daños por tormentas y erosión de las playas. Para realizar el pronóstico de olas y de este modo poder predecir situaciones potencialmente peligrosas se utilizan modelos numéricos, que simulan su comportamiento.

Para caracterizar las olas de manera realista, y así poder estudiarlas, se toma un punto del reticulado (*grid*) y para ese punto se calcula en forma cuantitativa la energía asociada al espectro de olas haciendo una descomposición en frecuencia y dirección. Se lo conoce como el "espectro de energía de ola" o simplemente "espectro de ola" y constituye la variable de pronóstico[1] del modelo.

Las olas son lo que se denomina un "problema de forzante", lo que quiere decir que hay un fenómeno externo que hace que se desarrollen. En el caso de las olas es principalmente el viento; también están afectadas por el hielo marino, topografía de fondo marino y temperatura del agua. Entonces cuando el forzante del fenómeno actúa el modelo propaga la energía. Por otro lado, las olas se organizan en grupos llamados "trenes de olas", con una dirección común.

Existen varios tipos de modelos de olas; los de tercera generación son los más modernos y no piden ningún tipo de restricción a la forma del espectro, como así lo hacían los modelos anteriores. Por parte de los oceanógrafos se ha hecho un esfuerzo importante al intentar incluir en sus cálculos los efectos físicos que se consideran importantes.

Lorenz en 1963 realizó estudios que pusieron de manifiesto **el comportamiento caótico de la atmósfera**[Lorenz, 1963][2] y por lo tanto la presencia de incertidumbre al tratar de modelarla numéricamente. La misma aumenta cuanto mayor es el plazo de pronóstico y desde los primeros estudios él estimó un plazo de dos semanas como máximo pronosticable del sistema. Se hizo evidente que no es posible conocer con precisión el futuro del sistema; se invirtió mucho trabajo en conocer el comportamiento de la incertidumbre, en reducir el error y en tratar de entender su comportamiento.

En una escala de metros las olas son un fenómeno aleatorio gausiano de superposición de ondas. Se representan por parámetros integrales como la hs. Por otro lado, en su generación, las olas son forzadas por la tensión del viento en superficie proveniente de fenómenos atmosféricos. Si bien la generación de olas se produce en una escala de tiempo pequeña, por ejemplo la duración de una tormenta, las olas generadas persisten mucho más allá en tiempo y se propagan muy lejos en el espacio. Por otro lado en una escala mayor, de kilómetros, las olas se organizan en grupos. Estos grupos cumplen con la definición de fenómeno caótico: dadas dos condiciones iniciales con una pequeña perturbación,

[1] Variable de pronóstico es la/s variable/s que caracterizan el fenómeno que se pretende modelar.

[2] Un efecto curioso de los sistemas, "Efecto Mariposa"; fue descubierto por el meteorólogo E. Lorenz en 1963; buscaba resolver las inestabilidades climáticas y conseguir un modelo climático preciso; como las computadoras eran muy lentas y no podían funcionar durante la noche, y los cálculos tomaban mucho tiempo, Lorenz hacía pausas en la ejecución y registraba los resultados parciales, que servían de datos iniciales para el siguiente día, pero al hacer esto "truncaba" los números. En este punto fue cuando se topó con soluciones divergentes. El término "sistema caótico" hace referencia a esta idea: partiendo de soluciones prácticamente idénticas se llega, en cierto periodo de tiempo, a estados muy diferentes.

por ejemplo en la dirección, producirán grupos de olas perturbados una vez transcurrido cierto tiempo. Ver [Tolman, 2008].

En 1969, Epstein introdujo el concepto de "pronóstico estocástico-dinámico"[Esptein, 1969], que consiste en determinar la función de distribución de la probabilidad de los estados posibles futuros del mar-atmósfera partiendo de un estado inicial. La solución para caracterizar la función de probabilidad consiste en realizar múltiples simulaciones del sistema (mar-atmósfera) usando salidas de modelos o condiciones iniciales levemente diferentes, de tal manera de poder obtener parámetros estadísticos que caracterizan la distribución (media y varianza). Este tipo de técnica se conoce como "pronóstico por conjunto o conjuntos". Una de las hipótesis de esta técnica es que si la manera de generar los distintos pronósticos es correcta, las características estadísticas del conjunto se parecerán a las de la función de distribución de los futuros estados del sistema mar-atmósfera. Ver [Macrina, 2004] y [Ruiz, 2008].

La asimilación de datos combina campos del modelo con observaciones para obtener un resultado que minimiza el error cometido de manera de lograr un campo más cercano al verdadero estado del sistema. Es decir, la asimilación de datos es el proceso mediante el cual se incorporan las observaciones en un modelo (matemático-computacional) de un sistema real de modo tal de generar nuevos valores de las variables que aproximen mejor la realidad.

Para realizar un pronóstico más preciso se han hecho grandes progresos, tratando de mejorar las condiciones iniciales [Kalnay, 2003]. Estas mejoras apuntan a reducir el error cometido buscando mejorar las herramientas para que el pronóstico sea más preciso.

Las observaciones disponibles que son pertinentes para las olas marinas son: la altura significativa de ola (hs) provista por los altímetros; la velocidad y dirección del viento provista por los escaterómetros. Si bien la hs no es la variable de pronóstico la primera se puede calcular a partir de la segunda.

A pesar de los grandes esfuerzos hay situaciones en las que los modelos no predicen correctamente la evolución de las olas; esto suele imputarse al forzante, siempre y cuando las parametrizaciones sean adecuadas.

En NCEP en 2004 fue implementado un sitema de asimilación de datos basado en Interpolación Óptima (IO) desarrollado por [Behringer, 1994], [Derber and Rosati, 1989] y [Behringer et al., 1998]. Este método fue previamente testeado para asimilar, en otro modelo, temperatura en la superficie del mar (SST) con buenos resultados [Chen et al., 2004]. Las variables asimiladas en NOAA WAVEWATCH III® son hs proveniente del satelite ERS-2 y datos provenientes de boyas. Se logró una mejora significativa en el pronóstico a 12 hs y un pequeño impacto ya a partir de las 24 hs. El forzante es el viento a 10 metros producido por GFS. La ventana de asimilación es de 1 hora. Actualemente no se esta realizando asimilación de datos pero hay planes para implementar un esquema de asimilación de en principio asimilando hs (comunicación personal).

También en Australia fué ensayado un esquema de asimilación [aus, 2010]. Utilizan AUSWAVE que es un modelo basado en la versión 3.14 de NOAA WAVEWATCH III®, como forzante utiliza los campos de vientos de un sistema basado en UK Met Office Unified Model/Variational Assimilation (UM/VAR). Si bien se hicieron esfuerzos es lograr un esquema de asimilación, hubo varios desarrollos, actualmente no se realiza asimilación de datos. Ver [Greenslade, 2004]. La razón es que se espera que la mejora proveniente de la asimilación en el modelo meteorológico devenga en una mejora en la estimación de los campos de viento que deberia redundar en una mejora en la estimación los campos de olas

haciendo evidente la confianza puesta en el forzante.

Por otro lado en ECMWF se hace asimilación de datos de altímetros [ecm, 2012b] y [ecm, 2012a] en el modelo de olas WAM, que está acoplado al modelo meteorológico. El ciclo de asimilación, del modelo meteorológico, es de 12 hs con una ventana la ventana es de 9 a 21 hs, de manera de obtener dos análisis diarios, a las 12 y a las 0 UTC. Un campo analizado de hs se obtiene vía IO, siguiendo los trabajos de [Lorenc, 1981]. Luego se reconstruye el espectro de ola 2D completo utilizando el espectro obtenido en la primera aproximación (pronóstico previo a la asimilación), separando mar de viento y mar de fondo para realizar esta tarea.

El objetivo fue estudiar la factibilidad de un sistema que complete eficientemente el ciclo de asimilación/pronóstico de un conjunto, a una escala global, es decir estudiamos la aplicación de la asimilación de datos en un conjunto de olas a escala global para mejorar el campo inicial de olas, en el cual se realiza un análisis enriquecido de viento[3]. Para ello tuvimos que programar un *ciclo de asimilación*. El desarrollo incluye la tarea de coordinar dos programas principales, el modelo de olas y el software que realiza la asimilación, conocido como filtro de asimilación, como así también el desarrollo completo de un sistema de adquisición, decodificación y procesamiento de observaciones satelitales para tres satélites, al cual le dimos la característica de ser modular y estar diseñado para que la extensión a nuevas fuentes de datos observacionales sea inmediata. El modelo tiene varias etapas previas a su ejecución, preprocesamiento de datos por ejemplo; para cada etapa utiliza archivos de configuración; el ciclo de asimilación requiere varias ejecuciones del modelo. También tuvimos que analizar su funcionamiento estructurado en módulos. En la etapa de corrección de la variable de pronóstico (espectro de ola) tuvimos que programar completamente el algoritmo. Para el filtro el trabajo fue más extenso y complejo, se agregó la capacidad de asimilar variables nuevas, en campos nuevos, observaciones de intensidad y dirección de viento y altura de olas.

Incluimos el tratamiento de las condiciones de borde cíclicas que se requieren para la asimilacion de datos en un dominio global.

El software existente para visualizar los mapas no tenía la versatilidad necesaria; creamos un sistema que permite visualizar los resultados del trabajo: mapas con campos de olas, viento, trazas satelitales sobre estos.

[3] A diferencia de la asimilación, el análisis enriquecido recibe la información del modelo para componer el análisis pero no necesariamente lo realimenta.

Estructura de la tesis

En el capítulo 2 se hace una descripción del problema de modelado de olas en general. Del capítulo 3 al capítulo 6 se hace un desarrollo teórico. En el capítulo 7 describimos el cluster Elementos, adquirido por el proyecto PIDDEF 46/10 y utilizado para realizar el trabajo. En el capítulo 8 se describe el trabajo realizado y en la sección 8.10 se presentan los experimentos. En el capítulo 9 mostramos las conclusiones. En el capítulo 10 tratamos sobre futuros trabajos.

En el apéndice A se agrega una lista de publicaciones producto de esta tesis.

Si bien en el capítulo 8 abordamos el desarrollo completo del sistema, en la figura 1.1 presentamos un esquema a modo introductorio.

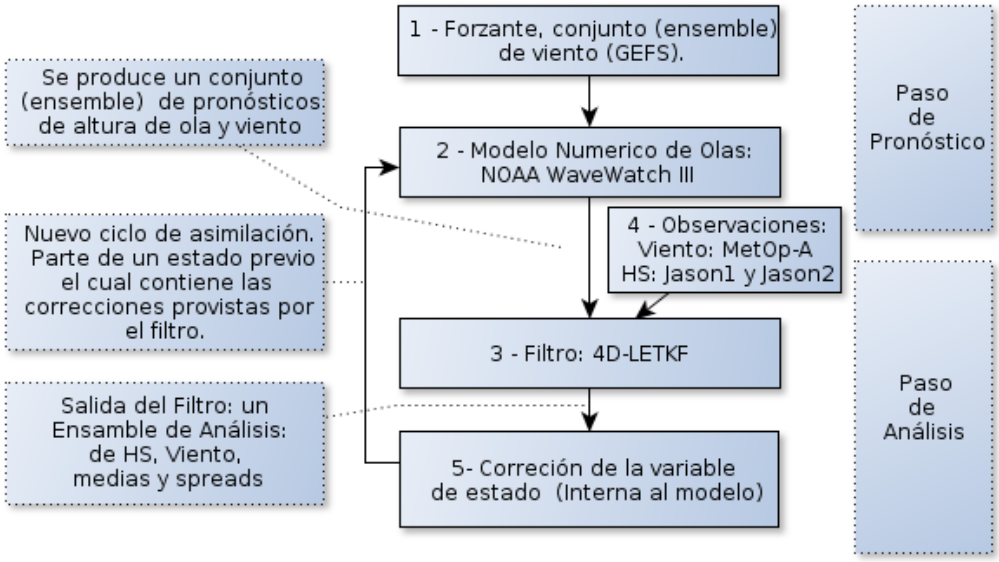


Fig. 1.1: Esquema del ciclo de asimilación que se implementó en el presente trabajo.

Primero se recibe el forzante del modelo proveniente del GEFS; con esta información se alimenta el modelo ejecutando la simulación. En forma independiente se obtienen, decodifican y preparan las observaciones. Una vez hecho esto el filtro lee las salidas del modelo y las observaciones, y produce un análisis que es utilizado nuevamente por el modelo para corregir el estado interno. La siguiente tabla hace referencia a los nodos de la figura 1.1.

	Breve introducción	Desarrollo
1 - Forzante: Ensamble de viento (GEFS)	capítulo 3	Sec. 8.1
2 - Modelo Numérico de Olas...	capítulo 4	Sec. 8.2
3 - Filtro: LETKF	capítulo 5	Sec. 8.4 y 8.5
4 - Observaciones	capítulo 6	Sec. 8.6
5- Corrección de la variable de estado	-	Sec. 8.7

La palabra guess o forecast se usan indistintamente y significan conocimiento previo o pronóstico.

Lugar de trabajo

Este trabajo de tesis se enmarca en el Proyecto conjunto del Servicio de Hidrografía Naval (SHN) y Servicio Meteorológico Nacional (SMN) del Programa de Investigación y Desarrollo para la Defensa PIDDEF No. 46/10 "Integración de la modelación marina y asimilación de datos", del cual formé parte como becario hasta diciembre de 2012 y del cual formo parte actualmente como personal del SMN. El equipamiento adquirido a tal efecto fue instalado en el Servicio Meteorológico Nacional, bajo el convenio marco correspondiente, por lo que el lugar de trabajo disponible es la sede central del Servicio Meteorológico Nacional, 25 de Mayo 658, C.A.B.A, u otra sede según dispongan las autoridades del/los organismos.

2. EL PROBLEMA DEL MODELADO NUMÉRICO DE OLAS OCEÁNICAS

2.1. Descripción estadística del campo de olas

Reproducimos aquí la descripción dada en [Etala et al., 2009]: "La olas oceánicas son ondas de gravedad producidas en la superficie del mar por las perturbaciones de pequeña escala de la presión atmosférica y por la tensión del viento sobre la superficie. El conjunto se caracteriza mediante parámetros integrales de altura, periodo y dirección. En general, la altura $H_{1/n}$ describe la altura media de la enésima porción de las olas más altas. El valor más utilizado para el pronóstico de olas en el océano es $n = 3$, ya que la altura significativa $H_{1/3}$ es la que más se asemeja a la altura característica (H_c) que un observador experimentado estima visualmente en un campo de olas.

No existe el mismo nivel de consenso para la estimación del periodo en el campo de olas. En el caso de registros continuos, es común dividir la longitud del mismo por la cantidad de veces en que este cruza el nivel cero, ya sea en sentido ascendente o descendente. Esta estimación resulta en una medida del periodo medio T_z. De la misma manera que para la altura, el periodo $T_{1/n}$ es el promedio de los periodos de la fracción $1/n$ de las olas más altas. Se denomina periodo significativo a $T_{1/3}$ y se asemeja al periodo del máximo de energía espectral T_p obtenido a partir de los modelos numéricos. El periodo característico T_c estima el periodo medio mediante la observación visual de un conjunto de 10 a 20 olas no consecutivas."

En la figura 2.1 puede verse la distribución estadística de las olas, en el eje X se muestra la altura de ola y en el eje Y la cantidad de olas. La hs se define como el **promedio** del tercio de olas más altas (en azul).

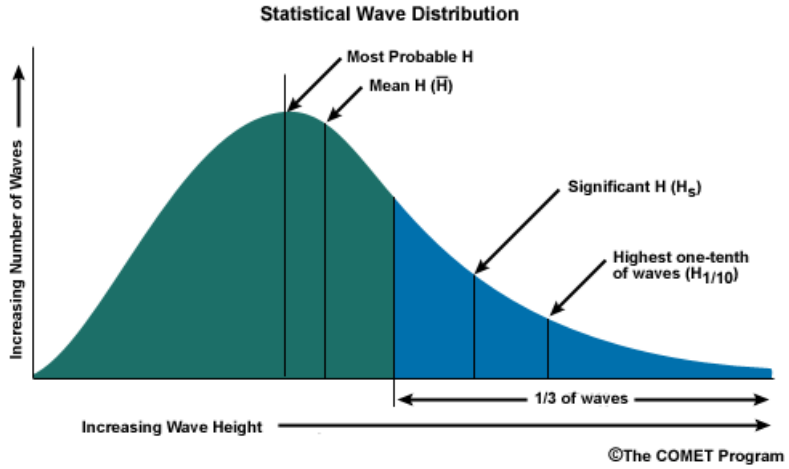


Fig. 2.1: Distribución estadística de las olas. La hs es la media del tercio mayor de olas (en azul claro). Fuente: www.comet.ucar.edu

2.1.1. Representación espectral

Continuamos aquí con párrafos de [Etala et al., 2009]: "Con los mismos argumentos que en la sección anterior, la distribución de probabilidad de las elevaciones de las superficie del mar, consideradas como el apartamiento del nivel de reposo en cualquier lugar y tiempo, es prácticamente gaussiana. Esta propiedad permite aproximar la probabilidad de encontrar un determinado rango de elevaciones entre dos puntos distintos en espacio $(x1, x2)$ y tiempo $(t1, t2)$ mediante la covarianza de la elevación en $(x1, t1)$ y $(x2, t2)$. La transformada de Fourier de esta covarianza es el denominado **espectro de olas**. La interpretación física del espectro resulta ser la función de densidad de energía de las olas (E), que permite determinar la distribución de la energía en las componentes de distintas frecuencias o números de onda. Su integral representa la energía total de las olas por unidad de área. En coordenadas geográficas, el espectro de energía de las olas se describe en general en el espacio (lat, lon, f, θ) , en donde (lat, lon) son la latitud y longitud, f es la frecuencia y θ, la dirección. Los modelos numéricos de uso más extendido para la escala oceánica se basan en la descripción de la evolución del espectro bidimensional (frecuencia y dirección) de energía de las olas en un punto.

La altura significativa se infiere a partir de la energía total y el periodo del pico corresponde al representativo del intervalo de máxima energía en el espectro unidimensional (integrado para todas las direcciones en cada intervalo de frecuencia)."

$$hs(lat, lon) = 4\sqrt{\int\int E(lat, lon, f, \theta) df d\theta}$$

Por otro lado las olas se organizan en dos tipos: mar de viento y mar de fondo. Mar de viento se denomina al mar donde el viento está actuando sobre el agua, transfiriendo energía, a diferencia del mar de fondo, que es el tipo de mar donde el viento ya no actúa y las olas continúan su trayecto sin ser influenciadas. Esta información no está almacenada en el modelo, el mismo solo almacena el espectro de energía y el campo de viento; con los mismos se puede reconstruir dado un punto que tipo de mar es.

3. CONJUNTO DE CAMPOS DE VIENTO DEL GFS (GEFS)

Como se describe en la introducción, el primer paso es obtener los campos de vientos producidos por el Global Ensemble Forecast System (GEFS), de manera de alimentar al modelo de olas con un forzante.

El Global Forecast System (GFS) es un Numerical Weather Prediction (NWP) creado y utilizado por la National Oceanic and Atmospheric Administration (NOAA)[1]. Se hacen cuatro pronósticos diarios con un plazo de pronóstico de hasta 16 días. Para producir el conjunto o ensemble del modelo oceánico utilizaremos el conjunto de campos de viento que ponen a disposición National Centers for Environmental Prediction (NCEP)[2]. Se produce utilizando el método breeding. Actualmente el conjunto cuenta con 80 miembros[3][Wei et al., 2008]. Los campos de viento son campos analizados resultado del sistema de asimilación de NCEP.

Con cada uno de estos campos de hielo y viento se hacen las simulaciones sin otra perturbación extra.

El GFS tiene disponibilidad prácticamente total en cuanto a la cobertura, cuya producción de salidas de predicciones están disponible gratuitamente a través de Internet y consisten en datos de dominio público.

Por otro lado en el European Centre for Medium-Range Weather Forecasts (ECMWF) se producen conjuntos de pronósticos a escala global. Lamentablemente, las salidas útiles para realizar el pronóstico, de al menos cada 6 hs, sólo son accesibles para los Estados Miembros[4].

Los principales centros meteorológicos realizan pronósticos por conjuntos y utilizan los vientos en superficie que producen, para realizar los correspondientes pronósticos de olas.

Dada una hora de pronóstico (0, 6, 12 ó 18) se cuenta con los 20 campos analizados de viento en superficie [5] y los correspondientes 20 pronósticos a 6 hs., en formato GRIdded Binary or General Regularly-distributed Information in Binary form (GRIB2)[6] con una resolución de $1^o \times 1^o$. De esta manera contamos con una densidad de forzantes suficientemente alta lo que permite obtener una simulación.

La figura 3.1 es un ejemplo de la variabilidad del conjunto. Las líneas representan una curva de nivel a 3 metros del campo de hs de cada miembro. Cada color representa un miembro del conjunto, lo que busca mostrar es que a pesar de ser mayormente similares presentan cierta variabilidad y esta es la característica buscada.

[1] www.noaa.gov

[2] www.ncep.noaa.gov

[3] El conjunto cuenta con 80 miembros de los cuales sólo se publican 20 por ciclo (ver capítulo 10).

[4] El centro ECMWF es una organización intergubernamental independiente financiada por los estados: Austria, Bélgica, Dinamarca, Finlandia, Francia, Alemania, Grecia, Islandia, Irlanda, Italia, Luxemburgo, Holanda, Noruega, Portugal, Eslovenia, España, Suecia, Suiza, Turquía, Reino Unido y otros estados asociados.

[5] En realidad es a 10 metros por convención.

[6] http://www.wmo.ch/pages/prog/www/DPS/FM92-GRIB2-11-2003.pdf

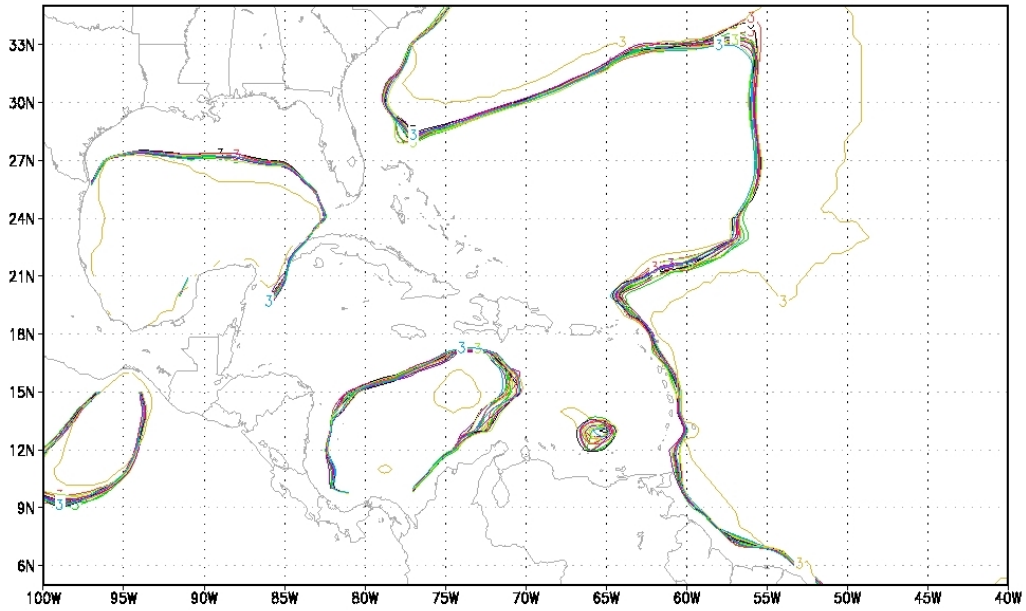


Fig. 3.1: Como ejemplo de variabilidad del conjunto de olas, se presenta un mapa "Spaguetti" de hs producto del modelo de olas WWIII forzado con el conjunto de NCEP. Son curvas de nivel de $hs = 3$ metros, cada color corresponde a un miembro del conjunto. Si bien se preserva la misma forma también hay cierta variabilidad.

