UiT
THE ARCTIC UNIVERSITY OF NORWAY

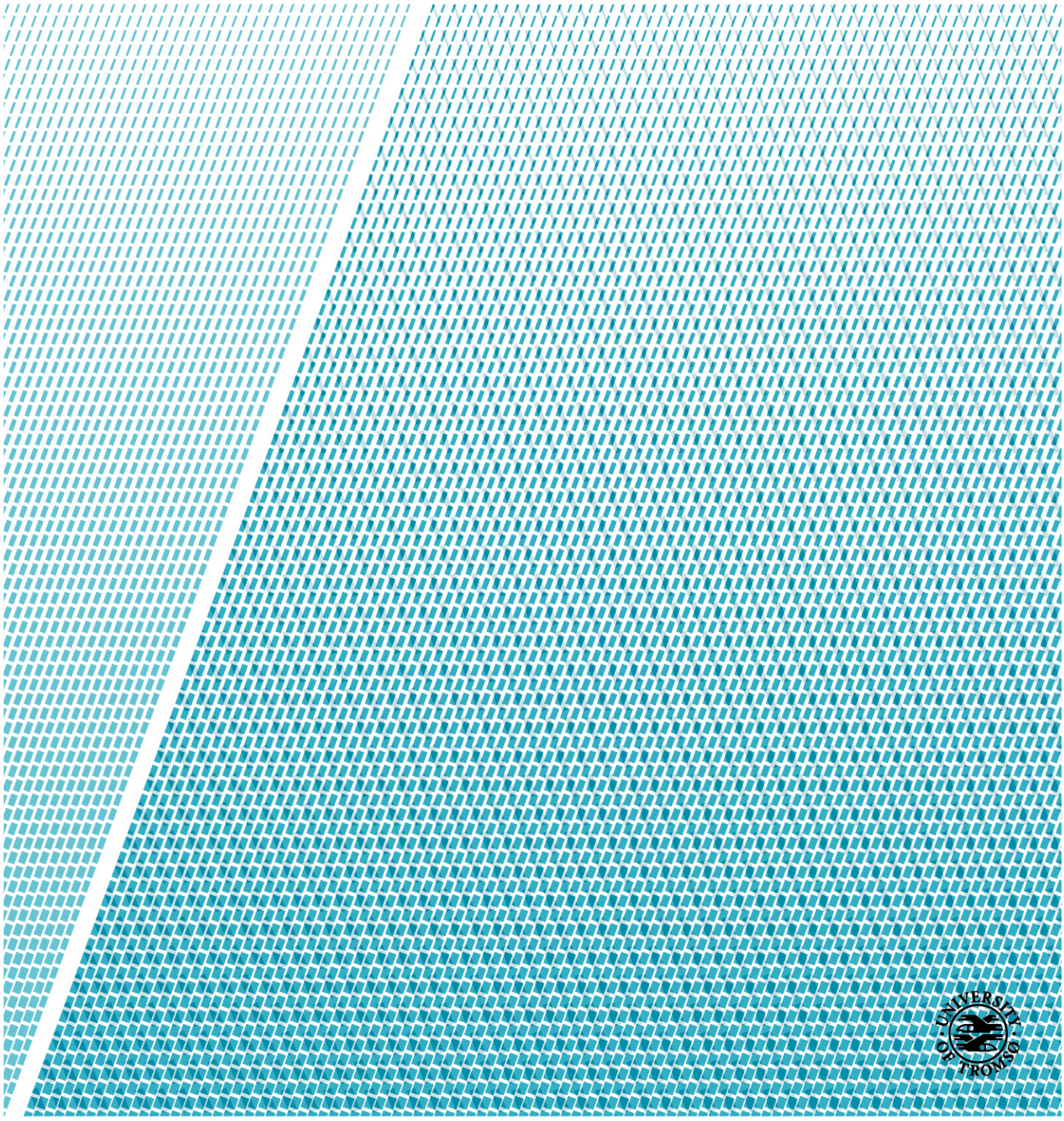
Faculty of Science and Technology - Department of Geosciences

Processing and interpretation of the Svyatogor 2016 high-resolution P-Cable 3D seismic dataset

Investigating the dynamics of a sub-seabed gas hydrate system with a potential abiotic methane source

Rowan Romeyn

Master's thesis in Geology GEO-3900 – 1st June 2017



Abstract

A new high-resolution P-Cable 3D seismic dataset was acquired in July 2016 targeting a seafloor pockmark cluster at the northern end of Svyatogor Ridge, offshore west Svalbard. The processing and interpretation of this dataset formed the primary focus of this thesis. The seismic processing sequence was designed to enhance the signal-to-noise ratio of the data while preserving the useful signal bandwidth and was implemented using the RadexPro 2016.3 software package. For example, burst-noise filtering allowed useful signal to be extracted from channels that would otherwise have been discarded, improving the overall trace density of the dataset. The suppression of bubble effects, ghost waves and random noise lead to a significant improvement in the useable bandwidth. In addition, "high-resolution static" correction proved an important means of improving reflector continuity and suppressing acquisition footprint noise caused by tides and streamer depth variations. Considerable effort was also spent on improving the receiver geometry based on a least squares type inversion of direct wave arrivals and produced a noticeable, if subtle, improvement in reflector continuity. This method of assigning geometry also has a potential application in 4D (time-lapse) seismic processing where the small-magnitude but extensive quantitative differences in amplitude, compared to the conventional method of assigning geometry, may become more critical.

Interpretation of the Svyatogor 2016 3D dataset indicates that the gas-hydrate, free-gas system has been relatively stable with respect to leakage and the seafloor pockmarks have been inactive and infilling for some time. However, significant evidence of paleo fluid migration was observed and the continued re-opening of fractures at fault tips or fault-segment junctions may be an important mechanism facilitating focussed, vertical fluid migration. The episodic fluid flow regime is postulated to be driven by 1) gas migration into the system along faults, probably as a dissolved phase 2) gas-hydrate formation at the base of the gas-hydrate stability zone (BGHSZ) produces a hydrate-cemented seal that results in a structurally enhanced trap 3) continued gas migration from depth and recycling of hydrate at the BGHSZ leads to accumulation of a free-gas phase contained beneath the BGHSZ 4) overpressure builds beneath the BGHSZ as gas charge continues eventually resulting in episodic gas release triggered in combination with dynamic stresses from earthquakes. The degree to which free-gas zone overpressure or external tectonic stresses control fault-slip is difficult to differentiate, but free-gas zone overpressures may significantly increase the slip tendency of faults at Svyatogor ridge.

It appears unlikely that the Svyatogor gas-hydrate/free-gas system could have been supplied by in-situ methane production alone. It remains difficult to conclusively rule out the contribution of a thermogenic source, but this would likely be dependent on lateral migration pathways that were not studied in detail in this thesis. However, it does appear plausible that the observed free-gas zone could have been charged by abiotic methane migrating along axial detachment faults during the period of active sedimentation on Svyatogor ridge assuming a similar flux rate to that reported by Cannat et. al. (2010) for the Rainbow hydrothermal field. A small elongated pockmark located above the lateral tip of an underlying fault may be associated with the most recent episode of seabed gas leakage. It is therefore considered the most promising target for future sediment coring aiming to recover gas or hydrate bearing samples for geochemical analysis to potentially resolve the methane source question.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would firstly like to thank the crew of the R/V Helmer Hanssen and the rest of the shipboard science team of July 2016. A great deal of effort goes into the acquisition of 3D seismic data and I am very grateful for the opportunity to work with this data and to be involved through the whole chain from acquisition to processing and interpretation. I would also like to thank my supervisors Stefan Bünz & Andreia Plaza Faverola for their guidance, encouragement and discussion throughout the project, which has helped to steer this thesis along a (relatively) smooth path.

I also owe a special thanks to Kate Alyse Waghorn who has been a fount of knowledge throughout and a source of great ideas through many fruitful scientific discussions and occasional arguments. The RadexPro support team also deserves a mention, it has been a nice experience to work with software that is so responsively supported and to see the great strides in capability that their very active ongoing development is achieving.

I would also like to thank Helen Dulfer for moving to the opposite side of the world with me where we have shared in discovering the Arctic paradise that is Tromsø and Northern Norway. The sense of community shared amongst our cohort of fellow masters students has also made the experience a truly positive one.

Finally, I would like to thank my family who have always supported me in everything that I've done and have simply wanted the best for me.

This thesis is dedicated to the memory of my dear mother, Jane Romeyn. Mum, I miss you every day and will never forget your caring, curious and adventurous spirit and only hope to live after your example.

1 Introduction

The main objective of this thesis was to process and interpret a newly acquired 3D P-Cable seismic survey that was shot over the northern part of Svyatogor Ridge in July 2016 with the R/V Helmer Hanssen (see Figure 1). This new 3D seismic dataset was acquired to address the following scientific objectives: 1) to improve the understanding of the development of Svyatogor Ridge and its stratigraphic and tectonic setting, 2) to understand the dynamics of the Svyatogor gas-hydrate free-gas system, both in terms of the factors controlling gas leakage to the seabed and to understand where the methane held in the system may have come from.

This introductory chapter presents a brief description of the motivation behind the present study, an overview of the study area and its significance. The following chapter presents the key parameters of the seismic acquisition and important observations made aboard the survey vessel. The seismic processing sequence that was designed to maximise the interpretability of the dataset is described in detail in the next chapter. Finally, the interpretation and discussion chapters focus on how the newly acquired 3D seismic dataset can be used to illuminate the scientific objectives stated above.

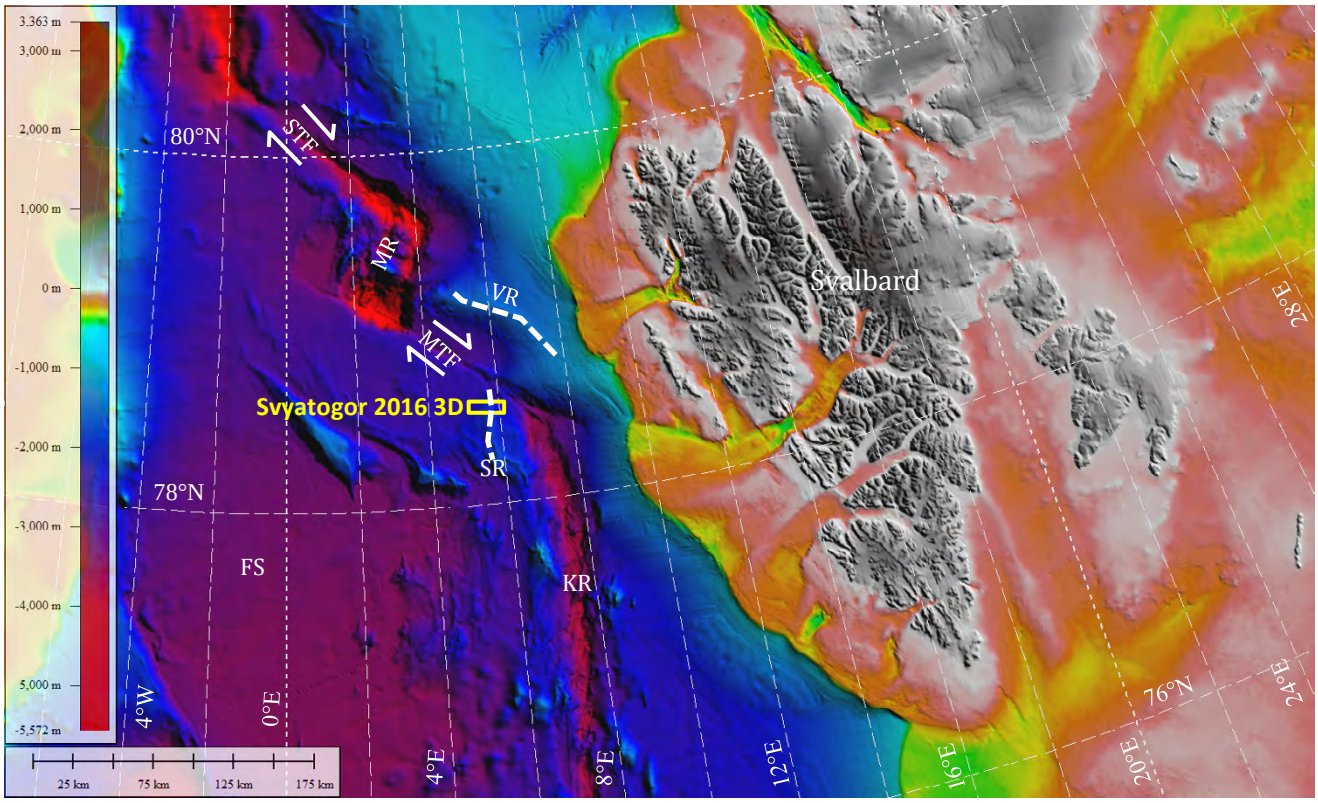


Figure 1 – Regional setting of the Svyatogor 2016 3D dataset (not shown at exact scale), offshore NW Svalbard, with respect to the important structural features of Svyatogor Ridge (SR), Vestnesa Ridge (VR), Knipovich Ridge (KR), Molloy Transform Fault (MTF), Molloy Ridge (MR), Spitsbergen Transform Fault (STF) and the Fram Strait (FS) plotted on IBCAO V3 bathymetry (500m grid).

Svyatogor Ridge lies within the Fram Straight to the west of the Knipovich Ridge (KR) and south of the Molloy Transform Fault (MTF) as illustrated in Figure 1. The Fram Strait is the only deep-water

connection between the North Atlantic and the Arctic Ocean and acts as a narrow gateway channelling inflows and outflows between the two ocean basins, including the northward flowing West Spitsbergen Current (WSC) that flows in the vicinity of the study area. Sediments along the eastern flank of the Fram Strait are mainly deposited as contourites under the influence of the WSC (Eiken & Hinz, 1993; Howe, et al., 2008).

Svyatogor Ridge has been hypothesized to have once been a part of Vestnesa Ridge before being offset along the MTF over the past 2Ma (Johnson, et al., 2015). Compared to Vestnesa, which is a large sediment drift hosting a highly active gas hydrate system, the sediment cover across Svyatogor Ridge is limited as displacement along the Molloy Transform Fault has moved the ridge further away from the path of the West Spitsbergen Current (WSC). The WSC is a dominant mechanism of sediment transport and deposition in this deep-water environment (Johnson, et al., 2015).

