



Máster en Ingeniería Computacional y Matemática

Trabajo Final de Máster

Metodología de reconstrucción de estructuras frontales

Nombre Estudiante: JOELSANS CUADRAT

Nombre Director/a: CARMÉ OLIVÉ / MIKHAIL EMELIANOV (ICM)/ JOSEP LLUIS PELEGRÍ (ICM)

Fecha de Entrega: 15 de Julio del 2018

Firma del director autorizando la entrega final del TFM:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2018 Joel Sans

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Joel Sans)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>METODOLOGÍA DE RECONSTRUCCIÓN DE ESTRUCTURAS FRONTALES</i>
Nombre del autor:	<i>JOEL SANS CUADRAT</i>
Nombre del director/a:	<i>CARME OLIVÉ / MIKHAIL EMELIANOV (ICM) / JOSEP LLUIS PELEGRÍ (ICM)</i>
Nombre del PRA:	<i>JUAN ALBERTO RODRÍGUEZ VELÁZQUEZ</i>
Fecha de entrega (mm/aaaa):	<i>07/2018</i>
Titulación:	<i>MÁSTER EN INGENIERÍA COMPUTACIONAL Y MATEMÁTICA</i>
Área del Trabajo Final:	<i>OCEANOGRAFÍA FÍSICA</i>
Idioma del trabajo:	<i>CASTELLANO</i>
Palabras clave	<i>Oceanografía, Física, Frentes</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El presente trabajo fin de máster tiene como objetivo principal contribuir al desarrollo del conocimiento de las ciencias marinas en general y en la oceanografía física en particular.

El objetivo es desarrollar una metodología que nos permita la reproducción de estructuras de dinámica oceánica que nos permita comprender sus mecanismos de funcionamiento.

En este caso en concreto, se trata de desarrollar una metodología para la reproducción de un sistema frontal existente en la confluencia de las corrientes de Brasil y Malvinas.

En abril del 2017 se realizó una campaña oceanográfica a bordo del Hespérides en la zona de confluencia cerca de la costa argentina. En abril y mayo del 2018 se realizó otra campaña oceanográfica dentro del marco del proyecto VA-DE-RETRO en la zona de retroflexión, cerca de la costa de Brasil. Ambas campañas nutren de datos este proyecto de reconstrucción.

Y no sólo eso. Una vez realizada la reconstrucción del frente, se analiza su comportamiento desde el punto de vista del intercambio transfrental. Un punto de vista muy relacionado con la geometría pues como se observa, la posición de las isolíneas es determinante para el citado intercambio.

De este modo, se contribuye también al desarrollo de software para el procesado de datos oceanográficos. En este caso, se emplea el software científico Matlab para la creación de funciones y scripts.

Abstract (in English, 250 words or less):

The present Final Master's Project has the main goal to contribute to develop the knowledge of the marine science in general and in the Physical oceanography in particular.

The aim is develop a methodology that let us to rendering of the dynamical oceanography structures that let us to understand their operating mechanisms.

In this case, we had the need to develop a methodology to render of a frontal system that exist in the confluence of Brazil from north and Malvinas currents from south.

In April 2017 was made a oceanographic cruise on board of RV Hesperides on the confluence zone near the Argentina coast. At April and May 2018 has been made another cruise from VA-DE-RETRO Project, in this case at Brazil coast in a retroflexión zone. Both cruises give us enough data to make this rendering front project.

But this is not all. Once ended the rendering Project of the front, has made an analysis of its behaviour from point of view of transfrontal transferences. A point of view related with mathematical geometry how we will observe. The position of the isolines is determinant to do the frontal exchange.

Of this way, we contribute to develop software to the oceanographic data processing. In this case, we have used the scientific Matlab software to program the all functions and scripts to processing data.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	1
1.4 Planificación del Trabajo.....	2
1.5 Breve resumen de productos obtenidos.....	2
1.6 Breve descripción de los otros capítulos de la memoria.....	4
2. Contextualización del trabajo.....	6
2.1. Equipos de caracterización y muestreo de la columna de agua de mar. Perfiladores verticales y perfiladores remolcados.....	6
2.1.1. CTD - Conductivity, Temperature and Depth.....	7
2.1.2. SeaSoar.....	9
2.1.3. Otros equipos de interés. El ADCP y el LADCP.....	10
2.2. Estrategia y diseño de las campañas oceanográficas.....	11
2.3. Estructura de datos y ficheros.....	13
3. Metodología de reconstrucción de la estructura frontal de la BMC.....	18
3.1. Datos GPS y distancia recorrida.....	18
3.2. Promedio de la distancia recorrida y la presión.....	19
3.3. Promedio de las variables físicas del agua de mar.....	22
3.4. Diseño de las matrices de las secciones verticales.....	24
3.5. Interpolación lineal en secciones verticales.....	25
3.6. El rumbo de la navegación es importante.....	28
3.7. gestor de las llamadas a las funciones.....	29
3.8. Resultados gráficos de la reconstrucción vertical.....	30
3.9. Algunas reflexiones sobre los datos presentados.....	42
3.10. Las secciones horizontales y definición de la matriz.....	42
3.11. Diseño de la matriz horizontal.....	43
3.12. Interpolación lineal en secciones horizontales.....	47
3.13. Resultados gráficos de la reconstrucción horizontal.....	48
4. Análisis del intercambio transfrontal.....	52
4.1. Isolíneas representativas del frente.....	52
4.2. Relación entre las inclinaciones de las isopícnas (σ_t), isoterma (θ) y isohalinas(S).....	54
4.3. Ruido.....	66
6. Glosario.....	67
7. Bibliografía.....	68

Lista de figuras

No se encuentran elementos de tabla de ilustraciones.

Figure 1: Imágenes del perfilador vertical, CTD	Pag.8
Figure 2: Control de vuelo. En rosa el vuelo teórico y en marrón el vuelo real	Pag.9
Figure 3: : Datos de Temperatura, Salinidad, Oxígeno, Clorofila. También la posición GPS y Tiempo	Pag.9
Figure 4: Equipo científico SeaSoar	Pag.10
Figure 5: Mapa de altimetría	Pag.11
Figure 6: : Imagen Satélite. Temperatura superficial	Pag.11
Figure 7: Mapa de navegación	Pag.12
Figure 8: distribuciones de distancia y presión a lo largo del transecto	Pag.19
Figure 9: Temperatura, Salinidad y Densidad Transecto 1	Pag.31
Figure 10: Temperatura, Salinidad y Densidad Transecto 12	Pag.32
Figure 11: Temperatura, Salinidad y Densidad Transecto 2	Pag.33
Figure 12: Temperatura, Salinidad y Densidad Transecto 23	Pag.34
Figure 13: Temperatura, Salinidad y Densidad Transecto 3	Pag.35
Figure 14: Temperatura, Salinidad y Densidad Transecto 34	Pag.36
Figure 15: Temperatura, Salinidad y Densidad Transecto 4	Pag.37
Figure 16: Temperatura, Salinidad y Densidad Transecto 45	Pag.38
Figure 17: Temperatura, Salinidad y Densidad Transecto 5	Pag.39
Figure 18: Temperatura, Salinidad y Densidad Transecto 56	Pag.40
Figure19: Temperatura, Salinidad y Densidad Transecto 6	Pag.41
Figure 20: Distribución temperatura horizontal	Pag.49
Figure 21: Distribución salinidad horizontal	Pag.50
Figure 22: Distribución densidad horizontal	Pag.51
Figure 23: Gradientes de temperatura transecto 1	Pag.55
Figure 24: Inclinaciones γ_T	Pag.60
Figure 25: Inclinaciones γ_S	Pag.61
Figure 26: Inclinaciones γ_p	Pag.62
Figure 27: Rho	Pag.63

Figure 28: Baroclinicidad-Termoclinicidad	Pag.-64
Figure 29: Ejemplo de ruido a 180 metros	Pag.65

1. Introducción

1.1 Contexto y justificación del Trabajo

El Trabajo Final del Máster se enmarca dentro de la oceanografía física, pero sin abandonar ni renunciar a la esencia de la ciencia computacional.

En definitiva, lo que se pretende es contribuir y enriquecer la base del conocimiento mediante el desarrollo de software para la reconstrucción de un sistema frontal y un posterior análisis (breve) del comportamiento del fenómeno reconstruido.

1.2 Objetivos del Trabajo

El trabajo está distribuido en tres grandes bloques. El primero contextualiza el proyecto, para introducirnos en el mundo de la oceanografía. El siguiente bloque, que ocupa la mayor parte del trabajo, desarrolla todas las funciones y scripts (basados en Matlab) que nos permitirán reconstruir la zona de confluencia de las dos corrientes oceánicas de interés: La corriente de Malvinas por el sur y la corriente de Brasil por el norte.

Con todas las herramientas diseñada, vamos a reproducir a pequeña escala la estructura frontal de estudio. Para ello se realizarán secciones verticales y secciones horizontales de la temperatura, la salinidad y la densidad, así como su distribución dentro de los transectos.

Mediante la observación de las isolíneas podremos determinar su comportamiento desde un punto de vista de transferencia transfrental.

Y una vez reconstruido el frente, entraremos de lleno en el análisis de su comportamiento. Seguiremos desarrollando software, en este caso, para el análisis. Las funciones y scripts reproducen los axiomas geométricos postulados por el oceanógrafo *K. N. Fedorov* sobre las características termohalinas (temperatura y salinidad) de los frentes en el océano.

1.3 Enfoque y método seguido

El método para el desarrollo del software para la reconstrucción y el posterior análisis se inicia con la ejecución de las campañas oceanográficas para la caracterización de la zona de estudio. Sin campaña no hay datos, y sin datos no hay proyecto ni producto.

A partir de aquí se establece la mejor manera posible para afrontar el reto que supone el estudio del mecanismo en cuestión.

Se eligen las herramientas adecuadas y se aprende.

1.4 Planificación del Trabajo

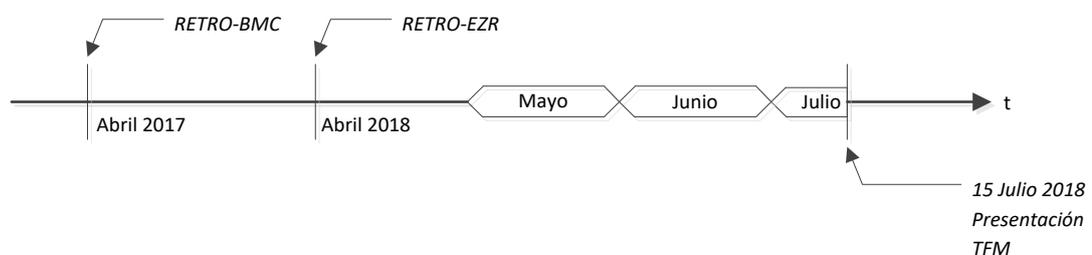
El Trabajo da comienzo cuando se inicia el proyecto del investigador de Instituto de Ciencias del Mar de Barcelona Josep Lluís Pelegrí, *VA-DE-RETRO* y sus correspondientes campañas oceanográficas BMC (Brazil-Malvinas Confluence) y EZR (Equatorial Zonal Retroflexión).

RETRO-BMC se ejecuta en abril del 2017 y RETRO-EZR se ejecuta en abril del 2018.

Una vez finalizadas estas dos campañas oceanográficas en las zonas descritas anteriormente, se procesan los datos obtenidos.

A partir de mayo del 2018 se constituye de un modo formal la necesidad encontrar los grados de termoclinicidad y baroclinicidad del frente y para ello se inicia el presente Trabajo Fin de Máster para la reconstrucción de la zona frontal mediante los datos obtenidos por los equipos científicos usados en campaña y con ello, en junio y julio del 2018, hallar las inclinaciones de las isoclinas y posteriormente, los índices de termoclinicidad y baroclinicidad del frente.

El diagrama temporal de los distintos hitos es el siguiente:



1.5 Breve resumen de productos obtenidos

Los productos que se obtienen del desarrollo del presente TFM son los códigos realizados en MATLAB para alcanzar los objetivos formulados.

Así, se obtienen como resultado el contenido gráfico de los mismos para poder analizar en detalle lo que sucede en la zona frontal.

Así, sin entrar en detalle, se desarrollan los algoritmos:

1. Transectos Verticales

1.1. *sw_dist.m*

Se halla la distancia recorrida entre puntos GPS. La distancia se muestra en kilómetros. La función también nos proporciona la fase del módulo de la distancia.

1.2. *Procesa_vp.m*

Aumenta la resolución de los puntos de GPS. El GPS tiene un *scan rate* de @1Hz. Con esta función se obtienen un scan Rate de @24Hz (1/24).

1.3. *Average_ps.m*

Promedio de todas las variables físicas, Temperatura, Salinidad y Densidad. El promedio se realiza cada metro de profundidad. También se promedia las distancias obtenidas por el GPS.

1.4. *Press2interp.m*

Diseño de la malla 2D cuyas filas representan la profundidad y las columnas, la distancia recorrida. Dicha malla cubre una distancia de unos 80 kilómetros (en función del transecto) y una profundidad de unos 400 metros.

1.5. *Interp_m2m.m*

Función que interpola los datos obtenidos por los equipos científicos para cubrir totalmente la matriz 2D vertical diseñada por la función anterior.

1.6. *Cambio_sentido.m*

El sentido de navegación del buque es importante. Siendo así, nos hemos tomado como referencia la navegación de Sur-Oeste (SO) a Norte-Este (NE). En los transectos que el buque navega de Norte a Sur, se llama la función (los transectos pares).

2. Superficies Horizontales

2.1. *geoavg.m*

Función que prepara la malla 2D para la representación horizontal del frente. Donde el eje x es la Longitud y el eje y es la Latitud. Se puede representar la superficie de cualquier profundidad.

2.2. *design_mh2d.m*

Se construye la matriz con 10000x10000 puntos (Longitud x Latitud).

2.3. *Interp_mh2d.m*

Función que interpola los datos obtenidos por los equipos científicos para cubrir totalmente la matriz 2D horizontal diseñada por la función anterior.

3. Análisis

3.1. *Iso_rep.m*

Script para encontrar la isolínea característica del frente.

3.2. *gran_escala.m*

Mediante esta función se realiza el cómputo de las diferenciales, y de las inclinaciones de las isolíneas. Calcula y grafica los grados de baroclinicidad y termoclinicidad del frente.

1.6 Breve descripción de los otros capítulos de la memoria

Tal y como se expuso en la propuesta del presente TFM, dicha memoria describe la metodología de reconstrucción de estructuras frontales.

Para hacerlo, primero, vamos a contextualizar el dicho Trabajo en su entorno, en el equipamiento científico usado, así como la estructura de datos generada por dichos equipos.

Así pues, el capítulo 2 define el entorno de trabajo, equipamiento, estructura de datos y archivos.

El capítulo 3 ya aborda la metodología de reconstrucción de la estructura frontal en estudio y se presentan las funciones creadas que lo hacen posible y los resultados gráficos de la reconstrucción desde el punto de vista de las isolíneas que caracterizan el frente. Se presentan los plots correspondientes a las secciones verticales como las secciones horizontales a profundidades dadas (constantes).

En el capítulo 4 se describe la reconstrucción de modo experimental de la estructura frontal desde el punto de vista baroclino o termoclino¹, desde el punto de vista de la transferencia. Se desarrollará la teoría de *K. N. Fedorov* sobre las *características termohalinas de los frentes en el océano*. En este caso, también se presentarán algunos plots de los resultados obtenidos.

En el capítulo 5 se abordan las conclusiones del Trabajo realizado.

¹ En el capítulo 4 se definirán de un modo descriptivo estos conceptos.

Reservamos el capítulo 6 para el glosario que define los términos y acrónimos usados en la presente memoria.

Capítulo 7 para la bibliografía convenientemente descrita.

Y finalmente el capítulo 8 se reserva para los agradecimientos.

2. Contextualización del Trabajo

Como ya se ha descrito anteriormente, tanto en la propuesta del TFM, así como en el capítulo anterior, es necesaria una contextualización más profunda del entorno que envuelve dicho trabajo. Y más aún, si el TFM propuesto es *ad-hoc* llevado al campo de la oceanografía.

2.1. Equipos de caracterización y muestreo de la columna de agua de mar. Perfiladores verticales y perfiladores remolcados.

Para poder caracterizar y describir convenientemente la zona de estudio, tanto las secciones horizontales (Latitud y Longitud) como las secciones verticales (distancia y profundidad) se han preparado dos equipos fundamentales para la detección y caracterización de estructuras físicas del agua de mar. Desde el punto de vista de la oceanografía física tienen gran interés las variables que identifican las distintas masas de agua, sus propiedades físicas y su distribución cinemática y dinámica. Estas distribuciones tienen un gran interés también en biología pues de ellas dependen también la distribución tanto de los nutrientes como los organismos que no tienen movilidad propia.

Las variables físicas de interés se obtienen de una forma directa y otras que se obtienen de una manera indirecta. Así pues, las variables directas se obtienen directamente de sensores y las indirectas se obtienen mediante métodos numéricos cuyas variables de entrada son los valores que nos proporcionan los sensores.

Tenemos como variables o magnitudes directas los sensores de:

Variables / Sensores	Unidades	Rango de medida
Temperatura	ITS-90 (°C)	-2 a 35°C
Conductividad	S/m (Siemens/metro)	0 a 7 S/m
Presión	dBar (Decibar)	0 a 6000 dBar

Las variables indirectas se calculan mediante los métodos numéricos normalizados (TEOS-10, www.teos-10.org) y se detallan en la tabla adjunta:

Variable calculada	Unidades	
Salinidad	Practical Scale (PSU)	
Densidad	Kg/m ³	

Destacar que son muchas las variables indirectas que se pueden calcular mediante los sensores primarios, pero en esta tabla solo se detallan las que condicionan el presente estudio.

Los equipos también tienen otros sensores para optimizar el muestreo y enriquecer o aumentar en número las variables medidas. En este caso estamos hablando de las variables de *Clorofila-a*, *Turbidez*, *Oxígeno*, *PAR* (*Photosynthetic Active Radiation*) y *transmitancia*, que enriquecen todo el abanico de variables bio-geo-químicas.

2.1.1. CTD – Conductivity, Temperature and Depth

El CTD (Conductivity, Temperature and Depth) és un equipo que, su principal característica desde el punto de vista de maniobra, se opera a buque parado. El buque una vez alcanzado el punto geográfico de muestreo se para, se aproa a la mar y viento, y realiza la maniobra de largado del CTD.

Otra característica del CTD (ver las imágenes adjuntas) es la capacidad de tomar muestras de agua a distintas profundidades gracias a las 24 botellas niskin de 12 litros y cierre externo que lleva instaladas.

Del CTD se obtienen datos a @24Hz. Eso quiere decir que tenemos 24 datos de presión, temperatura, Conductividad, Salinidad, Densidad... cada segundo. La resolución espacial vertical, si el equipo desciende a 1 m/s es de 24 datos cada metro, lo que significa tener un dato de Presión, Temperatura, Conductividad, Salinidad, Densidad... cada 0.042 metros \approx 0.050 metros = 5cm. Ello nos permite obtener una caracterización con suficiente resolución para hallar las magnitudes físicas que estamos buscando y su distribución en toda la columna de agua.

Una vez terminado la caracterización y el muestreo del punto geográfico, el buque se mueve hacia el próximo punto de estudio. Los puntos CTD (Latitudes y Longitudes) los proporciona el personal científico (habitualmente el Investigador Principal) que cumplen con los objetivos fijados para la campaña.



Sensores de temperatura,
Conductividad y presión

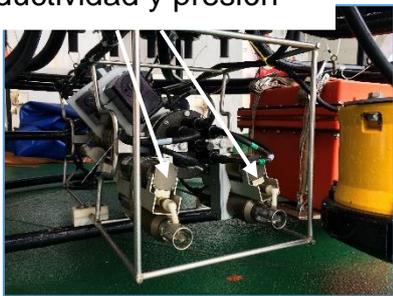
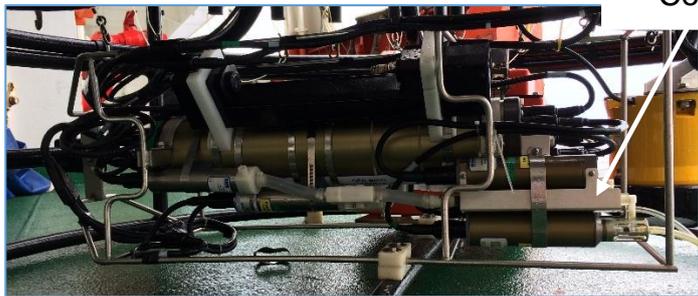


Figure 1: Imágenes del perfilador vertical, CTD

2.1.2. SeaSoar

El SeaSoar es un equipo que incorpora un CTD (igual que el perfilador vertical) pero en este caso está instalado en una plataforma que le permite volar entre aguas. La principal diferencia con un CTD descrito anteriormente es su capacidad de operar mientras el barco se mueve a rumbo fijo a una velocidad constante de 8 knts (nudos) mientras está realizando un vuelo programado de 0 a 400 metros de profundidad.

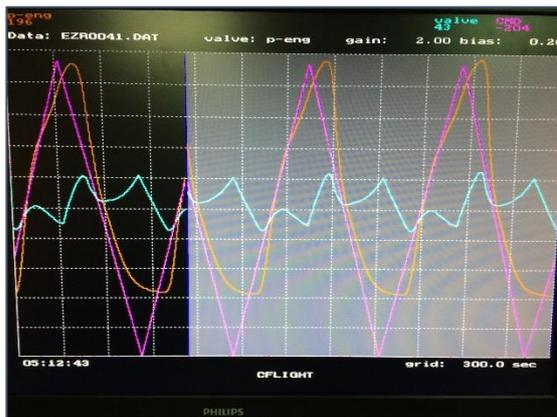


Figure 2: Control de vuelo. En rosa el vuelo teórico y en marrón el vuelo real

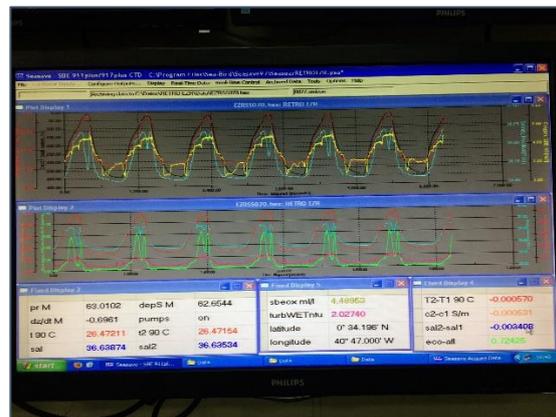


Figure 3: Datos de Temperatura, Salinidad, Oxígeno, Clorofila. También la posición GPS y Tiempo





Figure 4: Equipo científico SeaSoar

La principal ventaja del SeaSoar es que se puede cubrir un área de estudio en un tiempo razonablemente corto lo que permite hallar estructuras que tienen cierta dependencia del tiempo. El SeaSoar consigue datos de una muy alta resolución tanto espacial como temporal.