4. MODELO DE OLAS

Para realizar el pronóstico utilizamos modelo de olas WAVEWATCH III®
[Tolman, 2009] desarrollado por NOAA/NCEP. Fue instalado y puesto en funcionamiento en forma operativa en el Servicio de Hidrografía Naval (SHN)/Servicio Meteorológico Nacional (SMN) mediante una licencia de tipo open source[1]. Por el acuerdo de la licencia, las modificaciones al código que surjan deberán ser compartidas con NOAA, que decidirá si se incorporan a la siguiente versión. Se deberán respetar, por lo tanto, las condiciones establecidas en [Tolman, 2009].

- Está preparado para correr en paralelo usando bibliotecas Message Passing Interface (MPI).
- Se dispone del código fuente, permitiendo hacer modificaciones.
- Está programado en Fortran 90.
- Como parte del desarrollo fue compilado con el compilador Fortran de Intel®: ifort.

El autor del software, Hendrik Tolman, realizó una serie de pruebas y mediciones de escalabilidad en un servidor de similares características al nuestro (con procesadores Intel XEON®) donde puede observarse que escala en forma excelente; también muestra que el Hiperthreading® (nombre de Intel® para el Multithreading simultáneo) **no** es beneficioso [Tolman, 2003].

En primer lugar, cuando se instala el modelo se ejecuta una pre-compilación en la cual se permite que el usuario seleccione ciertas características que desea del programa final (parametrizaciones físicas, ciertas constantes, precisión de los números reales, entre otras cosas). Para esto se utilizan ciertas opciones obligatorias[2], en nuestro caso y para fines computacionales nos van a interesar:

El modelo de hardware

- *shrd* : Modelo de memoria compartida,
- *dist* : Modelo de memoria distribuida.

El protocolo de pasaje de mensajes

- *shrd* : **sin** pasaje de mensajes Modelo de memoria compartida,
- *mpi* : **con** MPI.

Es decir, si queremos usar MPI utilizamos *dist mpi* (memoria distribuida, pasaje de mensajes MPI); si queremos usar threads, utilizamos *shrd*, que vale para el primer y el segundo grupo. No está implementada la opción $threads + MPI$. Para más información se puede consultar el manual [Tolman, 2009].

[1] http://polar.ncep.noaa.gov/waves/wavewatch/license.shtml.
[2] Mandatory switches

5. ASIMILACIÓN DE DATOS

La pregunta que responde el método de asimilación de datos es:

¿Cómo combinar en forma óptima modelo y observaciones para obtener una estimación del estado de un sistema con el menor error posible?

5.1. Breve introducción a la teoría de asimilación de datos

Planteamos una muy breve introducción de la teoría que sustenta el método basado en [Kalnay, 2003].

Sea algún fenómeno que se quiere modelar, por ejemplo la temperatura de una placa de metal o el movimiento de algún líquido en un recipiente. La información del estado del fenómeno se puede almacenar en un vector X. De igual manera, también supongamos que contamos con algún conocimiento previo, almacenado en el vector X^b y alguna medición en Y^o. Dado que lo que se busca es incorporar información de las observaciones al conocimiento previo el análisis se define como una combinación lineal de los dos conocimientos:

$$X^a = wY^o + w_b X^b.$$

Pero, ¿qué valores hay que asignar a w y w_b de manera que X^a tenga el menor error posible? En principio podemos hacer una sustitución: $w_b = 1 - w$, con $0 < w < 1$

Entonces el análisis queda:

$$X^a = X^b + w(Y^o - X^b).$$

A $Y^o - X^b$ se lo conoce con el nombre de **innovación**. Todo el problema consiste en estimar lo más precisamente posible w.

De la teoría de errores tenemos que, si X^t es verdadero estado del sistema, generalmente desconocido, entonces contamos con los siguientes **errores**:

- Error del background (conocimiento previo): $\epsilon^b = X^b - X^t \Rightarrow X^b = \epsilon^b + X^t$,
- Error de la observación: $\epsilon^o = Y^o - X^t \Rightarrow Y^o = \epsilon^o + Y^t$,
- Error del análisis: $\epsilon^a = X^a - X^t \Rightarrow X^a = \epsilon^a + X^t$.

Para calcular w debemos contar con un conocimiento lo más preciso posible del comportamiento estadístico de los errores del background[1] y de las observaciones. Estas simples ecuaciones se pueden extrapolar a varias dimensiones, en cuyo caso ya no habrá un error sino una matriz de covarianza de errores del background(P^f), de la observaciones y del análisis.

W será un interpolador que permitirá obtener la combinación óptima entre el background y las observaciones de manera de minimizar el error cometido.

[1] Usaremos forecast, pronóstico o background indistintamente, ya que el background en un paso es el forecast/pronóstico del paso anterior.

- Los errores, en el sentido estadístico, están en función de los procesos físicos que gobiernan los procesos meteorológicos y los instrumentos.
- La varianza del error muestra la incertidumbre acerca del comportamiento del campo preliminar y de las observaciones.
- En métodos como 3D-var o Interpolación Óptima la estrategia es pensar que se mantienen constantes o acotados, y que se puede hacer una "climatología de los errores".
- La manera en la que se determinan las estadísticas de los errores constituye el problema central de la asimilación.

5.2. Ecuaciones de Kalman Filter extendido

Describimos la idea básica de Kalman Filter para modelos no lineales.

Paso 1: Pronóstico, ver figura 5.1.
Partiendo del estado analizado del paso de tiempo $k-1$, el modelo M_{k-1} evoluciona el estado de las variables X^a_{k-1}[2]:

$$X^f_k = M_{k-1}(X^a_{k-1}).$$

Luego evoluciona la matriz de covarianza de errores del pronóstico P^f. Usando la información de la matriz de covarianza de errores del modelo Q_{k-1} y la linealización del modelo en el paso de tiempo anterior L_{k-1} y la matriz de covarianza de los errores del análisis P^a_{k-1}.

$$P^f_k = L_{k-1}P^a_{k-1}L^T_{k-1} + Q_{k-1}.$$

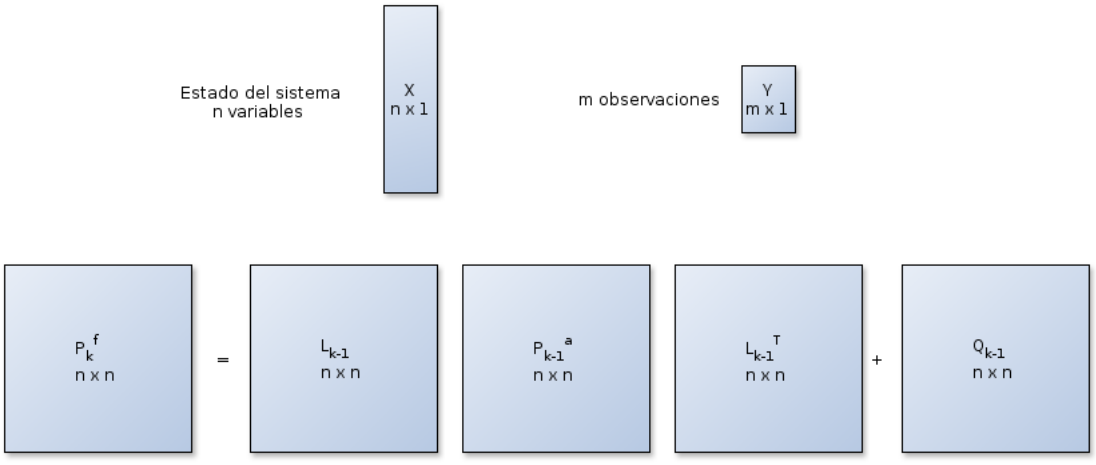


Fig. 5.1: Matrices en las ecuaciones del Kalman Filter extendido, paso 1.

[2] La ^{a} corresponde al estado analizado anteriormente.

Paso 2: Análisis, ver figura 5.2. Calcula la matriz de pesos de Kalman K_k para el paso de tiempo k, utilizando el operador de las observaciones H, la matriz de covarianza de errores del pronóstico P_k^f y la matriz de covarianza de errores de observación R_k. Por ejemplo de la función H que se utiliza en el presente trabajo consiste en una función que dado un punto de observación se busca el punto más cercano del modelo.

$$K_k = P_k^f H^T (H P_k^f H^T + R_k)^{-1}.$$

Para H lineal, si H no es lineal H^T es el modelo adjunto de H. Después hace el paso de asimilación donde se incorpora a la información previa X_k^f la **innovación** $Y_k - HX_k^f$, ponderada por la matriz de pesos de Kalman.

$$X_k^a = X_k^f + K_k(Y_k - HX_k^f).$$

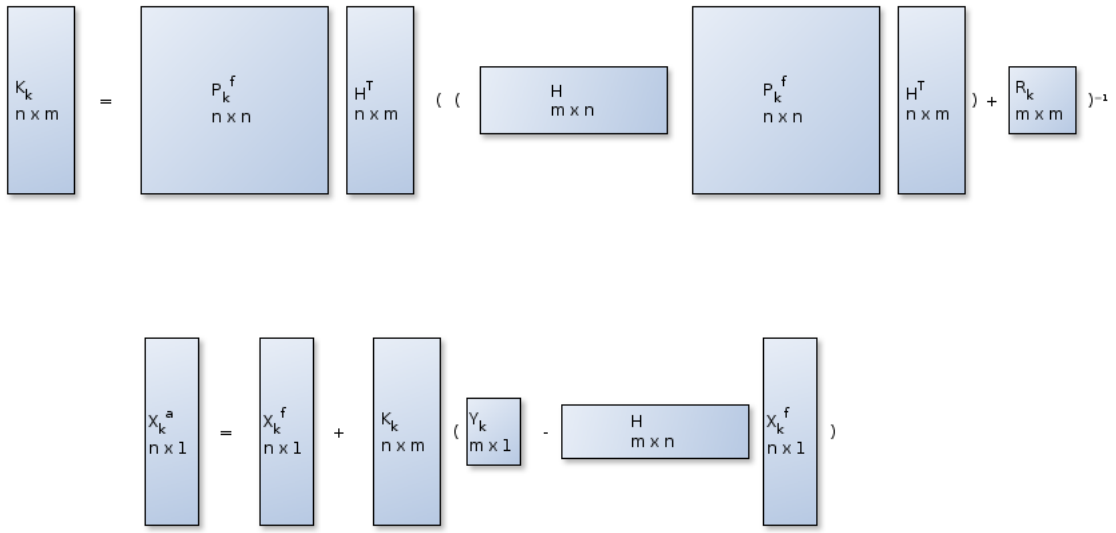


Fig. 5.2: Matrices en las ecuaciones del Kalman Filter, paso 2.

Paso 3: Incertidumbre de análisis, ver figura 5.3. Como último paso actualizamos la matriz de covarianza de los errores del análisis P_k^a.

$$P_k^a = (I - K_k H) P_k^f.$$

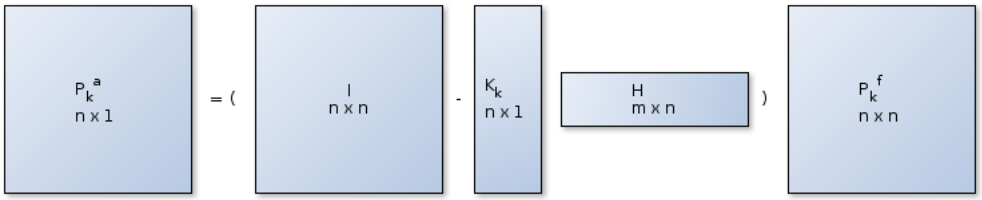


Fig. 5.3: Matrices en las ecuaciones del Kalman Filter, paso 3.

En la figura 5.4 se esquematiza la idea del análisis, se muestra cómo partiendo del análisis el modelo evoluciona las variables hasta llevarlas al nuevo estado; en ese punto la asimilación corrige ese pronóstico (FCST) para dar con un estado más cercano a la verdad del sistema (True state).

En la figura 5.4 y 5.5 se da una idea esquemática de que en los sistemas caóticos el error crece con el correr de las simulaciones.

En la figura 5.6 se puede ver una representación del tamaño del problema, donde d.o.f (degres of freedom) indica grados de libertad. P, en azul, muestra como se amplifica el error con el tiempo, en rojo se muestra el error de observación. Si 10^6 son los grados de libertad de vector de estado del sistema, la matriz de covarianza del pronóstico tendrá $(10^6)^2 = 10^{12}$.

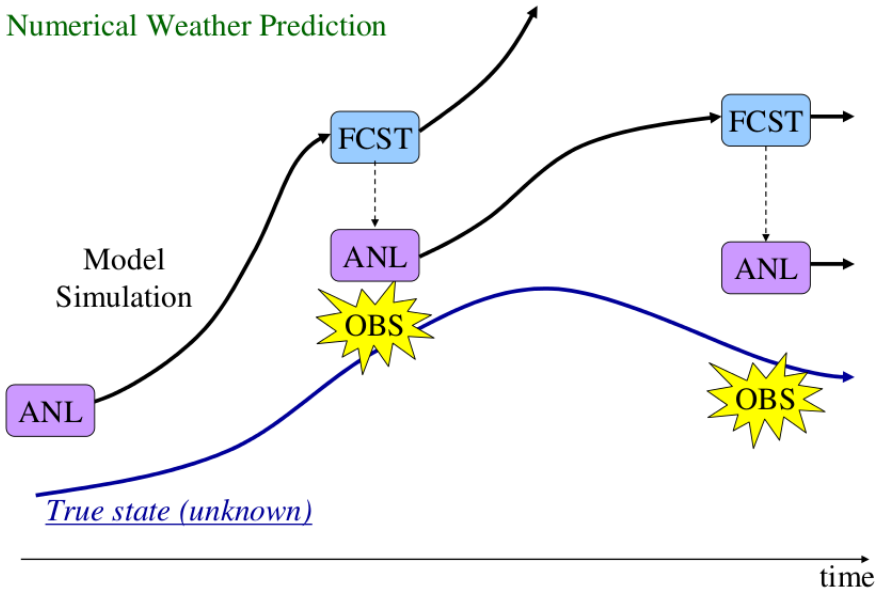


Fig. 5.4: Asimilación de datos en pronóstico numérico del tiempo. Fuente: [Kalnay et al., 2008].

En la asimilación de datos la calidad de las observaciones[3] le permite al método ajustar cuán importante será a la hora de influir en el resultado final. En términos prácticos:

Mucho error en la observación, poca influencia. Poco error, más influencia. Notar que el ajuste también depende de la varianza del error de Background que cambia día a día y de un punto a otro.

En los métodos como IO y 3D-VAR, la matriz de covarianza de los errores permanece constante en el tiempo y se obtiene mediante un estudio climatológico. No tienen en cuenta

[3] Utilizando el error cuadrático medio.

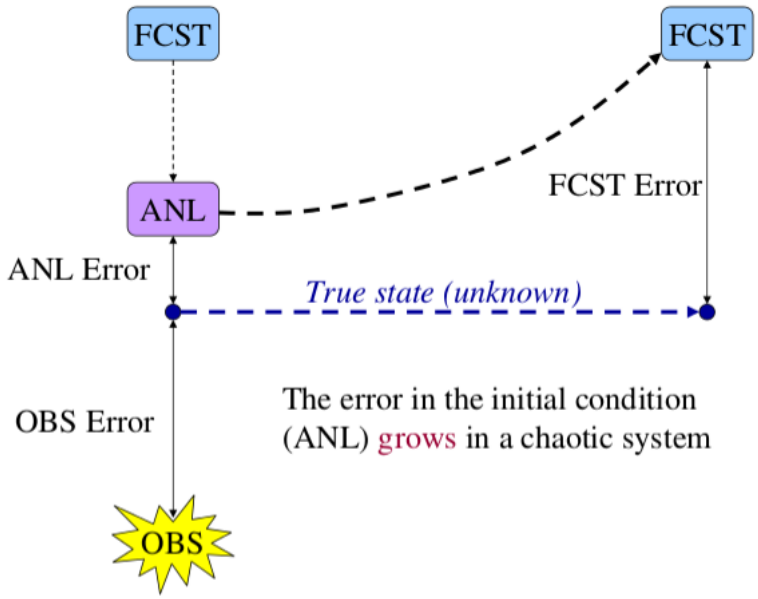


Fig. 5.5: Asimilación de datos = Análisis. Fuente: [Kalnay et al., 2008].

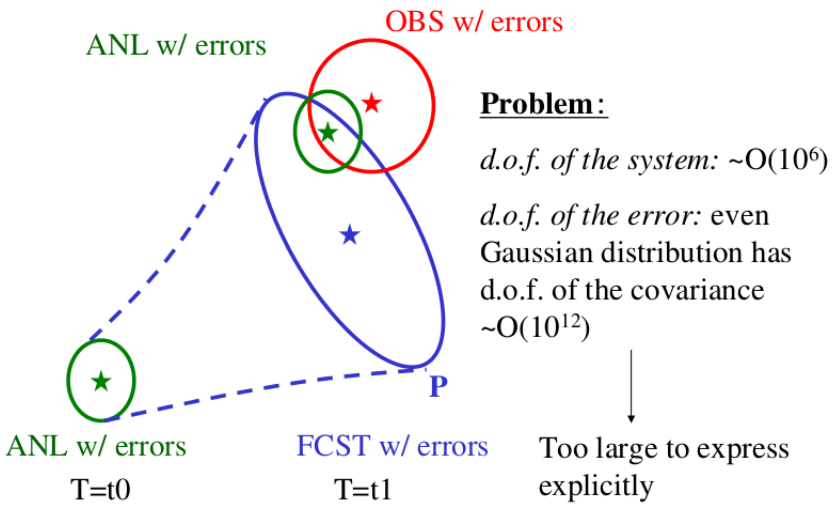


Fig. 5.6: Mirada probabilística. Fuente: [Kalnay et al., 2008].

que los errores pueden cambiar en el corto plazo.

5.3. Métodos de asimilación

Entre los métodos avanzados existentes y más utilizados están:

1. el método variacional 4-D (4-D Var);
2. el Filtro Kalman extendido (EKF);
3. el Filtro de Kalman por conjuntos o Ensemble Kalman Filter (EnKF).

El primero es el método que probablemente brinda los mejores resultados pero es extremadamente complejo y costoso en recursos computacionales y humanos. Requiere del modelo adjunto (en el sentido matemático del término) del modelo de pronóstico y, por lo tanto, su actualización ante modificaciones del mismo. El segundo es computacionalmente impracticable para los problemas meteorológicos dada las dimensiones del problema, ya que requiere el cálculo explícito de una matriz de covarianza en cada paso de integración, por ejemplo con $10^5 \times 10^5$ elementos en nuestro caso, además de necesitar el modelo tangente lineal. El EnKF plantea que la incertidumbre del pronóstico en el momento del análisis se puede conocer mediante un conjunto de estados del sistema levemente perturbados. Este conjunto se obtiene a partir de distintas ejecuciones del modelo o de varios modelos de pronóstico, con diferentes condiciones iniciales que permitan representar la incertidumbre de manera realista.

5.4. 4D-LETKF

Entre las numerosas variantes que tiene el EnKF para el paso de análisis, se seleccionó el 4D-LETKF (Local Ensemble Transform Kalman Filter) desarrollada por [Hunt et al., 2007], el cual constituye un método de asimilación por conjunto que además realiza una transformación. La elección de este método responde a que es un método avanzado, de resultados comparables al 4-D Var [Greybush et al., 2011] y, si bien también es computacionalmente costoso, es sustentable porque su formulación **no depende del modelo de pronóstico**. Dado que es un método por conjuntos la matriz de covarianza es estimada por el mismo y el método permite obtener una estimación de los futuros estados del sistema, lo que hace que no sea necesario contar con el modelo tangente lineal. Salvo algunos detalles particulares del problema que se está estudiando el filtro se utiliza como una caja negra. En este caso el modelo de olas recibe datos (guess y observaciones entre otros) y produce un análisis sin tener en cuenta de donde proviene ni qué es. Otra característica destacable es que implementa una solución local que permite paralelizar fácilmente el problema y desarrolla la solución en el espacio definido por las perturbaciones del conjunto. En particular utilizamos la implementación desarrollada como producto de la tesis doctoral de [Miyoshi, 2005]. Otra característica destacable es que 4D-LETKF también utiliza una técnica para descartar covarianzas espurias, la cual consiste en aplicar una función que aumenta el error de la observación en forma exponencial a la distancia al punto donde estamos realizando la asimilación.

6. OBSERVACIONES

En el presente capítulo vamos a describir los distintos instrumentos con los que se obtuvieron las observaciones que luego fueron asimiladas. Corresponden a información de tres satélites:

- La hs de los altímetros a bordo de los satélites **Jason1** y **Jason2**.
- La *velocidad y dirección del viento* del ASCAT a bordo del **MetOp-A**.

Además de hs de los altímetros, las componentes del viento wu y wv, del ASCAT. Se puede obtener, entre otros datos, el error cuadrático medio (RMS) de la medición, información de suma importancia para la asimilación de datos, como se vió en el capítulo 5.

6.1. Altímetros

Citamos a continuación una breve explicación sobre los altímetros:

"El altímetro emite una onda radar y la analiza después de ser reflejada por la superficie. El nivel del mar es igual a la diferencia entre la distancia satélite-superficie (deducida del tiempo que tarda la onda en ir y volver) y la posición del satélite respecto a una superficie de referencia arbitraria (el centro de la Tierra, o una superficie regular que se aproxime a su forma real, el elipsoide de referencia). Además del nivel del mar, podemos conocer la altura de las olas y la velocidad del viento (a partir de la amplitud y de la forma de onda reflejada)."

"Aunque esto parezca simple sobre el papel, su realización es compleja. Para obtener una medida de precisión centimétrica (sobre una distancia de varias centenas de kilómetros), se necesita una enorme precisión en el cálculo de la posición del satélite dentro de la órbita. Sistemas de localización como Doris permiten reducir a 2 cm esta incertidumbre. También se ha de tener en cuenta cualquier mínima perturbación que sufra la onda radar. El agua, los electrones en la atmósfera, el estado del mar, etc. modifican el tiempo de su recorrido, provocando un cálculo de la distancia incorrecto. La medida de estos efectos perturbadores (a través de instrumentos anexos o mediante el uso de varias frecuencias) o su estimación a partir de los modelos, permite corregir la medida altimétrica."

Fuente: http://www.aviso.oceanobs.com/es/altimetria/principio.html

Video que ilustra el funcionamiento de los altímetros: http://goo.gl/qWQH4e

6.1.1. Satélite JASON 1

Poseidon 2 - Altímetro de radar con apuntamiento a Nadir[1] usando la banda C y KU para medir altura de ola. El altímetro opera sobre los 13.575 GHz (banda Ku) y 5.3 GHz (banda C); es el sensor primario de la misión Jason-1.

[1] En astronomía se denomina nadir (del árabe: nadir o nathir, opuesto) a la intersección entre la vertical del observador y la esfera celeste. Es decir: si imaginamos una recta que pasa por el centro de la Tierra y por nuestra ubicación en su superficie, el nadir se encuentra sobre esa recta, por debajo de nuestros pies. En sentido contrario se encuentra el cenit. Quiere decir que mira hacia la tierra.