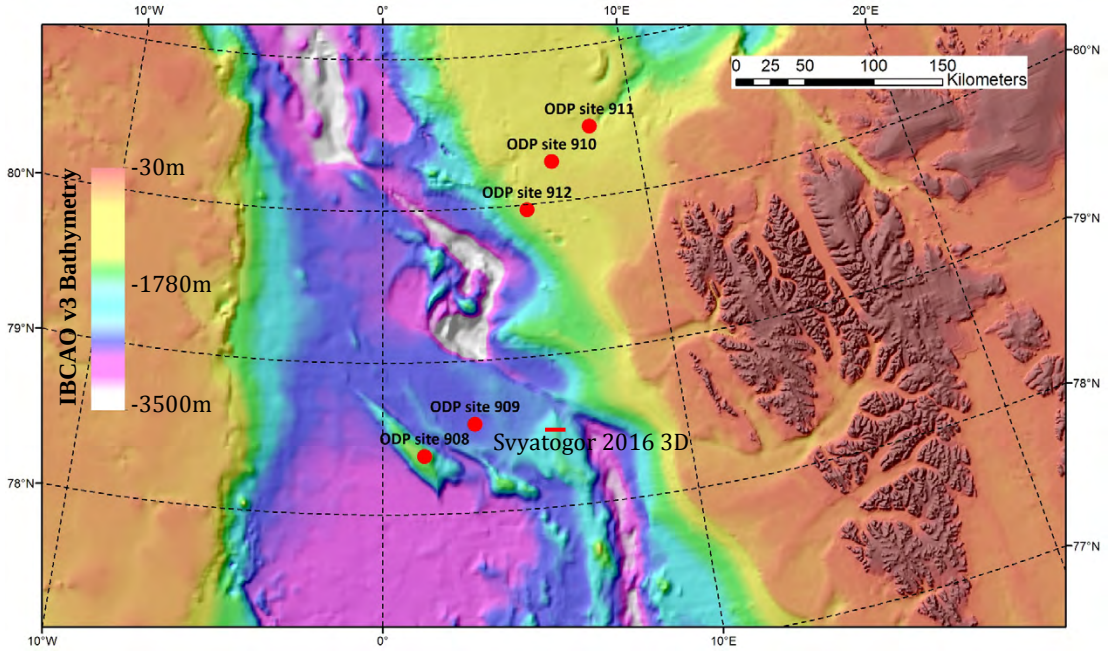


Figure 2 – Location of ODP boreholes 908-912 from survey Leg 151 (Ocean Drilling Program, 2005) that constrain the regional seismostratigraphic units YP-1, YP-2, YP-3 (Eiken & Hinz, 1993; Hustoft, et al., 2009; Mattingsdal, et al., 2014). The ODP boreholes were used in the present study to constrain rough estimates of sediment physical properties such as average bulk density and porosity around the sub-bottom depth of the base of the gas-hydrate stability zone, which has not been drilled at Svyatogor Ridge.

Sedimentation on the nearby Vestnesa Ridge has been punctuated by episodic turbiditic flows and increased sediment supply during glacial periods due to the combination of lower sea-level and the proximity of a fast-flowing ice stream from the Kongsfjordrenna extending to, or near, the shelf edge (Howe, et al., 2008). This sediment supply is unlikely to be a significant factor at the more distally located Svyatogor Ridge. However, a shift towards higher sedimentation rates around 2.7 Ma may have also influenced sedimentation at Svyatogor Ridge. This shift has been attributed to the intensification of Northern Hemisphere glaciation, subaerial exposure of the Barents shelf and expansion of ice sheets in the northern Barents Sea and Svalbard causing enhanced erosion and fluvial input along the pathway of the WSC (Gebhardt, et al., 2014). Despite the younger underlying crust and limited sedimentation, the

presence of a gas-hydrate system along Svyatogor Ridge has been inferred using seismic data, although it displays different dynamics and is thought to be driven by different mechanisms to the Vestnesa Ridge gas-hydrate system (Waghorn, et al., 2015; Johnson, et al., 2015).

Svyatogor Ridge remains a frontier area of study where the main stratigraphic control is provided by a regional seismostratigraphic framework tied to ODP Leg 151 boreholes 908-912 (see Figure 2). These boreholes have been used to constrain the chronology of the three seismic units (YP-1, YP-2 & YP-3) originally defined on the southern Yermak Plateau (Eiken & Hinz, 1993; Geissler & Jokat, 2004) and later correlated across the Molloy Transform Fault to ODP holes 908 & 909 (Knies, et al., 2009; Mattingsdal, et al., 2014). The base of the YP-3 corresponds with the base of glacial deposits with an age of ~2.7 Ma (Knies, et al., 2009; Mattingsdal, et al., 2014) and the package consists of glaciomarine contourites and turbidites. The chronology of the YP-2/YP-3 boundary (~2.7 Ma) is reasonably well constrained by correlation with the ODP boreholes (Knies, et al., 2009; Mattingsdal, et al., 2014). By contrast, the base of the contouritic YP-2 sequence has not been directly constrained by drilling results, but has been estimated to lie between 11 Ma (Mattingsdal, et al., 2014) and 14.6 Ma (Geissler, et al., 2011). At Svyatogor Ridge, the underling oceanic crust is relatively young, constrained to within the range 9.8-2.8 Ma by magnetic anomaly chrons 5 & 2A (Engen, et al., 2008). Consequently, the sedimentary sequence at Svyatogor Ridge is expected to consist of a package of YP-3 sediments and an incomplete section of YP-2 sediments.

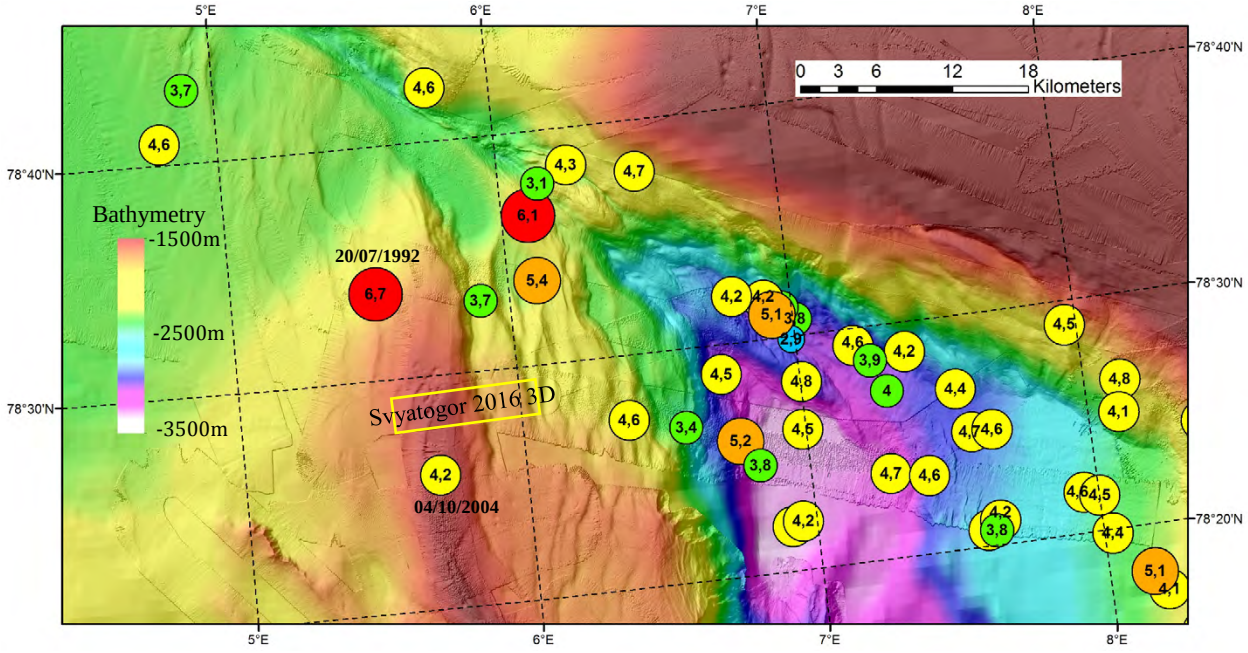


Figure 3 – Earthquakes in the vicinity of the Svyatogor 2016 3D dataset from the ANSS Comprehensive Earthquake Catalog (ComCat) spanning the period 1953-2016 (USGS, 2017), plotted by estimated magnitude. Positional uncertainty was not available for these earthquakes but is likely to be at least several km.

Svyatogor Ridge is located in an interesting tectonic setting; at the northern end of Knipovich Ridge and in close proximity to the Molloy Transform Fault. Knipovich ridge is a magma-limited ultraslow spreading ridge that is one of the slowest and most obliquely spreading mid-ocean ridges in the world

(Curewitz, et al., 2010). The northern end of Knipovich ridge trends N-S and is highly oblique to the spreading direction as given by a modelled plate motion vector of ~307° (Sella, et al., 2002; Curewitz, et al., 2010). In addition, the proximity of the Molloy Transform Fault to the study area may be expected to significantly perturb the otherwise ridge-normal extensional stress field. Shearing along an active transform fault has been proposed to perturb the stress field sufficiently to explain both the rotation and asymmetry in faulting observed at the ends of slow-spreading ridge segments (Behn, et al., 2002). Further, the inside corners of slow spreading ridge-transform systems are typically associated with thinned lithosphere consisting of an aggregate of variably deformed and intruded lower crustal and upper mantle rocks (due to large scale detachment faulting) that are likely to have exhibit complex failure patterns (Tucholke & Lin, 1994) and enhanced seismicity (Wolfe, et al., 1995; Smith, et al., 2002). The study area remains tectonically active to the present as illustrated by the distribution of modern earthquakes recorded over the period 1953-2016 (see Figure 3).

1.1 Significance of gas hydrates

Gas hydrates are a solid, crystalline, ice-like mixture of gas and water where gas molecules are held within a framework or cage of water molecules (Kvenvolden, 1998). Gas hydrates are stable under low-temperature, high-pressure conditions and are typically found in regions of permafrost or beneath the sea in continental margins and ocean basins (Sloan, 1998). The volume of methane trapped within gas hydrates globally is estimated to lie within the range of 170-12,700Gt (Dickens, 2011), which represents a significant reservoir in the global carbon cycle. It is important to understand the stability and fluxes of this reservoir since methane and other gases trapped in gas hydrates are potent greenhouse gases with the potential to contribute to long-term climate change if released to the atmosphere (Kvenvolden, 1993; 1998).

The temperature and pressure conditions required for gas hydrate formation are referred to as the gas hydrate stability zone (GHSZ). In a typical deep-water continental margin setting the upper limit of gas hydrate occurrence is the seafloor, although it is uncommon for gas hydrates to be located at the seafloor, except in areas of high gas-flux such as cold vents (MacDonald, et al., 1994). On the other hand, the lower limit of gas hydrate occurrence depends primarily on the limit of the gas hydrate stability zone (GHSZ), which is controlled by the overall water depth, seafloor temperature and the local geothermal gradient (Sloan & Koh, 2008). The presence of gas hydrates also depends on the kinetics of hydrate formation and dissociation, which in turn depends critically on the supply and composition of gas and liquid water within the pore space of sediments (Portnov, et al., 2016). In addition, the importance of gas migration from deep sources into the GHSZ has only been recognized since the late 1990s (Milkov, 2004). As a result, it is important to understand in detail the different mechanisms and spatial distribution of methane generation and migration through sediments in order to improve estimates of the size of the global gas hydrate reservoir (Milkov, 2004) and better understand its dynamics.

1.2 Identification of gas hydrates on seismic profiles

Gas hydrates begin to dissociate at the base of the gas hydrate stability zone (BGHSZ) so that any gas present below the BGHSZ will exist in a free-gas state. Seismic P-wave velocity decreases strongly in the presence of small quantities of free-gas (Holbrook, et al., 1996). Conversely, the presence of hydrate tends

to increase the P-wave velocity of sediments, although the effect is small at low concentrations until the crystals begin to interact with the granular skeleton at ~40% hydrate concentration (Yun, et al., 2005). As a result, a sharp decrease in velocity typically occurs at the BGHSZ along with a corresponding sharp decrease in acoustic impedance. This results in high-amplitude, bottom-simulating reflections (BSRs) on seismic profiles, with reversed polarity relative to the seafloor reflection. BSRs are so-called because they typically run parallel to the seafloor but at increasing sub-bottom depth with increasing water depth, due to increasing hydrostatic pressure and decreasing water-bottom temperature, which make gas hydrates become stable to greater depths (Shankar, et al., 2010). The BSR characteristically cross-cuts normal sedimentary strata reflections but can also be identified by distinct lateral transitions in seismic amplitude that follow the trend of the seafloor (see Figure 4).

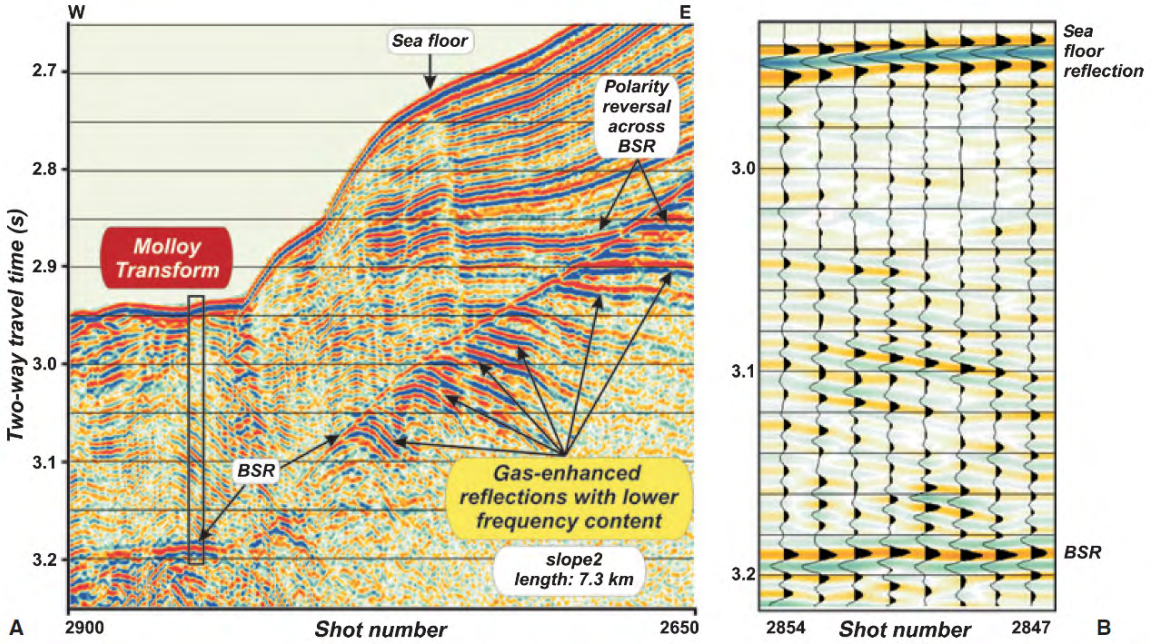


Figure 4 – A well-defined example of a bottom-simulating reflector (BSR) from Vanneste, et al., (2005). (a) The BSR runs parallel to the seafloor and cross-cuts the normal sedimentary reflections, abrupt termination of gas enhanced reflections is observed and (b) polarity is revered compared to the seabed reflection. This example is from the western Svalbard margin ~35km north of the Svyatogor 2016 3D dataset on the opposite side of the Molloy Transform Fault.