Otra diferencia remarcable y observable es la inexistencia de botellas niskin. El SeaSoar no es capaz de tomar muestras de agua.

En este caso, obtenemos también un scan rate de @24Hz igual que el CTD perfilador vertical, 24 datos de Temperatura, Conductividad, Salinidad, Densidad... pero uno de los datos que limita el cómputo es el dato del GPS que tenemos un scan reate de @1Hz (un dato de posición por segundo).

2.1.3. Otros equipos de interés. El ADCP y el LADCP

Lo que hemos visto hasta ahora es que el CTD y el SeaSoar, mediante la medida de las presiones, temperaturas, salinidades, densidades pueden calcular las velocidades (módulo y ángulo) geostróficas, velocidades basadas en la fuerza de Coriolis que depende de la latitud y la velocidad generada por los gradientes de presión (efectos termohalinos).

Para obtener una velocidad global compuesta por todas y cada una de las fuerzas que someten a una masa de agua a moverse deberíamos de usar un equipo que usa el efecto dopper para conocer las velocidades y todas sus componentes.

El ADCP (Acoustic Doppler Current Profiler) está instalado en la quilla del barco y mide las velocidades de las corrientes marinas. El CTD también incorpora un LADCP (Lowed Acoustic Dopper Current Profiler) para la adquisición de velocidades de las corrientes que se originan debajo y arriba del equipo.

Todo ello para describir la cinemática y la dinámica de la zona de estudio.

En todo caso y para el desarrollo de este trabajo, solo se considerarán los datos que proporcionan los *CTDs* y el *SeaSoar*.

2.2. Estrategia y diseño de las campañas oceanográficas

El proyecto VA-DE-RETRO lo constituyen dos campañas oceanográficas de oceanografía física a bordo del buque de investigación Hespérides: *RETRO-BMC Brazil-Malvinas Confluence* en abril del 2017 y *RETRO-EZR Equatorial Zonal Retroflexion*, en abril del 2018.

¿Porque en abril? En abril coexisten dos factores importantes: 1) Las estructuras de confluencia y retroflexión de las zonas de estudio están claramente acentuadas y localizadas y 2) El buque Hespérides navega por la zona porque está de regreso de su campaña de apoyo a las Bases Antárticas, rumbo a España.

Mediante el uso de modelos de altimetría e imágenes de satélite que miden la temperatura en la superficie del agua, se determina la ubicación y la zona de muestreo.

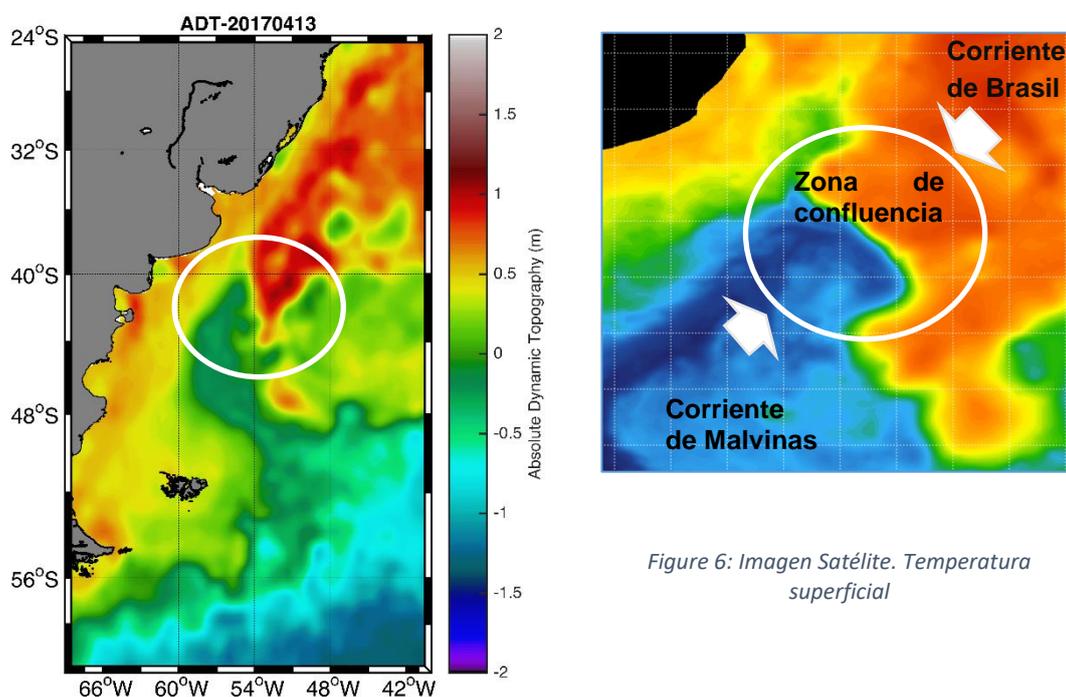


Figure 5: Mapa de altimetría

Una vez autorizado el uso de la plataforma de investigación y conociendo la zona a la que se necesita ir para alcanzar los objetivos planteados, sólo nos queda como diseñar la campaña y prever el uso del equipamiento para sacar un máximo rendimiento a los datos obtenidos.

Para atacar de la mejor manera posible una estructura frontal, el muestreo y los transectos del seasoar serán perpendiculares al frente. Con esa idea, se diseñan los siguientes transectos.

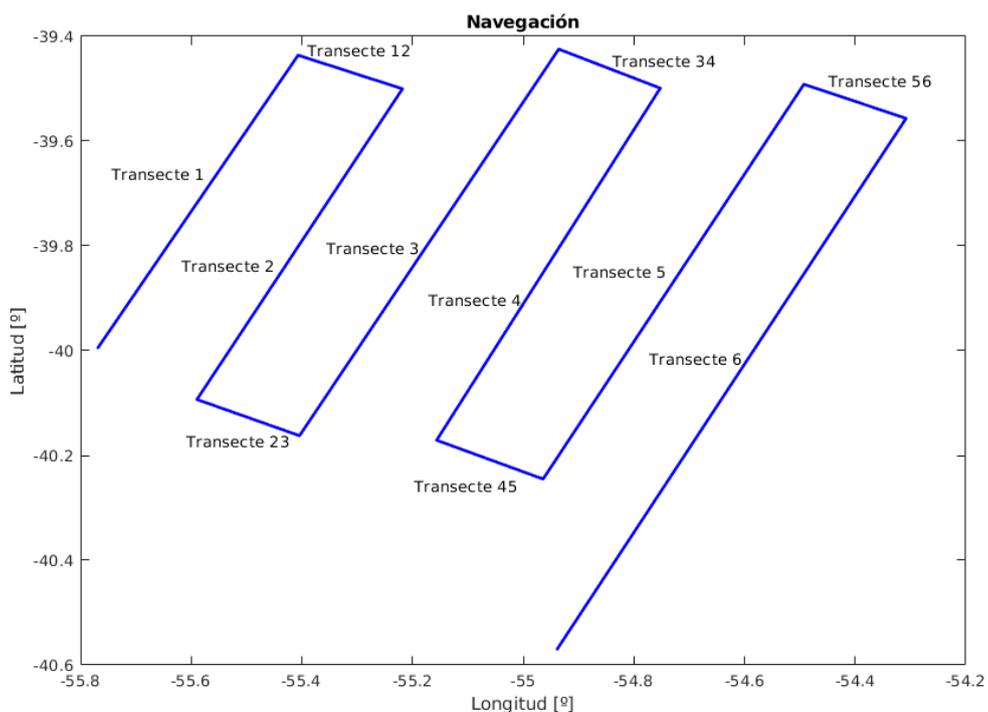


Figure 7: Mapa de navegación

Y sus coordenadas:

# transecto	Latitud Inicial	Longitud Inicial	Latitud Final	Longitud Final
transecto 1	-39.99450	-55.76850	-39.43616	-55.40670
transecto 12	-39.43616	-55.40670	-39.50050	-55.21820
transecto 2	-39.50050	-55.21820	-40.09310	-55.58954
transecto 23	-40.09310	-55.58954	-40.16224	-55.40424
transecto 3	-40.16224	-55.40424	-39.42458	-54.93572
transecto 34	-39.42458	-54.93572	-39.49912	-54.75224
transecto 4	-39.49912	-54.75224	-40.17090	-55.15642
transecto 45	-40.17090	-55.15642	-40.24500	-54.96400
transecto 5	-40.24500	-54.96400	-39.49178	-54.49270
transecto 56	-39.49178	-54.49270	-39.55662	-54.30792
transecto 6	-39.55662	-54.30792	-40.56946	-54.93844

Table 1: Datos de posición geográfica en grados y décimas de grado

2.3. Estructura de datos y ficheros

En esta sección, presentamos la estructura de datos, así como los ficheros del SeaSoar. Creemos que es importante destacar la tipología de datos pues a partir de ahora todo el trabajo depende de ellos.

Los datos que genera el SeaSoar son de tres tipos:

i. .hdr

El fichero *.hdr* es un fichero *header* y en él se describen las características del muestreo así como las características técnicas del equipo usado. Así, aparecen detalles como el nombre del fichero generado, su ubicación en disco, la fecha y la hora, también datos de posición, latitud y longitud. Aparecen también en detalle los sensores instalados, etc. El aspecto del fichero *.hdr* es

```
* Sea-Bird SBE 9 Data File:
* FileName = C:\Datos\RETRO-BMC\Data\SeaSoar001.hdr
* Software Version Seasave V 7.23.1
* Temperature SN = 4746
* Conductivity SN = 3361
* Number of Bytes Per Scan = 37
* Number of Voltage Words = 4
* Number of Scans Averaged by the Deck Unit = 1
* System Upload Time = Jan 08 2003 05:18:55
* NMEA Latitude = 39 59.67 S
* NMEA Longitude = 055 46.11 W
* NMEA UTC (Time) = Apr 17 2017 02:06:56
* Store Lat/Lon Data = Append to Every Scan
* SBE 11plus V 5.2
* number of scans to average = 1
* pressure baud rate = 9600
* NMEA baud rate = 4800
* Latitude/Longitude added to scan
* GPIB address = 1
* advance primary conductivity 0.073 seconds
* advance secondary conductivity 0.073 seconds
* autorun on power up is enabled, do not add lat/lon to scan
* S>
** Ship: HESPERIDES
** Cruise: RETRO BMC
* System UTC = Jan 08 2003 05:18:55
*END*
```

Cabecera del
fichero

ii. .hex

El fichero *.hex* contiene dos partes claramente diferenciadas. En primer lugar, contiene la cabecera del fichero y en segundo lugar contiene los datos del

muestreo. Hay que tener en cuenta que el SeaSoar proporciona un scan rate de @24Hz. La estructura es la que se muestra a continuación:

```
* Sea-Bird SBE 9 Data File:
* FileName = C:\Datos\RETRO-BMC\Data\SeaSoar001.hex
* Software Version Seasave V 7.23.1
* Temperature SN = 4746
* Conductivity SN = 3361
* Number of Bytes Per Scan = 37
* Number of Voltage Words = 4
* Number of Scans Averaged by the Deck Unit = 1
* System Upload Time = Jan 08 2003 05:18:55
* NMEA Latitude = 39 59.67 S
* NMEA Longitude = 055 46.11 W
* NMEA UTC (Time) = Apr 17 2017 02:06:56
* Store Lat/Lon Data = Append to Every Scan
* SBE 11plus V 5.2
* number of scans to average = 1
* pressure baud rate = 9600
* NMEA baud rate = 4800
* Latitude/Longitude added to scan
* GPIB address = 1
* advance primary conductivity 0.073 seconds
* advance secondary conductivity 0.073 seconds
* autorun on power up is enabled, do not add lat/lon to scan
* S>
** Ship: HESPERIDES
** Cruise: RETRO BMC
* System UTC = Jan 08 2003 05:18:55
*END*
0D886B1419A9824E950D73DF146D9F881FFFFB8FFFFFFDFFFFFFF1E836D2A8C49C06137D1
0D886E1419A6824E950D73E6146D9B880FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D2
0D886C1419A8824E950D73DF146D9D881FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D3
0D886E1419A7824E950D73DF146D9B880FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D4
0D886B1419A5824E950D73E2146D9A880FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D5
0D886C1419A6824E950D73E3146D9A880FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D6
0D886B14199E824E8D0D73DC146D9D87FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D7
0D886514199F824E950D73DC146D94880FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137D8
0D886C1419A5824E8D0D73DF146D9B880FFFFB6FFFFFFDFFFFFFF1E836D2A8C49C06137D9
0D88681419A5824E950D73DC146D9B880FFFFB6FFFFFFDFFFFFFF1E836D2A8C49C06137DA
0D88681419A2824E8D0D73D9146D94880FFFFB7FFFFFFDFFFFFFF1E836D2A8C49C06137DB
0D886C1419A4824E8D0D73DC146D9B880FFFFB6FFFFFFDFFFFFFF1E836D2A8C49C06137DC
0D88681419A5824E8D0D73DF146D9F880FFFFB3FFFFFFDFFFFFFF1E836D2A8C49C06137DD
0D88681419A2824E8D0D73DF146D9B880FFFFB1FFFFFFDFFFFFFF1E836D2A8C49C06137DE
0D886C1419A8824E870D73E5146D9D880FFFFB1FFFFFFDFFFFFFF1E836D2A8C49C06137DF
0D88651419A7824E8D0D73E3146D97880FFFFB3FFFFFFDFFFFFFF1E836D2A8C49C06137E0
0D88651419A2824E8D0D73DF146D98880FFFFB5FFFFFFDFFFFFFF1E836D2A8C49C06137E1
0D88681419A7824E860D73E3146D9D880FFFFB5FFFFFFDFFFFFFF1E836D2A8C49C06137E2
```

Cabecera del fichero

Zona de datos del SeaSoar

iii. *.xmlcon

Y el último fichero que genera es el *.xmlcon* que contiene tanto las configuraciones de los sensores conectados al CTD del *SeaSoar* así como los coeficientes de calibración para transformar en valor HEXA del dato a valores ASCII completamente entendibles.

```
<?xml version="1.0" encoding="UTF-8"?>
<SBE_InstrumentConfiguration SB_ConfigCTD_FileVersion="7.23.0.1" >
  <Instrument Type="8" >
    <Name>SBE 911plus/917plus CTD</Name>
    <FrequencyChannelsSuppressed>0</FrequencyChannelsSuppressed>
    <VoltageWordsSuppressed>0</VoltageWordsSuppressed>
    <ComputerInterface>0</ComputerInterface>
    <!-- 0 == SBE11plus Firmware Version >= 5.0 -->
    <!-- 1 == SBE11plus Firmware Version < 5.0 -->
    <!-- 2 == SBE 17plus SEARAM -->
    <!-- 3 == None -->
    <DeckUnitVersion>0</DeckUnitVersion>
    <ScansToAverage>1</ScansToAverage>
    <SurfaceParVoltageAdded>0</SurfaceParVoltageAdded>
    <ScanTimeAdded>0</ScanTimeAdded>
    <NmeaPositionDataAdded>1</NmeaPositionDataAdded>
    <NmeaDepthDataAdded>0</NmeaDepthDataAdded>
    <NmeaTimeAdded>0</NmeaTimeAdded>
    <NmeaDeviceConnectedToPC>0</NmeaDeviceConnectedToPC>
    <SensorArray Size="13" >
      <Sensor index="0" SensorID="55" >
        <TemperatureSensor SensorID="55" >
          <SerialNumber>4746</SerialNumber>
          <CalibrationDate>01-Apr-16</CalibrationDate>
          <UseG_J>1</UseG_J>
          <A>0.00000000e+000</A>
          <B>0.00000000e+000</B>
          <C>0.00000000e+000</C>
          <D>0.00000000e+000</D>
          <F0_01d>0.000</F0_01d>
          <G>4.34842546e-003</G>
          <H>6.38698074e-004</H>
          <I>2.11926621e-005</I>
          <J>1.82864760e-006</J>
          <F0>1000.000</F0>
          <Slope>1.00000000</Slope>
          <Offset>0.0000</Offset>
        </TemperatureSensor>
      </Sensor>
      <Sensor index="1" SensorID="3" >
        <ConductivitySensor SensorID="3" >
          <SerialNumber>3361</SerialNumber>
          <CalibrationDate>13-Jul-16</CalibrationDate>
```

Sin embargo, no podemos trabajar con estos archivos. Será necesario convertirlos en *ASCII*. Para ello se usa el software *SBE Data Processing*⁶ del fabricante del CTD *SeaBird*⁷. Este software convierte el fichero *.hex* a un fichero *.cnv* mediante el uso del fichero *.xmlcon*. Previamente seleccionaremos los datos que son de nuestro interés. Así, la estructura del fichero *.cnv* es:

```

1 -39.99450 -55.76850 107.088148 98.118 5.8658 5.8667 33.9432 33.9406 26.7353 26.7332 5.8717 2.4420e-03 0.433 0.000e+00
2 -39.99450 -55.76850 107.088149 98.118 5.8660 5.8671 33.9428 33.9399 26.7350 26.7326 5.8761 2.4420e-03 0.440 0.000e+00
3 -39.99450 -55.76850 107.088149 98.118 5.8659 5.8667 33.9431 33.9405 26.7352 26.7330 5.8721 2.4420e-03 0.440 0.000e+00
4 -39.99450 -55.76850 107.088150 98.118 5.8660 5.8667 33.9429 33.9403 26.7350 26.7329 5.8765 2.4420e-03 0.440 0.000e+00
5 -39.99450 -55.76850 107.088150 98.118 5.8658 5.8669 33.9429 33.9401 26.7351 26.7327 5.8770 2.4420e-03 0.440 0.000e+00
6 -39.99450 -55.76850 107.088151 98.118 5.8659 5.8669 33.9429 33.9400 26.7351 26.7327 5.8774 2.4420e-03 0.440 0.000e+00
7 -39.99450 -55.76850 107.088151 98.062 5.8658 5.8665 33.9424 33.9406 26.7347 26.7332 5.8822 2.4420e-03 0.440 0.000e+00
8 -39.99450 -55.76850 107.088152 98.118 5.8655 5.8665 33.9427 33.9400 26.7350 26.7327 5.8788 2.4420e-03 0.440 0.000e+00
9 -39.99450 -55.76850 107.088152 98.062 5.8659 5.8667 33.9429 33.9403 26.7350 26.7330 5.8790 2.4420e-03 0.446 0.000e+00
10 -39.99450 -55.76850 107.088152 98.118 5.8657 5.8665 33.9430 33.9405 26.7352 26.7331 5.8794 2.4420e-03 0.446 0.000e+00
11 -39.99450 -55.76850 107.088153 98.062 5.8657 5.8664 33.9428 33.9401 26.7351 26.7328 5.8797 2.4420e-03 0.440 0.000e+00
12 -39.99450 -55.76850 107.088153 98.062 5.8659 5.8665 33.9428 33.9405 26.7350 26.7331 5.8799 2.4420e-03 0.446 0.000e+00
13 -39.99450 -55.76850 107.088154 98.062 5.8657 5.8667 33.9431 33.9406 26.7352 26.7332 5.8802 2.4420e-03 0.464 0.000e+00
14 -39.99450 -55.76850 107.088154 98.062 5.8657 5.8667 33.9428 33.9403 26.7351 26.7330 5.8805 2.4420e-03 0.476 0.000e+00
15 -39.99450 -55.76850 107.088155 98.020 5.8659 5.8670 33.9431 33.9402 26.7352 26.7328 5.8806 2.4420e-03 0.476 0.000e+00
16 -39.99450 -55.76850 107.088155 98.062 5.8655 5.8669 33.9434 33.9398 26.7355 26.7325 5.8809 2.4420e-03 0.464 0.000e+00
17 -39.99450 -55.76850 107.088156 98.062 5.8655 5.8667 33.9430 33.9401 26.7352 26.7328 5.8811 2.4420e-03 0.452 0.000e+00
18 -39.99450 -55.76850 107.088156 98.013 5.8657 5.8669 33.9432 33.9403 26.7354 26.7329 5.8812 2.4420e-03 0.452 0.000e+00
19 -39.99450 -55.76850 107.088157 98.013 5.8657 5.8669 33.9430 33.9402 26.7352 26.7328 5.8814 2.4420e-03 0.452 0.000e+00
20 -39.99462 -55.76856 107.088157 98.013 5.8655 5.8667 33.9429 33.9405 26.7352 26.7331 5.8815 2.4420e-03 0.446 0.000e+00
21 -39.99462 -55.76856 107.088158 97.964 5.8657 5.8672 33.9431 33.9405 26.7353 26.7330 5.8816 2.4420e-03 0.446 0.000e+00
22 -39.99462 -55.76856 107.088158 98.013 5.8657 5.8669 33.9429 33.9402 26.7351 26.7328 5.8817 2.4420e-03 0.440 0.000e+00
23 -39.99462 -55.76856 107.088159 97.964 5.8659 5.8669 33.9431 33.9404 26.7352 26.7330 5.8817 2.4420e-03 0.440 0.000e+00
24 -39.99462 -55.76856 107.088159 97.964 5.8658 5.8669 33.9429 33.9402 26.7351 26.7328 5.8817 2.4420e-03 0.446 0.000e+00
25 -39.99462 -55.76856 107.088160 97.964 5.8658 5.8670 33.9429 33.9402 26.7351 26.7328 5.8817 2.4420e-03 0.446 0.000e+00
26 -39.99462 -55.76856 107.088160 97.964 5.8657 5.8667 33.9437 33.9404 26.7357 26.7330 5.8862 2.4420e-03 0.446 0.000e+00
27 -39.99462 -55.76856 107.088161 97.964 5.8658 5.8670 33.9434 33.9402 26.7355 26.7328 5.8866 2.4420e-03 0.440 0.000e+00
28 -39.99462 -55.76856 107.088161 97.964 5.8660 5.8669 33.9429 33.9407 26.7350 26.7332 5.8826 2.4420e-03 0.440 0.000e+00
29 -39.99462 -55.76856 107.088162 97.908 5.8663 5.8670 33.9428 33.9405 26.7349 26.7330 5.8868 2.4420e-03 0.440 0.000e+00

```

Donde la descripción de los datos se describe a continuación:

# Columna	Dato / sensor
1	Scan
2	Latitud
3	Longitud
4	Día juliano
5	Presión
6	Temperatura sensor 1
7	Temperatura sensor 2
8	Salinidad 1
9	Salinidad 2
10	Densidad 1
11	Densidad 2
12	Oxígeno
13	Fluorescencia
14	Turbidez
15	checksum

Table 2: Datos y sensores

⁶ <http://www.seabird.com/software/sbe-data-processing>

⁷ <http://www.seabird.com>

Como queremos analizar la estructura termohalina del frente, seleccionaremos los datos de temperatura, salinidad y densidad y como son sensores redundantes (dos sensores de temperatura, dos de salinidad), vamos a seleccionar los dos y haremos la media aritmética de ambos para obtener un dato de temperatura, salinidad y densidad por scan.