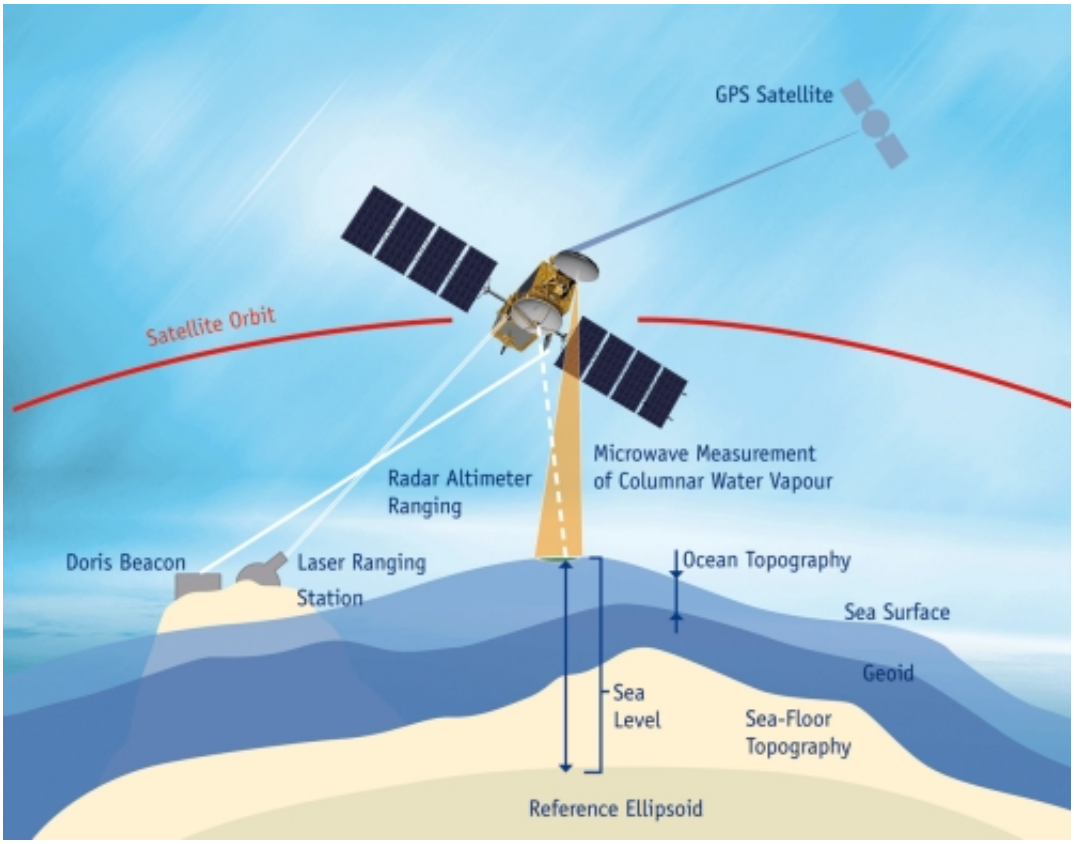


Fig. 6.1: Principios del funcionamiento de los altímetros de radar - Jason 1 y Jason 2. http://commons.wikimedia.org/wiki/File:Jason1_measurement_system.gif

Las mediciones hechas con las dos frecuencias combinadas obtienen las mediciones del altímetro, velocidad del viento (sólo intensidad, no dirección), hs y corrección ionosférica. El Poseidon-2 consiste en un altímetro dual redundante, con cada una de las partes diseñada de manera tal que su peso sea bajo y con poco consumo eléctrico.

Cada 10 días, Jason-1 mide la altura del mar a más del 90 % del océanos del mundo sin hielo; este altímetro de radar completa 127 revoluciones u órbitas alrededor de la tierra. Jason-1 tiene cinco instrumentos:

- Poseidon 2 - Altímetro.
- Jason Radiómetro de Microondas (JMR) - mide vapor de agua sobre la traza del satélite para corregir el retardo del pulso de microondas.
- DORIS (Doppler Orbitography and Radiopositioning Integrated by Satellite) para medir la orbita en 10 cm. o menos, y corrección ionosférica. para el Poseidon 2.
- BlackJack Global Positioning System.
- Laser retroreflector: trabaja con las estaciones de tierra para seguir el satélite, calibrar y verificar las mediciones del altímetro.

6.1.2. Satélite JASON 2

La Ocean Surface Topography Mission (OSTM), es una misión internacional conjunta de NOAA, National Aeronautics and Space Administration (NASA), Centre National d'Études Spatiales (CNES) y European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT), cuyo propósito es la observación satelital del planeta. Continua con la misión iniciada en 1992 en conjunto con la misión NASA/CNES TOPEX/Poseidon y NASA/CNES Jason-1 lanzada en 2001.

A bordo posee, entre otros instrumentos, un altímetro de radar, llamado Poseidon-3.

Fue lanzado el 20 de junio de 2008. Fue planeado para durar 5 años, hoy sigue funcionando.

- Altímetro Poseidon-3: Es el principal instrumento de la misión, provisto por el CNES derivado del poseidon-2, el altímetro del Jason-1, que funciona de la misma manera (ver la figura 6.1).
- También tiene un Advanced Microwave Radiometer (AMR) provisto por la NASA y mide la radiación proveniente de la superficie de la tierra en tres frecuencias (18, 21 y 37 GHz). Estas mediciones combinadas permiten medir el vapor de agua y el agua líquida, lo que permite aplicar la corrección necesaria.
- Entre otros, dos sistemas de localización se combinan para medir la posición exacta del satélite en órbita con una precisión de un centímetro en la componente radial.
  - Doppler Orbitography and Radio-positioning Integrated by Satellite (DORIS) provisto por la CNES.
  - Global Positioning System Payload (GPSP) y Laser Retroreflector Array (LRA) provistos por la NASA.
- Jason-2 tiene además, entre otros, dos nuevos instrumentos:
  - El instrumento de modelización y caracterización (Carmen-2) provisto por Francia y el Light Particle Telescope (LPT), provisto por Japón, que estudian la radiación en el entorno del satélite.
  - El Time Transfer by Laser Link (T2L2), también francés, consiste en detectores ultra precisos de tiempo de transferencia; T2L2 usa un link láser para la alta precisión y sincronización con los relojes en tierra.

Tanto Jason 1 como Jason 2 miden aproximadamente una observación cada 6 km. No tiene ancho de pasada.

6.2. ASCAT

6.2.1. Satélite METOP-A

Este satélite es parte del programa de satélites operacionales meteorológicos, Meteorological Operational Satellite Programme (MetOp); es un programa europeo que tiene como propósito proveer datos meteorológicos para el monitoreo del tiempo y la mejora en los pronósticos. El programa MetOp tiene una serie de tres satélites creados por

la European Space Agency (ESA) y por EUMETSAT, siendo parte del segmento Polar de la EUMETSAT (EPS). El programa consta de otros dos, MetOp-B y MetOp-C.

Los productos del ASCAT contienen la medición de la dirección y velocidad del viento sobre la superficie del mar, a 10 metros sobre el nivel del mar[2]. Utilizamos los datos procesados por el Koninklijk Nederlands Meteorologisch Instituut (KNMI) . El procesamiento es hecho lo más rápidamente posible (near-real time). Desde la última medición, procesar el dato toma aproximadamente 15 minutos. Dado que cada producto contiene datos de 30 minutos, los más antiguos tardan en estar disponibles aproximadamente 45 minutos.

Este producto está diseñado para NWP, donde la velocidad para obtener los datos es muy importante, a fin de poder realizar el pronóstico lo más rápidamente posible.

En la figura 6.2 se ve pasada del satélite en un intervalo de 22 hs; los colores representan intensidad de viento.

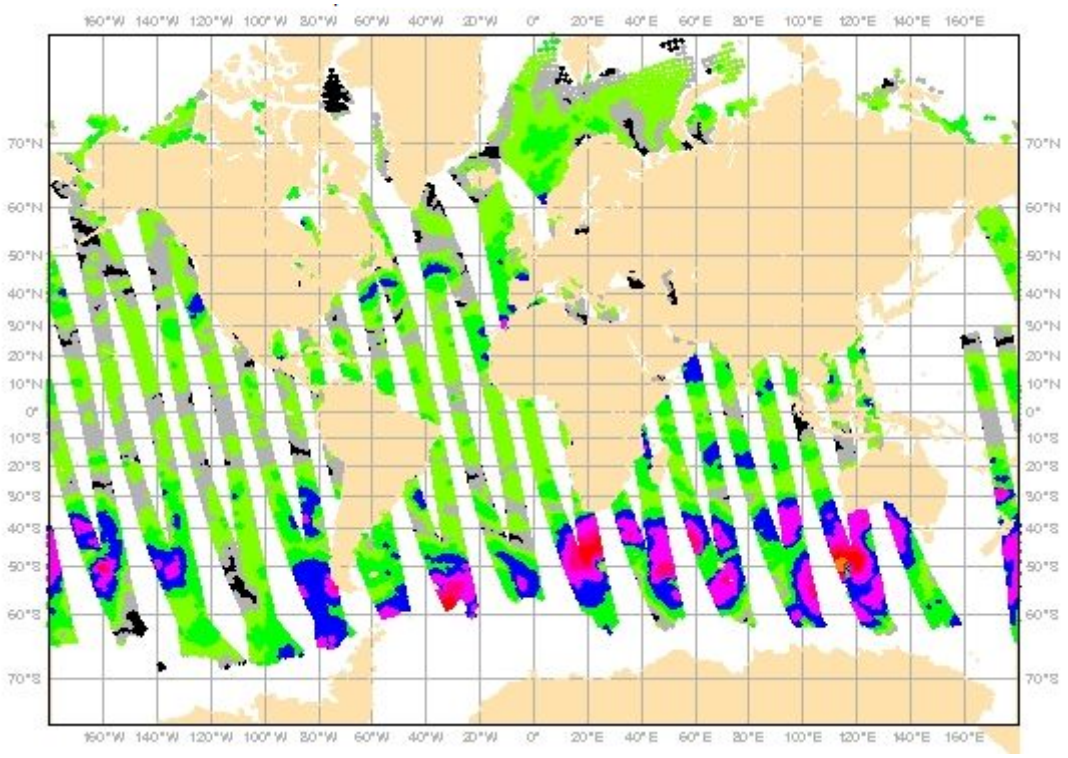


Fig. 6.2: Trazas del satélite Metop-A de 22 hs. Los colores indican intensidad del viento. Fuente: knmi.nl/scatterometer/

ASCAT utiliza un radar para medir la respuesta electromagnética de la rugosidad que produce el viento sobre el agua; usa microondas cuya longitudes de onda son de unos pocos centímetros; a partir de eso se procesa la señal de donde se infieren la dirección y velocidad del viento.

El ancho[3] de la pasada es aproximadamente de 500 km. La resolución espacial es de 12.5 ó 25 km. El tiempo en el que repite la órbita es de 29 días.

Una representación de cómo funciona puede verse en http://goo.gl/KoUjC4.

[2] Por convención.
[3] Swath.

6.3. Formato de archivos meteorológicos

6.3.1. BUFR

BUFR fue creado por la World Meteorological Organization (WMO). Técnicamente se conoce como "WMO Code Form M 94-IX Ext. BUFR". Este es el resultado de un comité, el cual produjo los primeros documentos en 1988. También es un código en el sentido que define un protocolo para la transmisión de datos cuantitativos, uno de un número de códigos creados por la WMO.

Fue diseñado para transmitir datos meteorológicos generalizados, pero debido a su flexibilidad puede usarse casi para cualquier cosa. De hecho, los archivos BUFR fueron diseñados para ser infinitamente extensibles, y para este fin están escritos en un lenguaje de descripción de datos único.

Los flujos de datos y archivos BUFR se adhieren a la especificación llamada WMO Standard Formats for Weather Data Exchange Among Automated Weather Information Systems.

6.3.2. NetCDF

NetCDF fue creado por UNIDATA, una empresa patrocinada por University Corporation for Atmospheric Research (UCAR), como formato estándar para que sea usado en los programas que ofrece a la comunidad científica. Este formato contiene, además de la información que se quería transmitir, metadata sobre ella: tipo de variable, unidades, dimensiones, institución que la creó, valores máximos, etc. Es un formato sumamente portable, se dispone de bibliotecas para casi cualquier plataforma.

6.4. Tiempo de procesamiento de las observaciones

Las observaciones son procesadas con cierto nivel de detalle, dependiendo de para qué se vayan a usar; si se necesitan a nivel operacional, esto es, para uso inmediato, entonces no hay mucho tiempo para ocuparse de esto; distinto es si se requieren para un estudio de la climatología, donde se puede esperar, para obtener datos de mayor calidad.

Antenas en tierra reciben la telemetría en crudo desde los satélites. Esta información debe ser procesada aplicando correcciones y complementando con otra información antes de ser utilizable.

Utilizamos datos de nivel 2, los cuales están georeferenciados, con cierta corrección y calibración. Están disponibles para ser utilizados en menos de 2 hs desde que el satélite pasa por la zona. Hay datos de nivel 0, 1, 2, 3 y 4. Cada nivel corresponde a un nivel de procesamiento, en el caso de 4 pueden estar disponibles hasta varios meses después y su interés es estadístico (climatológico).

7. CLUSTER

El cluster *Elementos*, ver figura: 7.1, compartido con el proyecto PIDEF 41/10, cuenta con dos nodos maestros[1] (Agua y Aire), uno de cada proyecto, que comparten tres nodos de trabajo (Madera, Tierra y Fuego) y de un servidor de almacenamiento masivo. Recientemente se incorporaros dos nodos más modernos (Hielo y Nieve). Por el momento y para realizar las pruebas, contamos con este último subconjunto del cluster completo, contando con un máximo de 16 cores.

Rol	Head Node
Nombre	**Agua**
Marca y Modelo	HP Proliant ML350 G5
Procesadores	2 x Intel® Xeon® CPU E5405 @ 2.00GHz, Cores 4, Threads 4
Memoria	16 GB DDR2 667 MHz Synchronous (4 x 4GB)
Storage	3 **320 GB** SAS 15K RPM en RAID 5 (disponible 2 x 320 GB)
Microarquitectura	Penryn

Rol	Head Node
Nombre	**Aire**
Marca y Modelo	HP Proliant ML350 G5
Procesadores	2 x Intel® Xeon® CPU E5405 @ 2.00GHz, Cores 4, Threads 4
Memoria	16 GB DDR2 667 MHz Synchronous (4 x 4GB)
Storage	3 **160 GB** SAS 15K RPM en RAID 5 (disponible 2 x 160GB)
Microarquitectura	Penryn

Rol	Node
Nombre	**Madera, Fuego, Tierra** 3 nodos
Marca y Modelo	HP Proliant ML350 G5
Procesadores	2 x Intel® Xeon®CPU E5405 @ 2.00GHz, Cores 4, Threads 4
Memoria	16 GB DDR2 667 MHz Synchronous (4 x 4GB)
Storage	2 x **250 GB** SAS 7,2K RPM en RAID 0
Microarquitectura	Penryn

Rol	Mass Storage
Nombre	**ms1**
Marca y Modelo	Dell PoweEdge R510
Procesadores	2 x Intel® Xeon®CPU E5503 @ 2.00GHz
Memoria	8 GB DDR3 1333 MHz Synchronous (2x 4GB)
Storage	5 x **2 TB** SAS 7,2K RPM en RAID 0

[1] head nodes

Rol	Head Node
Nombre	**Hielo**
Marca y Modelo	Dell PowerEdge R710
Procesadores	2 x Intel® Xeon®CPU E5620 @ 2.40GHz 4 cores
Memoria	12 GB DDR3 1333 MHz Synchronous Registered (Buffered) (6 X 2GB)
Storage	2 x 500 GB SAS 10k RPM
Microarquitectura	Nehalem

Rol	Node
Nombre	**Nieve**
Marca y Modelo	Dell PowerEdge R710
Procesadores	2 x Intel® Xeon®CPU E5620 @ 2.40GHz 4 cores
Memoria	12 GB 6 X 2GB DDR3 1333 MHz Synchronous Registered (Buffered)
Storage	2 x 500 GB SAS 7.2k RPM
Microarquitectura	Nehalem

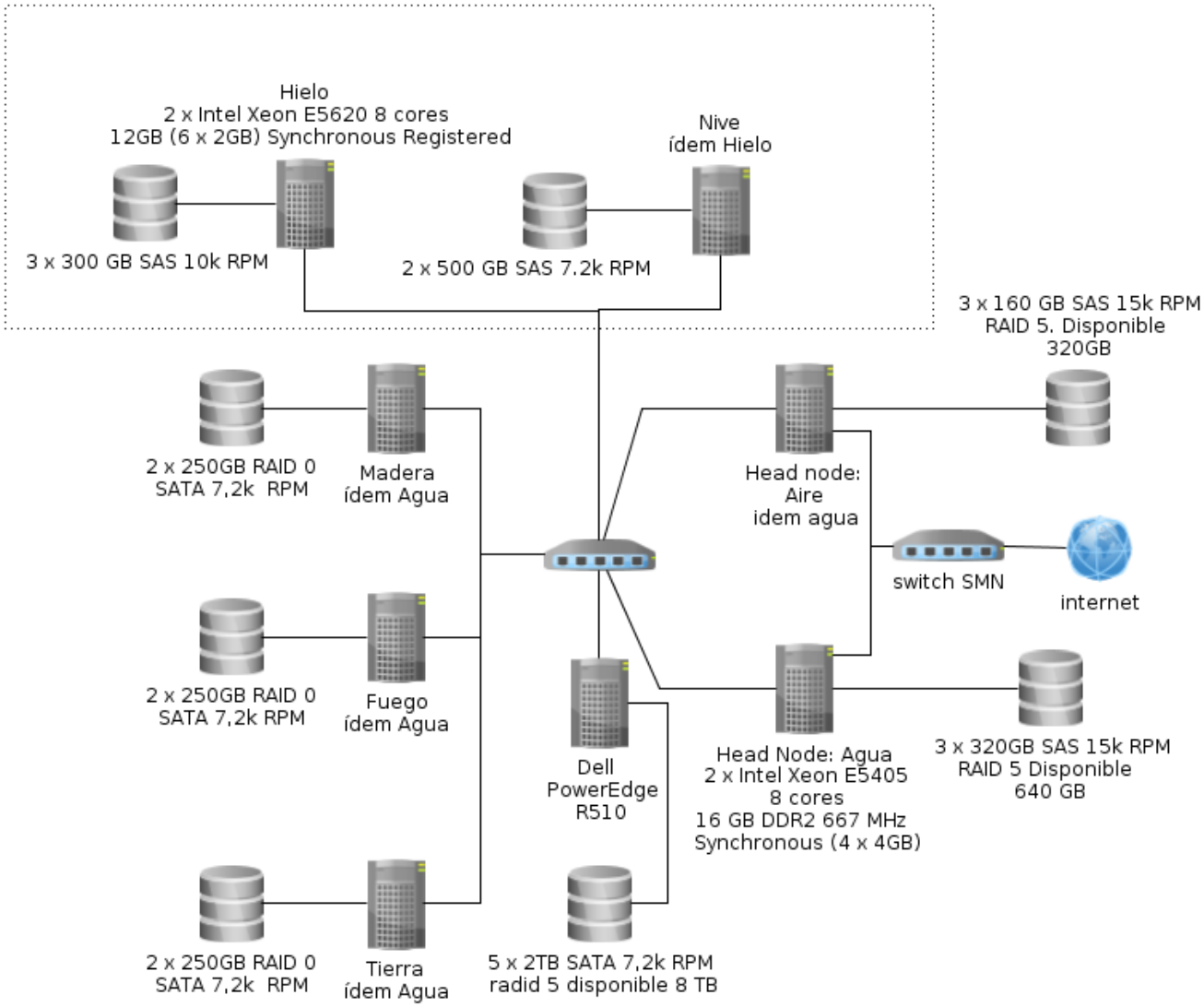


Fig. 7.1: Cluster Elementos.

8. DESARROLLO

Hasta este punto se hizo una descripción de las partes que conforman el problema. De ahora en más nos concentraremos en la implementación desarrollada partiendo del sistema que hoy está operacional en el SMN [Etala et al., 2012a]. Para esto nos planteamos los siguientes objetivos:

1. Desarrollar un método de adquisición sistemática de la información del conjunto de pronósticos de NOAA/NCEP;
2. Adaptar el sistema existente de pronóstico de olas para ejecuciones de conjunto de modelos basado en el conjunto de NOAA/NCEP;
3. Estudiar la dimensión del problema;
4. Instalar y adaptar el 4D-LETKF para la asimilación de altura de olas y viento;
5. Adaptar la interfase de entrada de datos para recibir la altura de olas hs obtenida por los satélites Jason-1, Jason-2 y vector viento (wu y wv) MetOp-A (ASCAT);
6. Incorporar el análisis como estado inicial del modelo de olas, modificando la interfase prevista para asimilación;
7. Cerrar el ciclo, utilizando el estado nuevo inicial para correr conjuntos de pronósticos;
8. Visualización de resultados.

Los lenguajes de programación elegidos fueron Bash, Python y Fortran. El uso de Fortran y Bash responde a razones históricas. En particular para Fortran una parte importante del software ya estaba programado (el modelo y el filtro). En otros casos sólo fue necesario modificar pequeñas partes de programas en funcionamiento. La elección de Python se debe, entre otros motivos, a la API para lectura de archivos (NetCDF y BUFR), facilidad para graficar siendo un lenguaje ideal para este tipo de aplicaciones, ya que permite optar un diseño orientado a objetos (sección 8.6) o bien scripting como Bash. Al optar por este lenguaje también se tuvieron en cuenta cuestiones culturales, dado que se puede programar en forma de *stripting*, esto ayuda a la migración. Se invirtió bastante esfuerzo en orientar el diseño a un uso operacional, respetando parte de los requerimientos.

Tanto el modelo como el filtro fueron compilados con Ifort de Intel, aplicando switches (-O2 -axSSE4.2) de optimización de nivel dos, y específicas para la arquitectura del cluster (ver capítulo 7).

En la figura 8.1 se puede ver un ejemplo de una etapa del ciclo de asimilación.

En la figura 8.2 puede verse un ejemplo de un ciclo de asimilación haciendo foco en la linea temporal y cada cuánto se ejecuta el modelo WWIII® para un miembro del conjunto. El ciclo de asimilación es de seis horas. La inicialización del conjunto se hace ejecutando el modelo con campos de viento del GEFS y campos de hielo del GFS.

8.0.1. Asimilación

Partiendo de un mar en reposo lo primero que se hace es correr una simulación sin asimilación de modo tal de obtener un estado el mar realista, el modelo utiliza dos inputs: campos de viento y hielo. El resultado es un conjunto de estados que sirven de punto de partida para la asimilación.

Dado un intervalo de fechas entre las cuales se quiere ejecutar el ciclo de asimilación, se itera para cada hora de análisis: 0, 6, 12 ó 18 hs. Greenwitch Mean Time (GMT) (ver figura 1.1). Produciendo un ciclo de asimilación de 6 hs, cuyos pasos son los siguientes:

Paso de pronóstico

1. Obtención de vientos del GEFS, el retroanálisis [1] para la hora de asimilación y el pronóstico a 6 hs.

2. Ejecución del modelo por un plazo de 9 hs. En este paso se genera un archivo restart.ww3[2] a la hora de análisis-1; el mismo será usando al final del ciclo, en el paso 5.

Paso de asimilación

3. Obtención y decodificación de observaciones. Discretización de las observaciones en slots de 1 hora. Preparación de observaciones para ser leídas por el filtro de asimilación.

4. Preparación de entorno y ejecución del filtro 4D-LETKF. Enlace a archivos con ganancia (inflation) aditiva (ver sección 8.4).

5. Ejecución del modelo para incorporar el resultado de la asimilación que corresponde a pasar del campo analizado de hs, wu y wv al espectro de energía.

6. Almacenamiento de los resultados para este ciclo de asimilación y del estado del modelo almacenado en los archivos restart.ww3, uno por cada miembro del conjunto (ensemble).

Este algoritmo está implementados en el script: *scripts/asimilacion/runcycle.py*.

Las observaciones de viento incorporan información al modelo dos maneras, una es vía las covarianzas cruzadas hs/wu hs/wv que produce una corrección en hs que es utilizada en el paso 5 (WW3-DAS) y la otra es utilizando directamente el campo de viento analizado en la distribución de la energía en el espectro de ola (todo se explica en la sección 8.7).

[1] En ingles Hindcast es mejorar un pronóstico pasado con los datos observacionales; lo que da como resultado es una mejor historia la cual servirá como base para futuros pronósticos.

[2] Este archivo almacena el estado del modelo.

8.0.2. Hora de análisis

A las 0, 6, 12 y 18 hs. GMT se realizan las mediciones de las estaciones (lectura de sensores, radio sondeos, etc.); por esta razón los centros meteorológicos mundiales realizan sus pronósticos coordinados a la misma hora. Por la misma razón es la hora en la que se realiza la asimilación de datos.