1.3 Significance of Svyatogor Ridge

Using carbon isotopes, it has been well established that microbial degradation and thermogenesis are important sources of methane in gas hydrates (typically methane comprises >99% of the gas held in gas hydrates) (Kvenvolden, 1998). A global review indicated that methane in gas hydrates is mainly derived by the microbial reduction of CO_2 from sedimentary organic matter, except in some areas like the Gulf of Mexico and the Caspian Sea where the methane is mainly thermogenic (Kvenvolden, 1995). Gas hydrates containing a mixture of microbial and thermogenic methane were also identified.

The unusual location of Vestnesa Ridge in close proximity to a mid-ocean ridge has important implications for the dynamics of the gas-hydrate and free-gas system. Geochemical measurements of gas from hydrates collected at the ridge have indicated a thermogenic source (Smith, et al., 2014). The high-

heat flow and tectonic activity of this setting, along with the thick sedimentary cover, lead to a shallow maturation window and an accelerated rate of biogenic and thermogenic gas production. A high degree of variability in gas supply, gas migration and gas hydrate build up and dissociation is also inferred (Bünz, et al., 2012).

More recently, an additional source of methane has been identified; termed 'abiotic' methane. It is generated in slow- to ultraslow-spreading mid-ocean ridge environments by the serpentinization of ultramafic rocks (Cannat, et al., 2010). Serpentinization involves the high temperature (200-350°C) hydrothermal alteration of the ultramafic rocks and leads to the production of hydrogen. The hydrogen produced during serpentinization can react with CO or CO_2, via Fischer-Tropsch type reactions, to produce abiotic methane (Proskurowski, et al., 2008).

Svyatogor Ridge is considered a prime candidate example of a gas hydrate-free-gas system that may have primarily been supplied by an abiotic source (Johnson, et al., 2015), in addition to a likely microbial contribution. The relatively thin sedimentary cover overlying young oceanic crust makes a thermogenic methane source much less plausible at Svyatogor Ridge compared to Vestnesa Ridge. Moreover, large-scale basement faults imaged on seismic data beneath Svyatogor Ridge are interpreted as oceanic detachment faults, which have exhumed relatively young, ultramafic oceanic-crust (Waghorn, et al., 2015; Johnson, et al., 2015).

Oceanic detachment faults dominantly occur at the inside corners of ridge-transform intersections of slow to ultraslow-spreading ridges (Buck, et al., 2005; Yu, et al., 2013). Globally, these detachment faults occur in areas of low magma supply at mid-ocean ridges, but at a local-scale they are found in areas with moderate magma supply i.e. magma supply is locally elevated (Yu, et al., 2013). The detachment faults at Svyatogor Ridge also follow this trend, occurring at the inside corner of the Molloy Transform Fault (MTF) and Knipovich spreading ridge. These detachment faults may act as conduits for fluid flow, allowing circulation of seawater to drive serpentinization and subsequently acting as pathways for fluids and abiotic methane to reach the shallow subsurface (Johnson, et al., 2015; Waghorn, et al., 2015). Many hydrothermal vents have been associated with detachment faults (Yu, et al., 2013), highlighting their importance to fluid flow.

The position of Svyatogor Ridge on the western flank of the northernmost part of Knipovich spreading ridge causes it to have an asymmetrical structure characterised by a relatively steep eastern flank with faults stepping down towards Knipovich ridge and a western flank with a slope that trends more gradually towards the Greenland-Norwegian Plateau (Chamov, et al., 2010). The structural setting of Svyatogor Ridge is illustrated in Figure 5, which shows the underlying oceanic crustal structures, detachment faults and previously interpreted BSR (reflection caused by the gas-hydrate to free-gas transition). The focus of the present study i.e. the Svyatogor 2016 3D dataset is located at the northern end of Svyatogor Ridge (see Figure 5) and targeted a pockmark cluster at the crest of the ridge that had been previously identified.

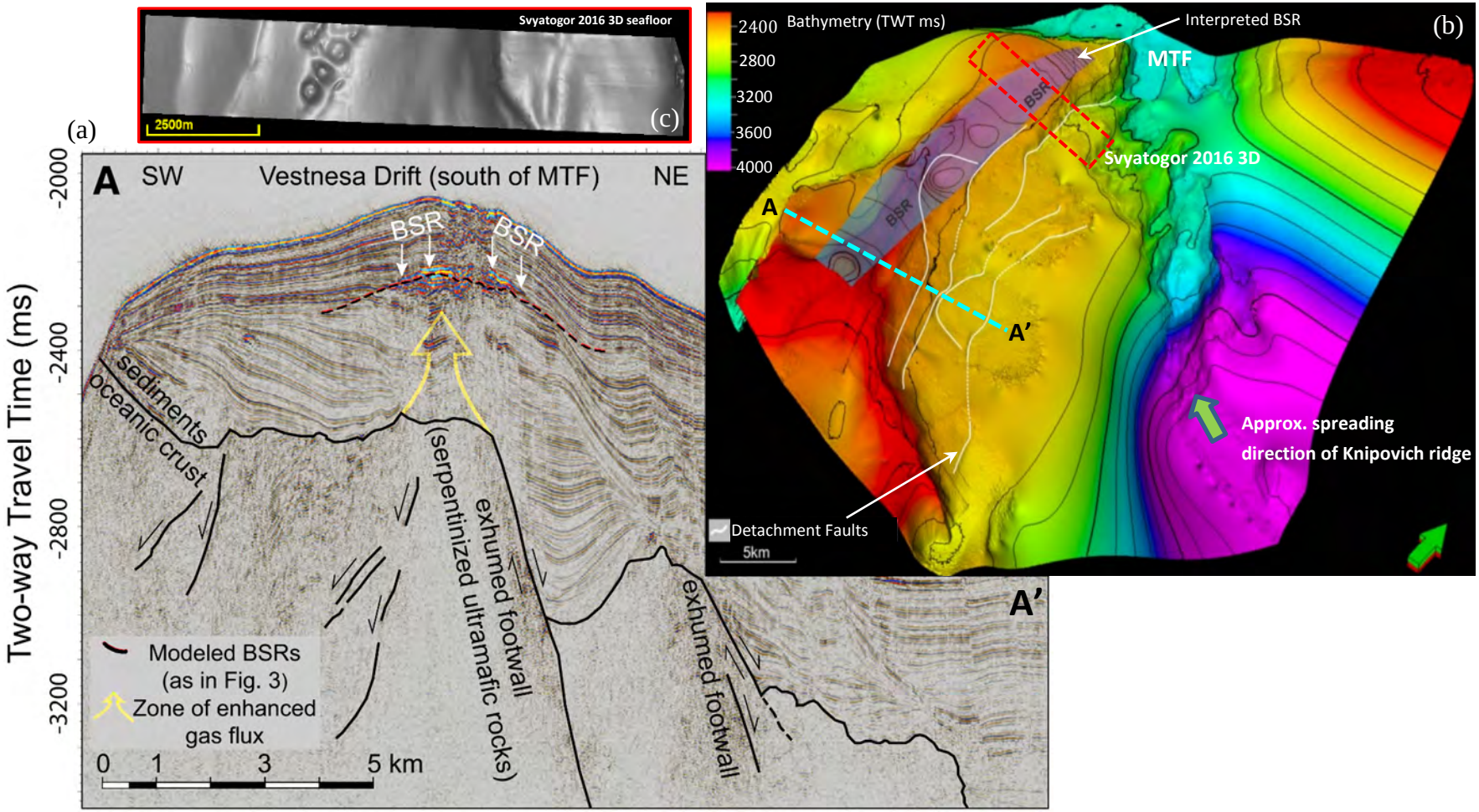


Figure 5 – High resolution 2D P-Cable seismic profile, modified from Johnson et. al. (2015) with oceanic crustal structures and BSR marked. (a) Note that what Johnson et. al. (2015) refer to as Vestnesa Drift South of Molloy Transform Fault (MTF), is referred to simply as Svyatogor Ridge in the present study. (b) Regional bathymetry map modified from Westvig (2015) to show the location of the 2D seismic profile (blue dashed line) and the location of the Svyatogor 2016 3D survey (red box), the extent of the BSR and location of axial detachment faults interpreted by Westvig (2015) are also shown. (c) Greyscale map of the seafloor from the Svyatogor 2016 3D dataset shows the ridge-crest pockmark cluster that the survey was designed to target

2 Seismic Acquisition

2.1 Introduction

The 3D seismic survey that constitutes the primary focus of this thesis was acquired during the CAGE-16-6 cruise conducted in July 2016 aboard the R/V Helmer Hanssen using a P-Cable high-resolution 3D seismic system and GI airgun source. The CAGE-16-6 cruise, sailing out of Longyearbyen (Figure 6), was led by Stefan Bünz and focussed on the acquisition of seismic data to study gas hydrate systems. The main activities included: (1) collection of Ocean Bottom Seismometers (OBS) following long term passive recording deployment at Vestnesa Ridge followed by redeployment for an active source OBS experiment. (2) Acquisition of 2D and 3D seismic data including large 3D data volumes in the Svyatogor and Storfjordrenna study areas. (3) Multibeam bathymetry mapping and mapping of gas flares in the water column using the recently upgraded multibeam system.

Figure 6 – Sailing out of Isfjorden from Longyearbyen at the beginning of the CAGE-16-6 cruise.

2.2 Survey Configuration

The P-Cable seismic system (Planke, et al., 2009; Petersen, et al., 2010) was developed to allow cost-efficient high-resolution 3D subsurface imaging. The system consists of an array of closely spaced and very short streamers that are towed via a cross cable that is spread by two large trawl doors (paravanes) and runs perpendicular to the direction of travel of the ship (Figure 7). The very high resolution that the system is capable of is best realized with small volume seismic sources, imaging relatively shallow targets (typically down to sub-bottom depths similar to the water depth) over focussed study areas of 10-50km^2 (Planke, et al., 2009).

During acquisition of the 2016 Svyatogor 3D seismic dataset the P-Cable system was deployed in the configuration shown in Figure 7, consisting of 14 streamers, each 25m long and capable of recording via 8 receiver groups. The key acquisition parameters are summarised in Table 1. Some variation from the ideal survey configuration was observed during acquisition of the Svyatogor 3D dataset, where one paravane would be offset relative to the other such that a line bisecting the two paravanes was not perpendicular to the sailing direction (see Figure 8). This variation in geometry was attributed to current from the north because the starboard paravane would tend to trail when sailing towards the

west and the port paravane would trail when sailing towards the east. It was also notable that the bearing of the ship was typically deviated 10-20° to the north from the direction of motion (Figure 8).

In addition, onboard QC of the geometry consistently showed the greatest misfit between modelled and observed direct-wave arrivals, which indicates poor fit of the assigned geometry, on the northernmost streamers independent of sailing direction (see Figure 9). Poor fit with the assigned geometry was also observed during times that the ship was not holding a steady course i.e. in the vicinity of wiggles in the ship track. The issue of poorly fitting geometry that was identified at sea during seismic acquisition was further addressed once back onshore during processing of the dataset (see section 3.5 pg.21).

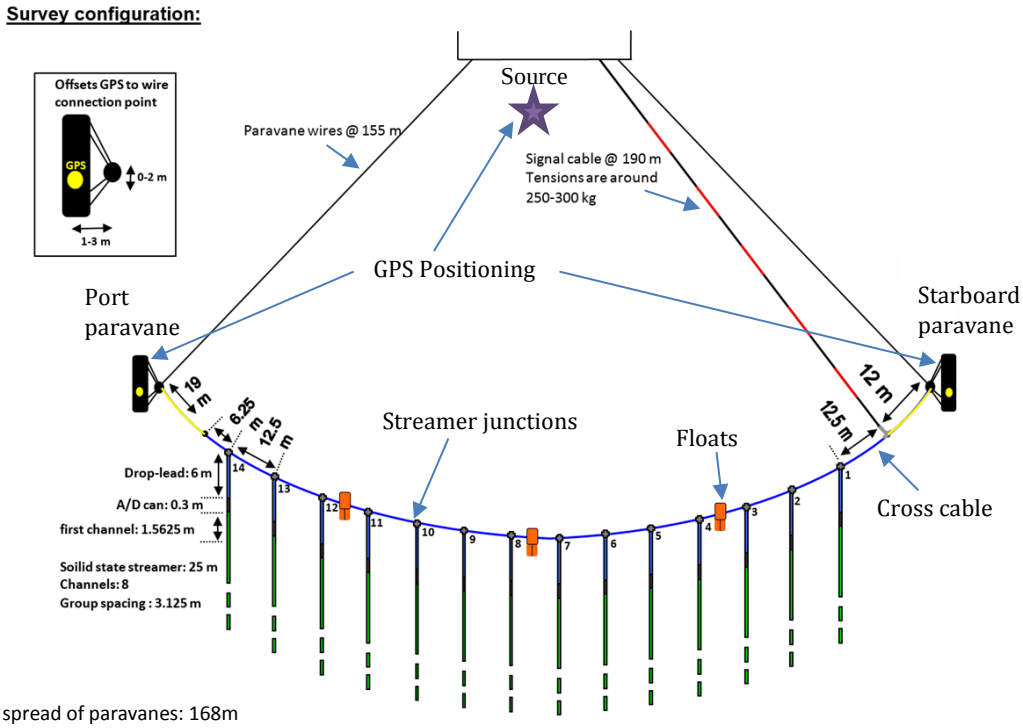


Figure 7 –Survey configuration of the P-Cable high-resolution 3D seismic system that was used to acquire the Svyatogor 2016 3D dataset.