3. Metodología de reconstrucción de la estructura frontal de la BMC

El primer objetivo a alcanzar, una vez realizado el muestreo de la zona de interés, es la reconstrucción de la estructura frontal en la Confluencia de las corrientes de Brasil y Malvinas.

Para la reconstrucción se realizarán unas secciones verticales cuyos ejes x será la distancia recorrida en metros y el eje y será la profundidad también en metros. Finalmente, el eje z se dibujará la variable física a representar. También se realizarán las secciones horizontales a distintas profundidades, donde el eje x será la longitud y el eje y será la latitud. En eje z será la variable física. Realizaremos las correspondientes *matrices* y los *contours* correspondientes para una visualización clara del frente muestreado.

3.1. Datos de GPS y distancia recorrida

Al tratar los datos, nos encontramos con una dificultad añadida. La tasa de muestreo de los datos de posición no es la misma que los datos físicos del SeaSoar, siendo el muestro de la Latitud y Longitud geográfica 24 veces menor que la tasa de muestro de los datos físicos.

Siendo así, tenemos dos opciones: 1) incremento el scan rate del GPS o 2) disminuimos el scan rate de los datos físicos⁸.

No se contempla la opción de disminuir el scan rate de los sensores del SeaSoar pues ya es costoso mover un barco y el personal – tripulación, técnicos y científicos - para que al final haya que reducir los datos físicos muestreados. Y no solo eso, no se puede reducir el tamaño de los datos si queremos unas resoluciones espacio-temporales grandes.

Para aumentar el muestreo del GPS haremos el ajuste de la resolución espacial del posicionamiento (GPS) a la frecuencia de muestreo con el SeaSoar:

:

$$GPS = (pos2 - pos1)/24$$

Así, obtenemos:

$$pos1, pos1 + GPS, pos1 + 2 \cdot GPS, pos1 + 3 \cdot GPS, \dots, pos1 + 24 \cdot GPS = pos2$$

⁸ Temperatura, Salinidad y Densidad

Haciendo esta pequeña maniobra, ahora cada dato de presión, temperatura, salinidad y densidad está relacionado con un dato de posición y distancia recorrida.

Para realizar los perfiles verticales, en el eje x estará la distancia y en el eje y se encuentra la profundidad (distancia, profundidad), transformaremos las posiciones GPS a distancia en metros. Para ello usaremos la función *sw_dist.m* que utiliza el método *Plane Sailing* para realizar la transformación.

La función proporciona la distancia recorrida entre puntos GPS y no es acumulativa. Es por ello que hay que tener en cuenta la suma del recorrido realizado entre puntos GPS.

El muestreo sigue el siguiente esquema:

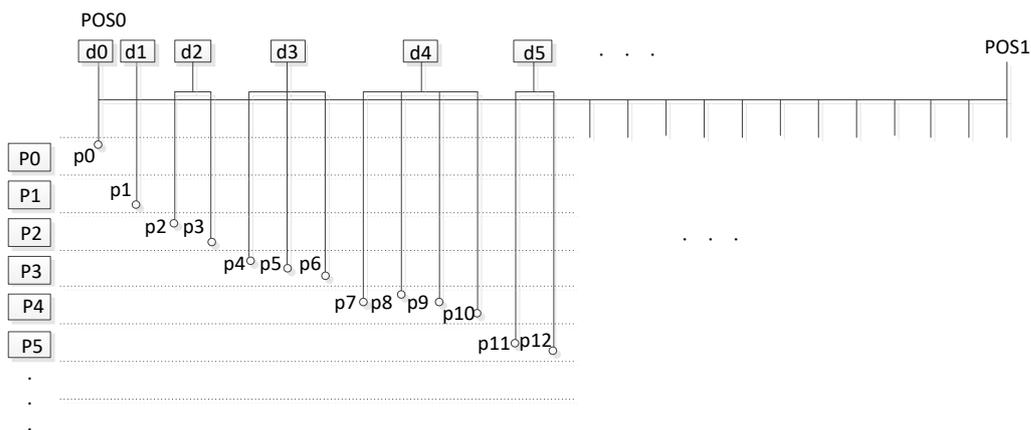


Figure 8: distribuciones de distancia y presión a lo largo del transecto

3.2. Promedio de la distancia recorrida y la presión

Teniendo en cuenta lo anteriormente expuesto, ahora será necesario promediar los datos de presión. Es muy habitual, para reconstrucciones tipo *contour* promediar la presión (o profundidad) cada metro. Hay que pensar que si obtenemos 24 datos de presión cada segundo y la velocidad de ascenso y descenso del equipo es de $1m/s$, obtenemos 24 datos de presión dentro de 1 metro recorrido. Así pues, una buena práctica es promediar cada metro de profundidad.

Pero hay otro factor que hemos de tener en cuenta, la distancia recorrida. Hay que pensar en mantener también las proporciones. Siendo así, y si promediamos cada metro en profundidad, promediaremos cada 5 metros en distancia si no hay cambios en la profundidad. Si hubiese cambios en la profundidad antes de alcanzar los 5 metros de distancia, el promediado será antes.

Esto nos conduce a pensar que, si bien las profundidades son equidistantes, cada valor de la distancia recorrida resultante no será constante, en tanto en cuanto, a la práctica, la velocidad del buque tampoco lo es.

Una vez consideradas estas cuestiones podemos plantear la siguiente estructura de ejemplo que sigue la figura 3. La estructura es una matriz cuyas columnas es la distancia recorrida y la presión, las filas.

$$\begin{bmatrix}
 p_0 & NaN & NaN & NaN & NaN & NaN & \dots \\
 NaN & p_1 & NaN & NaN & NaN & NaN & \dots \\
 NaN & NaN & (p_2 + p_3)/2 & NaN & NaN & NaN & \dots \\
 NaN & NaN & NaN & (p_4 + p_5 + p_6)/3 & NaN & NaN & \dots \\
 NaN & NaN & NaN & NaN & (p_7 + p_8 + p_9 + p_{10})/4 & NaN & \dots \\
 NaN & NaN & NaN & NaN & NaN & (p_{11} + p_{12})/2 & \dots \\
 NaN & NaN & NaN & NaN & NaN & NaN & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{bmatrix}$$

Y no solo eso, también podemos plantear estructuras similares para las distintas variables medidas. Así, la temperatura quedaría:

$$\begin{bmatrix}
 t_0 & NaN & NaN & NaN & NaN & NaN & \dots \\
 NaN & t_1 & NaN & NaN & NaN & NaN & \dots \\
 NaN & NaN & (t_2 + t_3)/2 & NaN & NaN & NaN & \dots \\
 NaN & NaN & NaN & (t_4 + t_5 + t_6)/3 & NaN & NaN & \dots \\
 NaN & NaN & NaN & NaN & (t_7 + t_8 + t_9 + t_{10})/4 & NaN & \dots \\
 NaN & NaN & NaN & NaN & NaN & (t_{11} + t_{12})/2 & \dots \\
 NaN & NaN & NaN & NaN & NaN & NaN & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{bmatrix}$$

Y la salinidad:

$$\begin{bmatrix}
 s_0 & NaN & NaN & NaN & NaN & NaN & \dots \\
 NaN & s_1 & NaN & NaN & NaN & NaN & \dots \\
 NaN & NaN & (s_2 + s_3)/2 & NaN & NaN & NaN & \dots \\
 NaN & NaN & NaN & (s_4 + s_5 + s_6)/3 & NaN & NaN & \dots \\
 NaN & NaN & NaN & NaN & (s_7 + s_8 + s_9 + s_{10})/4 & NaN & \dots \\
 NaN & NaN & NaN & NaN & NaN & (s_{11} + s_{12})/2 & \dots \\
 NaN & NaN & NaN & NaN & NaN & NaN & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{bmatrix}$$

Y la densidad:

ρ_0	NaN	NaN	NaN	NaN	NaN	NaN	...
NaN	ρ_1	NaN	NaN	NaN	NaN	NaN	...
NaN	NaN	$(\rho_2 + \rho_3)/2$	NaN	NaN	NaN	NaN	...
NaN	NaN	NaN	$(\rho_4 + \rho_5 + \rho_6)/3$	NaN	NaN	NaN	...
NaN	NaN	NaN	NaN	$(\rho_7 + \rho_8 + \rho_9 + \rho_{10})/4$	NaN	NaN	...
NaN	NaN	NaN	NaN	NaN	NaN	$(\rho_{11} + \rho_{12})/2$...
NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
.
.

También podríamos hacer lo mismo con el oxígeno, fluorómetro y turbidímetro⁹.

En este caso, las columnas serían las distancias recorridas y las filas la profundidad alcanzada por el equipo SeaSoar.

Para conseguir estas estructuras primero y calculadas las distancias no acumuladas, hay que calcular las mismas distancias, pero en este caso sí serán acumuladas. Las distancias acumuladas representarán las columnas de las matrices. Para hacerlo, hemos construido la siguiente función en Matlab llamada *procesa_vp.m*

<i>procesa_vp.m</i>			
<i>Variables de salida</i>		<i>Variables de entrada</i>	
<i>ddifm</i>	<i>Vector de posiciones</i>	<i>Dist</i>	Dato que proporciona la anterior función <i>sw_dist.m</i>
<i>pdifr</i>	<i>Vector de fases</i>	<i>Phase</i>	Ángulo. Coordenadas polares en lugar del Lat, Lon.

Función: *procesa_vp.m*

```
function [ddifm,pdifr]=procesa_vp(Dist,Phase)
%mediante esta función, en lugar de eliminar los ceros del vector de
%distancias generado por sw_dist, lo que hacemos es otorgar una
distancia a
%cada 0 para darle continuidad a la posición y adaptar los @24Hz del
CTD a
%1Hz del GPS
nzd=0; np=0;
xi=Dist(1); %distancia inicial
ai=Phase(1); %ángulo inicial
ddif(1)=xi; pdif(1)=ai;
for i=2:length(Dist)
    if Dist(i)==0
        nzd=nzd+1;
        %pdif(i)=pdif(i-1);
    else
        if Phase(i)~=0
```

⁹ Sensores del CTD del SeaSoar auxiliares para aumentar la capacidad de adquisición de datos en las disciplinas de Biología, Química y Geología.

```

        np=np+1;
        pdif(np)=Phase(i);
    %else
        %pdif(i)=pdif(i-1);
    end

    d=i-nzd;
    xf=Dist(i);
    xadd=xi+xf;
    xdif=(xadd-xi)/(nzd+1);
    for j=d:i
        ddif(j)=ddif(j-1)+xdif;
    end
    xi=xadd;
    nzd=0;
end
end
%La distancia está en Kilómetros. Como la presión es en dBar que es
igual
%al metro, los kilometros los pasamos a metros.
ddifm=ddif*1000;
pdifr=pdif*2*pi/360;
end

```

Una vez obtenidos los valores de las distancias y presiones deseadas, haremos los promedios de las distintas variables del SeaSoar.

3.3. Promedio de las variables físicas del agua de mar

Para realizar el promedio, se ha diseñado una función que promedia dichas variables cuando se encuentran a la misma distancia (Latitud y Longitud), y a la misma profundidad. La función *average_ps.m* tiene la siguiente tabla de variables de entrada y salida.

<i>Average_ps</i>			
<i>Variables de salida</i>		<i>Variables de entrada</i>	
DistAvg	En la salida vamos a tener las variables promediadas.	Ddifm	En la entrada tenemos todas las variables.
PressAvg		Press	
TempAvg		Temp	
SaltAvg		Salt	También vamos a tener la distancia acumulada calculada por la función anterior <i>procesa_vp.m</i>
SigmaAvg		Sigma	
OxAvg		Ox	
FluorAvg		Fluor	
TurbAvg		Turb	

```

function
[DistAvg, PressAvg, TempAvg, SaltAvg, SigmaAvg, OxAvg, FluorAvg, TurbAvg]=...
average_ps (Ddifm, Press, Temp, Salt, Sigma, Ox, Fluor, Turb)
%Esta funcion es igual que el average pero tiene en cuenta no solo la
%presion, tambien tiene en cuenta el tiempo. Hemos visto que en las
zonas
%donde el SeaSoar sigue la misma isobara (sobre todo en los extremos
de su
%vuelo) puede pasar 80-100 metros sin variar su profundidad. Si no
tenemos
%en cuenta el tiempo, se producen unos "blancos" en esas zonas.
%
%El siguiente codigo pretende corregir este efecto.
PressR=round(Press); %Redondeo al metro de la profundidad
cp=1;
np=1;
dacu=0; davg=5;
PressAvg(1)=PressR(1);
TempAvg(1)=Temp(1);
SaltAvg(1)=Salt(1);
SigmaAvg(1)=Sigma(1);
OxAvg(1)=Ox(1); FluorAvg(1)=Fluor(1); TurbAvg=Turb(1);
xi=Ddifm(1); xf=0;
for i=2:length(Ddifm)
    if PressAvg(cp)==PressR(i) && Ddifm(i)<davg+dacu
        TempAvg(cp)=TempAvg(cp)+Temp(i);
        SaltAvg(cp)=SaltAvg(cp)+Salt(i);
        SigmaAvg(cp)=SigmaAvg(cp)+Sigma(i);
        OxAvg(cp)=OxAvg(cp)+Ox(i);
        FluorAvg(cp)=FluorAvg(cp)+Fluor(i);
        TurbAvg(cp)=TurbAvg(cp)+Turb(i);
        np=np+1;
    else
        xf=Ddifm(i);
        dacu=round(Ddifm(i));
        drec=(xf-xi)/2;
        DistAvg(cp)=xi+drec;
        TempAvg(cp)=TempAvg(cp)/(np);
        SaltAvg(cp)=SaltAvg(cp)/(np);
        SigmaAvg(cp)=SigmaAvg(cp)/(np);
        OxAvg(cp)=OxAvg(cp)/(np);
        FluorAvg(cp)=FluorAvg(cp)/(np);
        TurbAvg(cp)=TurbAvg(cp)/(np);
        cp=cp+1;
        PressAvg(cp)=PressR(i);
        TempAvg(cp)=Temp(i);
        SaltAvg(cp)=Salt(i);
        SigmaAvg(cp)=Sigma(i);
        OxAvg(cp)=Ox(i);
        FluorAvg(cp)=Fluor(i);
        TurbAvg(cp)=Turb(i);
        np=1;
        xi=xf;
    end
end
%Distancia
DistAvg(cp)=xi+drec;
%Temperatura

```

```

TempAvg (cp) =TempAvg (cp) / (np) ;
%Salinidad
SaltAvg (cp) =SaltAvg (cp) / (np) ;
%Densidad
SigmaAvg (cp) =SigmaAvg (cp) / (np) ;
%Oxigeno
OxAvg (cp) =OxAvg (cp) / (np) ;
%Fluorescencia
FluorAvg (cp) =FluorAvg (cp) / (np) ;
%Turbidez
TurbAvg (cp) =TurbAvg (cp) / (np) ;
end

```

3.4. Diseño de las matrices de las secciones verticales

Una vez hecho todo lo anterior, ya disponemos de todos los datos necesarios para realizar una buena reconstrucción de la zona frontal de confluencia, sólo será necesario distribuir los datos en la matriz y representarla.

Para distribuir los datos, observemos que cada dato de Temperatura, Salinidad y Densidad está asociado a una profundidad y a una distancia. Por lo tanto:

$$T(x, y) = T(d, p) = t$$

$$S(x, y) = S(d, p) = s$$

$$\rho(x, y) = \rho(d, p) = \rho$$

Se crean tantas matrices como variables tenemos. Así, se crearán *Tsver*, *Ssver* y *psver* (secciones verticales) para cada uno de los transectos realizados.

Para ello creamos la función *press2interp.m* que tiene una doble función; distribuir los datos en las matrices descritas y preparar las matrices para realizar una interpolación lineal con los datos verdaderos.

Así, la función *press2interp.m* tiene las siguientes variables de entrada y salida.

<i>press2interp.m</i>			
Variables de salida		Variables de entrada	
Xvec	Vector de distancia	DistAvg	Distancia Promediada
Yvec	Vector de profundidades	PressAvg	Presión promediada
Xmat	Matriz de X	TempAvg	Temperatura promediada
Ymat	Matriz de Y	SaltAvg	Salinidad promediada
Tsver	Sección vertical Temperatura	SigmaAvg	Densidad promediada
Ssver	Sección vertical Salinidad	OxAvg	Oxígeno promediado
Rsver	Sección vertical Densidad	FluorAvg	Fluorescencia promediada
Osver	Sección vertical Oxígeno	TurbAvg	Turbidez promediado
Fsver	Sección vertical Fluor		
Usver	Sección vertical Turb		

```

function [Xvec, Yvec, Xmat, Ymat, Tsver, Ssver, Rsver, Osver, Fsver, Usver, ...
Asver, Bsver]=press2interp(DistAvg, PressAvg, TempAvg, SaltAvg, ...
SigmaAvg, OxAvg, FluorAvg, TurbAvg, alpha, beta)
Xvec=DistAvg; %X vector (distancia)
Yvec=1:max(PressAvg); %Y vector (Presion)
[Xmat, Ymat]=meshgrid(Xvec, Yvec); %X matrix, Y matrix
k=1;
for i=1:length(Yvec)
    for j=1:length(Xvec)
        if PressAvg(j)==i
            Tsver(i, j)=TempAvg(j); %Tsver=Temperatura Seccion Vertical
            Ssver(i, j)=SaltAvg(j); %Ssver=Salinidad Seccion Vertical
            Rsver(i, j)=SigmaAvg(j); %Rsver=Densidad Seccion Vertical
            Osver(i, j)=OxAvg(j); %Osver=Oxigeno Seccion Vertical
            Fsver(i, j)=FluorAvg(j); %Fsver=Fluorescencia Seccion
Vertical
            Usver(i, j)=TurbAvg(j); %Usver=Turbidez Seccion Vertical
            Asver(i, j)=alpha(j);
            Bsver(i, j)=beta(j);
        else
            Tsver(i, j)=NaN;
            Ssver(i, j)=NaN;
            Rsver(i, j)=NaN;
            Osver(i, j)=NaN;
            Fsver(i, j)=NaN;
            Usver(i, j)=NaN;
            Asver(i, j)=NaN;
            Bsver(i, j)=NaN;
        end
    end
end
end
end

```

En esta función se incorporan dos variables más: α y β (alpha y beta). Ambas variables las calcula la función *gsw_alpha.m* y *gsw_beta.m* de L TEOS-10. Estas dos variables relacionan tanto la salinidad como la temperatura con la densidad y serán necesarias para el cálculo de la Rro que al final nos indicará el grado de baroclinicidad y termoclinicidad del frente. Pero estos conceptos ya se desarrollarán en el capítulo siguiente.

3.5. Interpolación lineal en secciones verticales

Hasta ahora se han diseñado las matrices que contienen los datos promediados adquiridos por los equipos. A partir de ahora, los huecos existentes en las matrices de datos serán sustituidos por los datos originados por las interpolaciones lineales propuestas.

Para resolver las interpolaciones, haremos uso de la función integrada del entorno de programación Matlab *interp1*.

Para realizar las interpolaciones se ha creado una función llamada `interp_m2m.m`.

Así, la función `press2interp.m` tiene las siguientes variables de entrada y salida.