8.0.3. Organización del código

La organización del código es la siguiente:

- **ensamble**: es carpeta de ejecución del filtro, contiene una carpeta por miembro ($1 \cdots 20$), se hace una copia de los archivos necesarios, los archivos *guess*[3] (20 miembros por 9 slots) y links a las observaciones (obs01.dat $\cdots$ obs09.dat), un archivo por cada slot, un archivo *gridsettings* con parámetros de grilla (longitud y latitud inicial y delta). También se hace una copia de cada archivo guess como addi*nnn*.hs.bin, addi*nnn*.wu.bin y addi*nnn*.wv.bin, que corresponde a la inflación aditiva que es leída por el filtro.
- **scripts:** organizados por tipo: obs, LETKF, WW3, etc.
- **WW3:** contiene el modelo (fuente y ejecutable).
- **LETKF:** contiene el filtro (fuente y ejecutable).
- **results:** se almacenan por fecha y hora de análisis los resultados del filtro (análisis, spread y media), archivos NOUT-000, con información proveniente del filtro, los archivos guess, las observaciones. Estos datos servirán para el estudio de resultados.
- **obs:** se almacenan los archivos con las observaciones.
- **data:** Esta carpeta almacena la información descargada de los servidores: los campos de viento del GEFS y la información de los tres satélites, tanto los datos en formato crudo como en formato descomprimido (producto de la decodificación).

[3] Es la traducción al inglés de *adivinar* y son los pronósticos que tuvieron como base el análisis del ciclo anterior.

8.1. Desarrollar un método de adquisición sistemática de la información del conjunto de pronósticos de NOAA/NCEP

Desarrollamos un script que se conecta con el servidor de NOAA (ftpprd.ncep.noaa.gov) y descarga los 20 miembros del conjunto (ensemble) de campos de viento. Obtiene el retroálisis (hindcast) para la hora de interés y el pronóstico a 6 hs. Por ejemplo si se obtuvo para las 12 hs, también obtenemos el pronóstico para las 18 hs.

Las salidas del GEFS son muy complejas, se dispone de varios días y de diversas variables. Para no hacer una descarga de archivos con datos que no son necesarios, y que además ocupan un tamaño no despreciable, NCEP desarrolló un script que permite seleccionar sólo los campos de interés en el servidor y sólo hacer esa descarga[4]. En caso de que haya algún problema en la comunicación, como estrategia para obtener tolerancia a fallas, se reintenta n veces (con $n = 5$). Una vez obtenidos los archivos con formato GRIB2, son procesados para obtener los campos de viento y dejarlos listos para el preprocesamiento del modelo. Para esto se utiliza el script *scripts/ww3/make_wnd.sh*; este script utiliza el programa wgrib2 para extraer los campos. Una vez extraído el campo, se llama a "plot_sflux.sh" y "plotWND_ensamble" cuyo propósito es construir un archivo con los siguientes datos:

- encabezado con la fecha y hora del input;
- la componente u;
- la componente v.

Se los deja listos para que el modelo los incorpore en la etapa de pre-procesamiento. El modelo realiza la interpolación de los campos para tener disponible el forzante cada hora.

8.1.1. TIGGE

Como fuente alternativa también utilizamos THORPEX Interactive Grand Global Ensemble (TIGGE)[5] que fue desarrollado para propósitos experimentales; es un programa mundial de investigación meteorológica que tiene como objetivo acelerar las mejoras en la precisión de 1 día a 2 semanas en el alto impacto en los pronósticos climáticos. El archivo TIGGE consiste en un conjunto de pronósticos de diez centros NWP, y contiene datos desde 2006. Se utiliza para realizar experimentos en el pasado. En NOAA no se preservan los datos mucho tiempo.

[4] http://www.cpc.ncep.noaa.gov/products/wesley/fast_downloading_grib.html.
[5] http://tigge.ecmwf.int/

8.2. Adaptar el sistema existente de pronóstico de olas para ejecuciones de conjuntos de modelos basado en el conjunto de NOAA/NCEP

Se adaptaron scripts que preparan el entorno de ejecución del sistema modelo WW3 (WAVEWATCH III ©), de manera que tome los vientos producidos por el GFS. En la figura 8.3 puede verse el diseño. El orden de ejecución es de arriba hacia abajo.

Como input recibe el intervalo de tiempo a ejecutar el modelo: fecha y hora, iniciales y finales. También el tamaño del conjunto que se van a utilizar, lo cual facilita el desarrollo de experimentos.

La ejecución del modelo consta de 3 etapas:

1. **Pre-procesamiento:** en esta etapa, se ejecuta el script descripto en la sección 8.1 donde se descargan y preparan los campos de hielo (uno por día) y de viento uno para cada miembro del conjunto para la hora de análisis y un pronóstico a 6 hs. En caso de tener los campos de viento obtenidos de otra fuente como TIGGE (ver sección 8.1.1), como es el caso de los experimentos del presente trabajo, se puede suprimir la llamada a esta función y todo sigue funcionando.

2. **Procesamiento:** en este punto se ejecuta el modelo para correr una simulación de 9 hs, de manera de disponer de campos de vientos y altura significativa de ola 6 hs hacia atrás de la hora de análisis y 3 hs en el futuro. Se almacena el archivo restart.ww3 (el responsable de almacenar el estado del modelo) una hora antes de la hora de análisis, este servirá de base, para la etapa de asimilación.

3. **Post-procesamiento:** se extrae del archivo de estado la salida: 3 campos, hs, wu y wv, para las 9 hs de ejecución.

El modelo WW3 utiliza archivos con extensión *inp*, los cuales sirven de configuración para la ejecución de sus diferentes componente o etapas, pre-procesamiento, procesamiento y post-procesamiento.

Para el pre-procesamiento el modelo utiliza el archivo "ww3_prep.inp" donde se especifica qué tipo de *input* se le dará; luego de esto se ejecuta *WW3/exe/ww3_prep*[6] y de esta manera se preparan los campos de entrada. En nuestro caso tenemos 2 tipos de *input*, los campos de viento y los campos de hielo provistos por el GEFS y GFS respectivamente.

El ejecutable del modelo *WW3/exe/ww3_shel* utiliza el archivo "ww3_shel.inp". Debe hacerse had-hoc para cada ciclo de ejecución ya que tiene especificado el intervalo temporal de ejecución, que cambia en cada ciclo.

Estas etapas se ejecutan una vez para cada miembro del conjunto. El modelo mantiene su estado en el archivo llamado *restart.ww3* se puede configurar para almacenarlo cuando sea necesario; en nuestro caso lo almacenamos cada "hora de análisis -1" la primera etapa del ciclo (paso 1 de la figura 1.1) y uno para "hora de análisis" en el paso de asimilación (paso 5 de la figura 1.1).

Tanto *restart.ww3* como los archivos *.hs* y *.wnd* son almacenados en la carpeta */ensamble/#numero de miembro* de manera de poder recuperar el estado para la siguiente ejecución y para poder leerlo al momento de la asimilación.

[6] Se debe ejecutar dentro en *WW3/work*.

8.3. Estudiar la dimensión del problema

La resolución que utilizamos es de $1^o \times 1^o$; esto hace que el modelo tenga una discretización de la superficie del globo en una retícula de 146×360 puntos, da la vuelta al mundo en longitud (360°) y va de 80° latitud sur y hasta 65° en latitud norte, dado que hacia el norte no hay datos de interés para el modelado oceánico. Contamos con aproximadamente 36000 puntos "de agua"; esto es aproximado porque el hielo marino no es estático.

Por otro lado, en cada punto de retícula el modelo almacena la energía, en 24 direcciones y 25 frecuencias, lo que se denomina *bins*. Entonces, el modelo contiene el valor de la energía para cada tupla (punto húmedo, frecuencia, dirección). Dado que son 360×146 puntos por 24×25 bines por punto, en total se almacenan en el orden de (10^6) elementos.

En la figura 8.11 puede observarse la cantidad observaciones asimiladas en cada ciclo[7] para el caso de diciembre y en la figura 8.10 las del caso de abril.

[7] Son nombrados con la fecha-hora[yyyymmdd-hh].

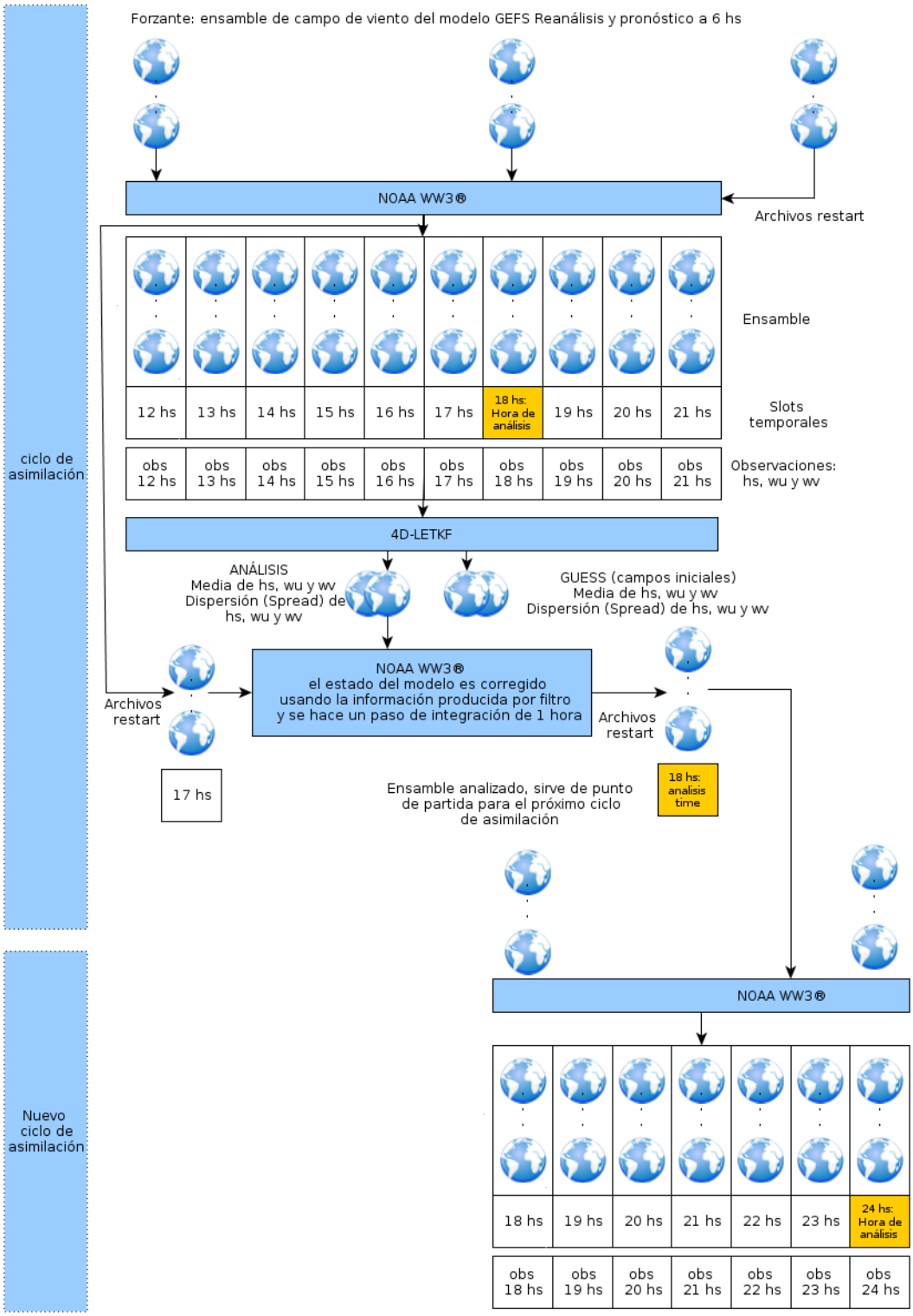


Fig. 8.1: Ciclo de asimilación para las 18 hs. El modelo recibe los campos de viento de las 12 y de las 18 como así también lee los archivos restart.ww3 (provenientes del ciclo anterior). Con este input genera un Ensemble de 20 miembros con hs, wu y wv para las 12, 13, ... , 21 hs. El paralelo las observaciones se descargan, decodifican y separan en slots, uno para cada hora (de 12 a 21). Esta información es leída por el filtro de asimilación el cual produce una salida de análisis para las 18 de hs, wu y wv que junto con un archivo restart.ww3, para las 17, es leída por el modelo el cual ejecuta la simulación de 17 a 18, donde se hace la incorporación de la información del análisis al modelo. Finalmente se cuenta con un Ensemble de archivos restart.ww3 con el estado completo del modelo para las 18 hs que sirve de punto de partida para el siguiente ciclo de asimilación.

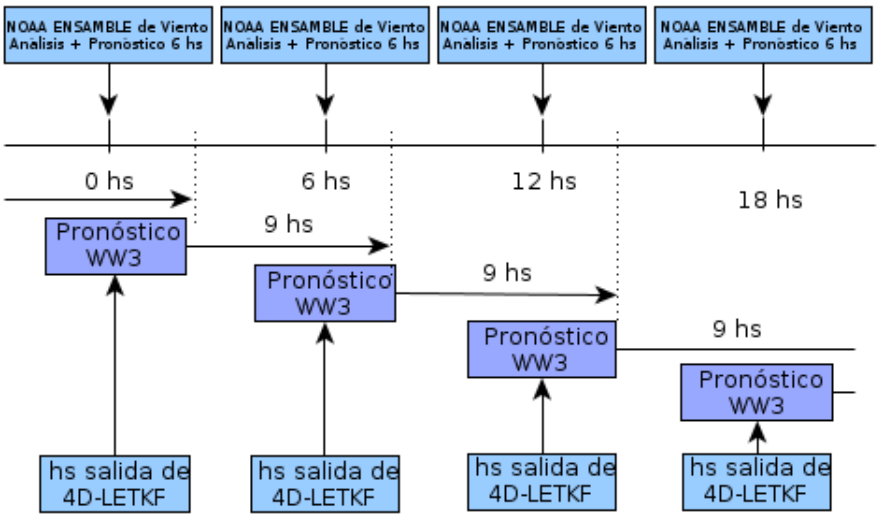


Fig. 8.2: Ciclo de asimilación - Pronóstico. Un ciclo cada 6 hs, pero el plazo de simulación del modelo dura 9 hs, es así para generar los datos que se utilizan en la ventana de asimilación

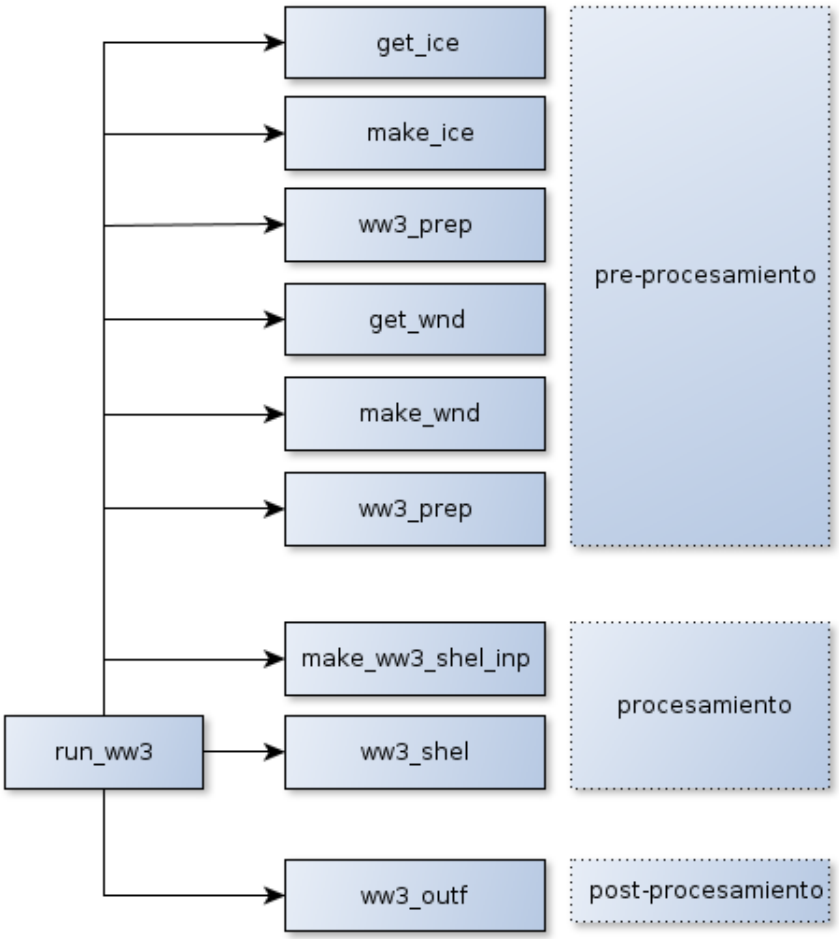


Fig. 8.3: Diseño del script que ejecuta un ciclo completo de pronóstico.

En la figura 8.4 se observa un campo de hs global (producto del modelo). No se modelan los mares interiores, como el mar Mediterráneo. También puede apreciarse, en verde, los campos de hielo marino cercanos a la Antártida. Se puede ver un punto con la información que es almacenada por el modelo para cada punto: el espectro de energía discretizado en frecuencia y dirección. Ese espectro lo que se puede ver para el punto cercano a África. Dado una latitud, una longitud, una frecuencia y una dirección se puede saber cuanta energía hay.

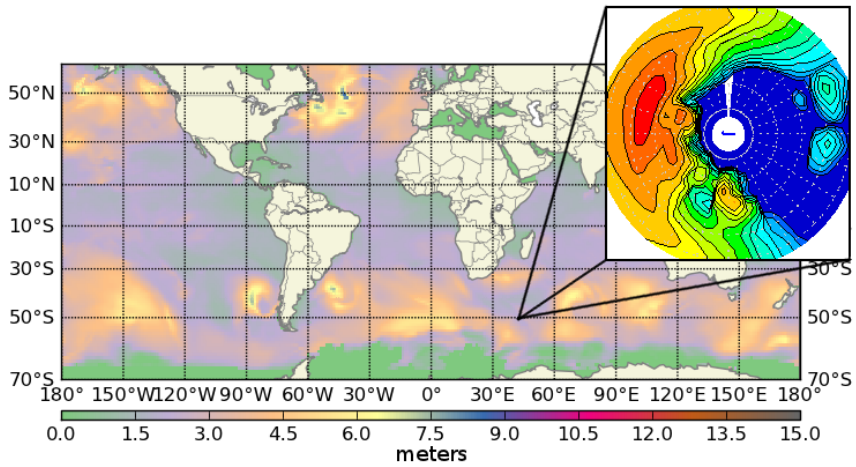


Fig. 8.4: Como ejemplo de como se modela el problema: Se puede ver un campo global de hs y en un punto, cercano a África, el espectro que muestra la descomposición en frecuencia y dirección de la energía de ola para ese punto.

Estadística de retícula:

Número de longitudes	360
Número de latitudes	146
Número de puntos de retícula	52560
Número de puntos de mar	36591 (69.6 %)
Número de puntos de tierra	14812
Número de puntos excluidos[8]	1157

Ejemplo de cantidad de observaciones (valor medio):

	wu	*wv*	*hs*
slot 1	11.005	9.063	2.668
slot 2	12.653	8.371	2.463
slot 3	12.199	9.292	3.491
slot 4	13.549	9.362	2.907
slot 5	12.397	8.938	3.024
slot 6	10.033	6.979	2.808
slot 7	9.343	7.764	3.181
slot 8	8.557	5.684	2.593
slot 9	7.105	6.150	3.011
Total	**96.841**	**71.602**	**26.146**

La diferencia entre wu y wv se debe a que son tratados como observaciones independientes. Hay casos en los que una componente pasa el control de calidad y la otra no o viceversa. Queda para trabajos futuros rediseñar el sistema para que sean tratadas como una misma observación.

8.4. Instalar y adaptar el 4D-LETKF para la asimilación de altura de olas y viento

Se utilizará como base el código para el 4D-Local Ensemble Transform Kalman Filter (4D-LETKF) de [Miyoshi, 2005], escrito es Fortran 90 y también disponible bajo licencia del MIT. La implementación de [Kosterlich, 2004], tal vez más cercana a la original de [Hunt et al., 2007], fue también cedida por el autor e instalada anteriormente. Sin embargo, la actualización de esta versión se encuentra algo discontinuada.

El primer código se probó sobre otro problema más sencillo y de menor dimensión, que es el de la modificación al nivel del mar por efecto meteorológico, con resultados preliminares satisfactorios [Etala et al., 2012b].

Para adaptar el filtro a las necesidades del nuevo modelo fue necesario agregar una nueva carpeta "LETKF/ww3". Se tomó como base la versión desarrollada para el modelo *ROMS*. Se suprimió todo el código sobre variables en 3D, dado que el modelo es en 2D; en el sentido que no tiene una discretización en alturas, nos limitamos a usar las rutinas de asimilación para variables en 2D.

El desarrollo teórico realizado por [Hunt et al., 2007] hace que en la implementación cada punto sea tratado independientemente, por lo que es naturalmente implementable en arquitecturas paralelas y en particular con MPI. En cada punto también se invierte una matriz.

8.4.1. Radio de localización

En el caso de olas oceánicas utilizamos 7 puntos de grilla (se setea en la variable sigma_obs en el archivo letf_obs.f90 y el valor corresponde a puntos de grilla) que corresponden a 700 km de nuestro modelado. La razón de esta elección se corresponde a la escala del fenómeno. Queda para trabajos futuros determinar el radio de localización óptimo.

8.4.2. Descripción de Módulos/Rutinas modificadas

A continuación damos una descripción de algunas de las partes del código, donde explicamos, brevemente, el propósito y las modificaciones introducidas.

- Programa principal: **4D-LETKF** en fuente en letkf.f90
  - Es el programa principal y es el encargado de hacer la llamada a la rutina **das_letkf**, Data Assimilation (DAS). Se incluyeron las rutinas propias del modelo con las sentencias USE common_ww3 y USE common_mpi_ww3.
- Módulo **letkf_tools**[9]
  - *das_letkf(gues2d,anal2d):* A esta rutina se le cambió la aridad[10]; antes también recibía el *gues3d* y devolvía *anal*3d. En esta rutina fue suprimido todo lo referente a 3D y un loop que itera sobre los niveles.

[9] Cada módulo está almacenado, en su correspondiente archivo, en este caso letkf_tools.f90.

[10] En el análisis matemático, la aridad de un operador matemático o de una función es el número de argumentos necesarios para que dicho operador o función se pueda calcular.

- *obs_local(ij,ilev,depth,hdxf,rdiag,rloc,dep,nobsl)* (hdxf_g,dep_g,rdiag_g) → (hdxf,dep,rdiag): En esta rutina es donde tuvimos que incorporar la "ciclicidad de dominio en la horizontal": el filtro no esta preparado para casos donde hay un dominio continuo como es nuestro caso. Puede verse el código en la sección 8.4.4.

- Módulo **common_mpi_ww3** : En este módulo se ataca el problema de la distribución de los datos en los diferentes procesos.

  - *set_common_mpi_ww3*: Se hace el cálculo de los puntos activos del proceso actual, **nijmax**.
  - *scatter_grd_mpi(nrank,v2dg,v2d)* : nrank → all, se ocupa de hacer un Scatter de los puntos, root envía a todos los otros procesos.
  - *gather_grd_mpi(nrank,v2d,v2dg)* all → nrank, root recibe los datos de todos los procesos
  - *read_ens_mpi(file,member,v2d)* En esta rutina se lee de disco el conjunto y se llama a scatter para distribuir los datos.
  - *write_ens_mpi(file,member,v2d)* Esta rutina se ocupa de llamar a gather y después almacenar los datos en disco.
  - *grd_to_buf(grd,buf)* y *buf_to_grd(buf,grd)* Este programa trabaja con los datos en un BUFFER interno y no con los datos reticulados. Estas rutinas hacen la conversión, en uno y otro sentido.
  - *write_ensmspr_mpi(file,member,v2d)* almacena el conjunto en disco.

- Auxiliares:

  - *countDryPoints(filename, cant, dryMasc)* Esta rutina cuenta los puntos secos.
  - *setLatLonXProceso (dryMasc, lonXproceso, latXproceso)*, Ésta se ocupa de asignar la longitud (lon.) y latitud (lat.) correspondientes a los puntos del proceso que esta ejecutando.
  - *setPuntosXProceso (dryMasc, riXproceso, rjXproceso)* ídem pero para puntos i y j.