Table 1 – Summary of key survey parameters for the Svyatogor 2016 3D dataset

Number of sailing lines	31
Number/length of streamers	14 streamers of 25m length
Number of channels per streamer	8
Approx. line length	12km
Approx. survey area	22km^2 (acquired in ~69 hours)
Gun volume and pressure	45/45 in^3 (harmonic GI mode) at 150-160 bar
Shot interval and ship speed	6 seconds at 4 ± 0,3 knots
Shot point distance	12 ± 1 m
Record length	4 seconds

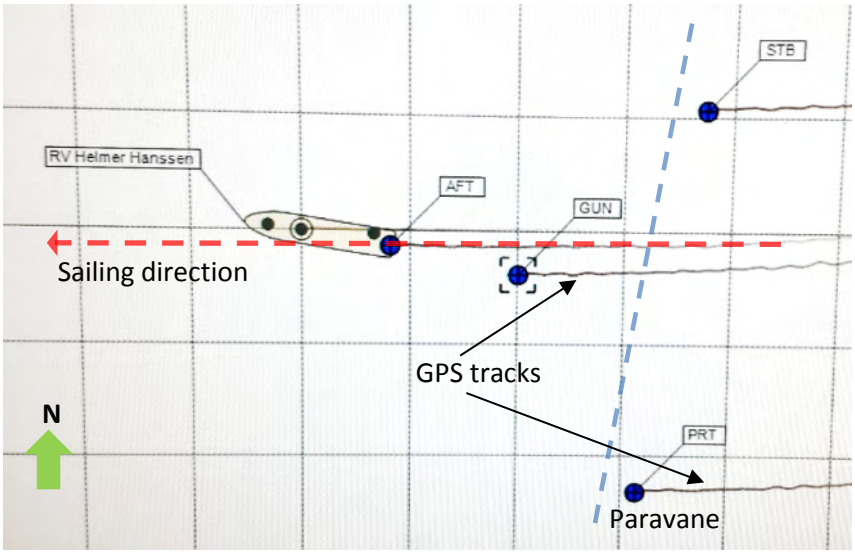


Figure 8 – Deviation from the ideal acquisition configuration during an actual sailing line of the Svyatogor 3D dataset, most likely caused by current from the North.

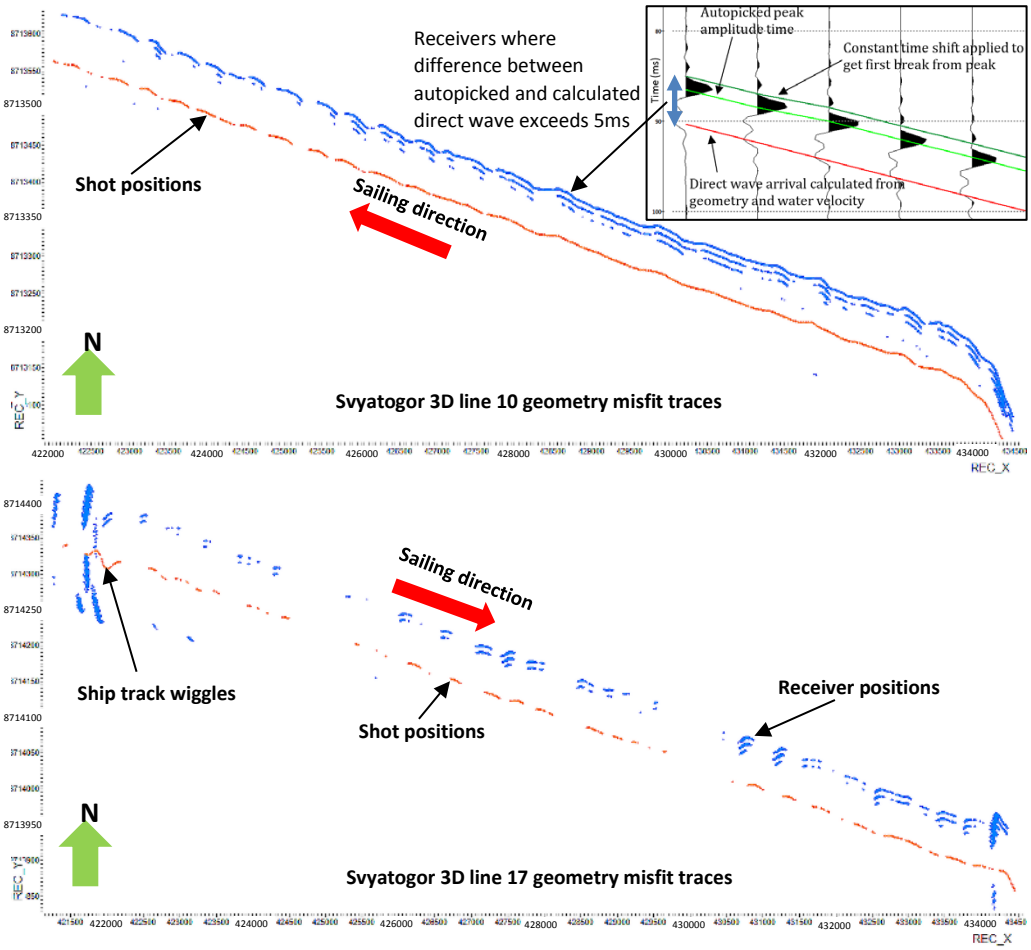
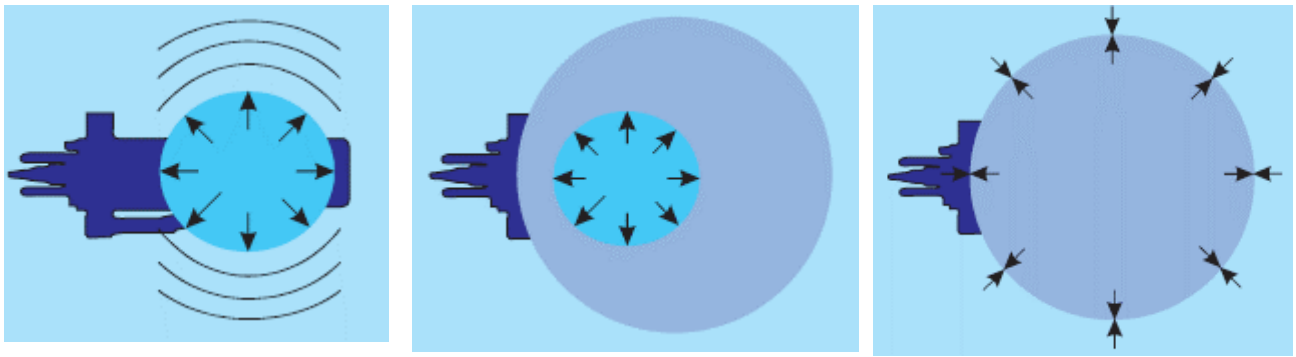


Figure 9 – Shipboard geometry QC plot (in UTM Zone 32N coordinates) of traces where calculated direct wave arrivals (based on geometry) differ from observed (autopicked) by more than 5ms (see inset). The largest misfit was consistently observed on streamers to the north of the sailing line (indicated by the shot positions) and is interpreted to be caused by current from the north causing some distortion from the ideal survey configuration. This issue was further addressed once back onshore during final processing of the dataset (see section 3.5 pg.21).

2.3 Seismic source

The Svyatogor 2016 3D dataset was acquired using GI airguns (meaning generator-injector). These guns were developed to reduce or suppress bubble oscillations when using a single air gun by injecting air into the bubble created by the air gun when this bubble reaches its maximum volume (GI process). Depending upon the characteristics of the injection, the bubble oscillations can be reshaped and reduced, or totally suppressed (Landrø, 1992). One GI gun is therefore comprised of two independent air guns within the same casing. The first air gun is called the Generator, as it generates the primary pulse. The second one is called the Injector, as it injects air inside the bubble produced by the Generator (see Figure 10).



1. Generator fires blast of compressed air producing an expanding bubble.

2. Injector fires when bubble is near max. size and encompasses the injector reducing its internal pressure below hydrostatic.

3. The injection of air inside the bubble increases its internal pressure preventing rapid collapse and reducing subsequent oscillation.

Figure 10 – Brief explanation of the firing sequence of a GI (generator-injector) gun, diagrams from manufacturer technical documents (Sercel, 2006).

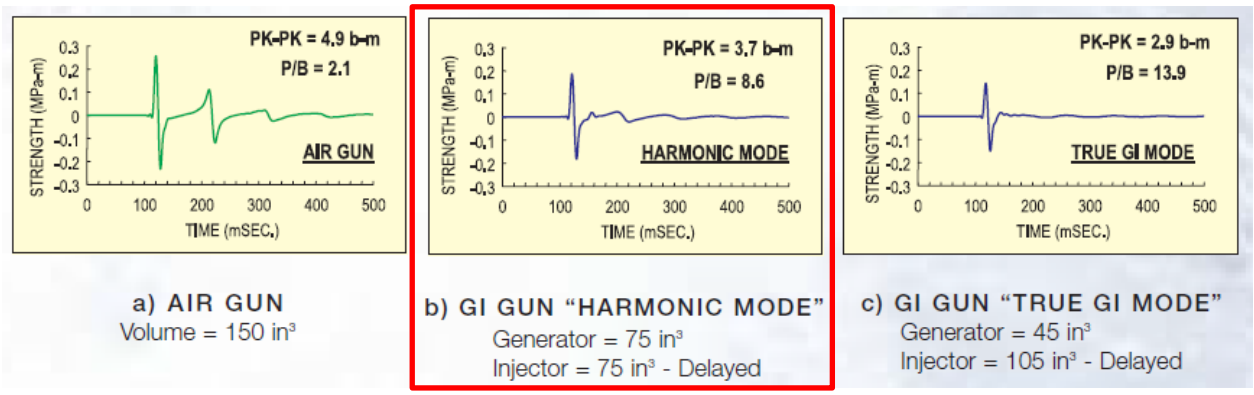


Figure 11 – Comparison of far-field signatures between different gun setups with the same total volume, from manufacturer Sercel technical docs (Sercel, 2006). The gun volume devoted to injection (b) and (c) results in decreased peak-peak amplitude compared to a conventional airgun (a) but dramatically reduced secondary pulses. During the CAGE-16-6 survey the guns were run in harmonic mode (highlighted in red). Signatures recorded with: Pressure = 2,000 PSI, Depth = 6.0 Meters, Filtered DFS 0-256 Hz 72 dB/o

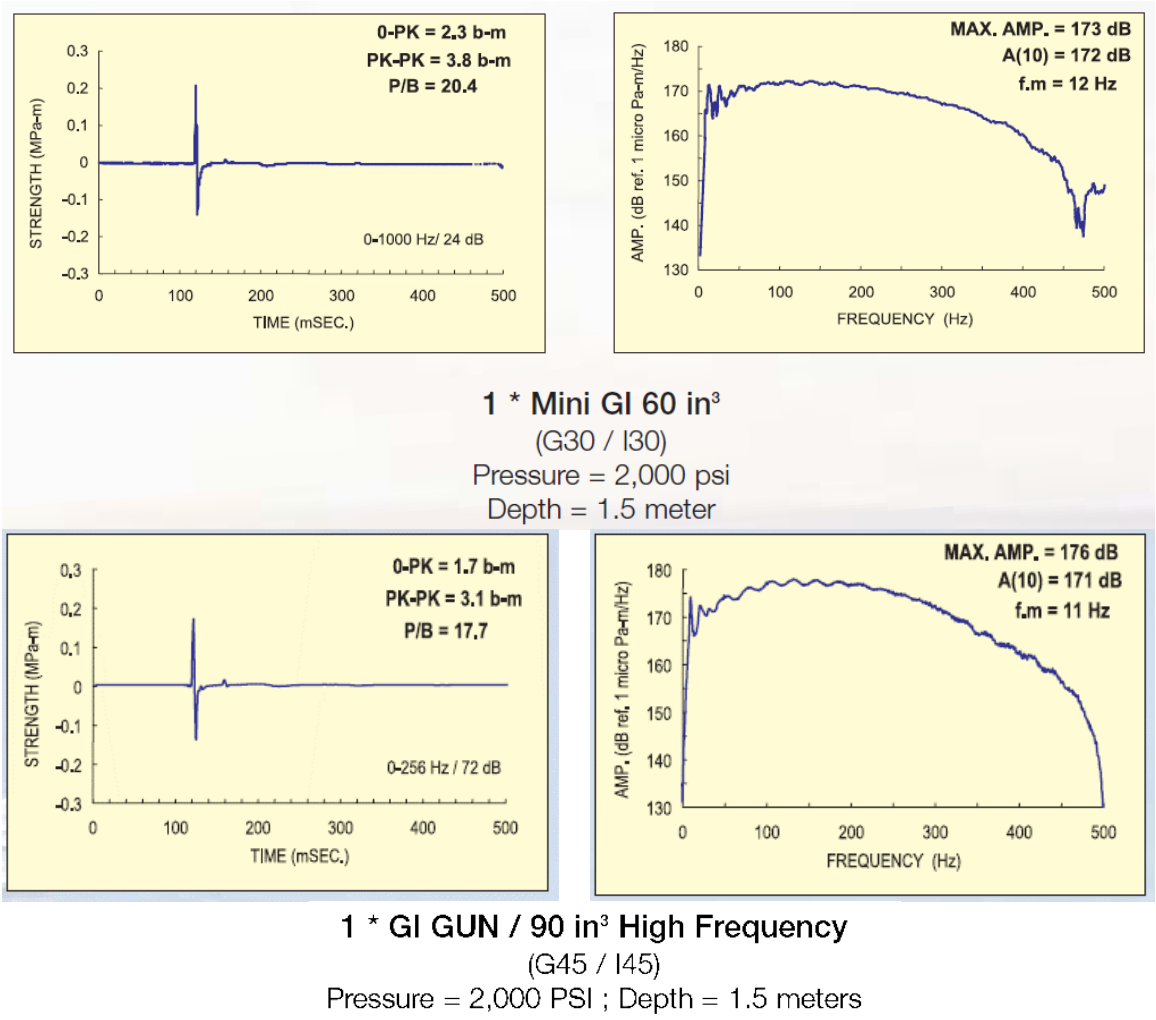


Figure 12 – Comparison of GI gun and mini-GI gun far field signatures and spectra from manufacturer technical docs (Sercel, 2006).

During the CAGE-16-6 survey, both GI and mini-GI guns were on board the R/V Helmer Hanssen. The mini-GI and GI systems produce different frequency spectrums with the mini-GI guns producing a spectrum with more power in the very high frequencies (see Figure 12), particularly when used in $15in^3/15in^3$ mode (not shown in the figure). Mini-GI guns also require less air so a more rapid rate of firing is possible with a given compressor. However, since the mini-GI guns were not firing reliably the GI guns were used for the acquisition of the Svyatogor 3D dataset.

During the survey, the GI airguns were run in harmonic mode (see Figure 11). This setup provides an attractive compromise between maximising peak-peak amplitude while minimising secondary bubble pulses and permitting a high firing rate. Plastic volume reducers were used in the GI guns to allow the $45in^3/45in^3$ high frequency shooting mode. With this setup, a firing rate of 5-6sec was possible at 160 bar pressure (2320 PSI) with the air compressor aboard the Helmer Hanssen. If run in true GI mode ($45in^3/105in^3$) the maximum firing rate would have been limited to around 10 seconds by the capacity of the air compressor. This setup produces a source wavelet that is approximately minimum phase.