<i>Interp_m2m.m</i>			
Variables de salida		Variables de entrada	
Tvli	Temperatura interpolada	Xvec	Vector de X (distancia)
Svli	Salinidad interpolada	Yvec	Vector de Y (profundidad)
Rvli	Densidad interpolada	Xmat	Matriz de X
Ovli	Oxígeno interpolada	Ymat	Matriz de Y
Fvli	Fluorescencia interpolada	Tsver	Sección vertical de Temperatura
Uvli	Turbidez interpolada	Ssver	Sección vertical de Salinidad
Avli	Alpha interpolada	Rsver	Sección vertical de Densidad
Bvli	Beta interpolada	Osver	Sección vertical de Oxígeno
		Fsver	Sección vertical de Fluorescencia
		Usver	Sección vertical de turbidez
		Asver	Sección vertical de alpha
		Bsver	Sección vertical de beta

Función: `interp_m2m.m`

```
function [Tvli, Svli, Rvli, Ovli, Fvli, Uvli, Avli, Bvli]=interp_m2m ...
    (Xvec, Yvec, Xmat, Ymat, Tsver, Ssver, Rsver, Osver, Fsver, ...
    Usver, Asver, Bsver)
for j=1:length(Yvec)
    t=[]; Tinterp=[]; x=[]; xq=[];
    s=[]; Sinterp=[];
    r=[]; Rinterp=[];
    o=[]; Ointerp=[];
    f=[]; Finterp=[];
    u=[]; Uinterp=[];
    a=[]; Ainterp=[];
    b=[]; Binterp=[];
    k=0; ini=0; inRange=0; h=0; p=0;
    for i=1:length(Xvec)
        if isnan(Tsver(j,i))
            if ini==1
                k=k+1;
                xq(k+1)=Xmat(j,i);
            end
        else
            if k~=0
                t(2)=Tsver(j,i);
                s(2)=Ssver(j,i);
                r(2)=Rsver(j,i);
                o(2)=Osver(j,i);
                f(2)=Fsver(j,i);
                u(2)=Usver(j,i);
                a(2)=Asver(j,i);
                b(2)=Bsver(j,i);
                x(2)=Xmat(j,i);
                xq(k+2)=x(2);
                Tinterp=interp1(x,t,xq);
                Sinterp=interp1(x,s,xq);
                Rinterp=interp1(x,r,xq);
```

```

Ointerp=interp1(x,o,xq);
Finterp=interp1(x,f,xq);
Uinterp=interp1(x,u,xq);
Ainterp=interp1(x,a,xq);
Binterp=interp1(x,b,xq);
for n=1:length(Tinterp)
    Tsver(j,i-length(Tinterp)+n)=Tinterp(n);
    Ssver(j,i-length(Sinterp)+n)=Sinterp(n);
    Rsver(j,i-length(Rinterp)+n)=Rinterp(n);
    Osver(j,i-length(Ointerp)+n)=Ointerp(n);
    Fsver(j,i-length(Finterp)+n)=Finterp(n);
    Usver(j,i-length(Uinterp)+n)=Uinterp(n);
    Asver(j,i-length(Ainterp)+n)=Ainterp(n);
    Bsver(j,i-length(Binterp)+n)=Binterp(n);
end
k=0; ini=1; xq=[];
t(1)=t(2); x(1)=x(2); xq(1)=x(2);
s(1)=s(2);
r(1)=r(2);
o(1)=o(2);
f(1)=f(2);
u(1)=u(2);
a(1)=a(2);
b(1)=b(2);
else
    t(1)=Tsver(j,i);
    s(1)=Ssver(j,i);
    r(1)=Rsver(j,i);
    o(1)=Osver(j,i);
    f(1)=Fsver(j,i);
    u(1)=Usver(j,i);
    a(1)=Asver(j,i);
    b(1)=Bsver(j,i);
    x(1)=Xmat(j,i);
    xq(1)=x(1);
    ini=1;
end
end
end
Tvli=Tsver;
Svli=Ssver;
Rvli=Rsver;
Ovli=Osver;
Fvli=Fsver;
Uvli=Usver;
Avli=Asver;
Bvli=Bsver;
end

```

3.6. El rumbo de navegación es importante

El rumbo es muy importante si queremos conservar cierta coherencia. Los transectos impares (1, 3 y 5), la navegación del buque y el muestreo de la columna de agua se realiza de SO a NE, sin embargo, cuando realizamos los transectos pares (2, 4 y 6) el buque navega de NE a SO. Siendo así, será absolutamente necesario crear una función que “gire” literalmente la matriz para que todos los transectos se desarrollen en el mismo sentido.

Se ha creado una función llamada *cambio_sentido.m* que se llamará cuando sea necesario.

En este caso, las variables de entrada serán las mismas que las variables de salida de la función.

<i>Cambia_sentido.m</i>			
Variables de salida		Variables de entrada	
Lat	Latitud	Latc	Distancia Promediada
Lon	Longitud	Lonc	Presión promediada
Press	Presión	Pressc	Temperatura promediada
Temp	Temperatura	Tempc	Salinidad promediada
Salt	Salinidad	Saltc	Densidad promediada
Sigma	Sigma	Sigmatc	Oxígeno promediado

Función: cambio_sentido.m

```
function [Lat, Lon, Press, Temp, Salt, Sigma]=cambio_sentido(Latc, ...
    Lonc, Pressc, Tempc, Saltc, Sigmac)
%Per canviar el sentit del transecte
for i=1:length(Latc)
    j=length(Latc)-i+1;
    Lata(i)=Latc(j);
    Lona(i)=Lonc(j);
    Pressa(i)=Pressc(j);
    Tempa(i)=Tempc(j);
    Salta(i)=Saltc(j);
    Sigmaa(i)=Sigmac(j);
end
Lat=Lata;
Lon=Lona;
Press=Pressa;
Temp=Tempa;
Salt=Salta;
Sigma=Sigmaa;
end
```

3.7. Gestor de llamadas a las funciones

Y finalmente para terminar con las funciones que generan las secciones verticales, se ha construido un gestor de llamadas y creador de gráficos tipo *contour* para desarrollar los distintos transectos.

Función: *vert_sect.m*

```
cambiosentido=1;
Lat=Lat_leg2; Lon=Lon_leg2;
Temp=(Temp1_leg2+Temp2_leg2)/2; Salt=(Salt1_leg2+Salt2_leg2)/2;
Sigma=(Sigma1_leg2+Sigma2_leg2)/2;
Press=Press_leg2;
Ox=Ox_leg2; Fluor=Fluor_leg2; Turb=Turb_leg2;
if cambiosentido==1
    [Lat, Lon, Press, Temp, Salt, Sigma]=cambio_sentido ...
        (Lat, Lon, Press, Temp, Salt, Sigma);
end
[Dist, Phase]=sw_dist(Lat, Lon, 'km');
[ddifm, pdifm]=procesa_vp(Dist, Phase);
[DistAvg, PressAvg, TempAvg, SaltAvg, SigmaAvg, OxAvg, FluorAvg, TurbAvg]=
...
    average_ps(ddifm, Press, Temp, Salt, Sigma, Ox, Fluor, Turb);
[alpha]=sw_alpha(SaltAvg, TempAvg, PressAvg, 'temp');
[beta]=sw_beta(SaltAvg, TempAvg, PressAvg, 'temp');
[Xvec, Yvec, Xmat, Ymat, Tsver, Ssver, Rsver, Osver, Fsver, Usver, Asver, ...
    Bsver]=press2interp(DistAvg, PressAvg, TempAvg, SaltAvg, SigmaAvg, ...
    OxAvg, FluorAvg, TurbAvg, alpha, beta);
[Tvli, Svli, Rvli, Ovli, Fvli, Uvli, Avli, Bvli]=interp_m2m(Xvec, Yvec, Xmat, ...
.
    Ymat, Tsver, Ssver, Rsver, Osver, Fsver, Usver, Asver, Bsver);
Ymat=-Ymat;
%Contour de TEMPERATURA
figure;
contour(Xmat, Ymat, Tvli, 'ShowText', 'on', 'Fill', 'on', 'LevelStep', 1, ...
    'LineColor', [0.15 0.15 0.15]);
title('Transecto 6 - Temperatura ITS90');
xlabel('distancia [m]');
ylabel('Profundidad [m]');
grid on
cmocean('thermal');
%Contour de SALINIDAD
figure;
contour(Xmat, Ymat, Svli, 'ShowText', 'on', 'Fill', 'on', 'LevelStep', 0.2, ...
    'LineColor', [0.8 0.8 0.8]);
title('Transecto 6 - Salinidad PSU');
xlabel('distancia [m]');
ylabel('Profundidad [m]');
grid on
cmocean('haline');
%Contour de DENSIDAD
figure;
contour(Xmat, Ymat, Rvli, 'ShowText', 'on', 'Fill', 'on', 'LevelStep', 0.2, ...
    'LineColor', [0.6 0.18 0.56]);
title('Transecto 6 - Densidad kg/m³');
xlabel('distancia [m]');
ylabel('Profundidad [m]');
```

```

grid on
cmocean('dense');
%Contour de OXÃ?GENO
figure;
contour(Xmat,Ymat,Ovli,'ShowText','on','Fill','on','LevelStep',0.2,...
'LineColor',[0.93 0.69 0.13]);
title('Transecto 1 - OxÃ-geno');
xlabel('distancia [m]');
ylabel('Profundidad [m]');
grid on
cmocean('oxy');
%Contour de FLUORESCENCIA
figure;
contour(Xmat,Ymat,Fvli,'ShowText','on','Fill','on','LevelStep',0.0025,
...
'LineColor',[0.6 0.18 0.56]);
title('Transecto 1 - Fluorescencia');
xlabel('distancia [m]');
ylabel('Profundidad [m]');
grid on
cmocean('algee');
%Contour de TURBIDEZ
figure;
contour(Xmat,Ymat,Uvli,'ShowText','on','Fill','on','LevelStep',0.001,.
..
'LineColor',[0.6 0.18 0.56]);
title('Transecto 1 - Turbidez');
xlabel('distancia [m]');
ylabel('Profundidad [m]');
grid on
cmocean('turbid');

```

Si se ejecuta el gestor de llamadas *vert_sect.m* y con los datos originales presentados en el capítulo anterior, se logra generar los *contous(X,Y,Z)* siendo la X la distancia, la Y la profundidad y la Z se correspondería a la variable física que queremos representar.

En la siguiente sección se muestran los resultados de la ejecución de todas las funciones descritas anteriormente.

Para no dibujar todas y cada una de las variables, sólo se dibujarán y se presentarán aquellas variables que tienen una influencia directa al comportamiento de la zona frontal. Estas son: la Temperatura, la Salinidad y finalmente la Densidad. No nos olvidemos de la presión.

3.8. Visualización de los resultados gráficos de las secciones verticales

Esta reconstrucción se basa en isolíneas, es decir, se ve perfectamente cómo se distribuye la temperatura, la salinidad y la densidad a lo largo de los transectos verticales diseñados.

Transecto 1

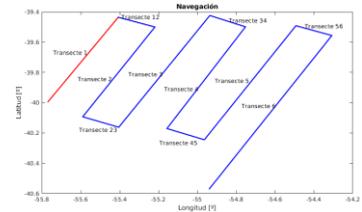
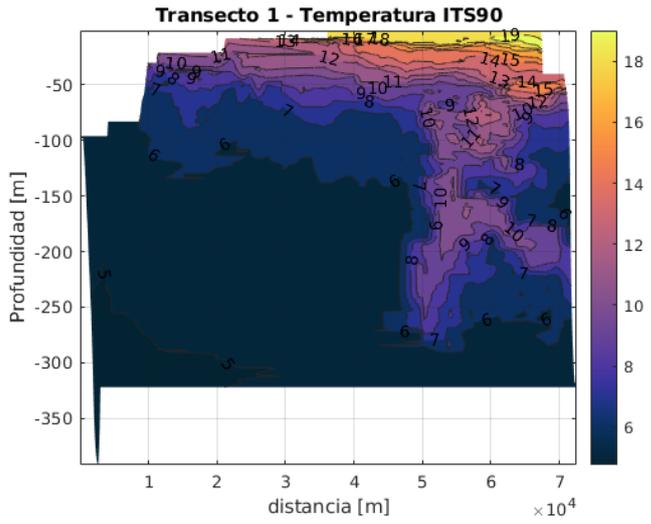
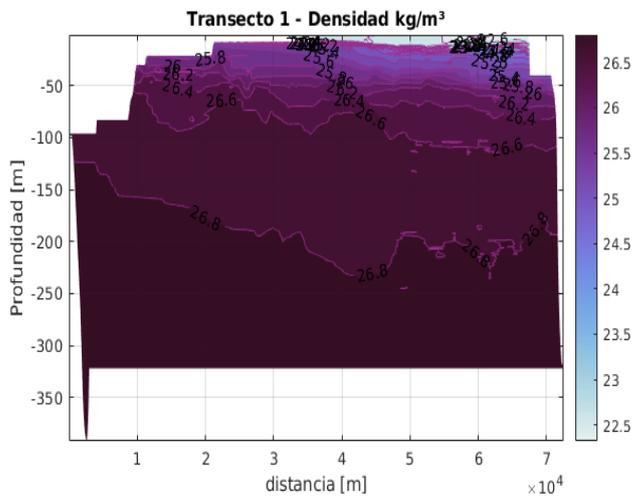
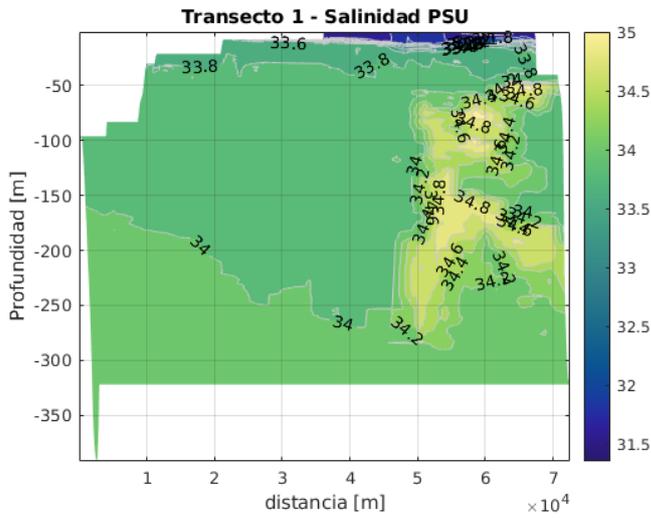


Figure 9: Temperatura, Salinidad y Densidad Transecto 1



Transecto 12

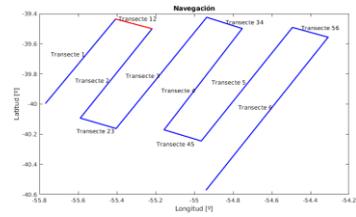
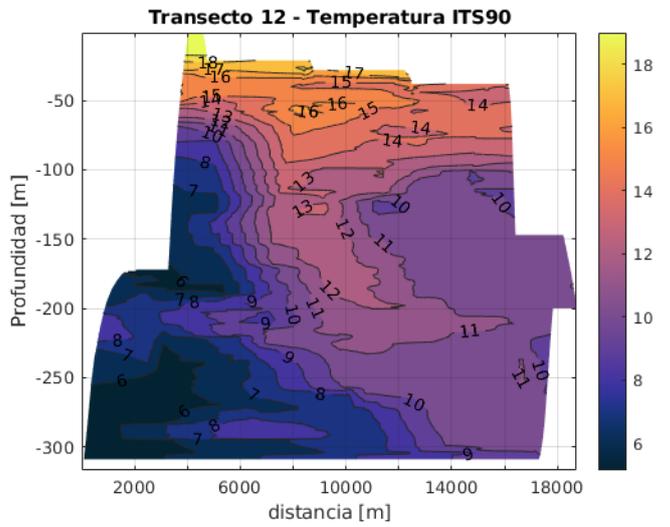
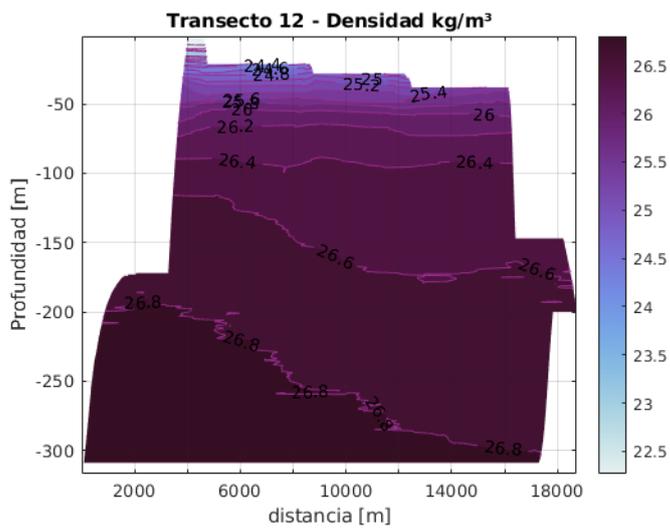
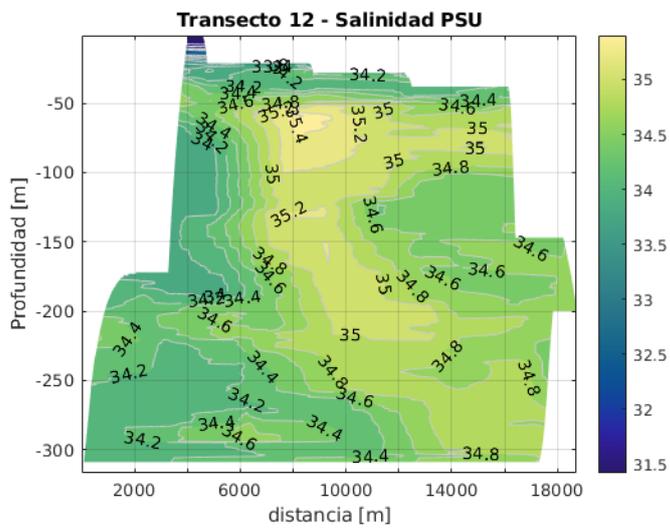


Figure 10: Temperatura, Salinidad y Densidad Transecto 12



Transecto 2

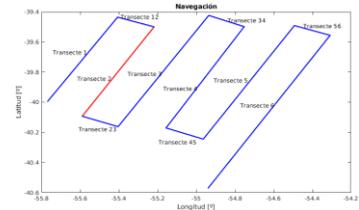
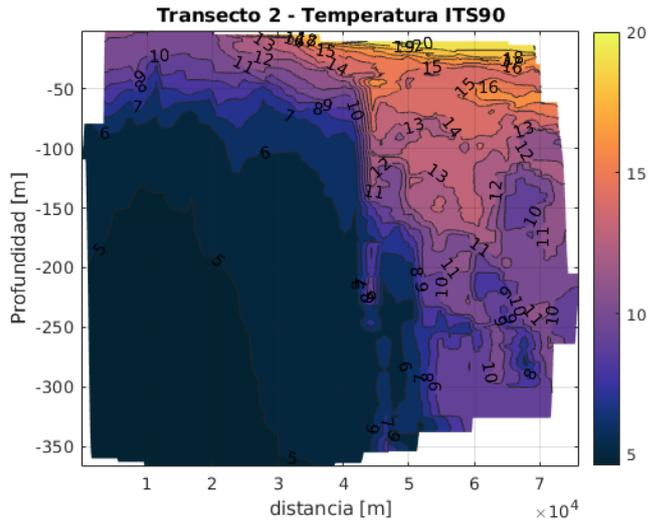
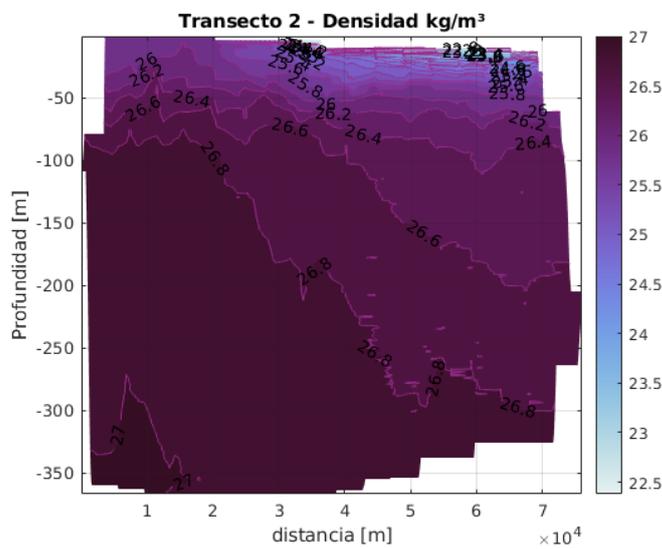
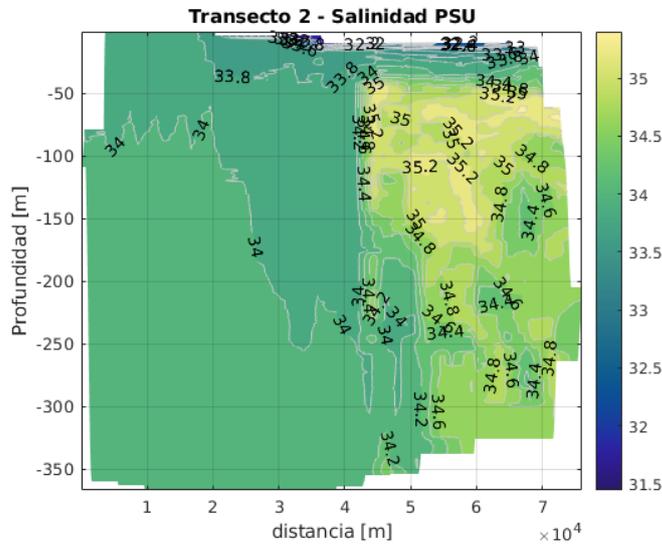


Figure 11: Temperatura, Salinidad y Densidad Transecto 2



Transecto 23

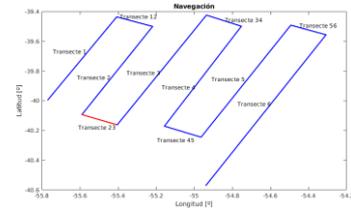
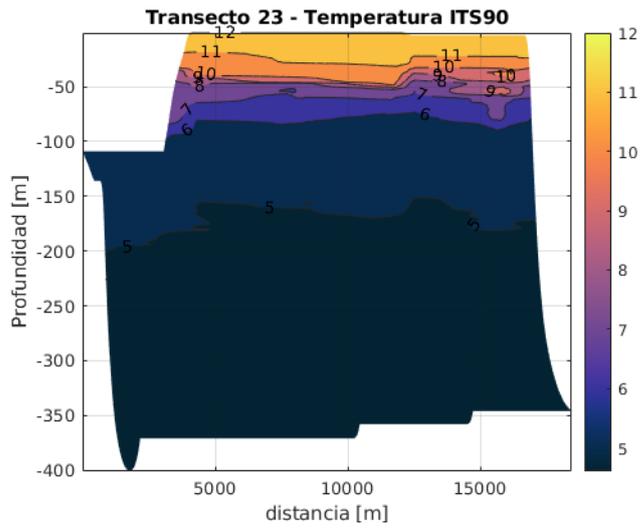
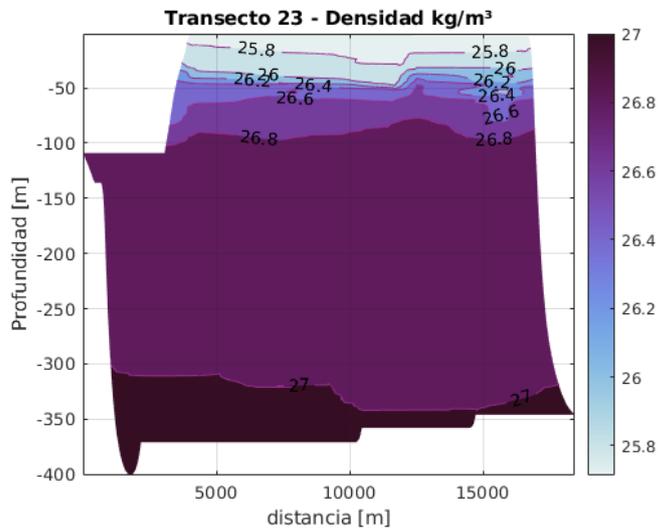
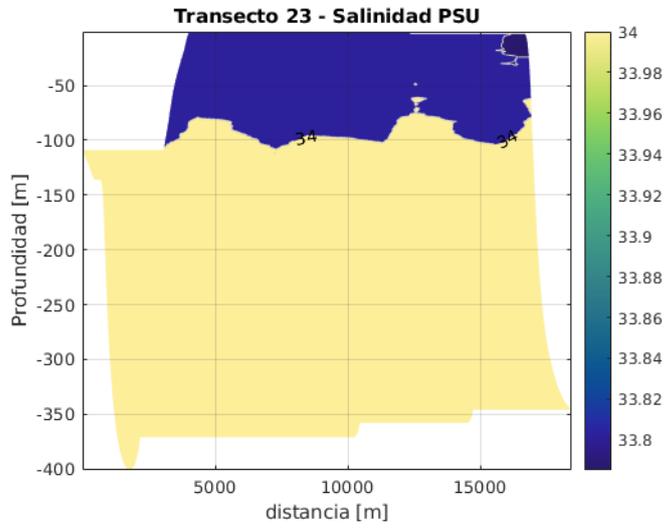