- Módulo **common_ww3**

  - *Parámetros generales (dimensiones el problema, niveles, variables, puntos totales, latitud, longitud).*
  - *set_common_ww3* En esta rutina se agregaron los tipos de observaciones que nos interesan (hs, wu y wv). Se lee la configuración de grilla: dónde empieza y el delta entre cada punto "gridsettigs". Con esa información se define el reticulado.
  - *read_grd(filename,v2d)*: Lee el archivo proveniente del modelo y lo almacena en la variable v2d (reales de doble precisión).
  - *read_grd4(filename, v2d)*: ídem pero v2d tiene tipo de dato real de precisión simple.
  - *write_grd(filename,v2d)*: Escribe en disco lo que este en v2d.

- *write_grd4(filename, v2d)*: ídem pero v2d tiene tipo de dato real de precisión simple.

Se agregó la siguiente subrutina en **common/common_letkf.f90** :

```
SUBROUTINE init_letkf()

  CHARACTER(len=32) :: arg

  call getarg(1, arg)
  read(arg, '(I3)')nbv

  RETURN

END SUBROUTINE init_letkf
```

También se sacó la declaración **PARAMETER** de la declaración de **nvb** y en **ww3/letkf.f90** se agregó la llamada:

```
call init_letkf()
```

Estas modificaciones agregan la posibilidad de pasar por línea de comando la cantidad de miembros que se usarán para la asimilación. De esta manera se pueden realizar experimentos y debuguear. Cabe destacar que si el valor está fijo esto le permitiría al compilador aplicar optimizaciones, desenrollo de ciclos, por ejemplo; con el que se lograría un *código de máquina* que aproveche mejor las características del hardware. Este código será suprimido una vez que el sistema pase a producción.

8.4.3. Variables "harcodeadas" en el 4D-LETKF

```
common_letkf.f90
!=====================================================
!  LEKF Model Independent Parameters
!=====================================================
 INTEGER,PARAMETER :: nbv=20     ! ensemble size
 common_ww3.f90
!-----------------------------------------------------
! General parameters
!-----------------------------------------------------
  INTEGER,PARAMETER :: nlon=360
  INTEGER,PARAMETER :: nlat=140
  INTEGER,PARAMETER :: nlev=1
  INTEGER,PARAMETER :: nv3d=0
  INTEGER,PARAMETER :: nv2d=3 ! wu, wv, hs
!  common_obs.f90
  INTEGER,PARAMETER :: id_hs_obs=2821
  INTEGER,PARAMETER :: id_wu_obs=2819
  INTEGER,PARAMETER :: id_wv_obs=2820
! common_obs_ww3: 267
! como hay levels, nivel=0
```

```
   rlev(n) = REAL(0,r_size)

!letf_obs.f90
  INTEGER,PARAMETER :: nslots=10! number of time slots for 4D-LETKF
  INTEGER,PARAMETER :: nbslot=7 ! basetime slot
  REAL(r_size),PARAMETER :: sigma_obs=5.0d0 ! grids
  REAL(r_size),PARAMETER :: sigma_obsv=50.0d0 ! meters
  REAL(r_size),PARAMETER :: sigma_obst=3.0d0 ! slots

!letkf_tools.f90
 CHARACTER(11) :: inflfile='infl_mul.nc'
  REAL(r_size),PARAMETER :: cov_infl_mul = 1.1d0 !multiplicative inflation
! > 0: globally constant covariance inflation
! < 0: 3D inflation values input from a GPV file "infl_mul.grd"
  REAL(r_size),PARAMETER :: sp_infl_add = 1.d0 !additive inflation
                                              !verdadero valor en gen_inflacio


! letkf.f90
CHARACTER(8) :: stdoutf='NOUT-000'
CHARACTER(4) :: guesf='gs00'

! letkf_obs.f90:52
  REAL(r_size),PARAMETER :: gross_error=1.0d0
```

8.4.4. Ciclicidad de Dominio Horizontal

```
 ! CICLICIDAD DE DOMINIO HORIZONTAL
 ! jmin, jmax se mantienen ya que no hay ciclicidad de dominio vertical
  ! caso 1
  ! el punto ij esta en un lugar menor en nlon, pero hay puntos en
  ! el radio que ``pasan para el otro lado''
  !    ------|
  !   |      |
  !   |   ij |
  !   |      |
  !    ------|
  !        nlon
  ! agrego estos puntos
  !   |---
  !   |   |
  !   |   |
  !   |   |
  !   |---
  !   1
   if ((ri1(ij) + dist_zero)> nlon) then
      imin = 1
      imax = (ri1(ij) + dist_zero)- nlon   ! lo que ``me sobra''
```

```
                                                    !va para el otro lado
      nnTMP = 1
      CALL obs_local_sub(imin,imax,jmin,jmax,nnTMP,nobs_useTMP)
      nnTMP = nnTMP - 1
      ! copio las nuevas observaciones al nobs_use verdadero
      j=1
      do i=nn+1, nn+nnTMP+1
        nobs_use(i)=nobs_useTMP(j)
        j=j+1
      end do
      nn=nn+nnTMP
  end if
  if (ri1(ij) - dist_zero < 0)then
  ! caso 1
  ! el punto ij es mayor que 1, pero hay puntos en el radio que
  ! ''pasan para el otro lado''
  !   |------
  !   |      |
  !   | ij   |
  !   |      |
  !   |------
  ! 1 o 0
  ! Agrego este pedacito:
  !    --|
  !   |  |
  !   |  |
  !   |  |
  !    --|
  !      nlon
      imin = nlon - (ABS (ri1(ij) - dist_zero))
      !lo que sobra, en valor absoluto, ya se que es negativo,
      !pero para no perder generalidad menos nlon, ese es el minimo
      imax = nlon  ! maximo

      nnTMP = 1
      CALL obs_local_sub(imin,imax,jmin,jmax,nnTMP,nobs_useTMP)
      nnTMP = nnTMP - 1
      ! copio las nuevas observaciones al nobs_use verdadero
      j=1
      do i=nn+1, nn+nnTMP+1
        nobs_use(i)=nobs_useTMP(j)
        j=j+1
      end do
      nn=nn+nnTMP
  end if
```

8.5. Adaptar la interfase de entrada de datos para recibir la altura de olas hs obtenida por los satélites Jason-1, Jason-2 y vector viento (wu y wv) MetOp-A (ASCAT)

- Módulo **common_obs_ww3**:
  Ubicado en el archivo LETKF/ww3/common_obs_ww3.f90.

  Se modificó el programa de manera de agregar las siguiente variables:

  - id_wu_obs = 2819,
  - id_wv_obs = 2820,
  - id_hs_obs = 2821,

  las que se corresponden con los identificadores de cada tipo de observación, componente u y v del vector viento y hs.

  - SUBROUTINE Trans_XtoY(elm,ri,rj,rlev,v2d,yobs): Dada un i, j, y tipo de elemento (wu, wv o hs); retorna, en $yobs$, el valor del modelo en ese punto.
  - SUBROUTINE itpl_2d(var,ri,rj,var5): interpolador.
  - monit_dep(nn,elm,dep,qc): Esta rutina se ocupa de calcular el error cuadrático medio y el sesgo de las observaciones menos el modelo (O-F)[11], lo que constituye una medida de la distancia entre el modelo y a las observaciones.
  - get_nobs(cfile,nn): Se utiliza para contar la cantidad de observaciones que hay en $cfile$.
  - read_obs(cfile,nn,elem,rlon,rlat,rlev,odat,oerr): Esta rutina lee las observaciones.

[11] Observación menos Forecast(pronóstico).

8.6. Adquisición y preparación de datos de observaciones de satélite

Originalmente se pensaban obtener observaciones de cuatro satélites; tres de ellos hs y uno información del viento (intensidad y dirección o su equivalente wu y wv). Lamentablemente durante el desarrollo del presente trabajo ENVISAT (uno de los satélites que mide hs) dejó de enviar información y de responder a cualquier comando de tierra, por lo que dejó de estar operativo[12].

Por otro lado, contamos con un nuevo satélite, el MetOp-B[13], que terminó su etapa de calibración, por lo que sus datos comenzaron a estar disponibles, como así también el OceanSat2[14]; esperamos poder incorporarlas al ciclo de asimilación en trabajos futuros. (Ver capítulo 10). En la figura 8.5 se puede ver un esquema del sistema de adquisición de observaciones.

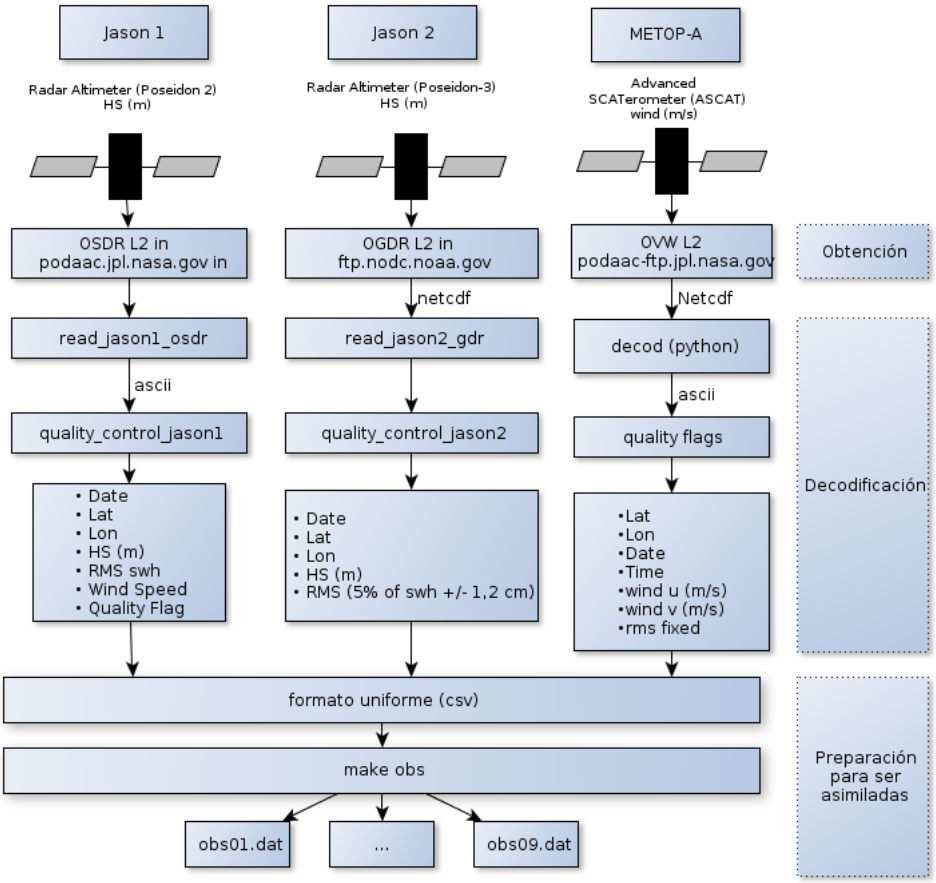


Fig. 8.5: Sistema de adquisición y procesado de observaciones. Se muestran los tres satélites que se utilizaron en la tesis. Las etapas de obtención desde el servidor ftp, decodificación y preparación para ser asimiladas por el filtro.

Esta etapa del trabajo se separa claramente en tres capas:

[12] Recientemente Jason 1 también dejó de estar operativo pero dado que no afecta a los propósitos prácticos de la tesis seguiremos desarrollándola como si esto no fuera así.
[13] http://www.eumetsat.int/metop-b/.
[14] http://www.isro.org/satellites/oceansat-2.aspx.

Obtención: corresponde a conectarse a distintos servidores y hacer la descarga de los archivos crudos, para la fecha de interés.

Decodificación: el formato de observaciones depende de cada centro así que esta etapa también es variable. Tiene su propia lógica y puede cambiar. También se hace un control de calidad utilizando flags que vienen junto a las observaciones.

Preparación de las observaciones para ser asimiladas: una vez obtenidas las observaciones es necesario preparar los datos de manera que el filtro las espera y separarlos en los archivos correspondientes a cada slot[15], los cuales se mapean a intervalos de tiempo de una hora.

8.6.1. Obtención y decodificación

El diseño elegido fue uno orientado a encapsular cómo se obtienen las mismas (en nuestro caso son tres servidores ftp, pero podría ser desde cualquier otra fuente). También la decodificación es no trivial y podría cambiar. En la figura 8.6 se puede ver un esquema de cómo se pensó e implementó el sistema.

Por esto se diseño una herencia, que tiene un objeto abstracto "Satellite" y de este se heredan las tres entidades (Jason1, 2 y MetOp A); en el script "getDecod.py" se crea una lista con estas entidades y usando polimorfismo se itera sobre estas llamando a los métodos, satellite.decod_obs(date) ó satellite.get_obs(date) según el caso; de esta manera extender el algoritmo a nuevos satélites o dejar de usar alguno es muy simple.

Obtención de observaciones:

Jason 1: se descarga desde *podaac.jpl.nasa.gov*, del archivo con la fecha de interés.

Jason 2: se descarga desde *ftp.nodc.noaa.gov*, con el pequeño detalle de que las observaciones están en carpetas numeras por ciclo, por lo que hay que calcularlo. Esto se hace de la siguiente manera: un ciclo se repite cada $9,9$ días aproximadamente; se comienza a contar desde que el satélite fue lanzado ($8/7/2008\ 00:00:00$); para calcular el número de ciclo de la fecha dada se programó una rutina que lo hace de acuerdo con [CLS et al., 2009]. Pero este cálculo no es exacto. Entonces hasta tanto no se disponga de otra fuente, dada una fecha se busca coincidencia con la fecha de la observación, que está en el nombre del archivo; ampliando el rango a $+1$ ciclo, -1 ciclo del que calculó en el script, de modo de asegurarnos de encontrar siempre el archivo. Quedará pendiente hasta tanto no se ponga operativo (en producción); por ahora no es preocupante ya que trabajamos con experimentos puntuales.

MetOp A: de este satélite originalmente se hacían las descargas desde el KNMI, pero dado que eran eliminadas relativamente rápido, decidimos utilizar **podaac-ftp.jpl.nasa.gov** entrando a: */allData/ascat/preview/L2/12km/**year/numeroDeDia***.

Todo el manejo de tiempo, cálculo de número de día del año, es muy simple utilizando Python; en las primeras pruebas y por compatibilidad y cultura del grupo intentamos utilizar Bash, pero encontramos muy poco productiva esta estrategia.

[15] Discretización temporal de una hora.

Decodificación de observaciones

De igual manera funciona la otra etapa que es la de **decodificación**, ya que contamos con distintos tipos de formatos (netCDF, Bufr, etc.). Cada satélite implementa la rutina de decodificación adecuada.

Jason 1: Utiliza un decodificador desarrollado por Kelley Case en 2001 en el California Institute of Technology; fue modificado para que tome los nombres de archivo por línea de comando. El código fuente se encuentra en *scripts/obs/jason1/scr/read_jason1_osdr.f* y está escrito en Fortran77. Graba el archivo en formato ASCII.

Jason 2: En este caso utilizamos un decodificador desarrollado por Eric Leuliette para NOAA Laboratory for Satellite Altimetry; consta de 2 rutinas: *read_gdr_netcdf* y *dump_gdr*; esta última fue modificada para guardar los datos de manera conveniente. El código fuente se encuentra en *scripts/obs/jason2/scr/read_jason2_osdr.f90* y está escrito en Fortran90. Como en el caso de Jason 1 también se pasa un control de calidad: se lee de *grilla_limits* y se ve que lat y lon estén en rango; también en los archivos de observaciones contamos con un campo llamado XX_fill, donde XX es la variable, que indica un valor de campo faltante; ante la ausencia de algún campo se elimina el registro completo. También se controla que hs esté entre 0 y 25 ($m.$), que el quality_flag $\neq 1$ y que el rms esté entre 0 y 60 (m/s).

MetOp A: En cuanto a este satélite, contamos con dos decodificadores; uno de ellos es el provisto por el KNMI que permite leer archivos BUFR y otro desarrollado íntegramente por nosotros que es utilizado para leer archivos NetCDF. El provisto por el KNMI fue modificado para tomar por línea de comandos los nombre de archivo *in* y *out*: <file, list of BUFR files> < output file u (ASCII) > < output file v (ASCII)> <Grads binary file>; originalmente decodificaba en un sólo archivo, pero el filtro toma cada componente por separado por lo que fue modificado para que se produzca directamente así. También se preparan salidas en Grid Analysis and Display System (GrADS). El desarrollado para el presente trabajo fue escrito en Python y utiliza la biblioteca **netcdf4-Python**[16], que permite acceder a los registros del los archivos netcdf como se explora un objeto, utilizando el punto para acceder a propiedades y valores, metadada, variables, etc..

Utilizamos los archivos ascci para almacenar las observaciones por que hacen muy sencillo el desarrollo; queda para el futuro pasar a un formato binario que es más eficiente en el uso del espacio en disco y la velocidad de I/O.

8.6.2. Control de Calidad de las Observaciones de Jason 1 y Jason 2

En el primer control de errores, se descartan las mediciones de hs fuera del intervalo: $0 < Hs < 50m$. Además del control específico de hs, se adopta un criterio general compatible con el producto Near Real Time (NRT) Sea Surface Height Anomalies (SSHA) desarrollado en [qcJ, 2003]. Dado que las observaciones OSDR no tienen un flag indicando si hay hielo o no, se utiliza el campo de hielo provista por el GFS.

[16] https://code.google.com/p/netcdf4-Python/.

Se realiza una verificación de consistencia horizontal de los datos a lo largo de la traza del satélite. En el conjunto de datos OSDR los puntos son aproximadamente equidistantes. Un grupo de 10 datos abarca aproximadamente $0,5^o \times 0,5^o$. El control consiste en comparar cada dato con el promedio de dicho grupo de datos, pesando cada uno en forma inversamente proporcional al RMS informado.

Citamos a continuación parte del trabajo de [Etala et al., 2005] donde se explica el control de calidad aplicado a los altímetros que utilizamos en el presente trabajo: "No se considera ninguna medida de la distancia en el peso, porque se privilegia la influencia del rms sobre la validez del dato. El dato cuya diferencia sea mayor que el 15 % del valor del promedio es señalado. Una vez completada la órbita, se repite el procedimiento, suprimiendo los datos marcados. De esta manera puede ser sospechado un dato anteriormente aceptado o, por el contrario, rehabilitado un dato marcado. Se repite una vez más el control, suprimiendo el nuevo conjunto de datos sospechados y con el resultado de esta tercera y última iteración se asigna el "flag" definitivo a cada dato ($0 = aceptado$, $1 = rechazado$, $9 = nosepudoaplicarelcontrol$). En promedio, aproximadamente un 13 % de los datos son rechazados, mientras que a un 6 % no se le puede aplicar el control según lo señalado anteriormente. Sólo se utilizan los datos con $flag = 0$".

8.6.3. Preparación para ser asimiladas

Una vez obtenidas y decodificadas las observaciones, se procede a "llenar" los archivos ubicando las observaciones en el slot temporal que le corresponde, de acuerdo a cuando fueron tomadas. En este caso los slots temporales van de 1 a 9.

El filtro recibe las observaciones con el siguiente formato:

idObs lat. lon. valor rms.

Para ello, como antes, se utiliza el polimorfismo y dos métodos de la clase "Satellite".

parsear() : Lee los archivos decodificados; si es necesario lleva la longitud al intervalo(-180,180).

prepararObs() : Lee los archivos, usando el parser, para una fecha dada y les agrega el id para esta observación (hs= 2821, wu= 2819, wv= 2820) devolviendo un array con todas las observaciones: id_obs, lat., lon., Observación (obs), RMS, Timestamp. En el array quedarán todas las observaciones de interés, luego se procede a separarlas en slots.

guardarSlots : Para la fecha y hora de análisis itera y aplica un "filter" de numpy sobre el array, de manera de separar en *slots*; en cada slot se almacenan observaciones con una discretización temporal de una hora. Luego es guardado en disco, dejando listas las observaciones para ser utilizadas por el 4D-LETKF.

Dado que las observaciones, una vez decodificadas, se almacenan en disco, tuvimos que implementar un parser para poder leerlas.

También se implementó un test de las siguientes propiedades:

1. Están tipados correctamente, todos *float* salvo el último que es *date*.
2. id obs es: 2819, 2820 ó 2821.

3. $-80 <= lat <= 60$.
4. $-180 <= lon <= 180$.
5. si es hs $\Rightarrow$ está entre 0 y 25 y si es wu o wv $\Rightarrow$debe ser $< 60\ m/s$ aproximadamente $200km/h$.
6. rms es $< 50\,\%$ de la obs.

Esta etapa fue el resultado de una primera iteración en el desarrollo; seguramente, en futuras mejoras, se suprimirá el uso de archivos intermedios en disco y se separará la funcionalidad de guardar y de distribuir en slots.

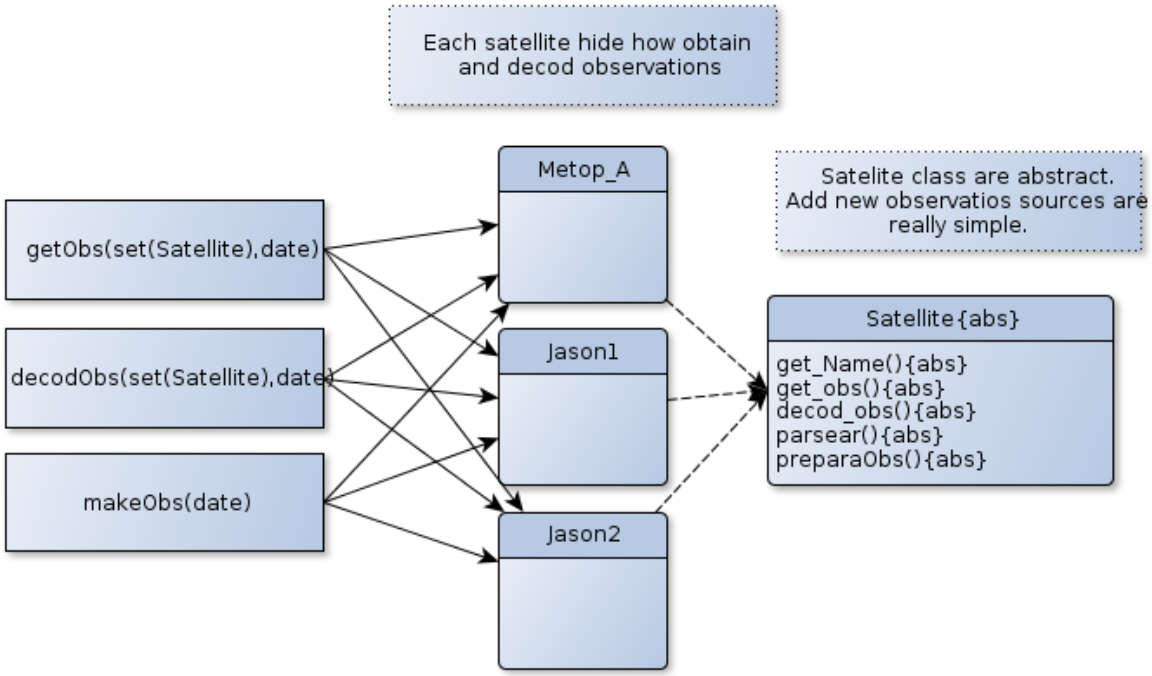


Fig. 8.6: Diseño del sistema de preparación de las observaciones. Utiliza polimorfismo

8.7. Incorporar el análisis como estado inicial del modelo de olas, modificando la interfase prevista para asimilación

Luego del paso de asimilación obtenemos, como resultado, el campo altura significativa de ola analizado (hs_a). Por otro lado, el modelo tiene almacenado (en los archivos "restart.ww3") la energía descompuesta en frecuencia y dirección para cada punto que es la variable de pronóstico. Entonces,

¿Cómo incorporar esta nueva información (hs_a) al modelo?