3 Seismic data processing

3.1 Introduction

Seismic data always consists of a signal component and a noise component (Elboth, et al., 2009). The primary objective in processing reflection seismic data is to enhance the signal-to-noise ratio while preserving the useful signal bandwidth at all stages in the analysis (Yilmaz & Doherty, 2001). An ideal seismic processing flow should minimize amplitude distortions while largely attenuating reverberations, multiples, random noise, produce a wavelet of known phase (Brown, 2011) and ultimately increase vertical and lateral resolution of the data (Yilmaz & Doherty, 2001). Stated in practical terms, the goal of seismic processing is to improve the interpretability of the data by providing the best possible imaging of subsurface geology i.e. by providing phase and amplitude measurements that represent as closely as possible the spatial variations in subsurface reflectivity (Brown, 2011). For example, improving the signal to noise ratio of the data can improve the apparent continuity of reflectors and reveal structural features and weak reflectors otherwise obscured by noise. In addition, ideal seismic processing will place reflectors in their true subsurface positions (though typically scaled vertically in the time domain) through good control over survey geometry and correct migration.

It is important that the seismic processing flow is carefully applied and appropriately tuned to the individual characteristics of the dataset so that the maximum amount of noise is attenuated, while minimizing the attenuation of reflections attributable to real geology and ensuring that artefacts are not introduced into the data.

This chapter begins with an overview of the complete seismic processing sequence followed by sections that go into further detail on the theory and motivation, parameter selection and before/after comparative results for each of the key processing stages in the same order as they are applied in the processing flow.

3.2 Outline of seismic processing flow

The processing of the Svyatogor 2016 3D seismic dataset was conducted using RadexPro 2016.3. The following is a list of the key steps in the complete processing flow, followed by a simplified brute-stack processing flow that was used to test the effect of adjusting the geometry assigned to the receivers (see section 3.5.3 pg.23). A flowchart that provides an overview of the complete processing flow is shown in Figure 13.

Complete processing flow

1. SEG-D input
2. Geometry assignment – RadexPro P-Cable Geometry module
3. Basic filtering e.g. bandpass and burst noise removal on specific channels
4. Direct wave arrival – theoretical (based on geometry) and observed (autopicked)
5. Geometry export
6. Adjustment of geometry using Python script (Appendix 1)
7. Reimport the adjusted geometry
8. Import tides

9. Merge sail lines to single dataset (and enumerate TRACENO), recalculate offset with adjusted geometry, recalculate and re-pick direct wave arrival, apply tide correction
10. Wavelet extraction and creation of bubble suppression filter
11. Broadband processing routines – debubble, denoise, deghost, predictive deconvolution
12. Pick seafloor
13. CDP binning - fold maps generated with python script (Appendix 2)
14. Calculate residual statics
15. Apply statics
16. Offset binning - fold maps generated with python script (Appendix 2)
17. 3D Regularization
18. Pre-stack migration (Kirchhoff Time Migration)
19. F-X-Y filtering
20. SEG-Y Output

Simple brute stack processing flow

1. SEG-D input
2. Geometry assignment – RadexPro P-Cable Geometry module
3. Basic filtering e.g. bandpass and burst noise removal on specific channels
4. *Direct wave arrival – theoretical (based on geometry) and observed (autopicked)
5. *Geometry export
6. *Adjustment of geometry using Python scripts
7. *Reimport the adjusted geometry
8. Merge sail lines to single dataset (and enumerate TRACENO), recalculate offset with adjusted geometry, recalculate and re-pick direct wave arrival.
9. CDP binning to 6.25mx6.25m grid
10. NMO correction (constant velocity of 1479m/s)
11. Stacking
12. Profile interpolation
13. SEG-Y Output

*one dataset was prepared including these steps and one was prepared without.

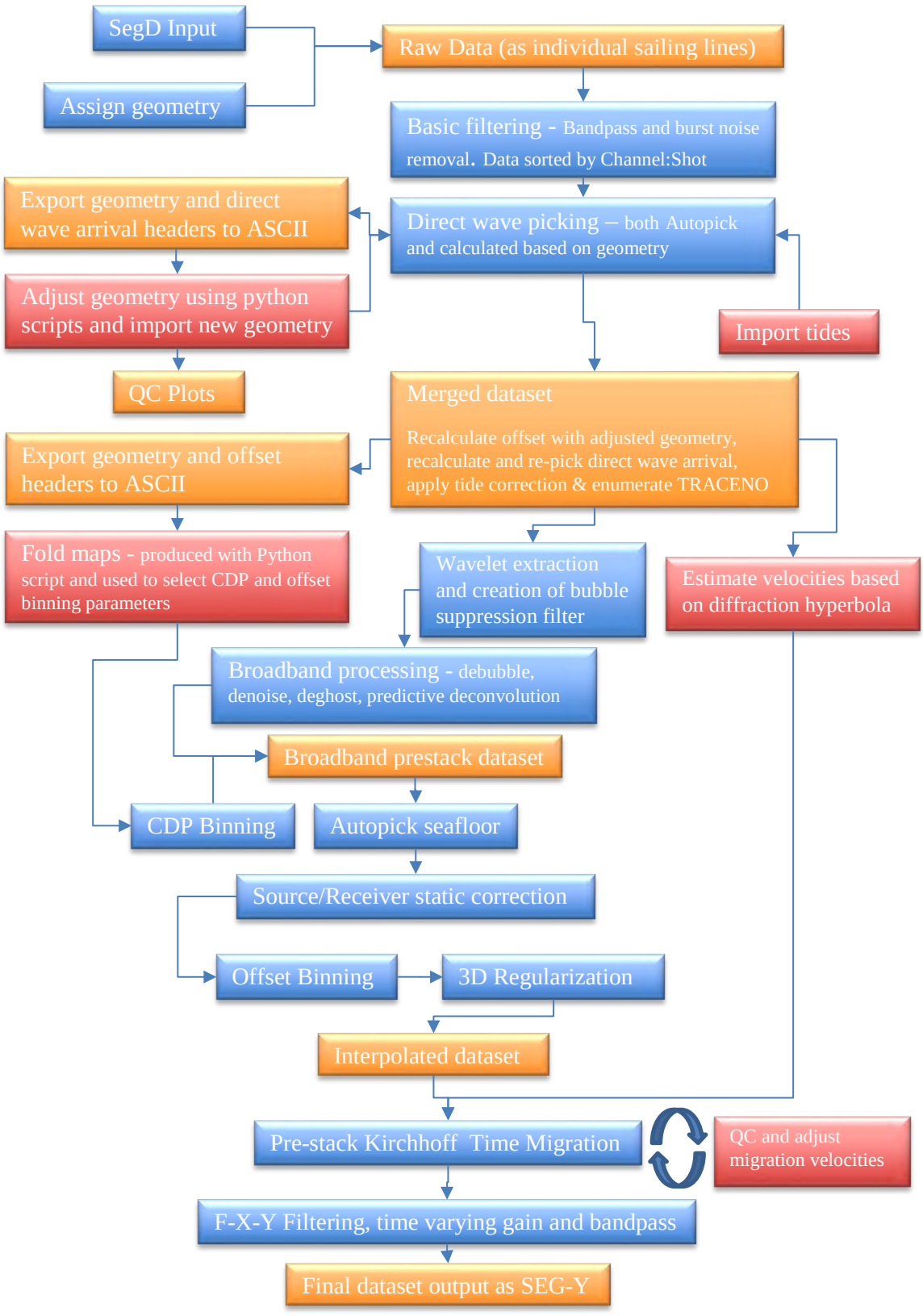


Figure 13 – Overview of Svyatogor 2016 3D processing flow, implemented with RadexPro 2016.3

3.3 Bandpass filtering to remove swell noise

A common source of noise in marine seismic data is caused by hydrostatic pressure variations produced by the changing height of the water column over the streamers due to ocean swells and streamer buckling. Typical ocean swells have wavelengths of around 50-100m and frequencies well below 1Hz. Such waves cause very large amplitude, low frequency noise on seismic data (Elboth, et al., 2009). In addition, another class of noise that manifests as vertical stripes on seismic data is a result of cross flow over the streamers that can be caused by ocean swell or currents (Elboth, et al., 2009). If the angle between the flow direction and the streamer is >15° the boundary layer around the streamer will become asymmetrical. This can lead to vortex shedding, which is an unsteady flow condition that creates strong alternating pressure fluctuations that are observed as high-amplitude noise in seismic sections (Elboth, et al., 2009). During the acquisition of the Svyatogor 2016 3D survey the presence of strong ocean currents from the north was noted (see section 2.2 pg.8) and may have contributed to the strong low-frequency noise that dominates in the unprocessed seismic sections (see Figure 14).

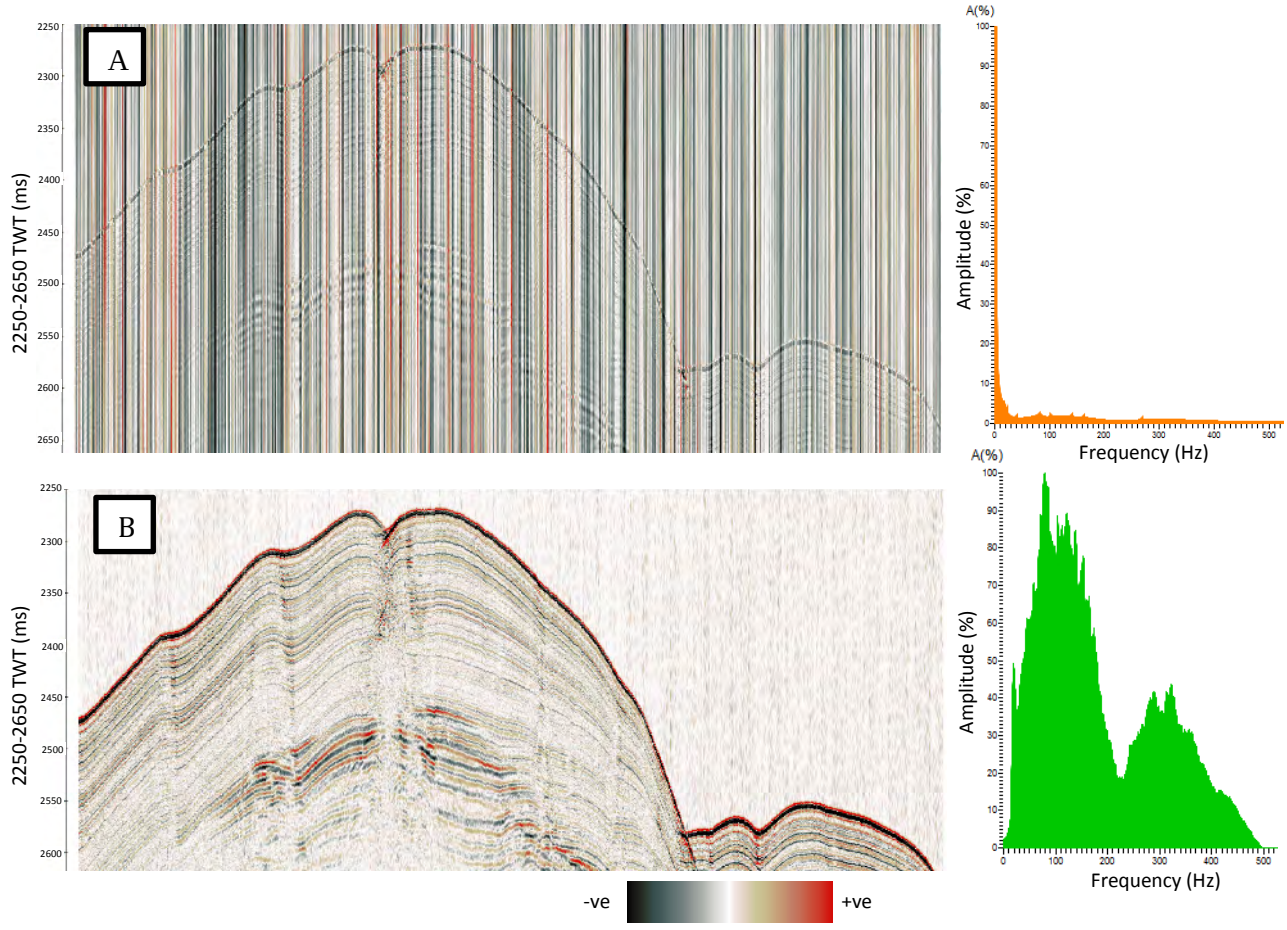


Figure 14 – Comparison of channel gather (sailing line 01, channel 27) (a) unprocessed raw data record and (b) after simple bandpass filtering with 15/20/450/500Hz (low-cut/low-pass/high-pass/high-cut). The raw record is dominated by low-frequency swell noise.

The swell noise was removed using bandpass filtering with low-cut/low-pass/high-pass/high-cut thresholds of 15/20/450/500Hz. RadexPro's "simple bandpass filter" (a trapeziform bandpass filter)

was selected to maintain the phase of the data (a zero-phase Ormsby filter should be avoided for minimum phase data because it would produce a mixed phase result).

3.4 Burst noise removal

During acquisition of the Svyatogor 2016 3D dataset it was observed that some channels would record high amplitude transient noise, likely caused by electrical interference due to streamer leakage, during some, but not necessarily all, survey lines. This class of noise was severe enough that either it must be removed by filtering or the channels would need to be completely disabled or discarded. The "burst noise removal" tool in RadexPro was found to be an efficient means of removing this kind of noise and once the filtering was applied a significant amount of usable reflection data was revealed. The tool calculates an average value of absolute amplitude, to which the high amplitude noise spikes are compared and removed if they exceed the average by more than a specified factor. The performance of this process was enhanced by running the noisy channel through the burst removal tool with two adjacent clean channels so that a reasonable average amplitude value could be found (if run only on the noisy channel the average amplitude value was biased by the noise, reducing the performance of the filtering). For the Svyatogor 3D dataset the best filter performance was achieved using the parameters in Table 2.

Table 2 - Parameters used in RadexPro burst noise removal module

Window size for average value calculation (traces)	11
Rejection percentage (%)	50
Do not change amplitudes lower than (%) of the average	60
Modify values when exceed average by more than N times	2.5

The performance of the burst removal tool was evaluated by assessing its effect on a noisy channel (Figure 16) compared to its effect on a clean channel (Figure 17). The filter performance was very good; it removes the vast majority of the noise while retaining a significant amount of useful signal when applied to a noisy channel and has minimal effect when applied to a clean channel. However, some attenuation of the direct wave occurred when applied to the clean channel and there was some modification of the amplitudes of strong reflectors e.g. the water bottom. As a result, a workflow was adopted where the burst noise filtered noisy channels were merged with clean channels that were not subjected to burst noise filtering (see Figure 15).