Figure 12: Temperatura, Salinidad y Densidad Transecto 23



Transecto 3

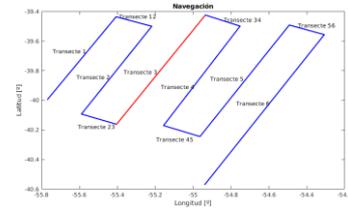
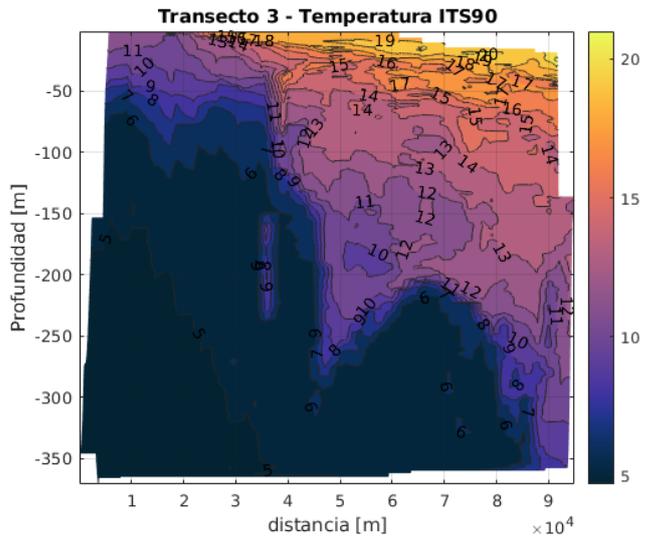
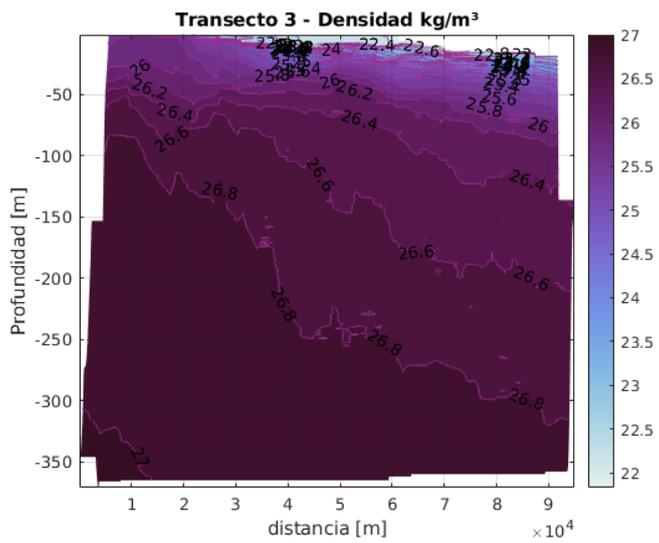
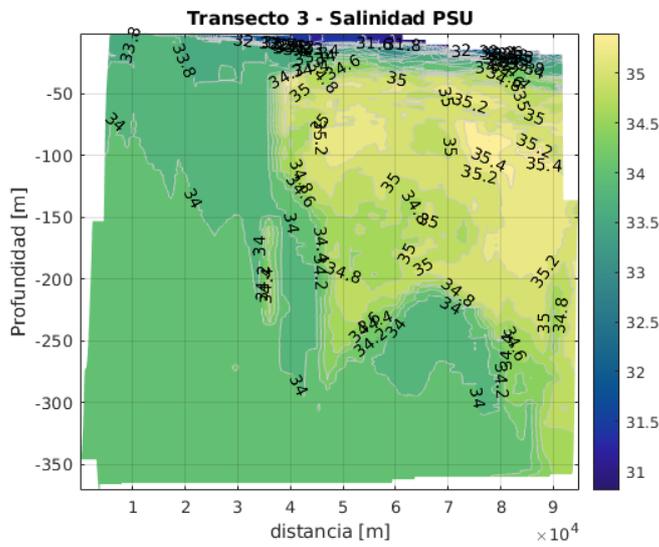


Figure 13: Temperatura, Salinidad y Densidad Transecto 3



Transecto 34

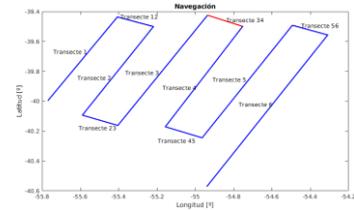
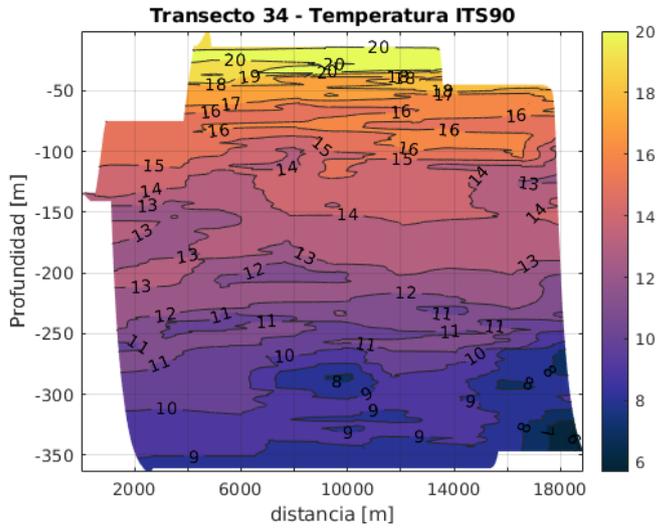
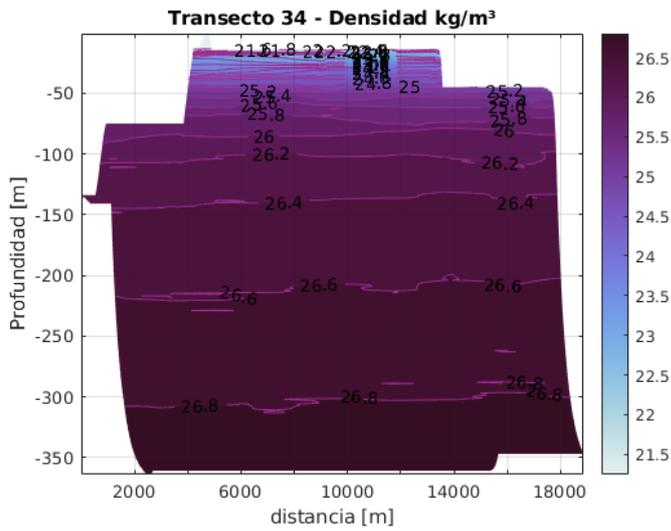
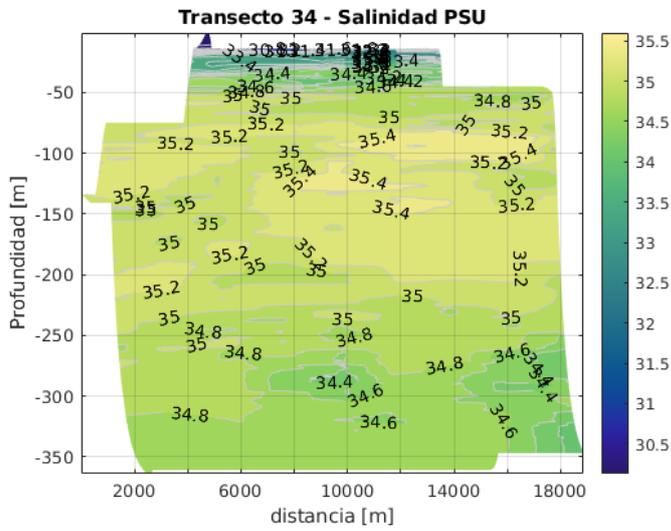


Figure 14: Temperatura, Salinidad y Densidad Transecto 34



Transecto 4

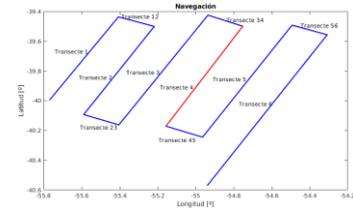
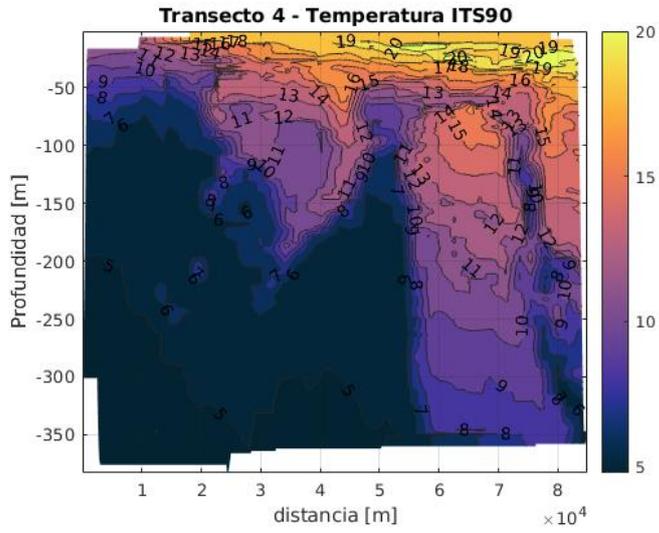
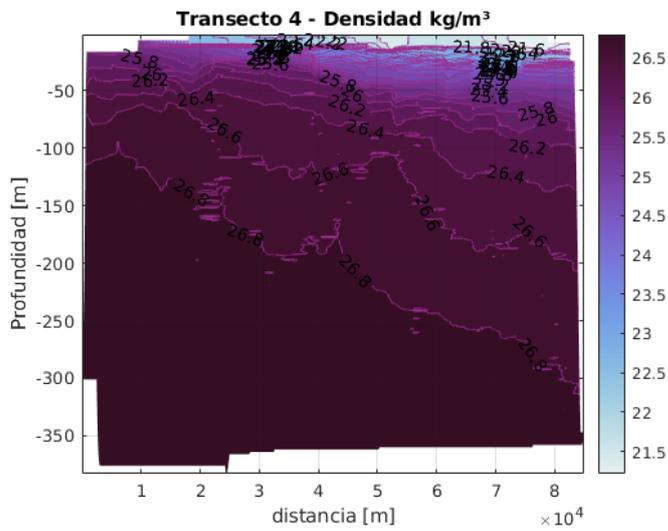
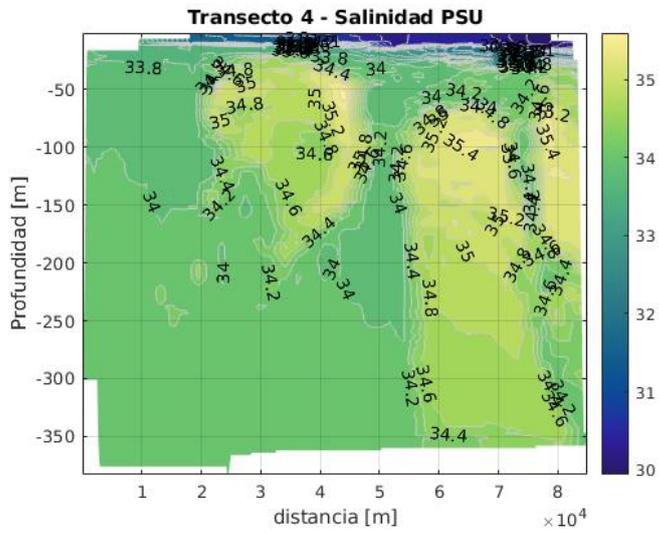


Figure 15: Temperatura, Salinidad y Densidad Transecto 4



Transecto 45

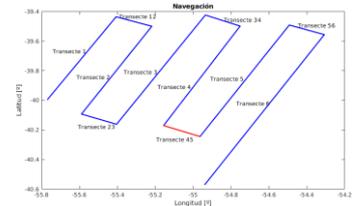
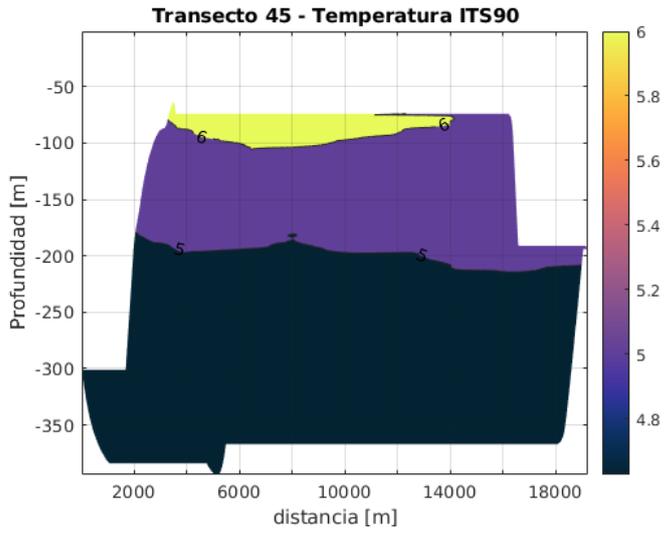
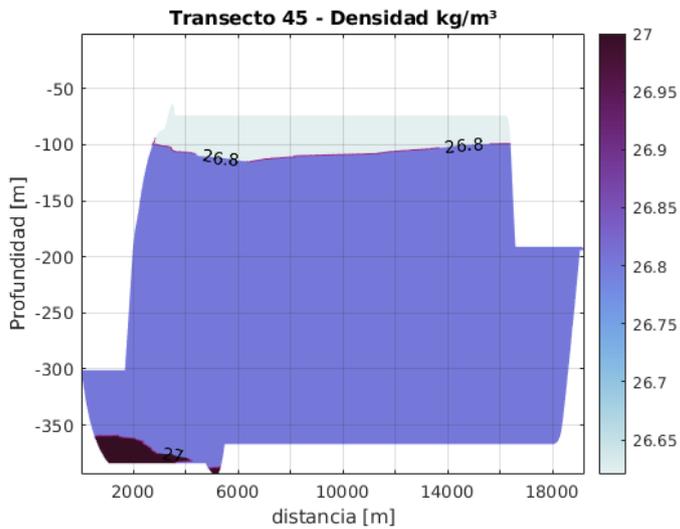
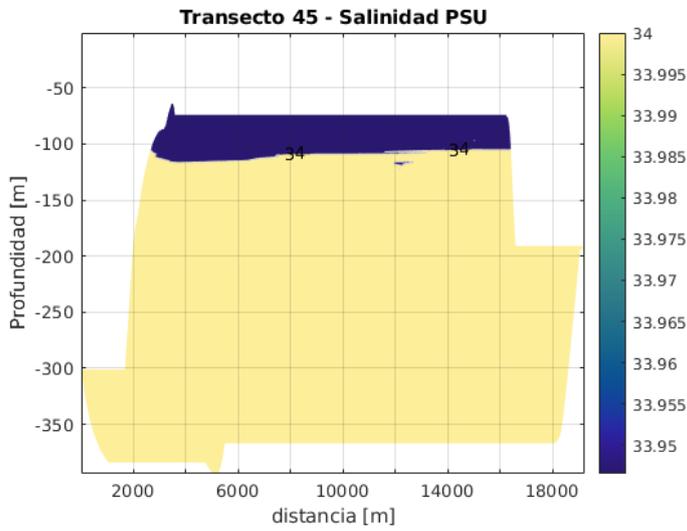


Figure 16: Temperatura, Salinidad y Densidad Transecto 45



Transecto 5

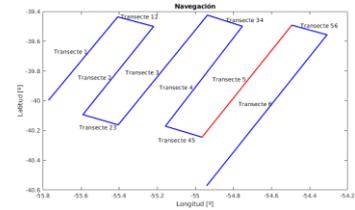
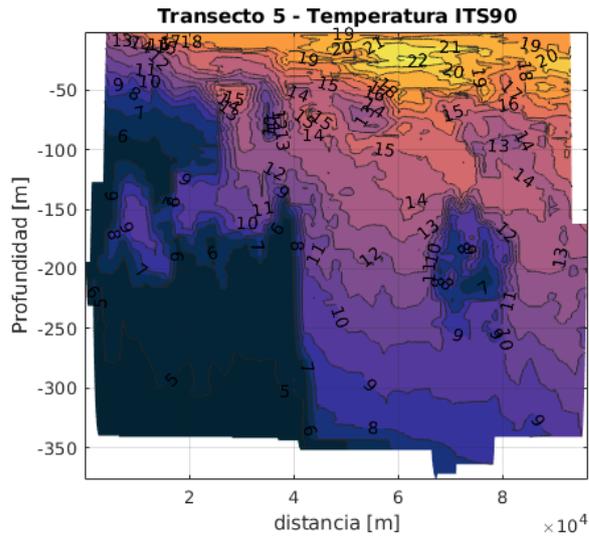
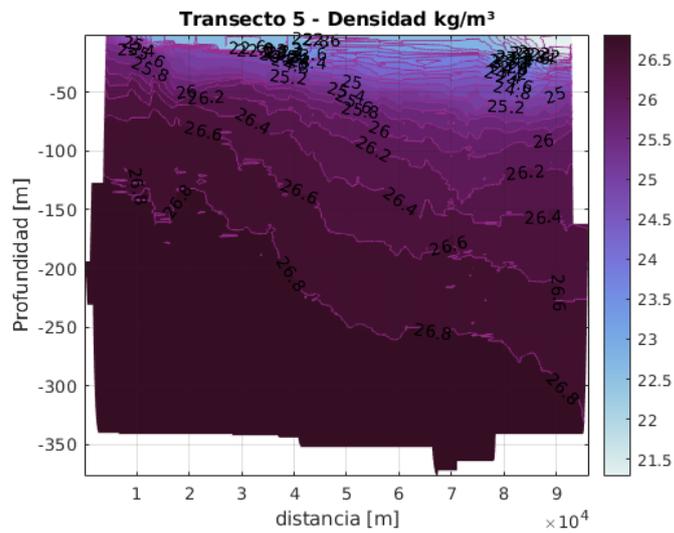
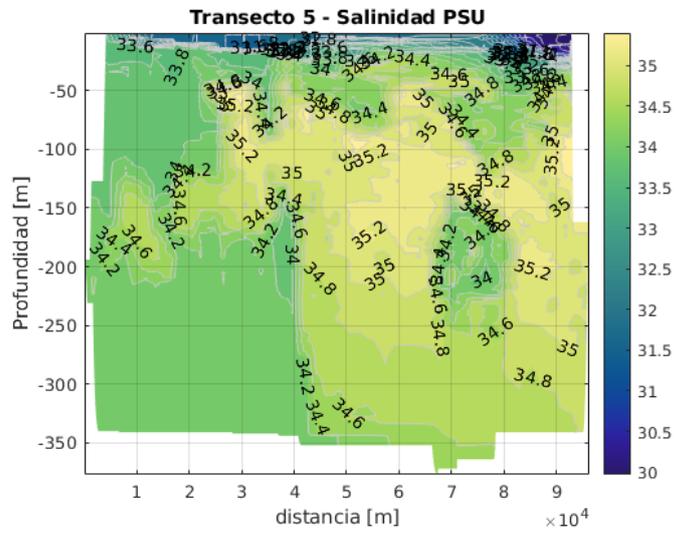