Existe una ecuación que relaciona la altura significativa de ola hs con la energía E:

$$hs = 4\sqrt{\int\int E_{f\theta} df d\theta}.$$

Para referirnos a la hs almacenada en el modelo usamos hs_f la f es por pronóstico, forecast en inglés. En este punto del desarrollo tuvimos dos iteraciones. En la primera, como se detalla más adelante, buscamos hacer una corrección del espectro escalándolo; lamentablemente con esta solución no se lograron buenos resultados. En la segunda etapa y siguiendo la bibliografía [ecm, 2012b], implementamos un algoritmo que separa Mar de viento y el Mar de fondo, para cada tupla: (lat, lon, dirección, frecuencia). Luego se calculan dos constantes A y B que sirven para obtener el nuevo valor de la energía dado por la ecuación:

$$E^a = AE^f(Bf, \theta).$$

8.7.1. Etapa 1 en el proceso de desarrollo: Escalar HS

Tanto hs_f como hs_a tienen la misma dimensión. Dados hs_a analizado y hs_f pronosticado existe una relación proporcional para cada punto:

$$\alpha hs_a = hs_f.$$

Usando la igualdad anterior:

$$4\sqrt{\int\int E^a_{f\theta} df d\theta} = \alpha \cdot 4\sqrt{\int\int E_{f\theta} df d\theta}.$$

E^a es la matriz con la energía que se quiere determinar, pero sólo conocemos el valor de la altura de ola resultado de la integral. Entonces, usando la relación que hay entre hs_f y hs_a podemos pasar adentro de la integral el valor α de manera de "escalar" la función:

$$hs_f = 4 \cdot \sqrt{\int\int E_{f\theta} df d\theta},$$

$$hs_a = \alpha \cdot (4 \cdot \sqrt{\int\int E_{f\theta} df d\theta}),$$

$$hs_a = 4\sqrt{\int\int \alpha^2 \cdot E_{f\theta} df d\theta},$$

Simplificamos entonces la relación, que queda expresada como

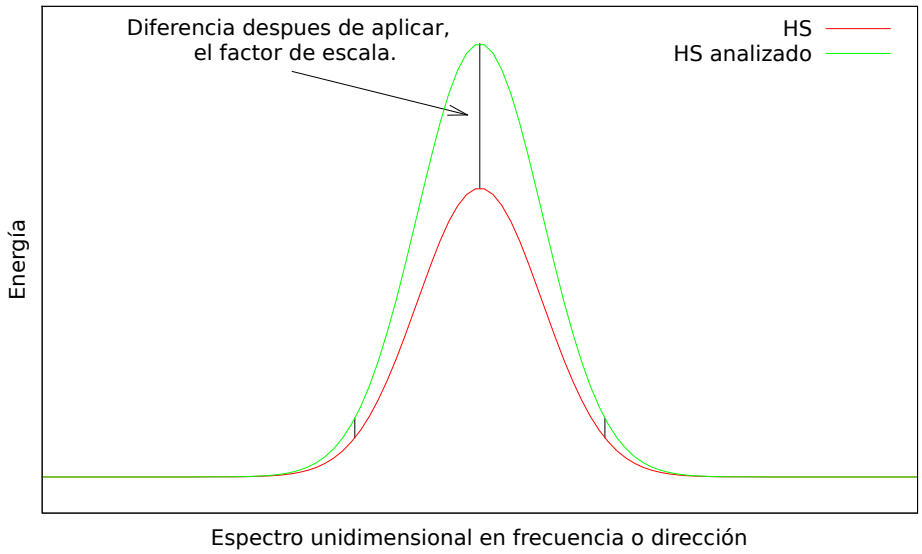


Fig. 8.7: Ejemplo de cómo escala el espectro manteniendo la misma forma y de tal manera que la hs final sea igual a la resultante del análisis.

$$\alpha^2 E^f = E^a.$$

De esta manera logramos tener una nueva función $E(f, \theta)$ que da una vez integrada el valor de la altura significativa de ola igual al resultado del análisis.

En la figura 8.7 se puede ver un ejemplo de una función E cualquiera la cual fue escalada con un valor constante.

8.7.2. Implementación

Para la implementación modificamos la rutina de asimilación de datos en el modelo (en el archivo w3wdasmd.ftn), de manera que pueda leer la nueva altura significativa de ola resultado del 4D-LETKF, a la que llamamos hs_a, y modifica variable de estado interna.

La altura significativa de ola, hs, es almacenada en una variable global, que se calcula para output con la rutina *W3OUTG*.

La implementación almacena, por razones de eficiencia, sólo para puntos húmedos. Estrictamente hay 3 categorías, excluidos, húmedos y puntos de borde activos.

Se cuenta con las siguiente funciones:

MAPSTA

- MAPSTA(IY,IX) = 0, puntos excluidos.
- MAPSTA(IY,IX) = 1, puntos húmedos.
- MAPSTA(IY,IX) = 2, puntos de borde activos.

MAPSF

- MAPSF(ISEA,1) = IX.

- MAPSF(ISEA,2) = IY.
- MAPSF(ISEA,3) = IXY.

MAPFS

- MAPFS(IY,IX) = ISEA.

Dado que el 4D-LETKF devuelve un mapa completo (lon x lat ó X x Y), lo primero que hacemos, después de leer el archivo, es pasarlo a la configuración de puntos activos $ISEA$, recorremos X é Y, asignando sólo a puntos activos: húmedos normales y de borde usando MAPSTA.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                     !!
!!        Todos leen HS resultado de LETKF                             !!
!!              v2d(nlon,nlat,nv2d)                                    !!
!!                                                                     !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

OPEN (IOFILE,file="anal.hs.bin",  iostat=istat , FORM='UNFORMATTED')
  read(IOFILE,iostat=istat) MY_HSA_BUFFER
CLOSE(IOFILE)


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                     !!
!!        Paso de Matriz a Vector                                      !!
!!        MY_HSA_BUFFER(NX,NY) -> MY_HSA(NSEAL)                        !!
!!                                                                     !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

DO IY=1, NY
 DO IX=1, NX
    IF (MAPSTA(IY,IX) == 1 .OR. MAPSTA(IY,IX) == 2)&
    .AND. (MY_HSA_BUFFER(IX,IY) > -990)then
        my_val = MY_HSA_BUFFER(IX,IY)
        my_isea= MAPFS(IY,IX)
        MY_HSA(my_isea) = my_val
    end if
 END DO
END DO
```

Después llamamos a una rutina que es usada en el módulo de output para forzar el cálculo de hs; de esta manera aprovechamos el código. Esto no causa problemas, ya que sólo se usa en ese módulo y su propósito es actualizar valores que durante el procesamiento del output serán utilizados; antes de hacerlo el sistema llama a esta rutina; por lo tanto llamarla antes hace que sea una operación idempotente.

```
!!!
```

```
!!              PASO MY_FACTOR a un vector local          !!
!!                                                        !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

DO JSEA=1, NSEAL
!/DIST                ISEA    = IAPROC + (JSEA-1)*NAPROC
!/SHRD                ISEA   =   JSEA

                MY_FACTOR_LOCAL(JSEA)= MY_FACTOR(ISEA)
END DO
```

Por último distribuimos uniformemente en cada punto el factor (al cuadrado). En la variable global VA son almacenados, dado un punto húmedo $ISEA$, los 600 puntos, que corresponden a la discretización del espectro de olas. Los mismos son el resultado de los 25 puntos de discretización para la frecuencia y los 24 puntos de discretización para el ángulo θ.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                        !!
!!                        MODIFICO VA                     !!
!!                                                        !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!



DO JSEA=1, NSEAL
   VA(:,JSEA)=(MY_FACTOR_LOCAL(JSEA)**2)*VA(:,JSEA)
END DO
```

8.7.3. Etapa 2 en el proceso de desarrollo: Corrección del espectro

Lamentablemente este primer acercamiento no dio los resultados esperados, por lo que tomamos la decisión de incorporar el resultado del análisis de manera más precisa, siguiendo la bibliografía [ecm, 2012b]. Consiste en realizar la corrección sobre cada par (frecuencia, dirección), llamado bin.

La corrección que se hace es: $E^a = AE^f(Bf, \theta)$; los factores A y B cambian si estamos en un bin de mar de viento o de fondo.

Separación del mar de viento y mar de fondo

Como primeros elementos describimos algunas con ciertas constantes y funciones que se utilizarán más adelante:

- En W3ADATMD:
  - WN: número de frecuencia (de 1 a 25).
  - U10(isea): velocidad del viento.
  - U10D(isea): dirección del viento, en radianes.
  - AS: parámetro de estabilidad.

- En W3IOSFMD:
  - UABS=U10 · ASF.
  - UDIR=MOD(270-U10d · RADE,360).
- En W3GDATMD:
  - TH: direcciones (radianes).
  - SIG: frecuencia para cada k ($rads^{-1}$).
  - NK: número de frecuencias.
  - NTH: número de direcciones.
- En W3WDATMD:
  - VA Matriz con la energía(E).
  - ASF.
- En WODATHMD:
  - WSMULT multiplicador del mar de viento.

Algoritmo para determinar mar de viento o mar de fondo

Adaptado del algoritmo del método PTMEAN en W3PARTMD

```
DO IK=1, NK
 DO ITH=1, NTH
  UPAR = WSMULT * UABS * MAX(0, COS(TH(ITH)-DERA*UDIR))
  IF UPAR < SIG(IK)/WN(IK)
   SWELL (mar de fondo)
  ELSE
   WINDSEA (mar de viento)
  END DO
END DO
```

Separación del análisis de hs en mar de viento (H^a_{ws}) y mar de fondo (H^a_{sw})

En primer lugar contamos con PWST en el módulo W3ADATMD; este valor indica qué proporción se corresponde con mar de viento y qué proporción con mar de fondo.

$$\frac{H^{f2}_{ws}}{H^{f2}} = PWST, \tag{8.1}$$
$$H^{f2}_{ws} = PWST \cdot H^{f2}. \tag{8.2}$$

La misma ecuación es válida en los valores del análisis

$$H^{a2}_{ws} = PWST \cdot H^{a2}. \tag{8.3}$$

$$H^{a} = \sqrt{H^{a2}_{ws} + H^{a2}_{sw}}, \tag{8.4}$$
$$H^{a2} = H^{a2}_{ws} + H^{a2}_{sw}, \tag{8.5}$$
$$H^{a2}_{sw} = H^{a2} - H^{a2}_{ws}. \tag{8.6}$$

Entonces utilizando lo anterior calculamos:

$$H^{a2}_{sw} = H^{a2} - PWST \cdot H^{a2}. \tag{8.7}$$

Entonces,

$$H^{a}_{ws} = \sqrt{PWST \cdot H^{a2}}, \tag{8.8}$$
$$H^{a}_{sw} = \sqrt{H^{a2} - PWST \cdot H^{a2}}. \tag{8.9}$$

Factores A, B en el caso Mar de Viento

En cada caso se calcula los factores A y B.

$$B = \left(\frac{U^f}{U^a}\right)^{0,3518} \left(\frac{H^a_{ws}}{H^f_{ws}}\right)^{0,6759}, \tag{8.10}$$
$$A = \left(\frac{H^a_{ws}}{H^f_{ws}}\right)^{1,363} \left(\frac{U^a}{U^f}\right)^{0,394}. \tag{8.11}$$

Factores A, B en el caso Mar de Fondo

$$B = \left(\frac{H_{sw}^{a}}{H_{sw}^{f}}\right)^{0,5}, \tag{8.12}$$

$$A = B\left(\frac{H_{sw}^{a}}{H_{sw}^{f}}\right)^{2}. \tag{8.13}$$

Entonces, una vez calculados A y B, debemos actualizar la energía con la siguiente ecuación:

$$E^{a} = AE^{f}(Bf, \theta).$$

- Buscamos el valor $B*f$. El modelo tiene 25 frecuencias y una función, SIG, que toma un número de frecuencia y devuelve el valor de dicha frecuencia, por lo que debemos calcular la inversa de manera de saber a qué **número de frecuencia** asignarle el valor. Dado que es exponencial despejamos:

  - $FR1 \cdot XFR^{(\alpha-1)}$,
  - $SIG(\alpha)$.

  Despejando α

$$SIG(IK) = SIG(IK-1)XFR\ 1, \tag{8.14}$$

$$\alpha = newIK, \tag{8.15}$$

$$newIK = NINT\left(1 + \frac{log(newSIG) - log(FR1)}{log(XFR)}\right), \tag{8.16}$$

$$FR1 = SIG(1). \tag{8.17}$$

  Una vez obtenido el nuevo IK se actualiza dependiendo de cada caso

$$E_{ws}^{a}(IK, ITH) = AE_{ws}^{f}(\alpha, ITH), \tag{8.18}$$

$$E_{sw}^{a}(IK, ITH) = AE_{sw}^{f}(\alpha, ITH). \tag{8.19}$$

De esta manera contamos con la "actualización" de la variable de estado interna del modelo, más cercana a la "verdad del campo"[18].

[18] Del inglés "True Field".

8.8. Cerrar el ciclo, utilizando el estado nuevo inicial para correr el conjuntos de pronósticos

En esta etapa implementamos el código que se ocupa de coordinar e integrar todo el desarrollo, puede observarse un esquema en la figura 8.8. Fue implementado en el archivo "runCycle.py". Esta implementación sigue la idea expresada en la figura 1.1. Como input recibe la fecha inicial y la fecha final, y se ocupa de ejecutar las llamadas necesarias para hacer la corrida para el periodo indicado.

1. **runWW3:** ejecuta el modelo para la hora de análisis dada.
2. **getDecod:** obtiene y decodifica las observaciones.
3. **makeObs:** construye los archivos con observaciones.
4. **runFilter:** prepara los archivos de input para el filtro, obteniendo el campo de análisis.
5. **runWW3DAS:** ejecuta el modelo para la hora de análisis actual, pero con la salvedad que ahora sólo lo hace para incorporar el resultado del filtro (4D-LETKF) a la variable interna del modelo (ver capítulo 8.7)
6. **copyResultsLETKF:** almacena el resultado del ciclo de asimilación; esta información es útil para estudiar el comportamiento del sistema, realizar verificaciones, etc.

El diseño fue orientado a un proceso secuencial, por pasos. Este tiene la ventaja de poder testear cada uno por separado. En caso de falla sólo hay que detectar dónde falló y continuar en el paso que corresponda.

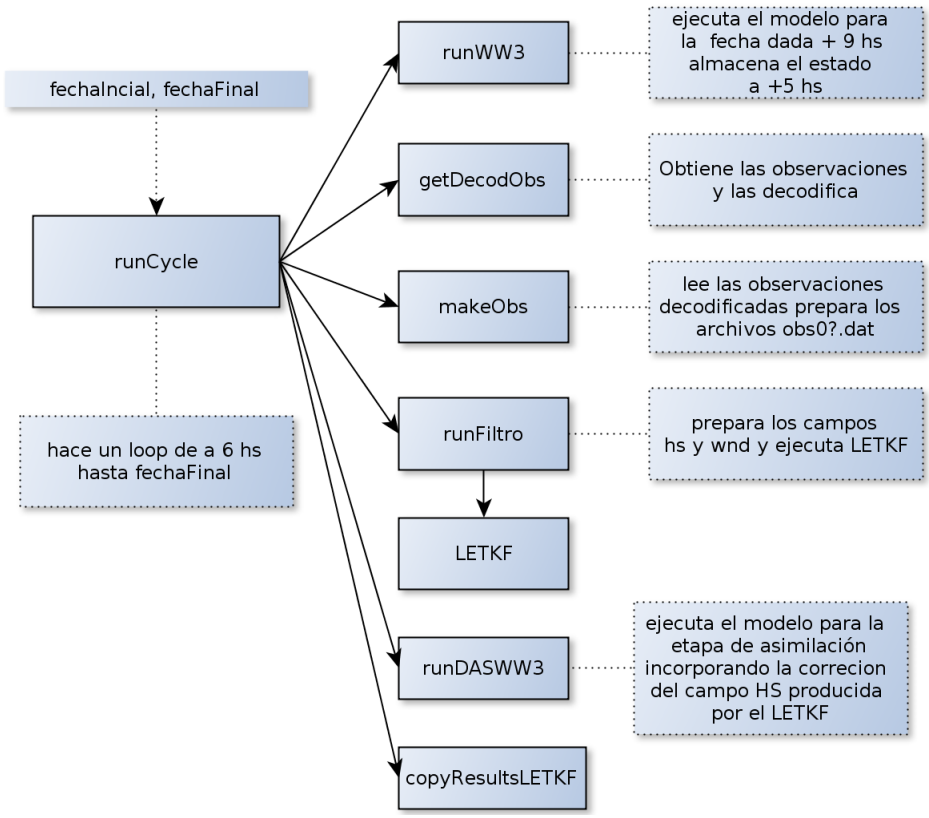


Fig. 8.8: Diseño del script que ejecuta un ciclo completo de asimilación.

8.9. Visualización de resultados

Dado que el software disponible no contaba con la versatilidad necesaria y existen herramientas muy bien desarrolladas en Python optamos por desarrollar una serie de scripts y clases que modelan el problema de la visualización de los campos meteorológicos-oceanográficos, esencialmente campos 2D.

En particular nuestros resultados son campos de hs, de tamaño fijo, que se pueden almacenar como matrices y graficar en un planisferio con proyección cilíndrica de Miller (es mercator modificada en 1949). Para la visualización utilizamos *matplotlib basemap toolkit*[19].

Debimos resolver varios problemas:

- el modelado de los campos de olas;
- el modelado de las trazas con las observaciones satelitales;
- el graficar todos los campos juntos o por separado;
- operar con ellos: calcular la *innovación* (resta del campo *guess* y el campo *análisis*).

Optamos por desarrollar una clase llamada *campo*, que modela un campo cualquiera. Por ahora contamos con dos subclases, *campoHS* y *campoWND*. Estos campos se grafican

[19] http://matplotlib.org/basemap/index.html

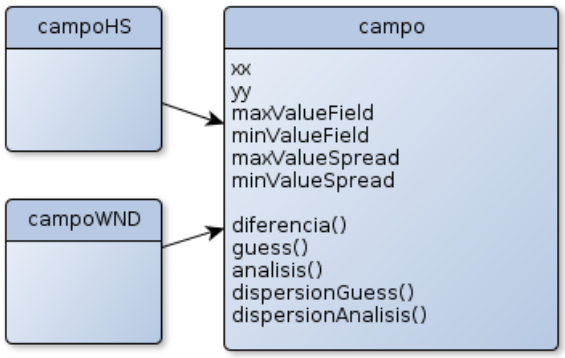


Fig. 8.9: Diagrama de clase Campo.

sobre un mapa (usando pcolor) y sobre éstos se pueden graficar las trazas de los satélites (usando scater). También podemos calcular la intensidad del viento en lugar de tener las dos componentes por separado. Nos encontramos con el problema de leer un archivo almacenado en disco desde Python, con formato binario el cual es generado por un programa escrito en Fortran. Para resolverlo utilizamos un *wrapper* de una función complementaria. Dado que estamos en la etapa de desarrollo en al que se hacen pruebas y el volumen de información para realizar experimentos aún es manejable no se cuenta con ninguna política para el manejo de los datos, se eliminan manualmente.

```
subroutine Olasbin2py(field, filename, nlon, nlat)


  IMPLICIT NONE
  INTEGER,PARAMETER  :: r_sngl=kind(0.0e0)
  INTEGER,INTENT(IN) :: nlon
  INTEGER,INTENT(IN) :: nlat
  CHARACTER(*),INTENT(IN) :: filename
  REAL(r_sngl),INTENT(OUT):: field(nlon,nlat)
  INTEGER :: iofile,istat

  iofile=26

  OPEN (iofile,file=filename, iostat=istat, FORM='UNFORMATTED')
  READ (iofile, iostat=istat) field


end subroutine OlasBin2Py
```

Instalamos el programa f2py que forma parte de numpy y generamos el módulo con el comando:

f2py -c -m OlasBin2Py OlasBin2Py.f90.

Una vez generado el .so este se importa desde Python con:

import OlasBin2Py,

Contamos con la siguiente signatura:

```
olasbin2py - Function signature:
  buf4 = olasbin2py(filenamein,nlon,nlat)
Required arguments:
  filenamein : input string(len=-1)
  nlon : input int
  nlat : input int
Return objects:
  buf4 : rank-2 array('f') with bounds (nlon,nlat)
```

Por otro lado implementamos las siguientes rutinas:

- **hacerMapa(nombreVentana):** toma el un *string* para identificar la ventana y construye un mapa global, con ciertos colores de fondo, paralelos, meridianos, etc. y cualquier otra propiedad que se le quiera dar al mapa.
- **dibujarCampo(campo, mapa, colorMap, myValMax, myValMin)**: esta función toma un mapa, un campo y dibuja ese campo, una instancia de campoWND o campoHS. También toma como parámetro paleta de colores y los valores máximos y mínimos, ya que en ciertos casos es necesario indicarle éstos para una correcta visualización de la información.
- **scaterData(datos,mapa, myValMax, myValMin):** toma una matriz con dato (de la forma latitud, longitud, dato) y hace un *scatter* sobre el mapa. También toma valores máximos y mínimos.

8.10. Experimentos y Resultados

Se diseñaron varios experimentos: lo primero que se hizo fue llevar el conjunto a un estado realista, partiendo del reposo haciendo una simulación de 15 días, sólo con vientos y sin asimilar observaciones; una vez llegado a este punto se comenzó a asimilar. Los experimentos realizados corresponden a los periodos del 17 al 29 de diciembre de 2012 y 1 al 16 de abril de 2013.

En las primeras pruebas pudimos verificar que la dispersión del conjunto se hacía cada vez más pequeña; esto no es bueno para el pronóstico por conjuntos, ya que con la diversidad en los distintos miembros se busca capturar la incertidumbre del sistema, por lo que decidimos agregar ganancia (inflation) aditiva. Este método agrega dispersión; como primera aproximación decidimos agregar un factor de 0,75, pero resultó ser demasiado, producía que la dispersión del conjunto creciera de manera poco realista. Luego pasamos a un factor de 0,20, el cual resultó ser adecuado. Queda como parte de trabajos futuros realizar un estudio más profundo de este punto.

Los campos que adicionamos son los campos *guess*, pero podría ser cualquier otro que represente de alguna manera el error del sistema. Los archivos son nombrados como "addiNNN.hs.bin", "addiNNN.wu.bin" y "addiNNN.wv.bin", NNN es el número de miembro del conjunto, ya que lo que se busca es agregar dispersión y de esta manera se agrega de manera coherente. En el futuro se puede hacer una permutación de los archivos para lograr de manera de lograr una perturbación realista.

En la figura en la figura 8.10 las del caso de abril y 8.11 puede verse la cantidad observaciones asimiladas en cada ciclo para el caso de diciembre.

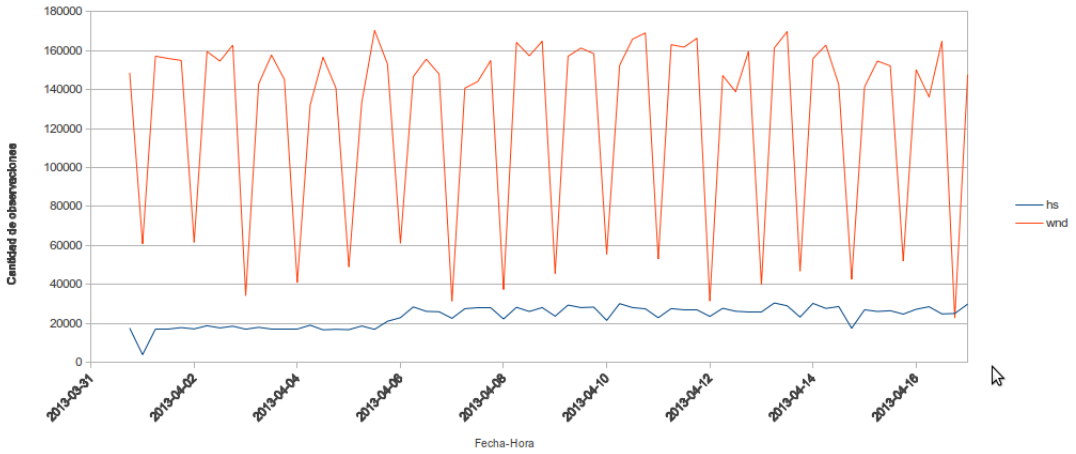


Fig. 8.10: Cantidad de observaciones asimiladas periodo 31/03/2013 al 17/04/2013. Entre 20.000 y 30.000 observaciones de hs y entre 100.000 y 140.000 de viento; se pueden apreciar los baches correspondientes a cuando el satélite sobrevuela tierra, totalizando entre 120.000 y 170.000 por ciclo de asimilación.