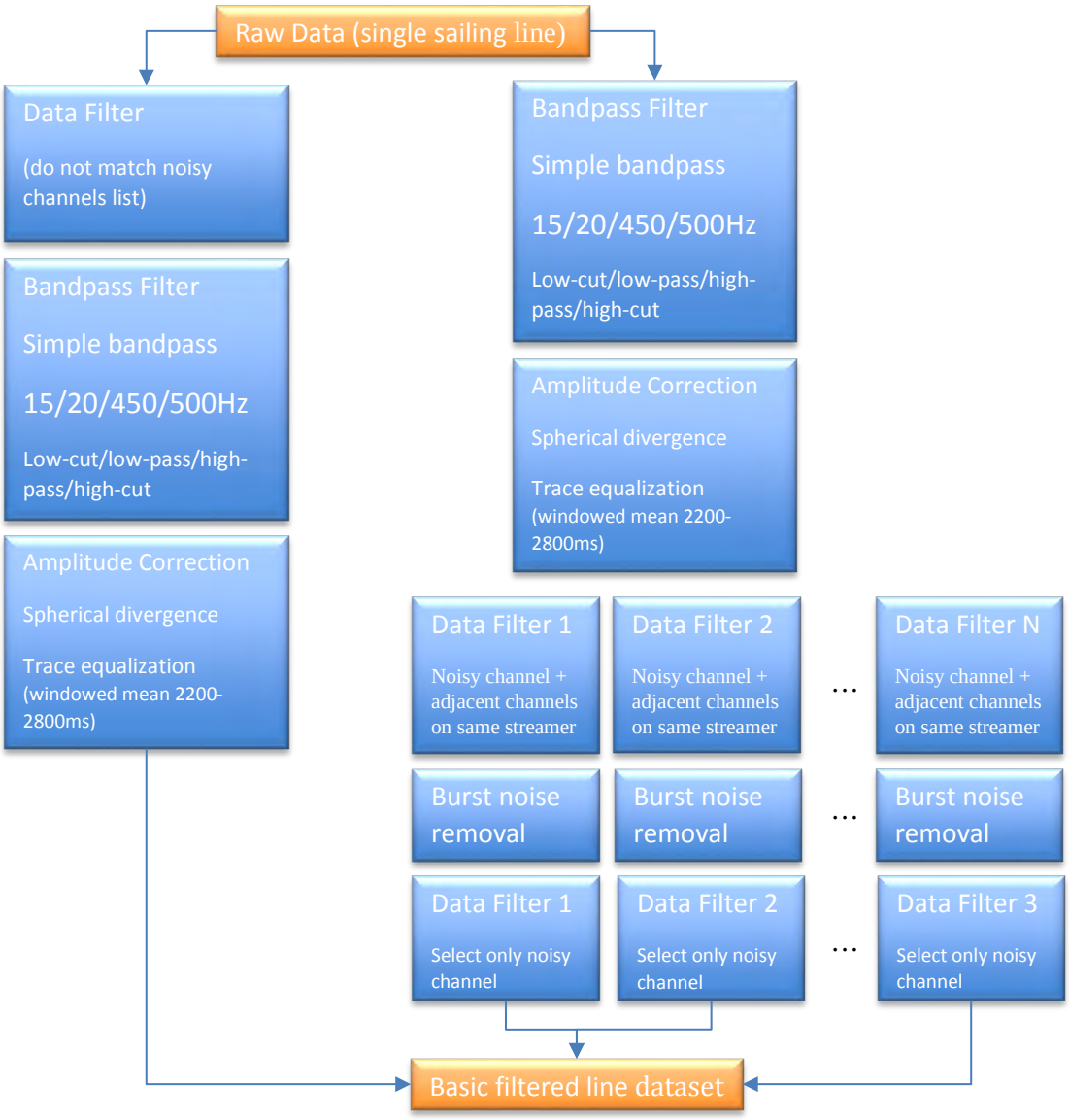


Figure 15 – Flowchart illustrating workflow to apply basic filtering procedure to raw sailing lines.

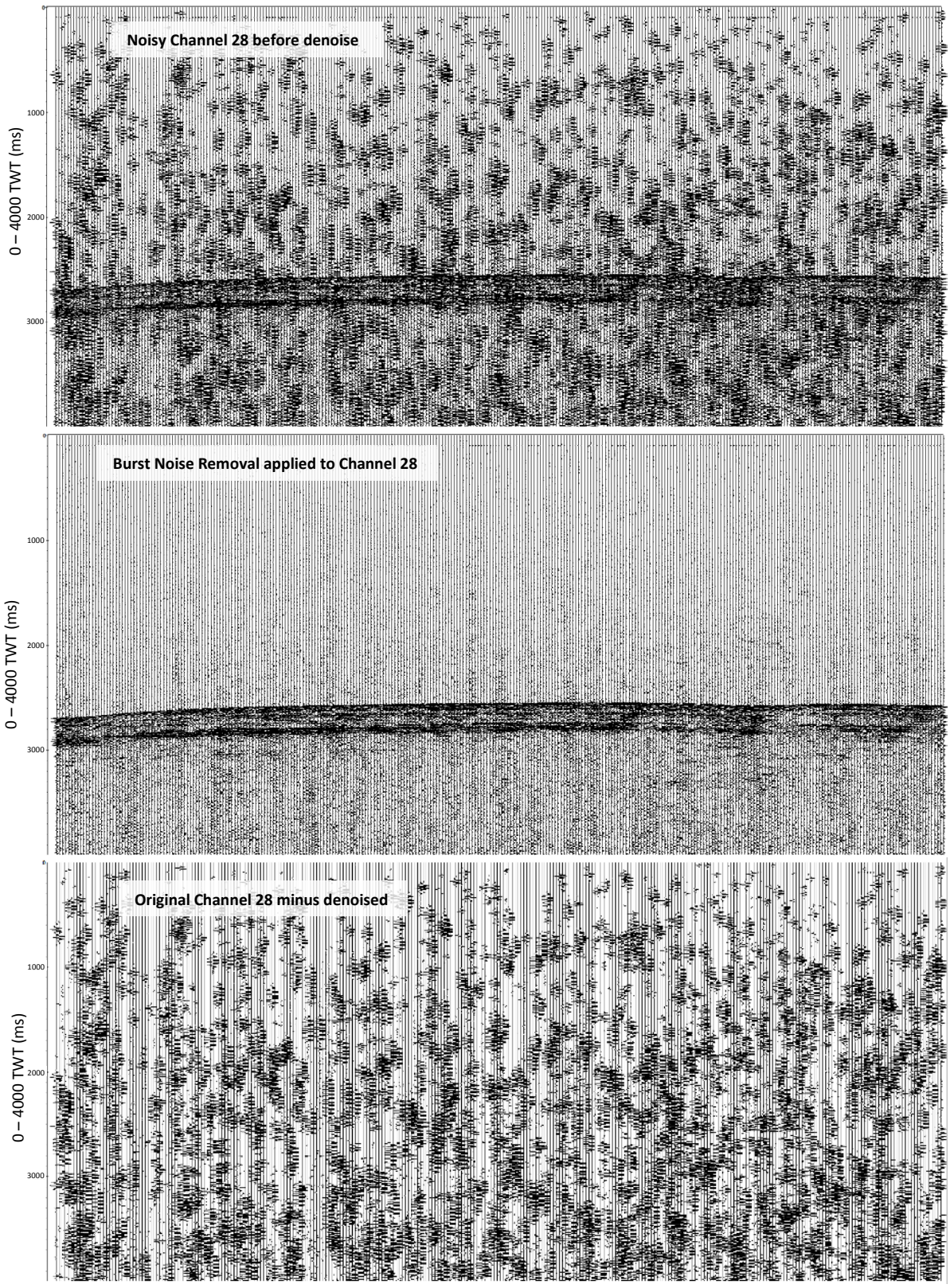


Figure 16 – Effect of burst noise removal on noisy channel 28. The filter does a good job of separating the signal from the noise

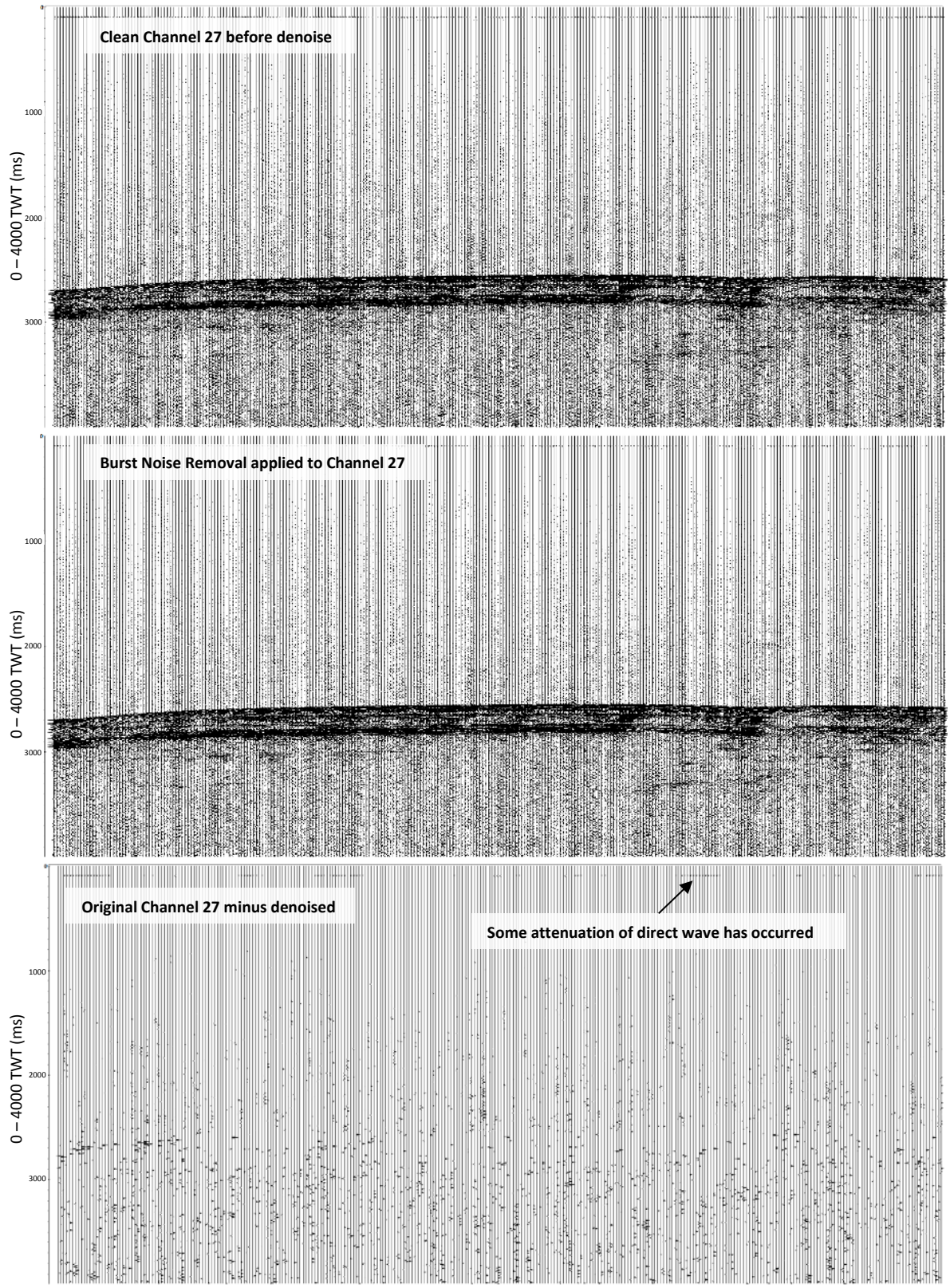


Figure 17 – Effect of burst noise removal on a clean channel (channel 27), which does not contain high amplitude noise spikes. The filter has a minimal effect on clean channels, although some attenuation of the direct wave occurs.

3.5 Geometry

3.5.1 Assigning Geometry

In seismic processing, the assignment of geometry concerns the specification of the surface positions of source and receiver corresponding to each recorded trace and accordingly defines the separation and azimuth between source and receiver (Yilmaz & Doherty, 2001). It is a very important step because it underpins all subsequent processes (Brown, 2011) and ultimately influences the extent to which the seismic section or volume represents the true subsurface reflectivity of the surveyed area. The accurate positioning of source and receivers in both vertical and horizontal planes has been described as the greatest limitation to high-resolution multi-channel 3D surveying (Mosher & Simpkin, 1999). Imperfect positioning of the recorded traces could lead to them falling into incorrect CDP bins potentially leading to smearing of structural features or decreased reflector continuity.

The Svyatogor 2016 3D survey was acquired with a P-Cable high resolution seismic system with the configuration shown in Figure 7. With this configuration, the position of the gun/source and the two paravanes are logged with differential GPS giving positional accuracy of <1m but the positions of the streamers and receivers must be calculated. The seismic processing software RadexPro contains a dedicated module for assigning P-Cable receiver positions based on a catenary model of the cross cable that is constrained by the positions of the paravanes and the known length of the cross cable. The fitted catenary defines the positions of the streamer junctions, which are placed according to their measured positions along the cross cable and the receivers are placed accordingly back along the streamers whose direction follows the trend of the ship heading.

To assign the initial geometry it was necessary to interpolate and QC the SeaTrack navigation files recorded by the survey vessel and then convert them to a format that could be more readily loaded into RadexPro. This was achieved using the Python script in Appendix 3.

3.5.2 Geometry QC

The source-receiver offsets defined by the assigned geometry are used to calculate the theoretical arrival of the direct wave, which travels through the water between the source and receiver in an approximately straight line at seawater velocity. A velocity of 1479m/s was used for this survey and corresponded to CTD profile measurements of sound velocity. These theoretical arrival times were then compared to the observed direct wave arrival times derived via wavelet autopicking to assess whether the assigned geometry fits the recorded data (see Figure 18).