Figure 17: Temperatura, Salinidad y Densidad Transecto 5



Transecto 56

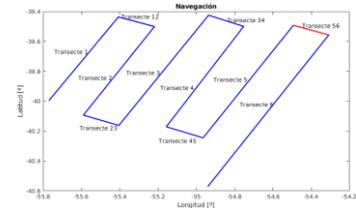
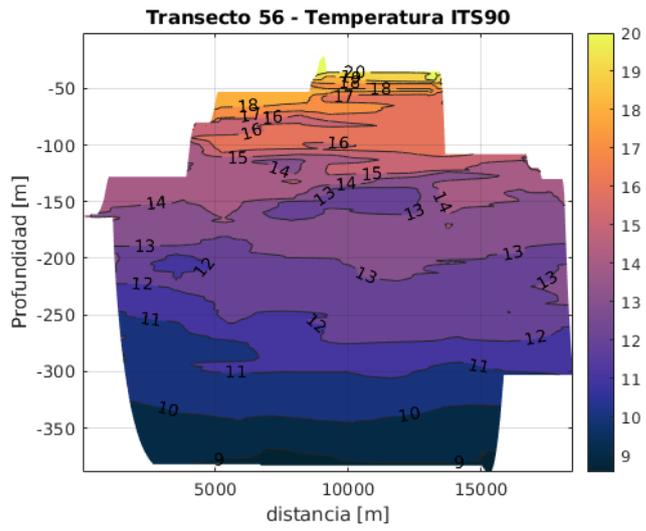
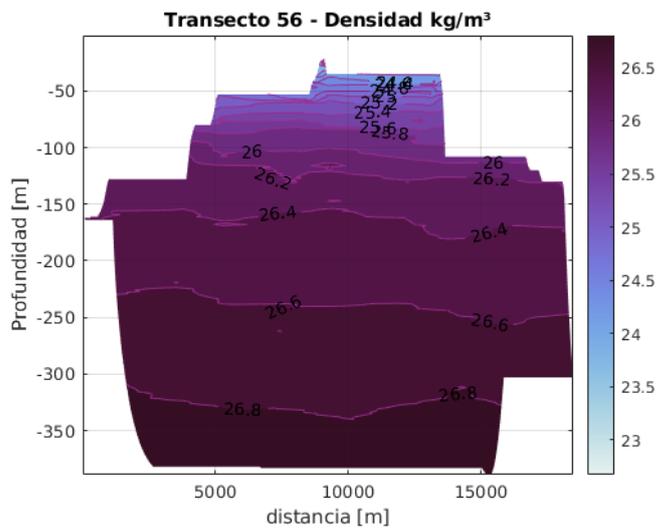
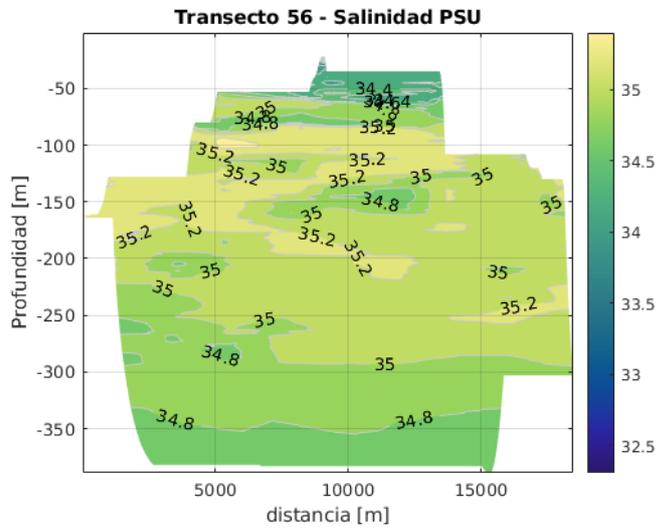


Figure 18: Temperatura, Salinidad y Densidad Transecto 56



Transecto 6

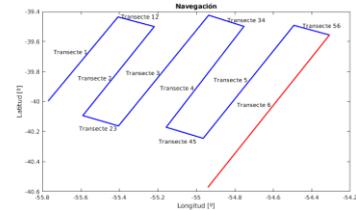
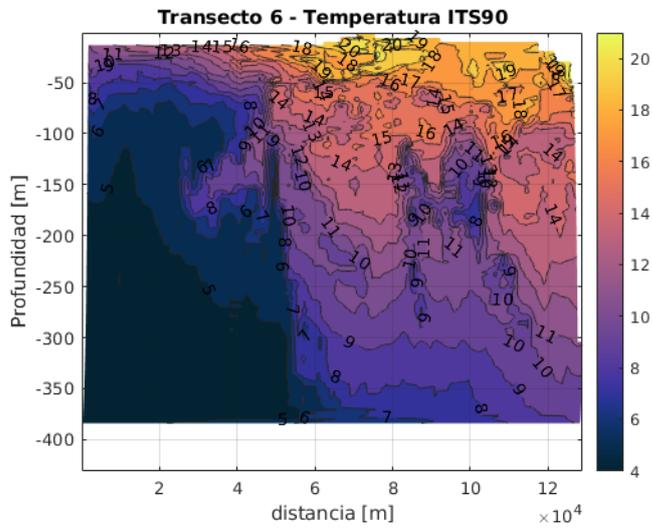
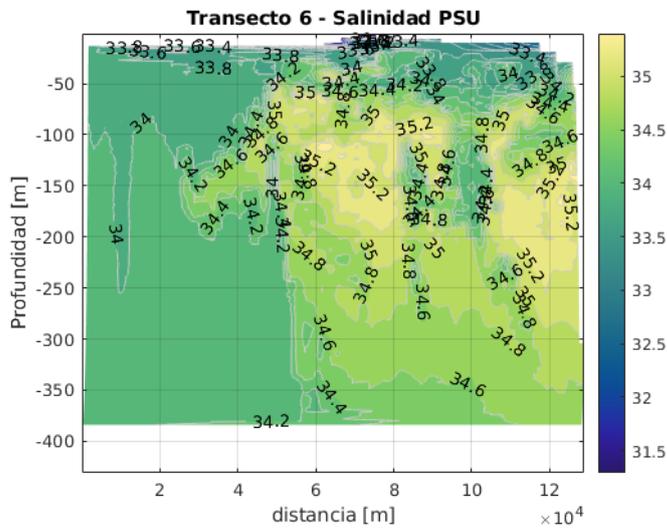


Figure 19: Temperatura, Salinidad y Densidad Transecto 6



3.9. Algunas reflexiones sobre los datos presentados

Como podemos apreciar en los *contours* de los distintos transectos mostrados en la sección anterior, vemos perfectamente la posición y ubicación de la zona frontal oceánica que se caracteriza por los gradientes de temperatura y salinidad elevados, lo que se refleja en el aumento de la densidad espacial de las isolíneas características. Es decir, el gradiente horizontal de temperatura (salinidad) dT/dx visualmente se representa en más isolíneas por la misma distancia.

Vemos caracterizada perfectamente la distribución de la temperatura, salinidad y densidad en todos los transectos y se puede apreciar perfectamente la zona interface entre las dos masas de agua, zona donde se crean unos gradientes de temperatura y salinidad muy grandes en el eje x .

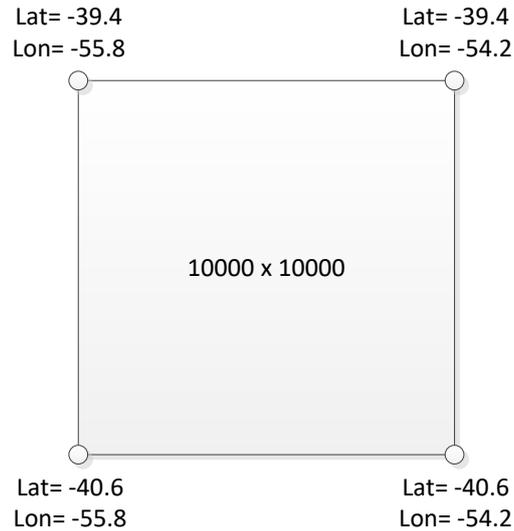
De hecho, la isolínea que mejor define y caracteriza la interficie del frente es la que tiene un gradiente máximo. Siendo así y haciendo el correspondiente cálculo diferencial podríamos hallar la correspondiente isolínea característica.

Pero, como también podemos apreciar en los *contours*, la interface del frente no es "suave" y continua, también está sometida a toda una serie de perturbaciones e intrusiones que dificultarán y mucho el cálculo y el desarrollo de las siguientes secciones.

A continuación, y siguiendo con la reconstrucción del frente, hallaremos las secciones horizontales.

3.10. Las secciones horizontales y definición de la matriz

Para dibujar las secciones horizontales, diseñaremos una matriz, en este caso cuadrada de 10000 x 10000 (filas x columnas) la cual contiene las posiciones Latitud y Longitud (ubicados en el plano horizontal) de cada uno de los valores de las variables muestreadas.



Con esta matriz podemos alcanzar resoluciones de 15 metros en el plano horizontal.

$$\text{distancia} = (\text{Latitud máxima} - \text{Latitud mínima}) \cdot 60 \cdot 1852 = 133344 \text{ metros}$$

$$\text{Resolución} = \frac{133344}{10000} = 13.3 \text{ metros}$$

$$\text{Resolución en grados} = \frac{\text{latitud máxima} - \text{latitud mínima}}{10000} = \frac{1.2}{10000} = 0.00012$$

Esta resolución es de Latitud.

Teniendo en cuenta lo anteriormente expuesto y teniendo en cuenta también que, si cada dato de temperatura, salinidad, densidad y presión está asociado a una latitud y longitud, ahora se van a ubicar cada dato físico a su ubicación correspondiente dentro de la matriz diseñada.

Para ello se han diseñado las siguientes funciones.

3.11. Diseño de la matriz horizontal

Para ejecutar este script deberemos de introducir que profundidad queremos para realizar el *contour* horizontal y definiremos la matriz con la variable a visualizar. Para la variable nos será muy útil las matrices verticales que se diseñaron para las secciones verticales.

La función se llama `design_mh2d.m` y se muestra a continuación.

```

prof=100;
m=10000;
n=10000;
Shli(1:m,1:n)=NaN;
latmax=(round(max(LatAvg3)*10)/10);
latmin=(round(min(LatAvg6)*10)/10);
lonmax=(round(max(LonAvg6)*10)/10);
lonmin=(round(min(LonAvg1)*10)/10);
ncellLat=(latmax-latmin)/m;
ncellLon=(lonmax-lonmin)/n;
veclat(1)=latmin;
veclon(1)=lonmin;
for i=2:m
    veclat(i)=veclat(i-1)+ncellLat;
end
for j=2:n
    veclon(j)=veclon(j-1)+ncellLon;
end
%Para el transecto 1
for i=1:length(Svli1)
    if ~isnan(Svli1(prof,i))
        latitude=LatAvg1(i);
        longitude=LonAvg1(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli1(prof,i);
                        break;
                    end
                end
            end
        end
    end
end
% TRANSECTO 12
for i=1:length(Svli12)
    if ~isnan(Svli12(prof,i))
        latitude=LatAvg12(i);
        longitude=LonAvg12(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli12(prof,i);
                        break;
                    end
                end
            end
        end
    end
end
% TRANSECTO 2
for i=1:length(Svli2)
    if ~isnan(Svli2(prof,i))
        latitude=LatAvg2(i);

```

```

longitude=LonAvg2(i);
for k=1:length(veclat)
    if veclat(k)>latitude
        for l=1:length(veclon)-1
            if veclon(l)>longitude
                Shli(k,l)=Svli2(prof,i);
                break;
            end
        end
        break;
    end
end
end
end
% TRANSECTO 23
for i=1:length(Svli23)
    if ~isnan(Svli23(prof,i))
        latitude=LatAvg23(i);
        longitude=LonAvg23(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli23(prof,i);
                        break;
                    end
                end
                break;
            end
        end
    end
end
% TRANSECTO 3
for i=1:length(Svli3)
    if ~isnan(Svli3(prof,i))
        latitude=LatAvg3(i);
        longitude=LonAvg3(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli3(prof,i);
                        break;
                    end
                end
                break;
            end
        end
    end
end
% TRANSECTO 34
for i=1:length(Svli34)
    if ~isnan(Svli34(prof,i))
        latitude=LatAvg34(i);
        longitude=LonAvg34(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli34(prof,i);
                        break;
                    end
                end
            end
        end
    end
end

```

```

end
end
break;
end
end
end
end
end
% TRANSECTO 4
for i=1:length(Svli4)
    if ~isnan(Svli4(prof,i))
        latitude=LatAvg4(i);
        longitude=LonAvg4(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli4(prof,i);
                        break;
                    end
                end
            end
            break;
        end
    end
end
end
% TRANSECTO 45
for i=1:length(Svli45)
    if ~isnan(Svli45(prof,i))
        latitude=LatAvg45(i);
        longitude=LonAvg45(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli45(prof,i);
                        break;
                    end
                end
            end
            break;
        end
    end
end
end
% TRANSECTO 5
for i=1:length(Svli5)
    if ~isnan(Svli5(prof,i))
        latitude=LatAvg5(i);
        longitude=LonAvg5(i);
        for k=1:length(veclat)
            if veclat(k)>latitude
                for l=1:length(veclon)-1
                    if veclon(l)>longitude
                        Shli(k,l)=Svli5(prof,i);
                        break;
                    end
                end
            end
            break;
        end
    end
end
end
end
end

```

```

for j=1:length(veclat)
    s=[]; Sinterp=[]; x=[]; xq=[];
    k=0; ini=0;
    for i=1:length(veclon)
        if isnan(Shli(j,i))
            if ini==1
                k=k+1;
                xq(k+1)=veclon(i);
            end
        else
            if k~=0
                s(2)=Shli(j,i);
                x(2)=veclon(i);
                xq(k+2)=x(2);
                Sinterp=interp1(x,s,xq);
                for n=1:length(Sinterp)
                    Shli(j,i-length(Sinterp)+n)=Sinterp(n);
                end
                k=0; ini=1; xq=[];
                s(1)=s(2); x(1)=x(2); xq(1)=x(2);
            else
                s(1)=Shli(j,i);
                x(1)=veclon(i);
                xq(1)=x(1);
                ini=1;
            end
        end
    end
end
end

```

3.13. Visualización de las distribuciones horizontales

Para la representación gráfica corremos el siguiente script que lanza en contour, esta vez encima de una proyección Mercator para una representación más real del frente en su proyección horizontal.

Lanzamiento del contour

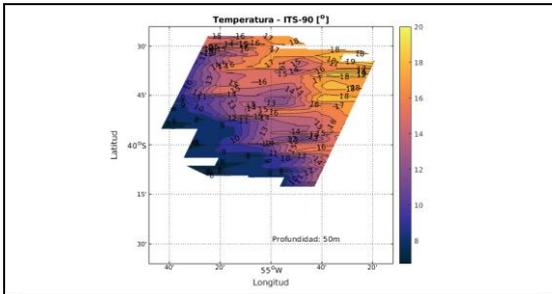
```

figure
m_proj('Mercator','long',[-55.8 -54.2],'lat',[-40.6 -39.4]);
m_contour(veclon,veclat,Thli);
m_grid;

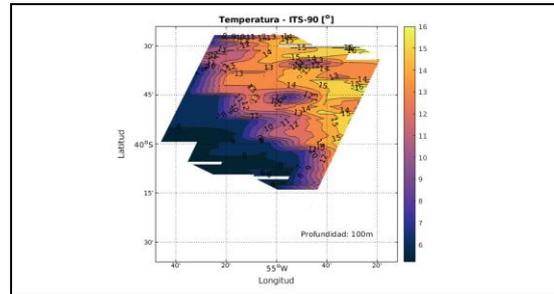
```

TEMPERATURA

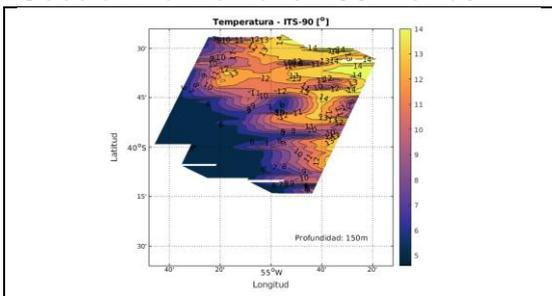
Sección horizontal a 50 metros



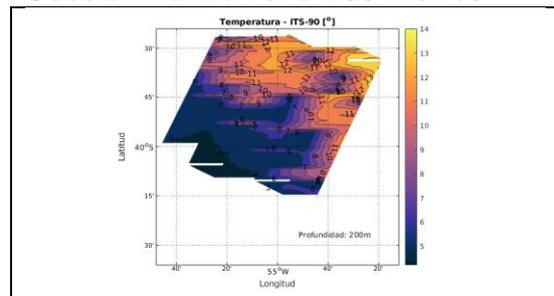
Sección horizontal a 100 metros



Sección horizontal a 150 metros



Sección horizontal a 200 metros



Sección horizontal a 250 metros

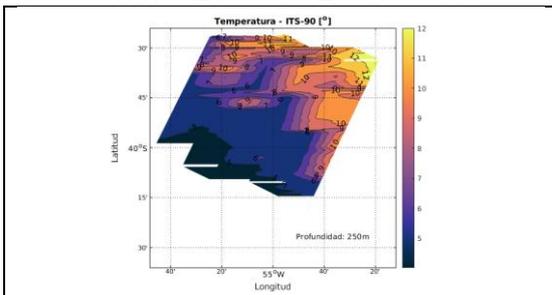


Figure 20: Distribución temperatura horizontal

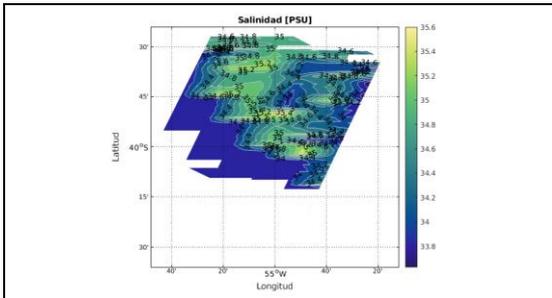
Se presentan los *contours* de las isotermas de las secciones horizontales de 50, 100, 150, 200 y 250 metros. Destacar que podríamos realizar cualquier *contour* a la profundidad deseada. Ahora se han escogido estas profundidades para observar el resultado de la ejecución de los scripts creados.

Presentación de los resultados que muestran, efectivamente la interface horizontal del frente observando que, en zonas de más al sur, la temperatura del agua es mucho más fría que la del norte. Ya se pueden ver las dos corrientes de estudio, Brasil y Malvinas.

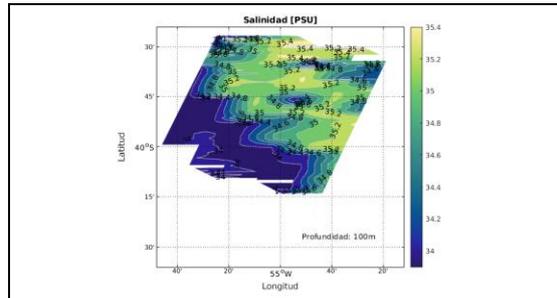
Como dos notas negativas de los *plots*, los extremos inferior y superior que no terminan de "pintarse" adecuadamente. Habría que repasar el código para ver qué ocurre en esas zonas, pero seguramente que como son esquinas de los transectos, pues faltan datos. Hay que mejorar la interpolación.

SALINIDAD

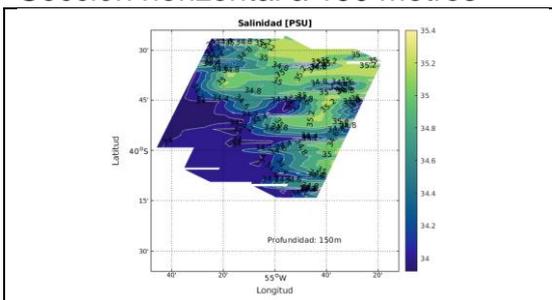
Sección horizontal a 50 metros



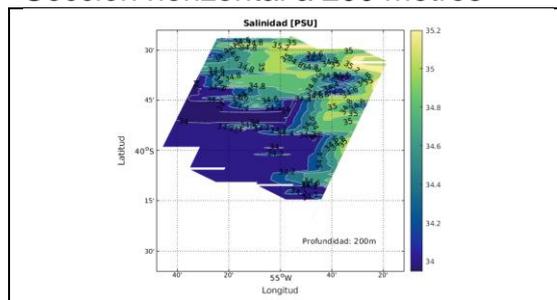
Sección horizontal a 100 metros



Sección horizontal a 150 metros



Sección horizontal a 200 metros



Sección horizontal a 250 metros

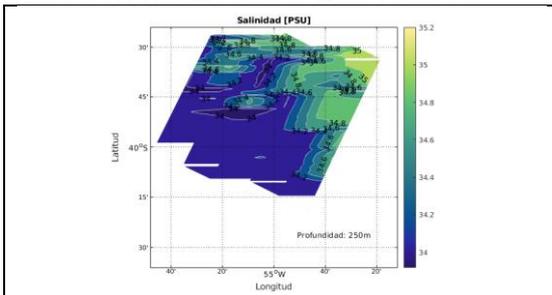


Figure 21: Distribución salinidad horizontal

También, las secciones horizontales de la salinidad se observa claramente la interface del frente. Vemos gradientes máximos que dividen las dos masas de agua.

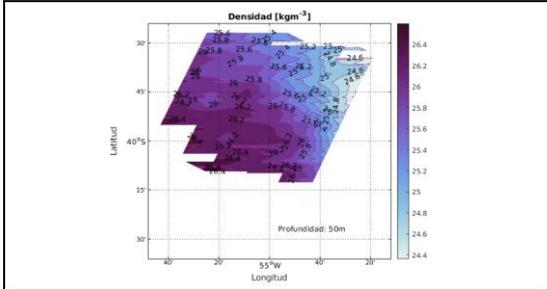
Notemos que cuando ganamos en profundidad, la pared de la zona frontal se desplaza hacia el norte.

Vemos también las partes superior e inferior de los *contours* que no terminan de cerrar bien la caja. Como hemos dicho antes con la temperatura, efectivamente faltan datos en los extremos que dificultan la interpolación y la representación de datos. Hay que mejorar el código.

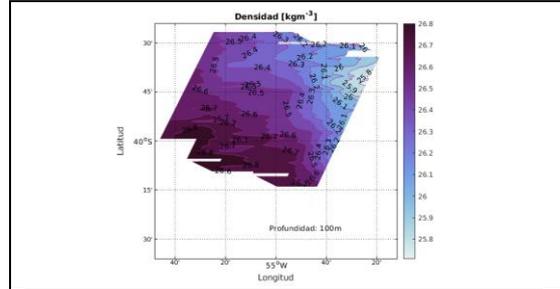
Notemos que el agua más salada se encuentra al norte y el agua menos salada está en el sur.

DENSIDAD

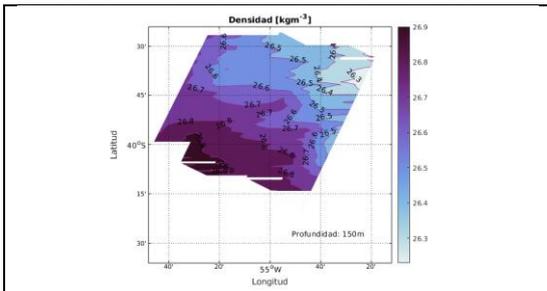
Sección horizontal a 50 metros



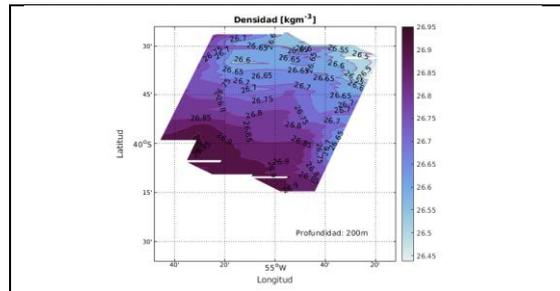
Sección horizontal a 100 metros



Sección horizontal a 150 metros



Sección horizontal a 200 metros



Sección horizontal a 250 metros

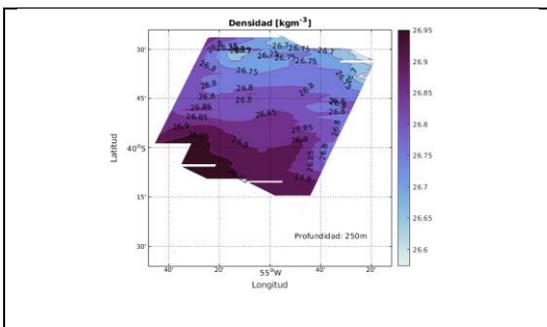


Figure 22: Distribución densidad horizontal

Y en cuanto a la densidad, se reproducen los mismos defectos en los *plots* que en el caso de las temperaturas y salinidades. Hay que resolver.