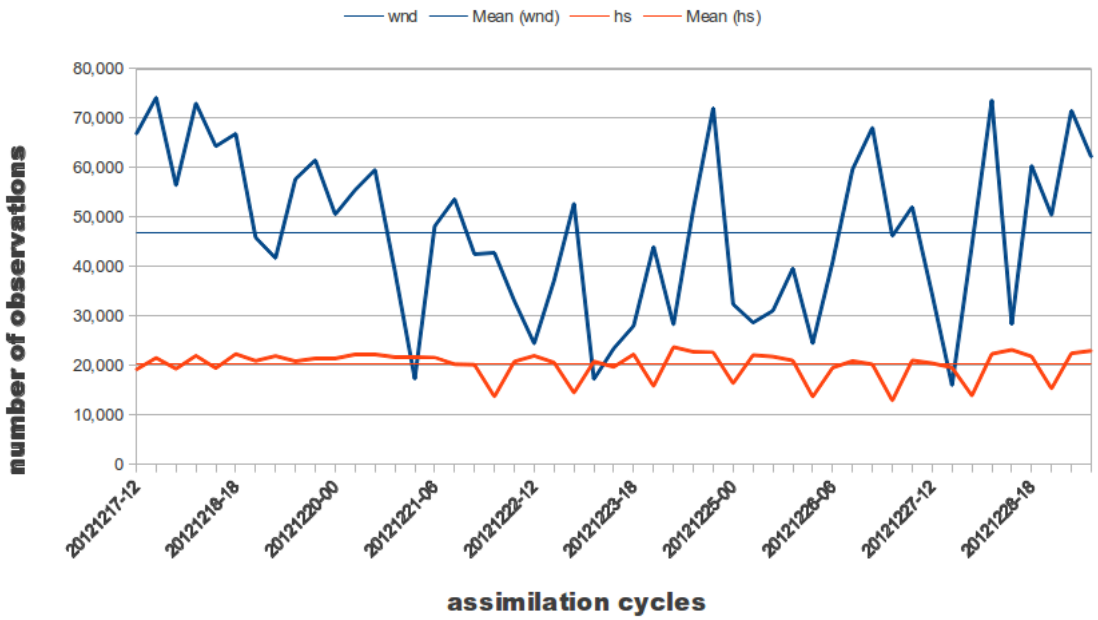


Fig. 8.11: Cantidad de observaciones asimiladas periodo 17/12/2012 al 29/12/2012. Hay una media de 20.000 observaciones de hs y una media 50.000 observaciones de viento, totalizando aproximadamente 70.000 observaciones por ciclo de asimilación.

Los siguientes ítems hacen referencia a la figura:

- **Incremento**: diferencia entre el campo guess y el campo análisis.
- **Guess media**: es la media del conjunto de los campos guess[20].
- **Análisis media** : ídem pero del campo análisis.
- **Dispersión guess**: el desvío estándar con respecto a la media del conjunto guess.
- **Dispersión análisis**: ídem pero para el campo análisis.

En la figura 8.12 puede observarse un ejemplo, la situación del día 2012/12/18 12hs GMT; en el Pacífico Norte, donde se observa una situación de gran corrección (a); en la imagen (b) podemos apreciar que hay varias trazas satelitales de hs (línea fina) y viento (más ancha) y por otro lado se ve en las imagen (e) y (f) que la dispersión de la zona no es menor; esto indica que la corrección que se está haciendo en zonas donde hay mayor incertidumbre. En las imágenes (c) y (d) mostramos el campo de hs antes y después del análisis.

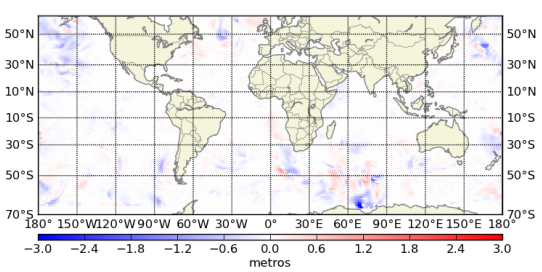
En el archivo *campos.html* dentro de *resultados/diciembre-2012* o *resultados/abril-2013* pueden verse más resultados de los distintos ciclos de asimilación.

8.10.1. Medida del error

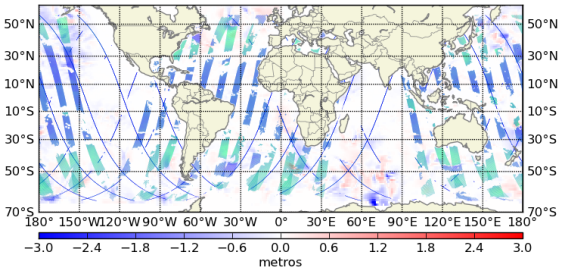
El 4D-LETKF también provee una medida de cuán alejado está el modelo de las observaciones en promedio. Se denomina apartamiento observacional O-F[21]. Para tomar esta media el algoritmo consiste en restar el valor del modelo al valor de la observación, para luego acumular esas diferencias y calcular el promedio y la varianza.

En las figura 8.13 puede observarse la evolución de estas medidas para el caso con asimilación (Análisis (ANL)) y sin asimilación (no ANL) en el periodo 20121217 12hs al 20121229 6 hs.

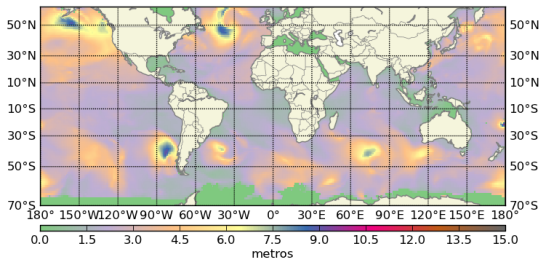
[20] Primera aproximación.
[21] Observación menos pronóstico (forecast).



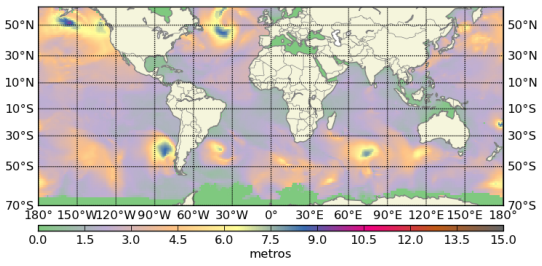
(a) Incremento.



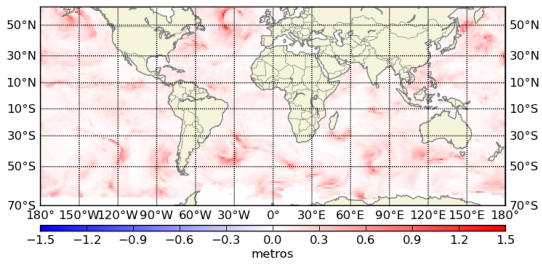
(b) Incremento con trazas satelitales.



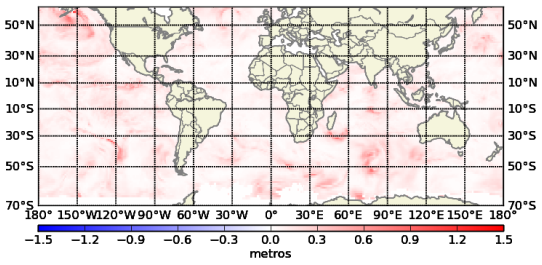
(c) Guess media.



(d) Análisis media.



(e) Dispersión del guess.



(f) Dispersión del análisis.

Fig. 8.12: Ejemplo de Mapas con hs para el ciclo de asimilación 18/12/2012 a las 12 hs. a) Incremento: diferencia entre el campo guess y el campo análisis. b) Guess media: es la media del conjunto de los campos guess. c) Análisis media: ídem pero del campo análisis. d) Dispersión guess: el desvío estándar con respecto a la media del conjunto guess. e) Dispersión análisis: ídem pero para el campo análisis. En el Pacífico Norte se observa una situación de gran corrección en a (a); en (b) podemos apreciar que hay varias trazas satelitales de hs (línea fina) y viento (más ancha) y por otro lado se ve en las imagen (e) y (f) que la dispersión de la zona no es menor; esto indica que la corrección se está haciendo en zonas donde hay mayor incertidumbre. En las imágenes (c) y (d) mostramos el campo de hs antes y después del análisis.

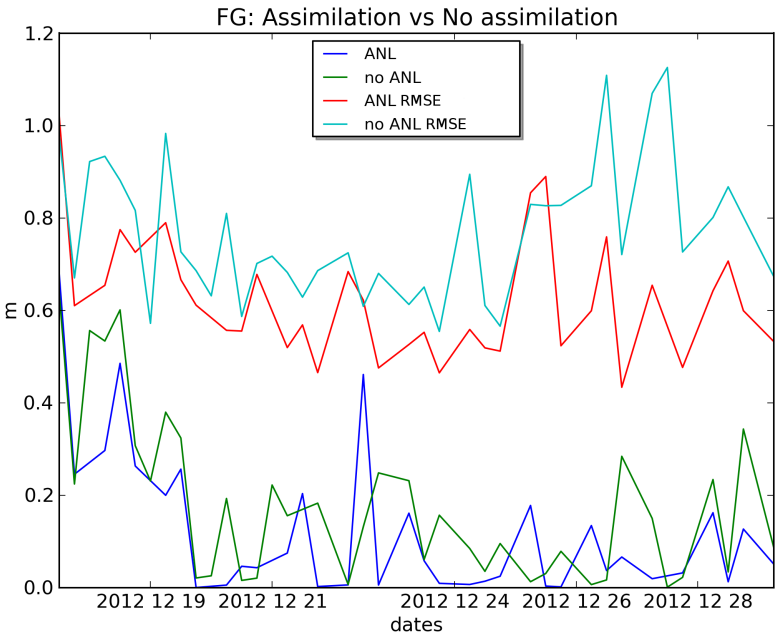


Fig. 8.13: Valor absoluto de Error Cuadrático Medio y Sesgo de los campos con asimilación (ANL) y sin asimilación (no ANL) en metros. Puede observarse que el apartamiento observacional es menor despues de asimilar (en azul).

En las figuras 8.14 y 8.15 puede observarse el apartamiento observacional, pero esta vez no está el caso sin asimilación. En el eje x se grafican los ciclos de asimilación y en el eje y el valor absoluto del error (en metros). Lo interesante a notar es que en ninguno de los dos casos el error crece. Si bien hay picos estos pueden deberse a que alguna observación no sea correcta y como resultado hace disparar la suma.

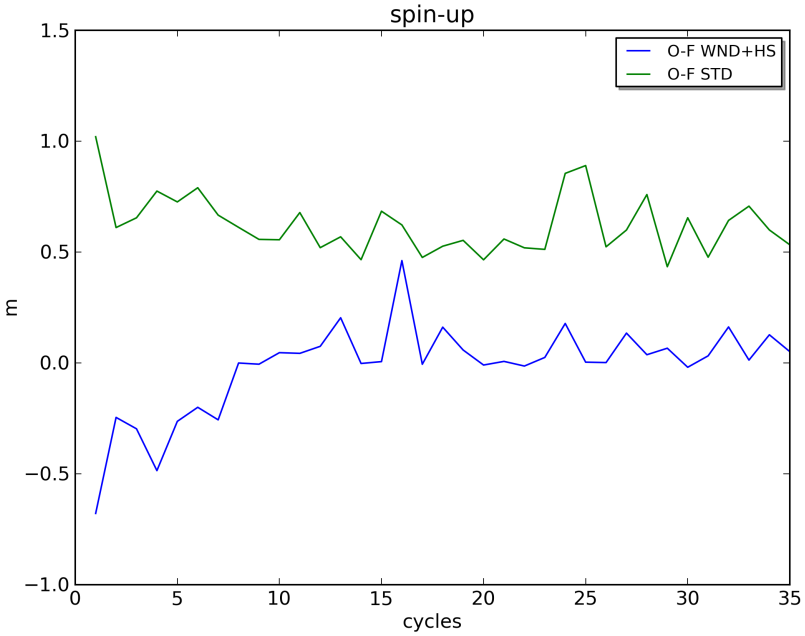


Fig. 8.14: Caso de Diciembre de 2012. Valor absoluto del error de hs (en metros) en función de los ciclos de asimilación.

Este es un primer indicio que nos sugiere que el sistema funciona, ya que puede verse que el apartamiento observacional (O-F) está más cercano al 0, teniendo en cuenta que el cálculo se hace con lo que fue el pronóstico del ciclo anterior, a 6 horas.

Para ejecutar el ciclo sin asimilación se ejecuta el mismo código; pero se suprime la llamada al paso *5-WW3DAS* y cambia la confección del archivo *ww3_shell.inp* con las configuración y en particular el guardado del archivo restart[22].

8.11. Performance

En cuanto a la performance se estudió el speedup[23] del 4D-LETKF, utilizando 1 a 16 cores[24]; para nuestra sorpresa el programa no presentaba buena performance. Más aún teniendo en cuenta que el mismo código fue utilizado en otros casos y que la formulación del problema es altamente paralelizable, dado que cada punto se trata independiente del resto.

En las figuras se grafican en el eje X la cantidad de cores y en el eje Y el tiempo insumido (en segundos). Para todos los casos, se muestran cuatro mediciones:

TOTAL tiempo total insumido por el programa

IDEAL tiempo ideal total, para un core de toma el tiempo TOTAL, para n cores lo insumido para uno dividido n. Es una medida de lo que pasaría si el speedup fuera

[22] Ese archivo es el que sirve de punto de partida para el próximo ciclo de asimilación.

[23] Speedup se define por la siguiente formula: $S_p = T_1/T_n$ donde T_1 es el tiempo insumido por el programa ejecutando con un proceso o el programa no paralelo y T_n es el tiempo insumido por el programa paralelo en n procesos.

[24] Máximo poder de cómputo del cluster de desarrollo del SMN/SHN Hielo-Nieve.

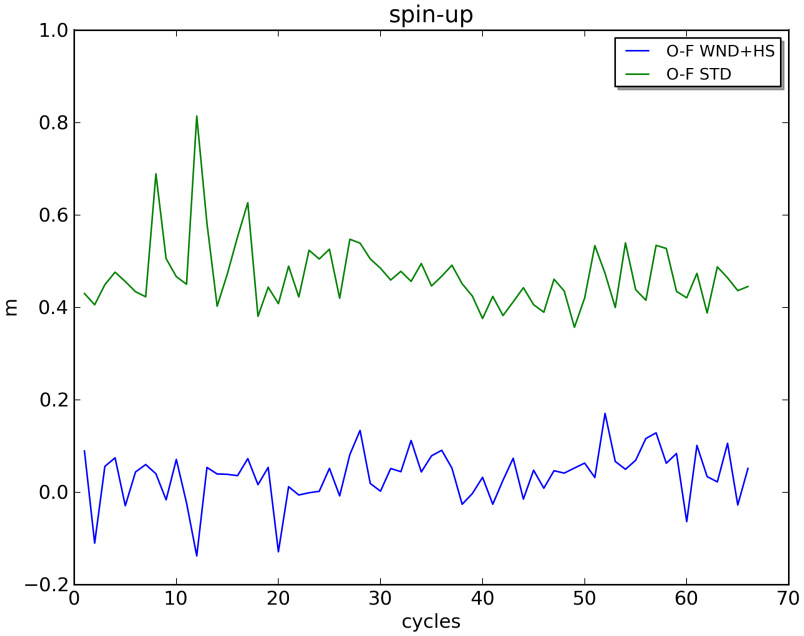


Fig. 8.15: Caso de abril de 2013. Valor absoluto del error hs (en metros) en función de los ciclos de asimilación. El error no crece a medida que pasan los ciclos de asimilación.

perfecto (cada procesador agregado divide en la tarea sin overhead, por lo que el costo es $1/n$).

READ_OBS tiempo insumido por la rutina de lectura de observaciones

LETKF core del sistema

Haciendo un estudio detallado del tiempo insumido en cada una de las etapas descubrimos que el core del procesamiento tiene un excelente speedup; en la figuras 8.16, 8.17 y 8.18 pueden observarse los gráficos con los tiempos versus cantidad de procesos, tomados en los laboratorios del Departamento de Computación (www.dc.uba.ar), en el CECAR[25] y en el Cluster de desarrollo del SMN/SHN(hielo-nieve). En la tabla 8.1 se muestra el detalle de los ambientes donde se realizaron las mediciones, en la tabla 8.2 se presenta el detalle de los procesadores y en la tabla 8.3 se pueden apreciar las versiones de software. Además mostramos solo para el caso Aire-Nieve el speedup con escala logarítmica (figura 8.19) de esta última la idea es comparar el LETKF vs. el caso IDEAL (tiempo 1 cores / tiempo n cores), comparación que en la escala lineal esto no se aprecia.

El autor del código, Takemasa Miyoshi, confirmó en una comunicación personal que no se había tenido en cuenta la performance en la etapa lectura de observaciones. La razón de ello es que en un típico caso meteorológico se cuentan con 38 niveles y 15 variables. La cantidad de observaciones es significativamente menor por lo tanto la mayor cantidad del tiempo es consumido por el core del filtro, que hace que la lectura de observaciones quede solapada.

[25] http://cecar.fcen.uba.ar

En el caso del problema de olas, a diferencia de un caso meteorológico, el problema tiene solo 2 dimensiones y procesamos 3 variables; por otro lado la cantidad de observaciones es significativamente mayor, por lo que la lectura de observaciones pasa a ser un problema, ya que el costo computacional de procesar la asimilacion no es tan grande como para enmascarar la lectura de observaciones.

	CECAR	Laboratorios DC	Hielo-Nieve
CPU's por host	2	1	2
Procesador	Xeon(R) 5030	I5-3550	Xeon(R) E5620
Cores por host	4	4	4
Red	INFINIBAND	Lan 10/100	Gigabit

Tab. 8.1: Clusters usado en las mediciones del 4D-LETKF.

Xeon(R) 5030	4M Cache, 2.66 GHz, 667 MHz FSB
I5-3550	6M Cache, up to 3.70 GHz
Xeon(R) E5620	12M Cache, 2.40 GHz, 5.86 GT/s Intel® QPI

Tab. 8.2: Detalle de los procesadores.

	CECAR	Laboratorios	Hielo-Nieve
Compilador	Ifort 14.0.0	Ifort 12.1.0	Ifort 12.1.5
MPI	OpenMPI 1.4.2	OpenMPI 1.4.3	MPICH2 1.2.1
OS	Debian 6.0.4	Ubuntu 12.04.2 LTS	CentOS 6.3

Tab. 8.3: Compiladores e implementación de MPI.

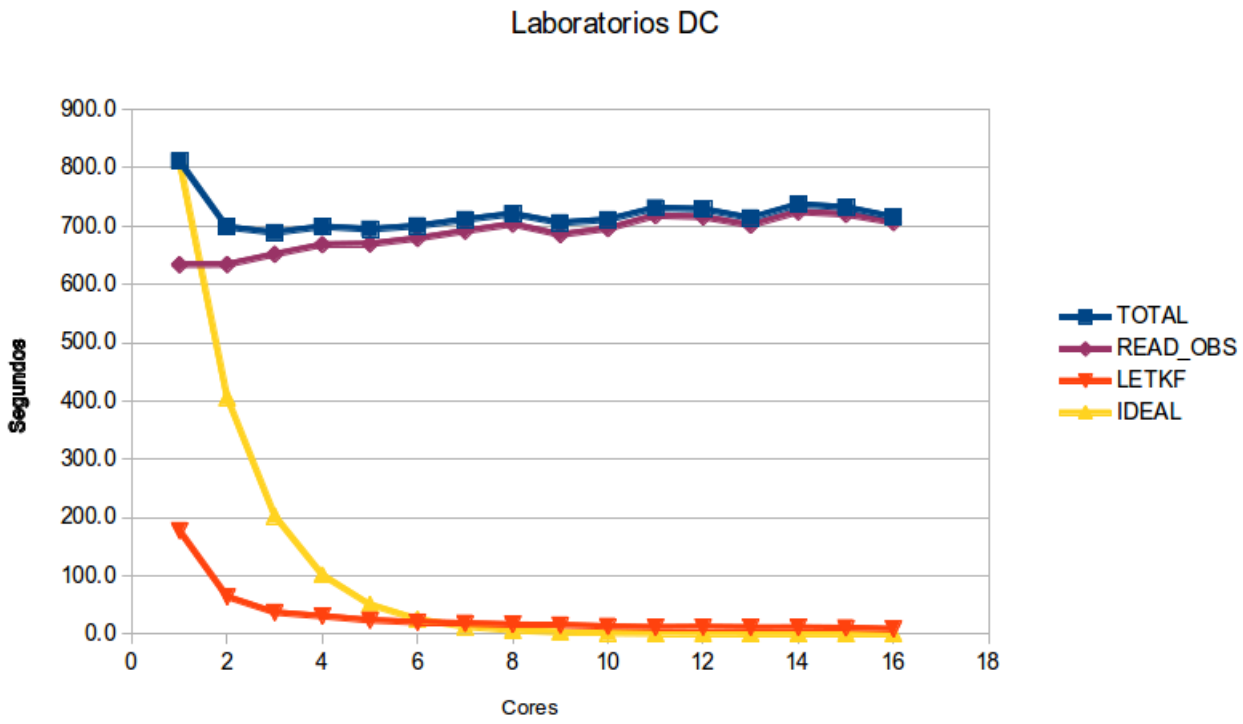


Fig. 8.16: Speedup 4D-LETKF en las computadoras del laboratorio del departamento de computación, el mayor costo lo insume la lectura de observaciones. No encontramos explicación para los "saltos".

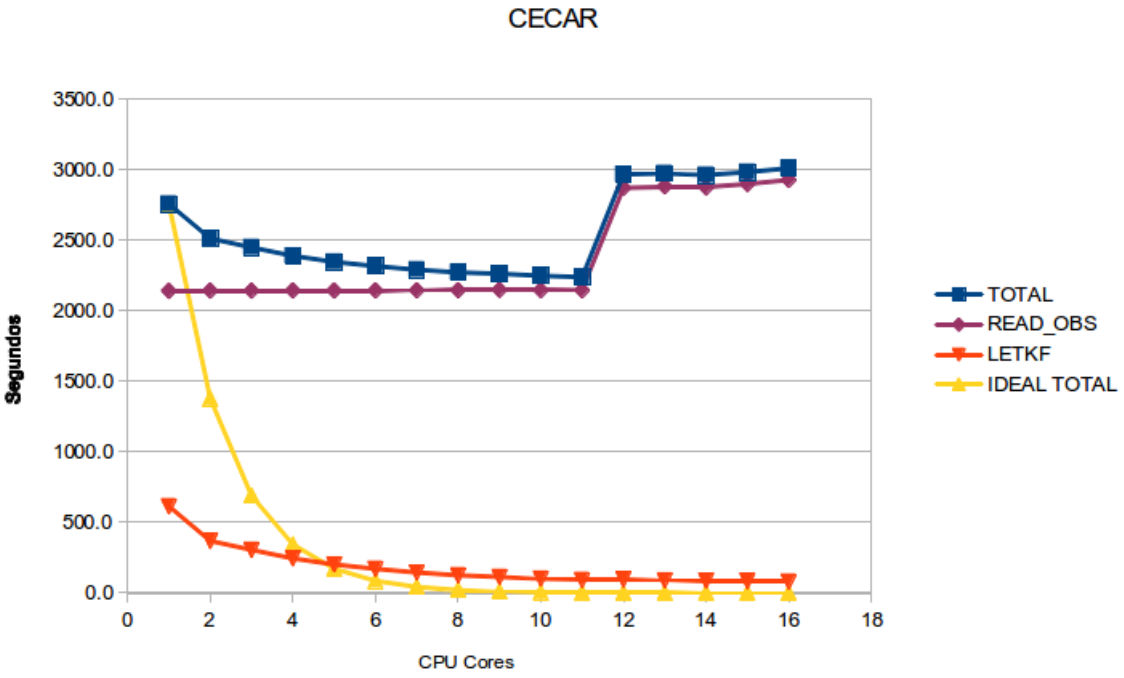


Fig. 8.17: Speedup 4D-LETKF en el CECAR, el mayor costo lo insume la lectura de observaciones.