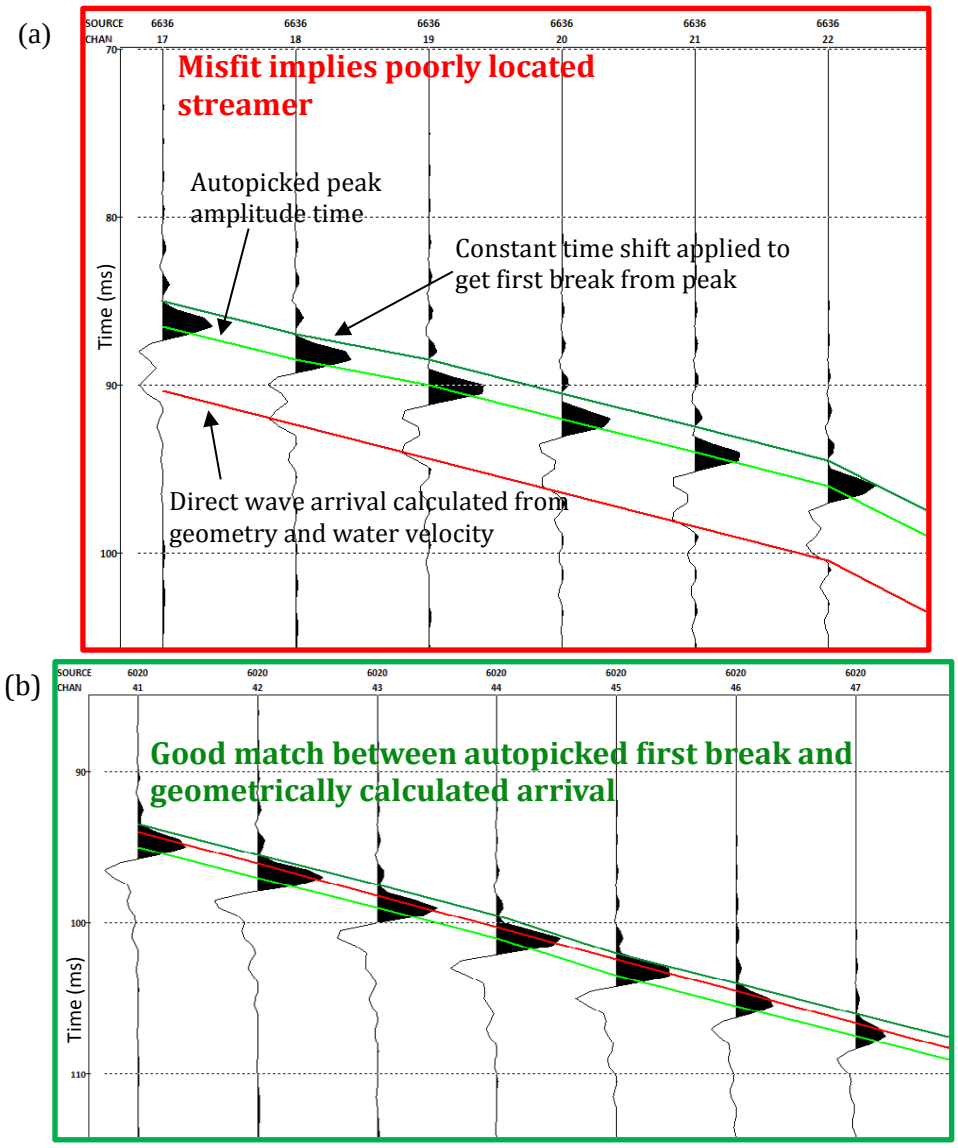


Figure 18 – examples of geometry QC showing (a) misfit between theoretical direct wave arrival (based on assigned geometry) and observed (autopicked) arrival implying that the streamer is poorly positioned and (b) good match between theoretical and observed direct wave arrivals implying that the streamer position is consistent with the recorded data.

One possible approach to the problem of geometry that does not fit the observed data would be to discard all traces where the misfit between observed and modelled direct wave arrivals exceeds a certain threshold (illustrated in Figure 19 using a threshold of 5ms). However, this leads to a significant reduction in fold and creates gaps in coverage since the outermost streamers consistently show the largest misfit (see Figure 19).

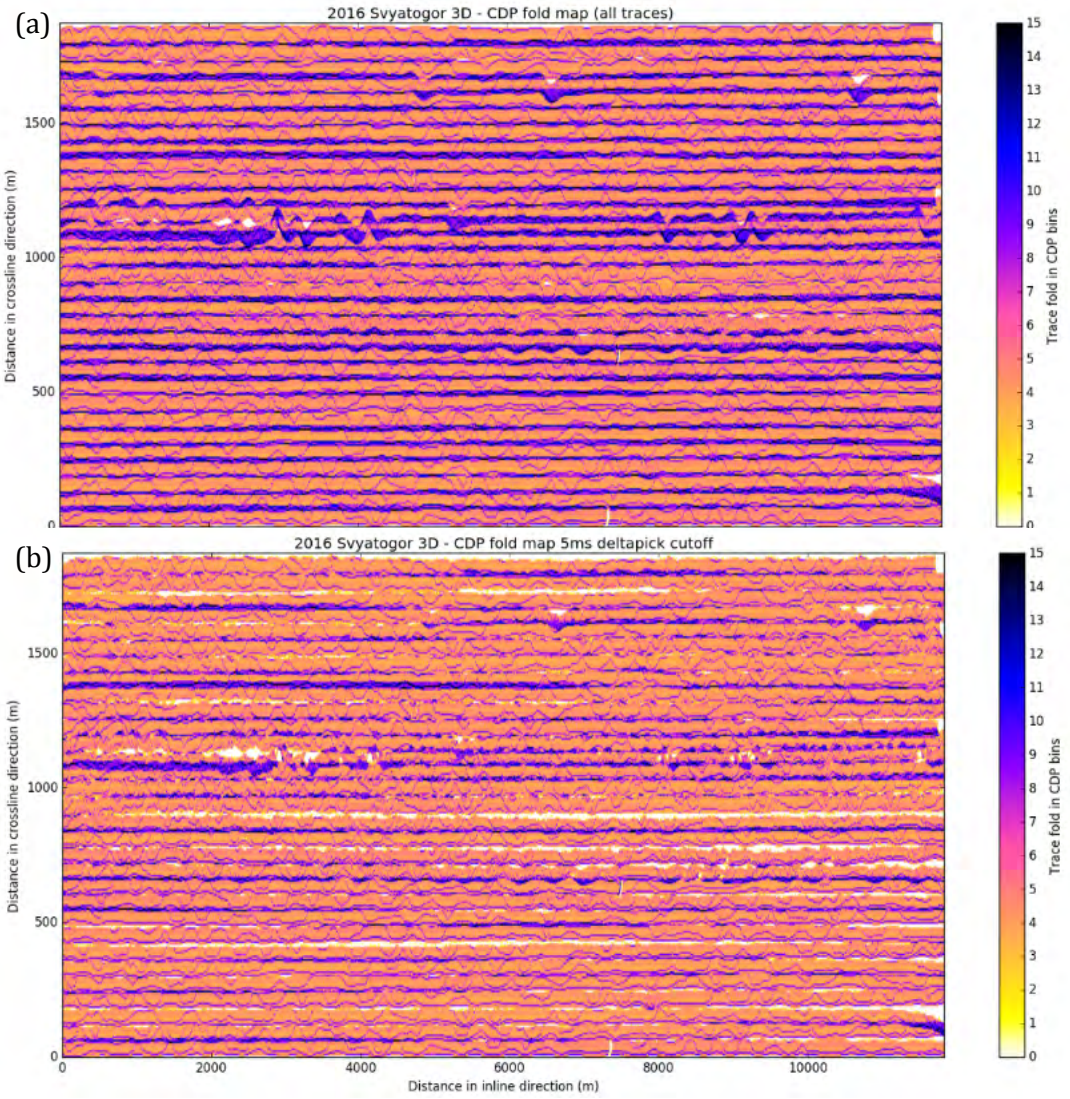


Figure 19 – comparison of CDP fold (a) all recorded traces and (b) traces where difference between observed and modelled direct wave arrival is <5ms. The fold is significantly reduced and gaps in coverage are created when the traces with large misfit are discarded.

3.5.3 Improving the geometry fit

It is useful to consider the spatial distance that the differences between observed (autopicked) and modelled (based on assigned geometry) direct wave arrivals represent. This distance can be considered the Δoffset:

$$\Delta offset = V_{seawater} \times (T_{observed\ direct\ wave} - T_{observed\ direct\ wave})$$

The example of misfitting geometry illustrated in Figure 18 shows that the direct wave arrival modelled based on the assigned geometry is coming in later than the observed direct wave. This implies that the streamer needs to be moved closer to the source to shift the modelled direct wave to an earlier arrival time (assuming the water velocity is correct). It follows that a general approach to improve the fit of the assigned geometry with the data is to use the calculated Δoffsets to reposition the receivers closer to their true locations. Importantly, since the Δoffset only provides a 1D constraint on the 2D receiver positioning problem (variation in receiver depth was ignored) it was necessary to

approach the problem in a least squares inversion sense using additional known geometrical parameters (e.g. cross cable length) as constraints. This approach is also relatively insensitive to mispicks of the direct wave arrival by the autopicking routine because the receivers are always considered as a group rather than individually. This approach has similarities with that described in Petersen et. al. (2010) who used source-receiver offsets derived from direct wave arrivals to update an initial triangular streamer geometry of a P-Cable system. However, one key difference is that the current approach aims to fit the geometry with all receiver channels whereas Petersen et. al. (2010) simply take the first channel for each streamer as constraints.

This solution was implemented by exporting navigation, offset and autopick headers to ascii files, which were then processed using a Python script (Appendix 1). The following sections provide an overview of the important steps and processes implemented in the Python scripts and designed to improve the receiver geometry.

3.5.3.1 Transformation to a new coordinate system

A key step that greatly simplifies the subsequent calculations is to define a transformation to a new coordinate system where the y-axis is aligned with the ships heading. Several methods of calculating the ship heading were trialled, including 1) taking the perpendicular to a line bisecting the two paravanes and 2) taking a linear trend through 2-3 successive shot points i.e. using gun position trend. However, the best performance was achieved simply by extracting the ships heading already assigned by RadexPro (which is based on an alpha trimmed average of the ship position) by fitting a linear trend through the one of the central streamers (RadexPro includes an option to make the streamers follow the ship track). The x-axis is arbitrarily placed at the average y-coordinate of the same central streamer used to calculate the heading. A rotation and translation is applied to transform the survey geometry to the new coordinate system as shown in Figure 20.

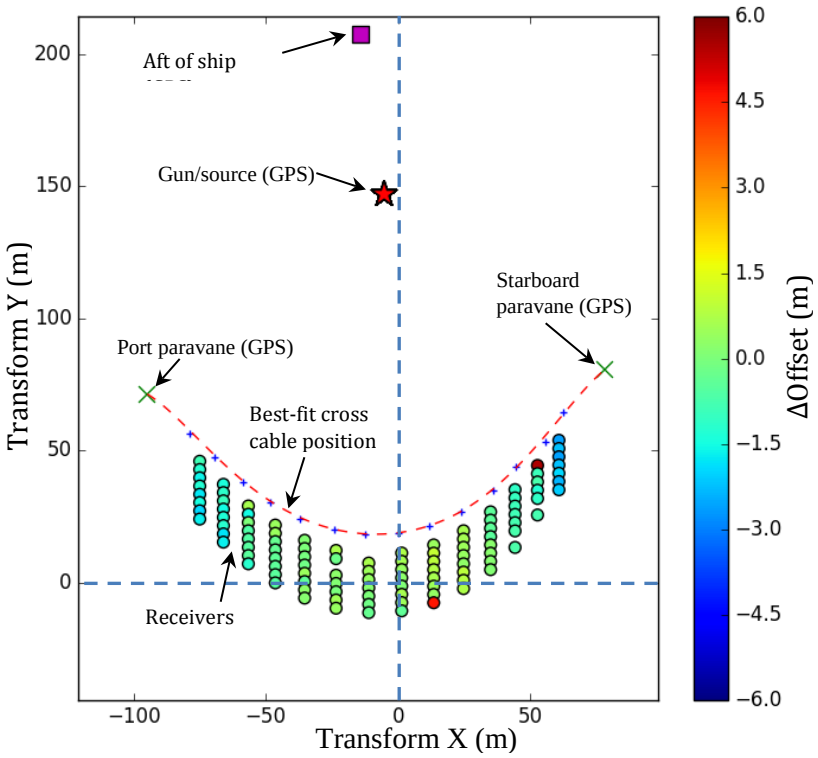


Figure 20 – An example of shot geometry in transformed coordinate system with Y-axis aligned with the ships heading

3.5.3.2 Locating the cross cable

The Δoffset values derived from autopicking the direct wave were used to invert for the position of the cross cable. For each receiver, we know that the true position of the cross cable should be at some radius corresponding to the distance of the receiver along the streamer plus the error in source-receiver offset corresponding to the initially assigned geometry i.e. the Δoffset. Therefore, for each streamer we wish to find the point that lies at the correct radius from all of the receivers on that streamer (this should be the true position of the streamer junction). Graphically, this is a simple problem corresponding to finding the convergence of sets of circles as shown in Figure 21. Mathematically, this problem was solved using non-linear least squares optimization to solve the following function for each streamer:

$$min \sum_{i=1}^{n} (x - x_{receiver})^2 + (y - y_{receiver})^2 - r^2$$

Where *i* is the streamer index, *n* is the number of receivers on the streamer and *r* is the distance of the receiver from the streamer junction (see Figure 7) plus the Δoffset.

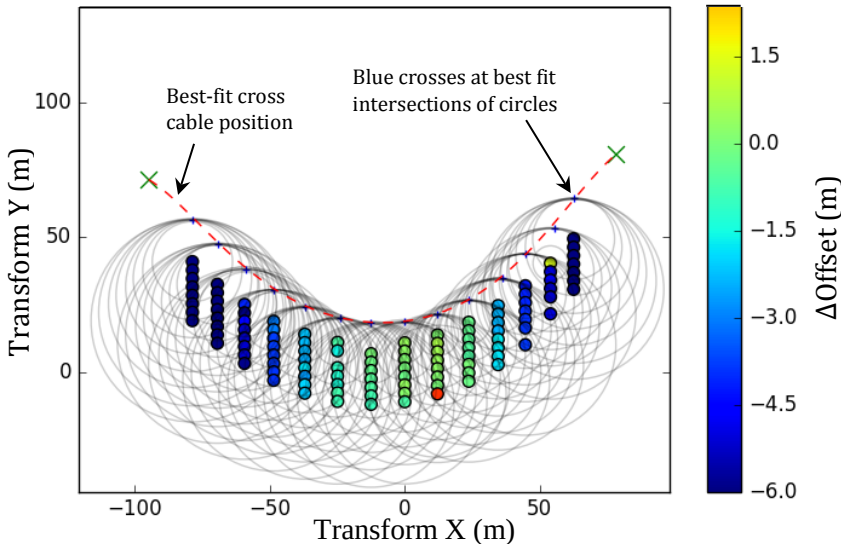


Figure 21 – example of locating streamer junctions using non-linear least squares optimisation to find the intersections of circles drawn with radius of receiver distance along streamer plus initial Δoffset.