El comportamiento de la densidad es distinto al comportamiento de la temperatura y la salinidad. Sin entrar a valorar más detalles, podemos ver que las isólicas, en este caso, isopicnas son más “suaves” y tienen un comportamiento más lineal y parece que no están tan sometidas a las intrusiones observadas en los perfiles de temperatura y salinidad.

Este efecto se observa tanto en los perfiles verticales anteriores como en los horizontales.

Vemos también que el agua más densa es del sur y el agua menos densa es del norte.

4. Análisis del intercambio transfrontal

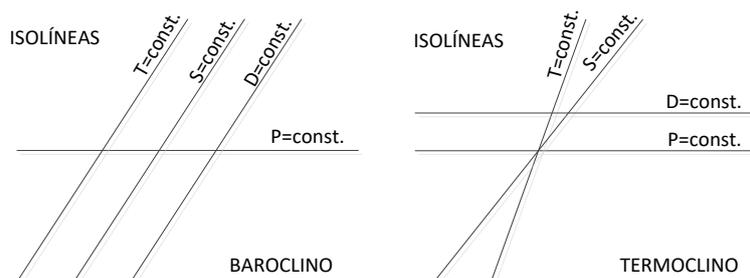
Este capítulo es especial y aunque parezca que no tenga mucho sentido en el contexto del Máster en Ingeniería Computacional y Matemática, cobra más sentido cuando se desarrolla y analiza su teoría.

K. N. Fedorov realizó un análisis teórico basado en la geometría y posición de las isolíneas y las relaciones entre las inclinaciones de isopícnas (γ_ρ), isoterma (γ_T) e isohalinas (γ_S) para hallar un número y catalogar así los frentes y poder compararlos en función de su intercambio transfrontal.

Describo brevemente que las inclinaciones de las isolíneas proporcionan los grados de termoclinicidad y baroclinicidad del frente los cuales nos indican su capacidad de unir o separar las distintas masas de agua a ambos lados de la interface frontal.

En este sentido, y una vez realizada la reconstrucción de la zona frontal y visto en detalle el frente, podemos realizar el análisis planteado. Ya sabemos que aspecto tiene, pero... ¿cómo funciona la confluencia Brasil-Malvinas desde el punto de vista del intercambio transfrontal?

La teoría sugiere que un frente será baroclino puro cuando las isolíneas de temperatura, salinidad y densidad son paralelas entre si y se cruzan con la isobara. Y un comportamiento termoclinico puro se desarrolla cuando las isopícnas y las isobaras son paralelas. Gráficamente tenemos:



4.1. Isolíneas representativas del frente

En este caso y una vez descrito el frente, se trata de localizar su isolínea representativa. Es decir, en cuanto al perfil de temperaturas, será la isoterma que caracteriza el frente.

¿Y, cual isoterma puede representar el frente? Serán las isotermas cuyo gradiente sea máximo, es decir, isolíneas muy juntas entre sí. La superficie que genera la isoterma será la que separa las dos masas de agua.

En este caso hallaremos la isoterma derivando la temperatura respecto el eje x (distancia) en todos los puntos de la sección vertical.

Así:

$$isoterma = \left(\frac{\partial T}{\partial x} \right)_{max}; \quad isohalina = \left(\frac{\partial S}{\partial x} \right)_{max}; \quad isopicna = \left(\frac{\partial \rho}{\partial x} \right)_{max}$$

Teniendo en cuenta una P constante.

Para ello se ha realizado la siguiente función que determina el gradiente máximo y calcula un polinomio de primer grado y lo sitúa en el perfil vertical de gradientes de temperatura. Con la línea, podemos calcular los ángulos respecto las isobaras y con respecto a las isohalinas e isopicnas.

iso_rep.m

```

o=0; p=0; q=0;
for i=1:length(Yvec)
    for j=1:length(Xvec)-1
        t=(Tvli_avgd(i,j+1)-Tvli_avgd(i,j))/(DistAvg(j+1)-DistAvg(j));
        s=(Svli(i,j+1)-Svli(i,j))/(DistAvg(j+1)-DistAvg(j));
        r=(Rvli(i,j+1)-Rvli(i,j))/(DistAvg(j+1)-DistAvg(j));

        if (t<0.0022) && (t>-0.0022)
            dTdx(i,j)=t;
        else
            dTdx(i,j)=NaN;
        end
        dTdx(i,j+1)=dTdx(i,j);

        if (s<0.00035) && (s>-0.00035)
            dSdx(i,j)=s;
        else
            dSdx(i,j)=NaN;
        end
        dSdx(i,j+1)=dSdx(i,j);

        if (r<0.00005) && (r>-0.00005)
            dRdx(i,j)=r;
        else
            dRdx(i,j)=NaN;
        end
        dRdx(i,j+1)=dRdx(i,j);
    end
end

for i=1:(length(Yvec)/20)-1
    for j=1:length(Xvec)

        t=(Tvli_avg(i+1,j)-Tvli_avg(i,j))/20;
        s=(Svli(i+1,j)-Svli(i,j));
        r=(Rvli(i+1,j)-Rvli(i,j));
    end
end

```

```

    if (t<0.0022) && (t>-0.0022)
        dTdy(i,j)=t;
    else
        dTdy(i,j)=NaN;
    end
    dTdy(i,j+1)=dTdy(i,j);

    if (s<0.00035) && (s>-0.00035)
        dSdx(i,j)=s;
    else
        dSdx(i,j)=NaN;
    end
    dSdx(i,j+1)=dSdx(i,j);

    if (r<0.00005) && (r>-0.00005)
        dRdx(i,j)=r;
    else
        dRdx(i,j)=NaN;
    end
    dRdx(i,j+1)=dRdx(i,j);
end

end
dTdy(i+1,:)=dTdy(i,j);

PressAvgn=-PressAvg;
for i=1:length(Yvec)
    for j=1:length(Xvec)-1

        u=dTdx(i,j);
        if ((u<0.00225) && (u>0.00218)) %|| ((u>-0.002) && (u<-
0.001980))
            o=o+1;
            PTx(o)=DistAvg(j);
            PTy(o)=PressAvgn(j);
            PTt(o)=TempAvg(j);
        end

        v=dSdx(i,j);
        if ((v<0.0003) && (v>0.0002975)) %|| ((v>-0.0003) && (v<-
0.00029925))
            p=p+1;
            PSx(p)=DistAvg(j);
            PSy(p)=PressAvgn(j);
            PSs(p)=SaltAvg(j);
        end

        w=dRdx(i,j);
        if ((w<0.00005) && (w>0.0000485))
            q=q+1;
            PRx(q)=DistAvg(j);
            PRy(q)=PressAvgn(j);
            PRr(q)=SigmaAvg(j);
        end

    end
end
end
%Cálculo de la isoterma representativa

```

```

PT=polyfit (PTx, PTy, 1);
PTqx=1:1000:DistAvg (length (DistAvg));
PTqy=polyval (PT, PTqx);
tiso=mean (PTt);

```

%Cálculo de la isohalina representativa

```

PS=polyfit (PSx, PSy, 1);
PSqx=1:1000:DistAvg (length (DistAvg));
PSqy=polyval (PS, PSqx);
siso=mean (PSs);

```

%Cálculo de la ISOPICNA representativa

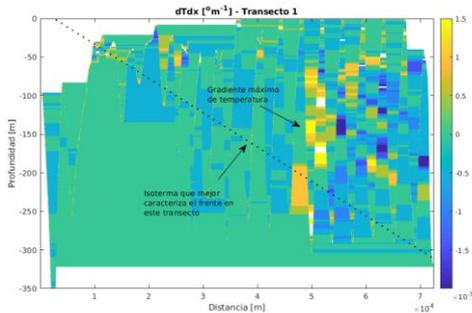
```

PR=polyfit (PRx, PRy, 1);
PRqx=1:1000:DistAvg (length (DistAvg));
PRqy=polyval (PR, PRqx);
riso=mean (PRr);

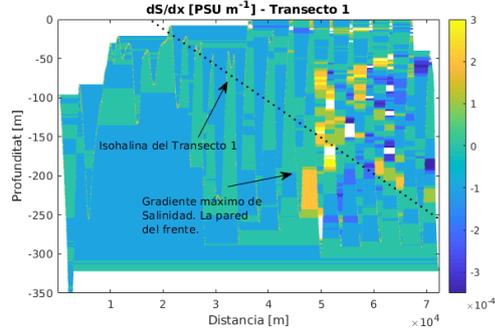
```

Y como resultado obtenemos:

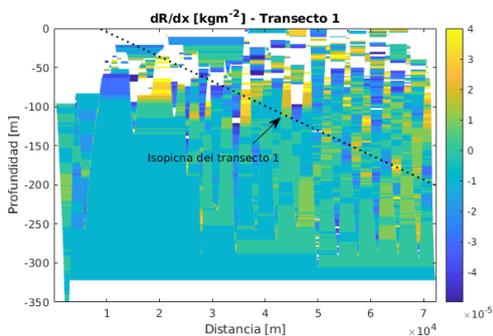
Gradientes de temperatura en transecto 1



Gradientes de salinidad en transecto 1



Gradientes de densidad en transecto 1



Inclinaciones de las isolíneas

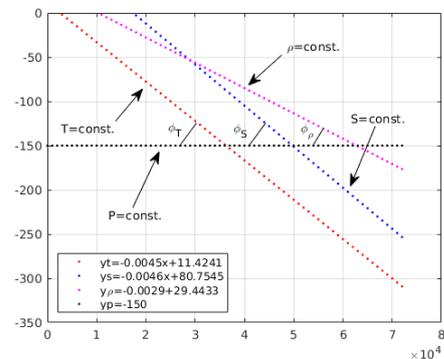


Figure 23: Gradientes de temperatura transecto 1

Y podemos ver que el comportamiento del frente es, en términos generales, mixto con cierto grado de termoclinicidad y baroclinicidad sin ser puros y las inclinaciones de las isotermas se calcularían mediante:

$$\gamma_T = \tan(\varphi_T); \quad \gamma_S = \tan(\varphi_S); \quad \gamma_\rho = \tan(\varphi_\rho)$$

4.2. Relación entre las inclinaciones de isopícnas (γ_ρ), isotermas (γ_T) e isohalinas (γ_S)

Otra manera de hallar los grados de baroclinicidad y termoclinicidad es buscando las inclinaciones γ_T , γ_S y γ_ρ con:

$$\gamma_T = \frac{\partial T / \partial x}{\partial T / \partial z}; \quad \gamma_S = \frac{\partial S / \partial x}{\partial S / \partial z}; \quad \gamma_\rho = \frac{\partial \rho / \partial x}{\partial \rho / \partial z} \quad (1)$$

Y usando la ecuación linealizada del estado del agua de mar para $P=cte$:

$$\rho = \rho_0 (1 - \alpha \Delta T + \beta \Delta S)$$

Podemos obtener:

$$\gamma_\rho = \frac{\beta \partial S / \partial x - \alpha \partial T / \partial x}{\beta \partial S / \partial z - \alpha \partial T / \partial z} = \frac{\gamma_T (R\rho - \gamma_S / \gamma_T)}{R\rho - 1} = \frac{\gamma_S (R\rho \gamma_T / \gamma_S - 1)}{R\rho - 1} \quad (2)$$

Y de estas expresiones se obtienen:

$$\alpha = -\frac{1}{\rho} \frac{\partial \rho}{\partial T}; \quad \beta = \frac{1}{\rho} \frac{\partial \rho}{\partial S}; \quad R\rho = \frac{\alpha \partial T / \partial z}{\beta \partial S / \partial z} \quad (3)$$

La relación γ_S / γ_T representa la característica de la proximidad de la situación frontal al tipo baroclino o termoclino. Podemos ver que el frente es puramente termoclino cuando:

$$\gamma_S / \gamma_T = R\rho = \frac{\alpha \partial T / \partial z}{\beta \partial S / \partial z} \quad (4)$$

Y será puramente baroclino cuando:

$$\gamma_S / \gamma_T = 1 \quad \text{o} \quad \gamma_S = \gamma_T \quad (5)$$

Cuando:

$$R\rho = \frac{\alpha \partial T / \partial z}{\beta \partial S / \partial z} < \gamma_S / \gamma_T < 1 \quad (6)$$

El frente tendrá un comportamiento mixto, baroclino-termoclino. Si realizamos un análisis a pequeña escala (como esa era la intención) veríamos que en determinadas zonas del transecto, el frente se comportaría de distinta manera, facilitando el intercambio transfrente en unas zonas y dificultando o impidiendo el intercambio en otras zonas.

Así, nos proponemos realizar una función completa que en primer lugar desarrolle el cómputo diferencial para las variables de temperatura, salinidad y densidad siguiendo las indicaciones de (1). En este sentido, se calculan las $\partial T/\partial x, \partial T/\partial z; \partial S/\partial x, \partial S/\partial z; \partial \rho/\partial x, \partial \rho/\partial z$.

No sólo eso, una vez realizado el cómputo diferencial, calculamos las inclinaciones siendo estas: $\gamma T = \partial T/\partial x/\partial T/\partial z$ y $\gamma S = \partial S/\partial x/\partial S/\partial z$.

De estas expresiones ya se puede extraer la primera conclusión: si $\gamma T = \gamma S$, la zona se comportará de un modo baroclino puro dificultando la transferencia transfrontal. Si nos fijamos, el hecho que $\gamma S/\gamma T = 1$, significa que las isolinneas son paralelas tal y como se anunció al principio de esta sección.

La función también calcula $R\rho$ para encontrar el grado de termoclinicidad puro o mixto.

La función que desarrolla la teoría aquí expuesta es:

```

t=0; s=0; r=0; index=0; p=0; kakona=0;
press_avg=20;
dist_avg=100;
for i=1:length(Xvec) %columnas, distancia
    t=0; s=0; r=0; a=0; b=0; index=0; p=0;
    for j=1:length(Yvec) %filas, presiÃ³n
        p=p+1;
        if p==press_avg
            index=index+1;
            t=t+Tvli(j,i);
            s=s+Svli(j,i);
            r=r+Rvli(j,i);
            a=a+Avli(j,i);
            b=b+Bvli(j,i);
            Tvli_avg(index,i)=t/press_avg;
            Svli_avg(index,i)=s/press_avg;
            Rvli_avg(index,i)=r/press_avg;
            Avli_avg(index,i)=a/press_avg;
            Bvli_avg(index,i)=b/press_avg;
            Press_gscale(index)=index*press_avg;
            t=0; s=0; r=0; a=0; b=0; p=0;
        else
            t=t+Tvli(j,i);
            s=s+Svli(j,i);
            r=r+Rvli(j,i);
            a=a+Avli(j,i);
            b=b+Bvli(j,i);
        end
    end
end
%una vez promediado en presion. vamos a suavizar la distancia
for i=1:length(Yvec)/press_avg %filas, presion
    t=0; s=0; r=0; a=0; b=0; index=0; d=0;
    for j=1:length(Xvec) %columnas, distancia
        d=d+1;
        if d==dist_avg

```

```

        index=index+1;
        t=t+Tvli_avg(i,j);
        s=s+Svli_avg(i,j);
        r=r+Rvli_avg(i,j);
        a=a+Avli_avg(i,j);
        b=b+Bvli_avg(i,j);
        Tvli_gscale(i,index)=t/dist_avg;
        Svli_gscale(i,index)=s/dist_avg;
        Rvli_gscale(i,index)=r/dist_avg;
        Avli_gscale(i,index)=a/dist_avg;
        Bvli_gscale(i,index)=b/dist_avg;
        Dist_gscale(index)=DistAvg(j);
        t=0; s=0; r=0; a=0; b=0; d=0;
    else
        t=t+Tvli_avg(i,j);
        s=s+Svli_avg(i,j);
        r=r+Rvli_avg(i,j);
        a=a+Avli_avg(i,j);
        b=b+Bvli_avg(i,j);
    end
end
end

%-----
--
%realizamos el cómputo de las diferenciales respecto x y respecto y de
%la temperatura: dT/dx y dT/dy y de la salinidad: dS/dx y dS/dy
for i=1:length(Press_gscale)
    for j=1:length(Dist_gscale)-1
        dTdx_gscale(i,j)=(Tvli_gscale(i,j+1)-
Tvli_gscale(i,j))/(Dist_gscale(j+1)-Dist_gscale(j));
        dSdx_gscale(i,j)=(Svli_gscale(i,j+1)-
Svli_gscale(i,j))/(Dist_gscale(j+1)-Dist_gscale(j));
        dDdx_gscale(i,j)=(Rvli_gscale(i,j+1)-
Rvli_gscale(i,j))/(Dist_gscale(j+1)-Dist_gscale(j));
        end
        dTdx_gscale(i,j+1)=dTdx_gscale(i,j);
        dSdx_gscale(i,j+1)=dSdx_gscale(i,j);
        dDdx_gscale(i,j+1)=dDdx_gscale(i,j);
    end

for i=1:(length(Yvec)/press_avg)-1
    for j=1:length(Xvec)/dist_avg
        dTdy_gscale(i,j)=(Tvli_gscale(i+1,j)-
Tvli_gscale(i,j))/press_avg;
        dSdy_gscale(i,j)=(Svli_gscale(i+1,j)-
Svli_gscale(i,j))/press_avg;
        dDdy_gscale(i,j)=(Rvli_gscale(i+1,j)-
Rvli_gscale(i,j))/press_avg;
        end
    end
    dTdy_gscale(i+1,:)=dTdy_gscale(i,j);
    dSdy_gscale(i+1,:)=dSdy_gscale(i,j);
    dDdy_gscale(i+1,:)=dDdy_gscale(i,j);
%-----
%Una vez realizado el cálculo diferencial a gran escala, es momento
de:
%1. Hallar las gammaT y gammaS
%2. Hallar las Rho

for i=1:length(Press_gscale)

```

```

    for j=1:length(Dist_gscale)
        gammaT(i,j)=dTdx_gscale(i,j)/dTdy_gscale(i,j);
        gammaS(i,j)=dSdx_gscale(i,j)/dSdy_gscale(i,j);
        gammaD(i,j)=dDdx_gscale(i,j)/dDdy_gscale(i,j);
        if gammaS(i,j)/gammaT(i,j)>0.95 &&
gammaS(i,j)/gammaT(i,j)<1.05
            kakona=kakona+1;
            Xgamma(kakona)=Dist_gscale(j);
            Ygamma(kakona)=-Press_gscale(i);
            Zgamma(kakona)=gammaS(i,j)/gammaT(i,j);
        end
    end
end
%-----
%Y ahora, realizado el c mputo de la Rho, alpha, beta y las gammas
%correspondientes a gammaT y gammaS, podemos buscar sus relaciones y
marcar
%en los transectos verticales el comportamiento del frente en funci n
de su
%baroclinicidad y termoclinicidad.
[Rho,alpha,beta]=gsw_rho_alpha_beta(Svli_gscale,Tvli_gscale,Y_gscale);

for i=1:length(Press_gscale)
    for j=1:length(Dist_gscale)
        rho(i,j)=gammaS(i,j)/gammaT(i,j);

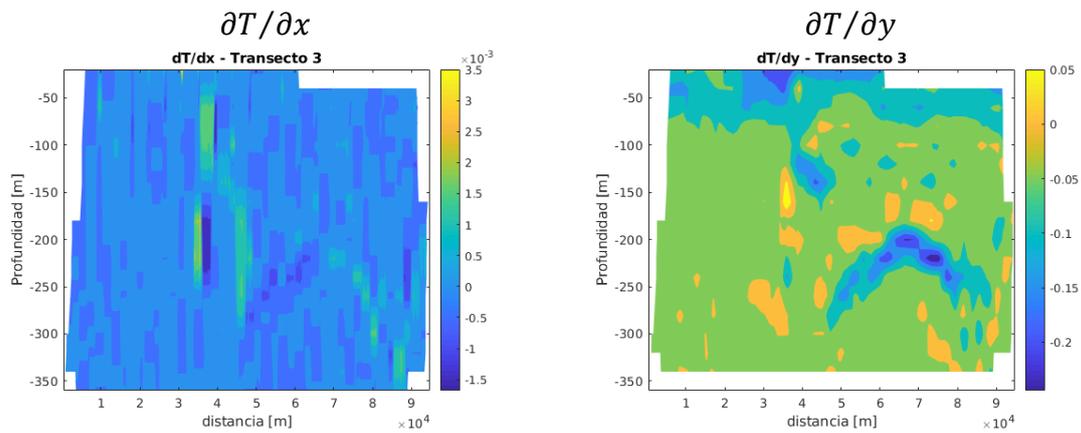
Rho_alt(i,j)=(alpha(i,j)*(dTdy_gscale(i,j)))/(beta(i,j)*(dSdy_gscale(i
,j)));

        %comportamiento TERMOCLINO
        if rho(i,j)==Rho(i,j)
            termoclino=termoclino+1;
            termoX(termoclino)=Dist_gscale(j);
            termoY(termoclino)=Press_gscale(j);
        end
    end
end
end

%preparamos los (x,y) de gran escala para los contours
[X_gscale,Y_gscale]=meshgrid(Dist_gscale,Press_gscale);
Y_gscale=-Y_gscale;

```

Y los resultados han sido los siguientes:



Inclinaciones: γT

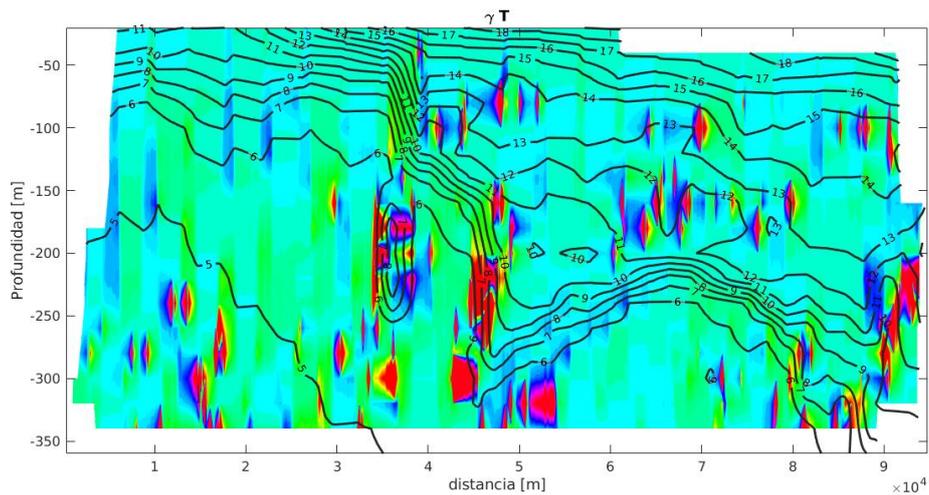
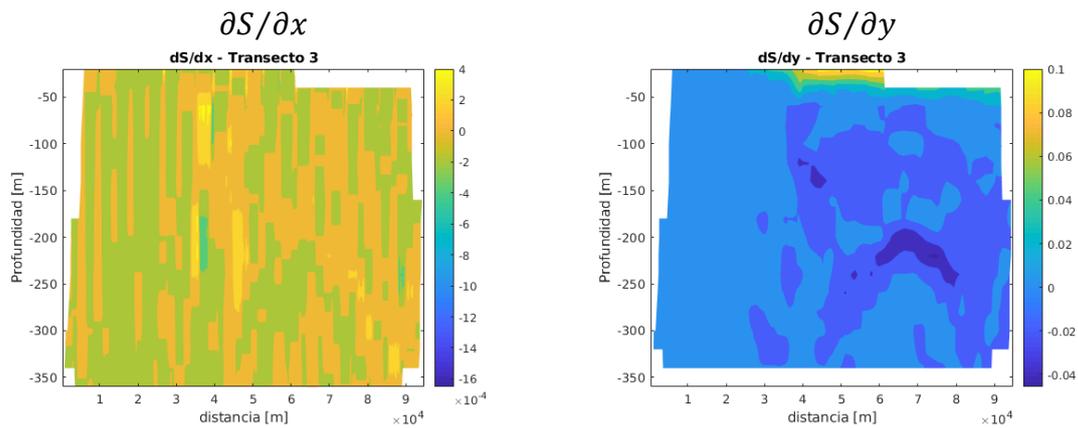


Figure 24: Inclinaciones γT

Los *plots* se corresponden a las diferenciales respecto el eje x (horizontal) y el eje y (vertical) de la temperatura. Podemos ver los gradientes máximos tanto en x como en y, pero también es cierto que la señal no muestra una intensidad importante. Esto es debido al ruido que genera la propia naturaleza del frente y al cómputo de las diferenciales. En este sentido hay que mejorar y aplicar un método de filtro paso bajo para eliminar el ruido sin alterar la señal de interés.