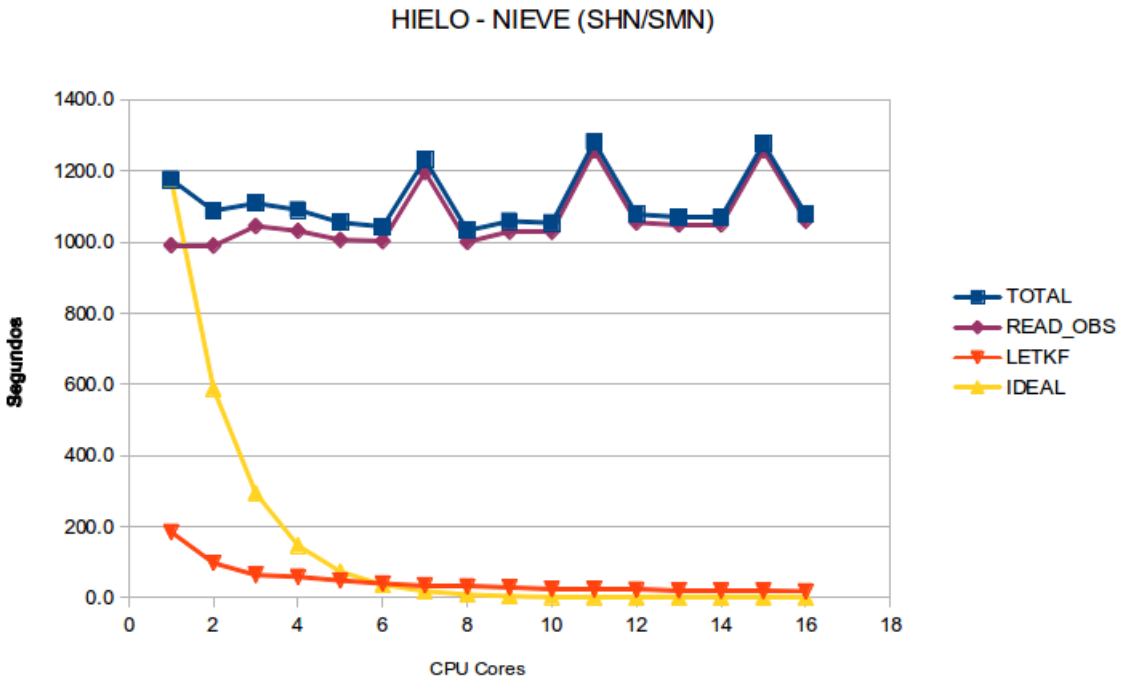


Fig. 8.18: Tiempo de cómputo 4D-LETKF en Hielo-Nieve, notar que el mayor costo lo insume la lectura de observaciones. No encontramos explicación para los "saltos".

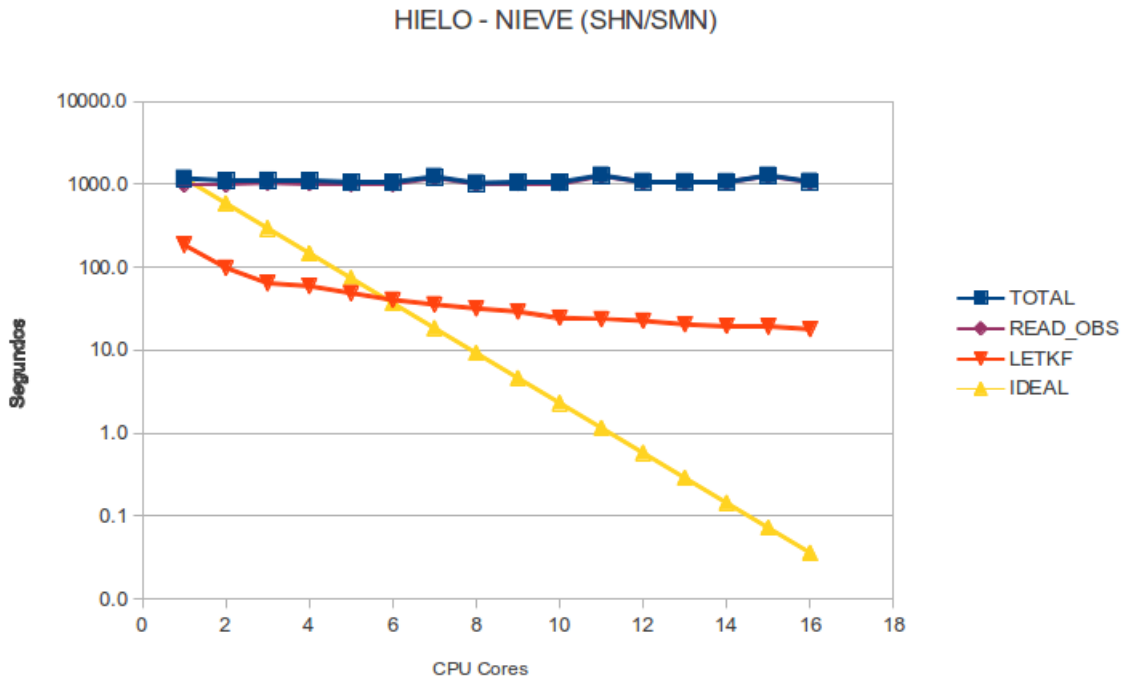


Fig. 8.19: Tiempo de cómputo 4D-LETKF en Hielo-Nieve. La paralelización del LETKF(core del proceso) es buena no llega a ser perfecta, lo cual es esperable.

9. CONCLUSIONES

El objetivo del trabajo fue la implementación de un ciclo de asimilación de seis horas utilizando vientos del *Ensemble* del modelo GFS de NCEP y observaciones de tres satélites polares, dos que miden altura de ola, y uno que mide intensidad y dirección de viento. El software desarrollado ejecuta un control de calidad de los datos. Completamos los pasos que permiten ejecutar el ciclo de pronóstico-asimilación.

Dotamos al software de una flexibilidad en su diseño que permite realizar experimentos, aumentar o disminuir la cantidad de miembros del conjunto, la cantidad de fuentes observacionales y la cantidad de cores. También desarrollamos un software que permite la visualización de resultados y por otro lado el diseño modular permite testear y modificar cada parte por separado.

Como resultado de los experimentos descubrimos que había un problema en la performance de la lectura de observaciones del 4D-LETKF. El autor de la implementación, Takemasa Miyoshi, confirmó que no lo había tenido en cuenta dada la naturaleza de los problemas a los que había aplicado el filtro (comunicación personal).

La rutina de transformación de viento y altura de ola analizada en espectro de energía, es una rutina avanzada de acuerdo a desarrollos recientes del centro europeo pero los efectos requieren aún de testeo en cuanto a lo que llamamos "ondas zombie": dado que al asimilar se corrige la forma y posición de los trenes de olas, es posible encontrar grupos de olas que no tengan sentido físico.

El trabajo es la primera experiencia que utiliza el 4D-LETKF en un modelo de olas asimilando mediciones de vientos y altímetros.

Los resultados preliminares sugieren que una mejora en las condiciones iniciales genera un pronóstico con menor error, en particular en el caso de mar de fondo, donde la propagación se desarrolla sin intervención del forzante. Por tal razón, la mejora proviene de las condiciones iniciales.

Creemos que la experiencia de estudiar el modelado de un fenómeno oceánico a escala global, y el funcionamiento de satélites pone en evidencia que la computación, a través del desarrollo de software específicos, puede contribuir en gran medida a las investigaciones en ciencias de la atmósfera y los océanos. En tal sentido, el aporte interdisciplinario puede ser sumamente beneficioso.

10. TRABAJOS FUTUROS

10.1. GPU

Evaluar, haciendo pruebas de performance, la utilización de bibliotecas matemáticas (BLAS, LAPACK etc) implementadas en GPU[1], dadas las características del problema, esencialmente las dimensiones y el uso de operaciones matemáticas, hacen suponer que los resultados pueden ser muy buenos. Aplica tanto al modelo NOAA WWIII® como al filtro LETKF. Las mediciones de consumo de memoria indican que el filtro utiliza 107 mb de memoria residente mientras que el modelo 335mb datos que indican que a priori es factible su uso.

10.2. Sistema Operativo

Continuar trabajando para lograr un sistema operativo de asimilación ajustando constantes y demás parámetros.

10.3. Cuellos de Botella

Estudiar cuellos de botella en el modelo de olas y mejoras en performance.

10.4. Unificar observaciones *wu wv*

El sistema no toma a las dos componentes del viento como una única observación; entonces cuando una es eliminada porque no pasa el control de calidad la otra podría quedar.

10.5. MetOp B y OceanSat-2

Se encuentran disponibles las observaciones de los escaterómetros a bordo de los satélite *MetOp B* y *OceanSat-2*.

10.6. Paralelizar lectura de observaciones de 4D-LETKF

Los tiempos de procesamiento son aceptables pero el sistema no escala. Aumenta mucho el tiempo de cómputo en el caso de agregar observaciones y no disminuye tanto como se podría para el caso de disponer de mayor poder de calculo.

10.7. Verificación

Implementar un sistema de verificación de resultados que permitan verificar permanentemente el funcionamiento del sistema utilizando boyas.

[1] https://developer.nvidia.com/magma

10.8. Sistema de pre y post-condiciones

Implementar un sistema de verificación de resultados de los pasos de la asimilación de manera de que si alguna falla el sistema puede tomar decisiones al respecto y registrar lo ocurrido.

10.9. Ensamble a 80 miembros

Extender el sistema a 80 miembros.

10.10. Radio de localización óptimo

Realizar testeos para encontrar el radio de localización óptimo.

Índice de figuras

1.1. Esquema del ciclo de asimilación que se implementó en el presente trabajo. 8

2.1. Distribución estadística de las olas. La hs es la media del tercio mayor de olas (en azul claro). Fuente: www.comet.ucar.edu 11

3.1. Como ejemplo de variabilidad del conjunto de olas, se presenta un mapa "Spaguetti" de hs producto del modelo de olas WWIII forzado con el conjunto de NCEP. Son curvas de nivel de $hs = 3$ metros, cada color corresponde a un miembro del conjunto. Si bien se preserva la misma forma también hay cierta variabilidad. 14

5.1. Matrices en las ecuaciones del Kalman Filter extendido, paso 1. 18
5.2. Matrices en las ecuaciones del Kalman Filter, paso 2. 19
5.3. Matrices en las ecuaciones del Kalman Filter, paso 3. 20
5.4. Asimilación de datos en pronóstico numérico del tiempo. Fuente: [Kalnay et al., 2008]. 21
5.5. Asimilación de datos = Análisis. Fuente: [Kalnay et al., 2008]. 22
5.6. Mirada probabilística. Fuente: [Kalnay et al., 2008]. 22

6.1. Principios del funcionamiento de los altímetros de radar - Jason 1 y Jason 2. http://commons.wikimedia.org/wiki/File:Jason1_measurement_system.gif . 26
6.2. Trazas del satélite Metop-A de 22 hs. Los colores indican intensidad del viento. Fuente: knmi.nl/scatterometer/ 28

7.1. Cluster Elementos. 33

8.1. Ciclo de asimilación para las 18 hs. El modelo recibe los campos de viento de las 12 y de las 18 como así también lee los archivos restart.ww3 (provenientes del ciclo anterior). Con este input genera un Ensemble de 20 miembros con hs, wu y wv para las 12, 13, ... , 21 hs. El paralelo las observaciones se descargan, decodifican y separan en slots, uno para cada hora (de 12 a 21). Esta información es leída por el filtro de asimilación el cual produce una salida de análisis para las 18 de hs, wu y wv que junto con un archivo restart.ww3, para las 17, es leída por el modelo el cual ejecuta la simulación de 17 a 18, donde se hace la incorporación de la información del análisis al modelo. Finalmente se cuenta con un Ensemble de archivos restart.ww3 con el estado completo del modelo para las 18 hs que sirve de punto de partida para el siguiente ciclo de asimilación. 42
8.2. Ciclo de asimilación - Pronóstico. Un ciclo cada 6 hs, pero el plazo de simulación del modelo dura 9 hs, es así para generar los datos que se utilizan en la ventana de asimilación 43
8.3. Diseño del script que ejecuta un ciclo completo de pronóstico. 43
8.4. Como ejemplo de como se modela el problema: Se puede ver un campo global de hs y en un punto, cercano a África, el espectro que muestra la descomposición en frecuencia y dirección de la energía de ola para ese punto. 44

8.5. Sistema de adquisición y procesado de observaciones. Se muestran los tres satélites que se utilizaron en la tesis. Las etapas de obtención desde el servidor ftp, decodificación y preparación para ser asimiladas por el filtro. . 52
8.6. Diseño del sistema de preparación de las observaciones. Utiliza polimorfismo 56
8.7. Ejemplo de cómo escala el espectro manteniendo la misma forma y de tal manera que la *hs* final sea igual a la resultante del análisis. 58
8.8. Diseño del script que ejecuta un ciclo completo de asimilación. 66
8.9. Diagrama de clase Campo. 67
8.10. Cantidad de observaciones asimiladas periodo 31/03/2013 al 17/04/2013. Entre 20.000 y 30.000 observaciones de hs y entre 100.000 y 140.000 de viento; se pueden apreciar los baches correspondientes a cuando el satélite sobrevuela tierra, totalizando entre 120.000 y 170.000 por ciclo de asimilación. 69
8.11. Cantidad de observaciones asimiladas periodo 17/12/2012 al 29/12/2012. Hay una media de 20.000 observaciones de *hs* y una media 50.000 observaciones de viento, totalizando aproximadamente 70.000 observaciones por ciclo de asimilación. 70
8.12. Ejemplo de Mapas con hs para el ciclo de asimilación 18/12/2012 a las 12 hs. a) Incremento: diferencia entre el campo guess y el campo análisis. b) Guess media: es la media del conjunto de los campos guess. c) Análisis media: ídem pero del campo análisis. d) Dispersión guess: el desvío estándar con respecto a la media del conjunto guess. e) Dispersión análisis: ídem pero para el campo análisis. En el Pacífico Norte se observa una situación de gran corrección en a (a); en (b) podemos apreciar que hay varias trazas satelitales de hs (línea fina) y viento (más ancha) y por otro lado se ve en las imagen (e) y (f) que la dispersión de la zona no es menor; esto indica que la corrección se está haciendo en zonas donde hay mayor incertidumbre. En las imágenes (c) y (d) mostramos el campo de hs antes y después del análisis. 72
8.13. Valor absoluto de Error Cuadrático Medio y Sesgo de los campos con asimilación (ANL) y sin asimilación (no ANL) en metros. Puede observarse que el apartamiento observacional es menor despues de asimilar (en azul). . 73
8.14. Caso de Diciembre de 2012. Valor absoluto del error de hs (en metros) en función de los ciclos de asimilación. 74
8.15. Caso de abril de 2013. Valor absoluto del error hs (en metros) en función de los ciclos de asimilación. El error no crece a medida que pasan los ciclos de asimilación. 75
8.16. Speedup 4D-LETKF en las computadoras del laboratorio del departamento de computación, el mayor costo lo insume la lectura de observaciones. No encontramos explicación para los “saltos”. 77
8.17. Speedup 4D-LETKF en el CECAR, el mayor costo lo insume la lectura de observaciones. 77
8.18. Tiempo de cómputo 4D-LETKF en Hielo-Nieve, notar que el mayor costo lo insume la lectura de observaciones. No encontramos explicación para los “saltos”. 78
8.19. Tiempo de cómputo 4D-LETKF en Hielo-Nieve. La paralelización del LETKF(core del proceso) es buena no llega a ser perfecta, lo cual es esperable. 78

Apéndice

A. PUBLICACIONES

- 2013 “Ensemble Kalman Filter based data assimilation in wave models” P. Etala y **P. Echevarria**, aceptado en 13th International Workshop on Wave Hindcasting and Forecasting and 4th Coastal Hazards Symposium, Banff, Alberta, Canada.
- 2013 “4D-LETKF Data Assimilation in a WAVEWATCH© Wave Model Ensemble” **P. Echevarria** y P. Etala, aceptado en modo poster a WMO Symposium on Data Assimilation, College Park, Maryland, EEUU al que fuí enviado por el Servicio Meteorologico Nacional.
- 2013 “Mapas de asimilación de datos en un modelo de olas”, **P. Echevarria** y P. Etala, 1ra. Conferencia Argentina y Latinoamericana de Python en la Ciencia (SciPyCon Argentina 2013), Puerto Madryn, Argentina.
- 2012 “First results and validation of marine surface wind speed obtained from SAC-D/ Aquarius MWR” , C.Tauro*, Y.Hejazin***, P. Etala**, **P. Echevarria******; M. Jacob*, L. Jones*** (*CONAE, **SHN, ***CFRSL, ****SMN), 2012 Fall Meeting, American Geophysical Union, San Francisco, EEUU.
- 2012 “Storm surge water level data assimilation along the Argentine coast” con M. Saraceno* y **P. Echevarria**** (*UBA/CONICET, **PIDDEF) en “Exploring the Use of Data Assimilation Methods for the Detection and Attribution of Climate Change”, Instituto Franco-Argentino sobre Estudios de Clima y sus Impactos, CNRS-CONICET-UBA, Buenos Aires, Argentina.
- 2012 “Towards assimilation of satellite altimetry products into a storm surge model ensemble along the Argentine coast” P. Etala, M. Saraceno y **P. Echevarria**, 6th Coastal Altimetry Workshop, Riva del Garda, Italia.
- 2012 “Progreso en el modelo de pronóstico de olas hasta un mosaico global multiescala” P. Etala, S. M. Alonso*, D. G. Souto*, C. Romero* y **P. Echevarria**** (*SHN,**PIDDEF), XI Congreso Argentino de Meteorología CONGREMET XI, trabajo completo en línea en http://www.congremet.prmarg.org/upload/etalapaula.pdf, 11 págs

Bibliografía

[qcJ, 2003] (2003). Near real time sea surface height anomaly products for jason-1 and topex/poseidon.

[aus, 2010] (2010). The implementation of the auswave wave model at theaustralian bureau of meteorology.

[ecm, 2012a] (2012a). Part ii: Data assimilation.

[ecm, 2012b] (2012b). Part vii: Ecmwf wave model. chapter 4 data assimilation.

[Behringer, 1994] Behringer, D. (1994). Sea surface height variations in the atlantic ocean: A comparison of topex altimeter data with results from an ocean data assimilation system. *Journal of Geophysical Research: Oceans (1978–2012)*, 99(C12):24685–24690.

[Behringer et al., 1998] Behringer, D. W., Ji, M., and Leetmaa, A. (1998). An improved coupled model for enso prediction and implications for ocean initialization. part i: The ocean data assimilation system. *Monthly Weather Review*, 126(4).

[Chen et al., 2004] Chen, H. S., Behringer, D., Burroughs, L. D., and Tolman, H. L. (2004). A variational wave height data assimilation system for ncep operational wave models. Technical Report MMAB/2004-04, MMAB.

[CLS et al., 2009] CLS, J. P. D., CLS, V. R., cnes, N. P., NASA/JPL, S. D. ., EUMETSAT, H. B., EUMETSAT, J. F., NOAA, J. L., and Altimetrics, R. S. (2009). *OSTM/Jason-2 Products Handbook*. CNES EUMETSAT JPL NOAA/NESDIS.

[Derber and Rosati, 1989] Derber, J. and Rosati, A. (1989). A global oceanic data assimilation system. *Journal of Physical Oceanography*, 19(9):1333–1347.

[Esptein, 1969] Esptein, E. S. (1969). Stochastic dynamic prediction. *Tellus*, XXI(6):739–759.

[Etala et al., 2005] Etala, P. M., Alonso, S. M., and Cristaldo, K. L. (2005). Etapa preoperacional de un sistema de pronóstico de olas para el atlántico sudoccidental y mar argentino.

[Etala et al., 2012a] Etala, P. M., Alonso, S. M., Souto, D., Romero, C., and H., P. E. (2012a). Progreso en el modelo de pronóstico de olas hasta un mosaico global multiescala. *CONGREMET XI*, page 11.

[Etala et al., 2009] Etala, P. M., Alonso, S. M., Souto, D., Romero, C., and Suaya, M. (2009). Pronóstico numérico marino en el shn/smn. In *Congremet X*.

[Etala et al., 2012b] Etala, P. M., Saraceno, M., and Echevarría, P. H. (2012b). Storm surge water level data assimilation along the argentine coast. In *Exploring the Use of Data Assimilation Methods for the Detection and Attribution of Climate Change*. Instituto Franco-Argentino sobre Estudios de Clima y sus Impactos, CNRS-CONICET-UBA, Buenos Aires, Argentina.

[Greenslade, 2004] Greenslade, D. J. M. (2004). The past, present and future of numerical modelling. Technical Report 104, Marine Forecasting at the Bureau of Meteorology.

[Greybush et al., 2011] Greybush, S. J., Kalnay, E., Takemasa, M., Ide, K., and Hunt, B. R. (2011). Balance and ensemble kalman filter localization techniques. *Monthly Weather Review*, 139(2):511–522.

[Hunt et al., 2007] Hunt, B. R., Kostelich, E. J., and Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: a local ensemble transform kalman filter. Technical Report physics/0511236, University of Maryland, College Park MD 20742 Arizona State University, Tempe AZ 85287-1804.

[Kalnay, 2003] Kalnay, E. (2003). *Atmospheric modeling data assimilation and predictability.* Cambridge University Press, Cambridge United Kingdom.

[Kalnay et al., 2008] Kalnay, E., Saulo, C., Ruiz, J. J., Miyoshi, T., Talagrand, O., and Cardinali, C. (2008). Intensive course on data assimilation.

[Kosterlich, 2004] Kosterlich, E. (2004). The local ensemble transform kalman filter (letkf). http://math.asu.edu/ eric/letkf/.

[Lorenc, 1981] Lorenc, A. C. (1981). A global three-dimensional multivariate statistical interpolation scheme. *Monthly Weather Review*, pages 701–721.

[Lorenz, 1963] Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of Oceanic and Atmospheric Technology*, 20:130–141.

[Macrina, 2004] Macrina, M. (2004). *On variational data assimilation in ocean wave models.* PhD thesis, Utrecht Univ.

[Miyoshi, 2005] Miyoshi, T. (2005). Ensemble kalman filter experiments with a primitive-equation global model. Master's thesis, University of Maryland, College Park.

[Ruiz, 2008] Ruiz, J. J. (2008). *Aplicación de los pronósticos por ensambles a la predicción del tiempo a corto plazo sobre Sudamérica.* PhD thesis, Centro de Investigaciones del Mar y la Atmósfera (UBA-CONICET).

[Tolman, 2003] Tolman, H. L. (2003). Running wavewatch iii on a linux cluster. Technical report, Enviromental Modeling Center - Marine Modeling and Analysis Branch.

[Tolman, 2008] Tolman, H. L. (2008). Practical wind wave modeling. In *CBMS Conference Proceedings on Water Waves: Theory and Experiment.* Howard University Word Scientific Publishers.

[Tolman, 2009] Tolman, H. L. (2009). *User manual and system documentation of WAVEWATCH III TM version 3.14.* U. S. Department of Commerce National Oceanic and Atmospheric Administration National Weather Service National Centers for Environmental Prediction 5200 Auth Road Camp Springs, MD 20746.

[Wei et al., 2008] Wei, M., ane Richard Wobous, Z. T., and Zhu, Y. (2008). Initial perturbations based on the ensemble transform (et) technique in the ncep global operational forecast system. *Tellus A*, 60(1):62–79.