Ahora ya podemos buscar las inclinaciones. El contour γT muestra, en primer lugar las isothermas y en segundo lugar, podemos ver perfectamente las inclinaciones en los lugares donde los gradientes y las inclinaciones son máximas. Se puede ver el color rojo intenso. En azul cielo podemos ver las inclinaciones a 0 o prácticamente 0.

Del mismo modo, podemos encontrar las inclinaciones de las isohalinas. En la página siguiente se muestran los resultados.



Inclinaciones: γ_S

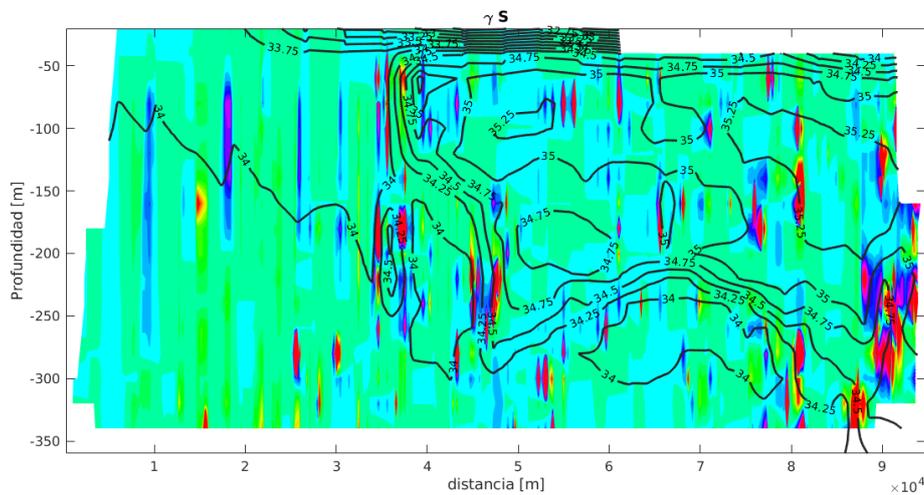
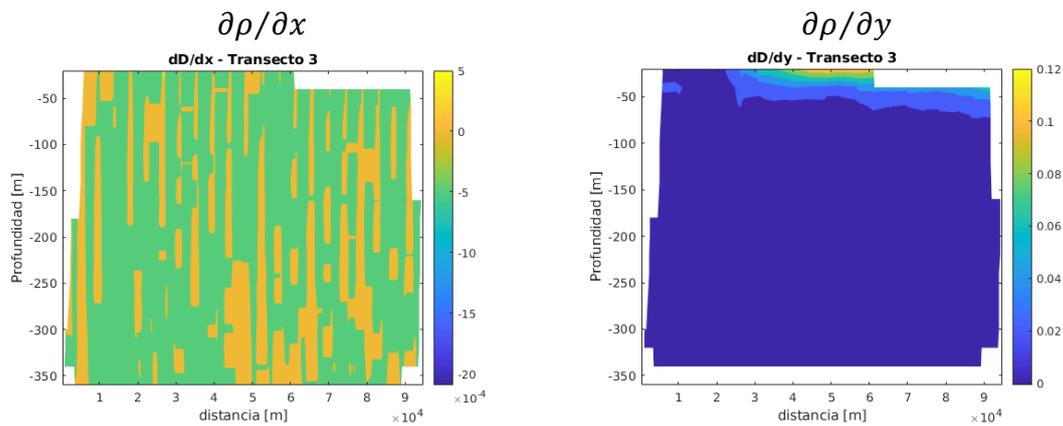


Figure 23: Inclinaciones γ_S

En este caso, los *contours* $\partial S/\partial x$ y $\partial S/\partial z$ muestran los gradientes en x e y en todo el transecto. En amarillo los gradientes máximos.

En esta ocasión ocurre lo mismo que en el caso anterior. Los *contours* están afectados por esa señal de ruido que impide una clara visualización de la señal de interés. Se debe corregir este efecto pues afecta, no sólo en el cómputo de las inclinaciones, también en el cálculo de la *Rho*.

El tercer *contour* muestra en primer lugar, las isohalinas y en segundo lugar las inclinaciones de las isólinas. Podemos ver cierta coincidencia, al igual que en la temperatura, allí donde hay una isohalina vertical, tenemos una “mancha” roja que indica esa inclinación máxima. En el resto del *contour* las inclinaciones son 0 o poco inclinadas en azul cielo.



Inclinaciones: $\gamma\rho$

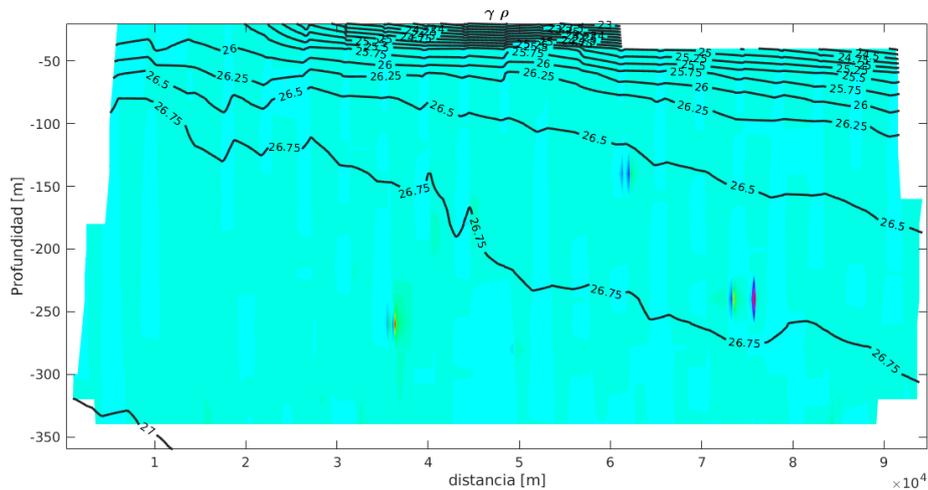


Figure 26: Inclinaciones $\gamma\rho$

Y finalmente, en el caso de la densidad, el cómputo diferencial no nos aclara su comportamiento, si bien nos dice que no hay máximos ni mínimos en los gradientes en x ni en y . En el caso de $\partial\rho/\partial y$, vemos en la superficie que hay un gradiente importante.

Lo vemos también en el contour de las isopícnas. Allí tenemos un gradiente máximo. En cuanto a las inclinaciones vemos un plot totalmente homogéneo, sin extremos.

Eso sí, las isopícnas cortan a las isobaras y por lo tanto tendremos, en general, una aproximación al frente baroclino.

Y ya para finalizar el análisis, mostramos dos contours más. El primero hace referencia al cómputo de la $R\rho$. Recordemos que si la relación de inclinaciones

$\gamma_S/\gamma_T = R\rho$ luego estaríamos hablando de un frente con un comportamiento termoclino puro.

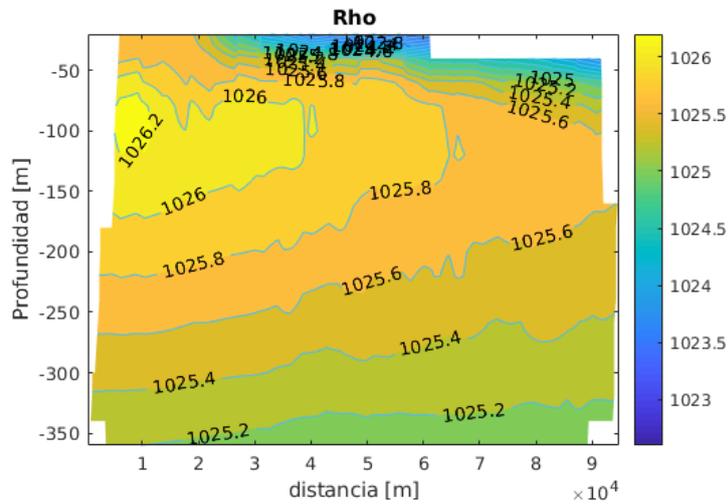


Figure 27: Rho

Lamentablemente, en esta ocasión no podemos calcular la relación γ_S/γ_T por los motivos descritos anteriormente de ruidos que enmarcaran la señal de interés y no podemos continuar sin aplicar algún tipo de criterio para elaborar un filtro que destruya el ruido pero que deje intacta la señal de interés. En el siguiente apartado se describe gráficamente la acción del ruido.

Pero si podemos calcular que zonas del frente se comportan como un frente baroclino puro. Así, cuando cumpla la relación $\gamma_S/\gamma_T = 1$, estaremos en zonas frontales con un carácter marcadamente baroclino puro.

Siendo así, mostramos:

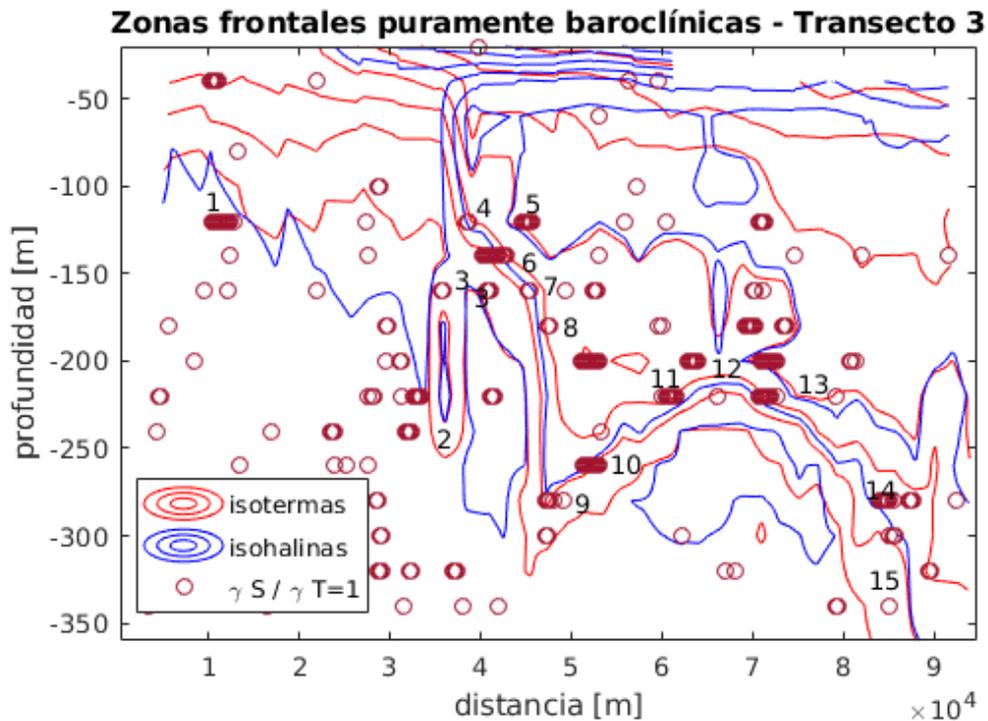


Figure28: Baroclinicidad-Termoclinicidad

En la imagen se han insertado las isoterms en rojo y las isohalinas en azul. Y como podemos observar, los puntos nos indican las zonas baroclínicas puras.

Y tiene su lógica pues la mayoría de puntos se encuentran en zonas donde las isolíneas de temperatura y salinidad van en paralelo.

4.3. Ruido

Durante el análisis y el proceso de la redacción de este trabajo, nos hemos encontrado con situaciones complejas.

Un sistema frontal de estas características está sujeto a perturbaciones e intrusiones que generan una gran variabilidad en sus propiedades termohalinas. Cuando, y este era el propósito de este trabajo, se analiza un sistema a pequeña escala pueden surgir estos problemas de ruido que anulan por completo la señal de interés.

El tema mejora cuando se realiza un análisis a gran escala, con promedios que ayudan a suavizar las curvas. Al final se ha recorrido al análisis de la gran escala debido a este efecto no deseado.

Para el estudio a pequeña escala hay que tener muy en cuenta el efecto de los ruidos.

A continuación, se presentan unas imágenes que nos han afectado al estudio:

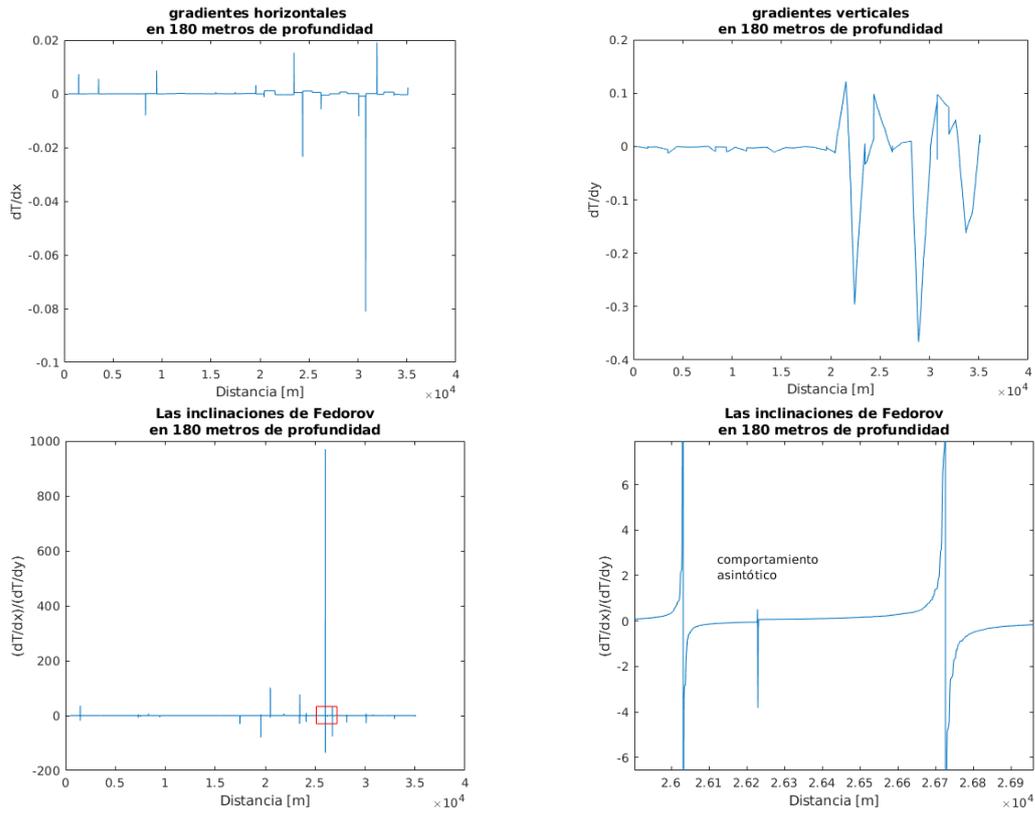


Figure 29: Ejemplo de ruido a 180 metros

5. Conclusiones

Este trabajo aborda tres temas importantes. La contextualización del trabajo, reconstrucción y el análisis posterior.

¿Porque han sido importantes estos tres bloques? El de la contextualización es importante por tratarse de un TFM ad-hoc y poco o nada el director/a y los miembros del tribunal que evalúan el presente trabajo conocen la naturaleza del mismo. De este modo, nos vamos introduciendo en el campo de la oceanografía sin alejarnos de la ciencia computacional (y matemáticas) para abordar el eje central de todo este trabajo: la reconstrucción de un sistema frontal (en este caso, la confluencia de las corrientes de Malvinas y Brasil).

Pero la reconstrucción por sí sola, tampoco conduce a nada. Es por ello que nos atrevemos a ir un paso más allá. Para el futuro análisis necesitamos la reconstrucción y necesitamos la reconstrucción para realizar un análisis.

Es por todo ello que, aunque el análisis no tiene nada que ver con el Máster de Ingeniería Computacional y Matemática, es clave para justificar la reconstrucción previa.

Pero concretemos.

El SeaSoar es un equipo científico capaz de medir a una alta resolución espacial y temporal estructuras físicas a pequeña escala tales como sistemas frontales (como el desarrollado aquí) sistemas de remolinos ciclonicos y anticlonicos, etc. Pero son pocos los investigadores usuarios de este equipo. Tanto es así que no hay código suficiente para el procesado de los datos obtenidos. Esta es pues una contribución al conjunto de software para el procesado de datos de sistemas basados en caracterización de la columna de agua remolcados.

Pero también hay que destacar, que las funciones presentadas habría que mejorarlas, en el sentido estricto de su funcionalidad. No es un paquete cerrado de funciones, todo lo contrario, más bien es un paquete abierto que está en desarrollo.

Como ya se ha indicado con anterioridad, hay que tratar mejor las señales ruidosas pues nos afectan mucho al análisis de la pequeña escala. Este ha sido el factor que nos ha obligado a tratar el análisis de la gran escala.

Tanto la reconstrucción del capítulo 3 así como el análisis del capítulo 4, no terminan aquí. A lo largo del desarrollo del Máster de Ingeniería Computacional y Matemática, ha ido creciendo también la inquietud por cursar el doctorado en el programa de ciencias del Mar, un programa convenido con la Universitat

Politécnica de Catalunya - UPC y el Instituto de Ciencias del Mar – ICM. Precisamente, el presente TFM es el embrión de la tesis doctoral que si todo va bien se iniciará el curso académico 18-19.

En este sentido, el actual trabajo aquí expuesto no termina, todo lo contrario, ha hecho sus primeros pasos

4. Glosario

Isoterma: Línea o superficie de temperatura constante

Isohalina: Línea o superficie de salinidad constante

Isopicna: Línea o superficie de densidad constante

Isobara: Línea o superficie de presión constante

Frente / Sistema frontal: zona de confluencia entre dos masas de agua de características termohalinas distintas

Características termohalinas: temperatura y salinidad

Baroclino: (muy breve) capacidad de separación

Termoclino: (muy breve) capacidad de unión

Alpha: Coeficiente de expansión térmica

Beta: Coeficiente de contracción salina

Hesperides: Buque de investigación oceanográfica

BMC: Brazil-Malvinas Confluence

EZR: Equatorial Zonal Retroflexion

SeaSoar: Equipo científico que mide temperatura, salinidad, densidad y presión. Equipo remolcado.

CTD: Equipo científico que mide temperatura, salinidad, densidad y presión. Perfilador vertical

5. Bibliografía

Artículos:

[1] K.N. Fedorov. Las características termohalinas de los frentes en el océano. 1988, T 302, Nº1.

[2] N. Kuzmina. On the structure and dynamical features of intrusive layering in the Eurasian Basin in the Arctic Ocean

[3] N. Kuzmina. The structure and driving mechanisms of the baltic intrusions. Journal of physical oceanography. Volume 35

Bibliografía:

[3] Climate and the oceans. Geoffrey k. vallis. ISBN:978-0-691-14467-2

[4] Applied numerical methods using Matlab. Won Young Yang. ISBN: 0-471-69833-4

[5] Introduction to physical oceanography. Robert H. Stewart. Department of Oceanography. Texas A & M University. Copyright 2008.

Enlaces web:

[6] <https://www.sciencedirect.com/science/article/pii/0198014979900888?via%3Dihub>

[7] <http://nptel.ac.in/courses/119102007/2>

[8] <http://www.oc.nps.edu/nom/day1/partb.html>