3.5.3.3 Fitting a curve to model the cross cable

Once the streamer junctions are located a model cross cable is derived by fitting a curve through the streamer junction and paravane positions. The cross cable is expected to follow the geometry of a catenary because the primary force acting on it is drag in the direction of travel as it is pulled through the water by the ship (although some distortion from an ideal catenary would be expected if cross track water currents exerted an appreciable force on the cross cable). A pure catenary has the following form:

$$y(x) = a\cosh\left(\frac{x}{a}\right)$$

We can use Taylor series expansion of the hyperbolic cosine i.e. $\cosh(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \cdots$ to derive the following expansion for a catenary with vertex at (x_0, y_0):

$$y = y_0 + \frac{(x - x_0)^2}{2a} + \frac{(x - x_0)^4}{24a^3} + \cdots$$

Keeping the first 3 terms and expanding yields a 4th degree polynomial (4 terms would give a 6th degree polynomial etc) of the following form:

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4$$

Consequently, it was possible to define a realistic model for the cross cable by simply fitting a 4th degree polynomial through the streamer junction and paravane positions. However, improved performance was achieved by analysing the standard error associated with the polynomial and if it exceeded a certain threshold i.e. the polynomial fit was poor, then a 6th degree polynomial was attempted and retained if it gave a better fit.

However, we also have some further geometrical knowledge about the cross cable that we can use to constrain the curve fitting i.e. its total length, which has been measured. As a result, the curve fitting by least squares optimisation includes a penalisation factor so that the fitted curve will be constrained towards keeping the correct total cross-cable length. If y=f(x) is the polynomial approximation of the cross cable then its length can be calculated as:

$$curve\ length = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$

Finally, the performance of the curve fitting was also substantially improved by using weighting factors, whereby the curve fitting is weighted to favour the middle streamers Figure 22. The geometrical basis for this weighting is demonstrated by considering the convergence envelope i.e. the region where the circles denoting the range of the cross cable from each receiver approach one another (see Figure 22). These envelopes can be considered uncertainty ranges in the positioning of the streamer junctions. The convergence envelopes of the central streamers lie roughly parallel to the fitted cross cable curve so the shape of the curve would not change significantly if the streamer junctions were shifted slightly along the envelope of convergence. Conversely, convergence envelopes of the outer streamers intersect the fitted cross cable more obliquely, so a shift of the streamer junction position along the envelope would lead to a noticeable change in the shape of the curve. The central streamers thus represent a more reliable control on the position of the cross cable than the outer streamers and are weighted higher accordingly (see Figure 22). The paravane positions are also highly weighted as they have GPS control so are expected to be reliable (although there is some uncertainty in the precise offset of the GPS to the cross-cable connection; (see Figure 7).

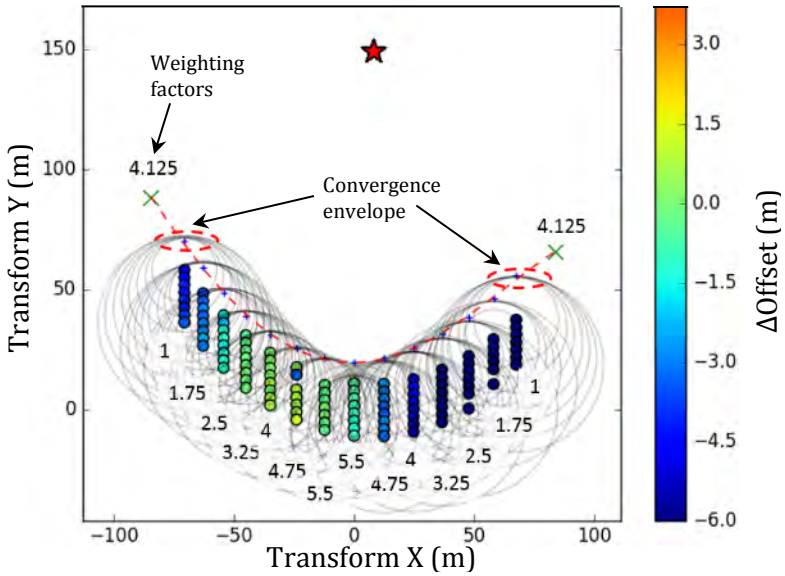


Figure 22 – example illustrating weighting factors used during polynomial curve fitting to bias the fitting towards the central streamers and paravanes. The convergence envelope can be thought of as a range of uncertainty in the streamer junction positions and is most significant for the outer streamers because it intersects the fitted curve obliquely.

3.5.3.4 Integrating along the curve to locate streamer junctions

The fitted curve represents the least-squares best fit model of the cross-cable subject to all available constraints: paravane GPS positions, range from receivers consistent with direct wave arrivals and total length of cross cable. From this model, the final positions of the streamer junctions are calculated by integrating the known distances along the cross cable i.e. distance from starboard paravane to first streamer, distance to next streamer etc. using the arc length equation:

$$s = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$

Where 's' is the length of each section. In this case, the equation is solved numerically to find the value of 'b' that satisfies the expression. Using this method, the streamers are always placed the correct distance apart from one another along the cross cable (see Figure 23) and the sensitivity to direct wave mispicks on a single streamer are further reduced as they are effectively averaged across the entire array.

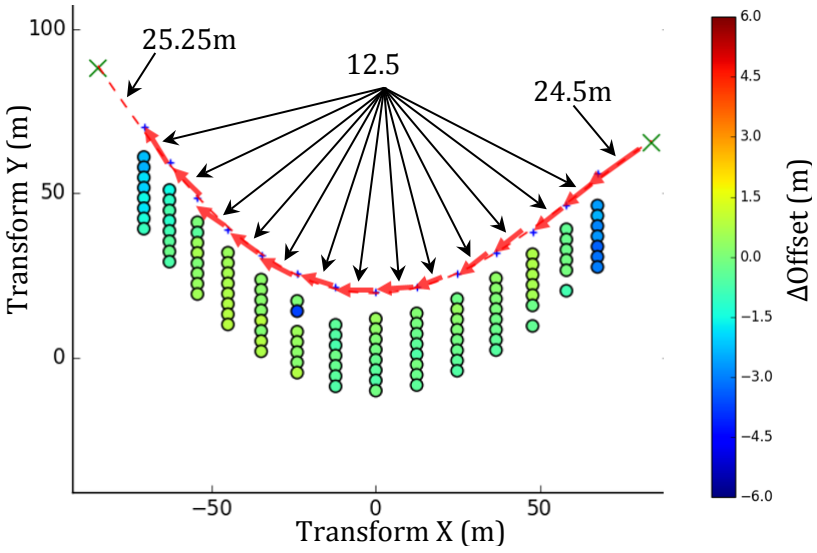


Figure 23 – Example illustrating the final placement of the streamer junctions determined by integrating the specified distances along the fitted curve representing the cross cable.

3.5.3.5 Positioning receivers and applying inverse coordinate transformation

The final position of the receivers is determined by simply shifting them along the y-axis (which is aligned with the ships heading) by their known position along the streamer (see Figure 7). For the Svyatogor 2016 3D survey, 14 streamers were used with 8 receiver channels per streamer so the distance between each receiver channel and the streamer junction is given by the series:

$$6 + 0.3 + 1.5625 + 3.125\times(n-1)$$

Where *n* is the positional index of the receiver channel i.e. the integer series from 1-8. This process is illustrated in Figure 24.

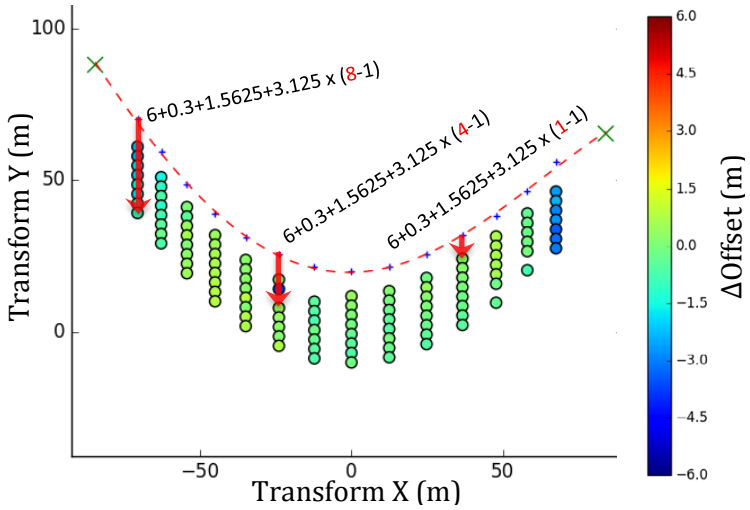


Figure 24 – illustration of the final positioning of receivers by shifting along the y-axis by their known position along the streamer (red numbers show the positional index of the indicated receivers).

Once the final positioning of the receivers has been achieved, the inverse of the rotation and translation coordinate transformation is applied to return to the original projected UTM coordinate system.

3.5.3.6 Assessing performance with QC metrics

The process of adjusting the receiver geometry guided by the autopicked direct wave arrivals produces geometries that show much greater consistency with the observed arrivals and remains a physically reasonable model that is consistent with known geometrical constraints such as length of cross cable and streamer spacing along the cross cable. The difference in geometry for a single shot from sailing line 31 is shown in Figure 25.

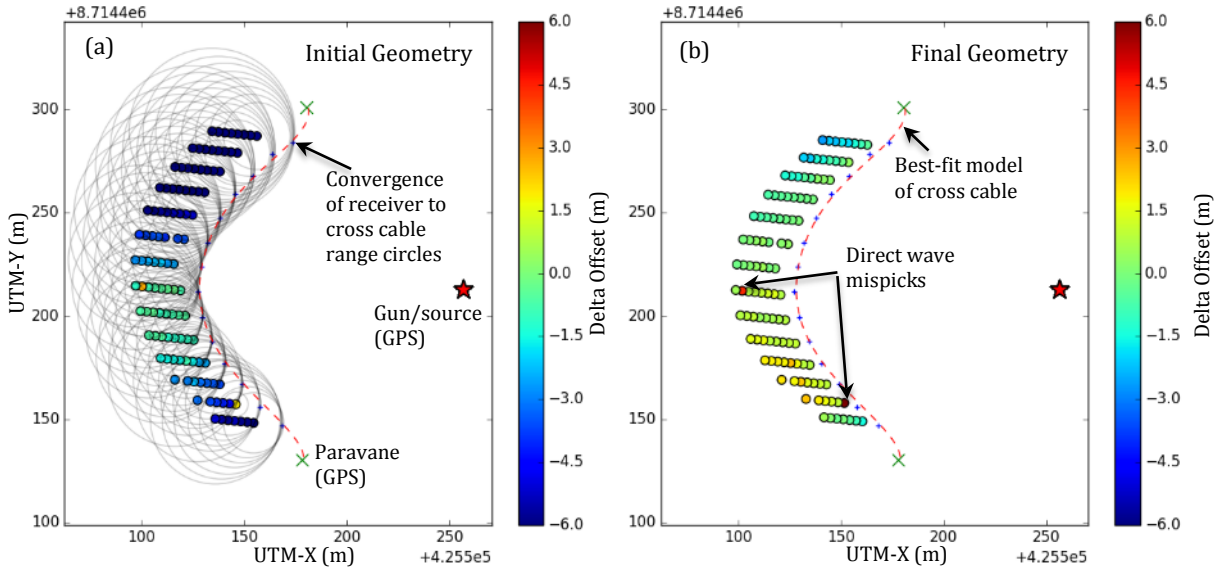


Figure 25 – Comparison of a typical receiver geometry from a single shot from sailing line 31 (a) before and (b) after geometry adjustment *(the modelled position of the cross cable is shown on both to aid comparison).The Δoffsets have been substantially reduced and indicate that the final geometry is more consistent with the observed direct wave arrivals. Several receivers show mispicks of the direct wave leading to anomalous Δoffset values but these do not have a substantial influence on the result.*

To more broadly assess the results of the geometry adjustment the total absolute misfit was calculated for each shot by summing the absolute values of Δoffset for all receivers:

$$Total\ absolute\ misfit = \sum_{i=1}^{n} |\Delta offset_i|$$

Where *i* denotes the active receivers for each shot. The total number of receivers (*n*) that were successfully recording for a given shot was typically 106 (because some channels were disabled). A typical example from sailing line 18 is shown in Figure 26 where the final geometry shows a dramatic improvement compared to the initial geometry (lower misfit). It was typical that the misfit was much higher at the beginning and/or ends of the sailing lines because recording was usually started before the ship reached the target survey area and had not fully straightened onto the line.

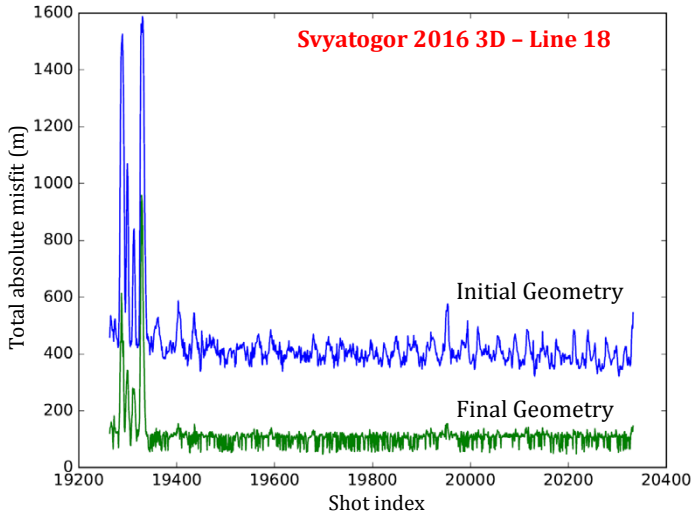


Figure 26 – comparison of geometry before and after adjustment for sailing line 18 showing a dramatic improvement in misfit. The start of the sailing line has high misfit because recording began before the turn was complete.

For QC purposes, it was also useful to plot histograms representing the delta offsets for all receivers and all shots on a given sailing line before and after the geometry adjustment (see Figure 27). It was typical amongst all of the sailing lines that the histogram representing the initial geometry showed a peak around zero corresponding to the central streamers, which generally showed acceptable fit. However, the initial geometry also typically produced histograms with a broad profile that were often skewed towards negative values as a result of the relatively poor fit observed for the outer streamers (possibly due to strong crossline currents; see Figure 8 & Figure 9). These problems have been resolved by the geometry adjustment as shown by the final geometry histogram, which is close to a gaussian distribution centered around zero Δoffset (see Figure 27). We must also recall that the shots at the start and end of the line are also included in these histograms and these shots are really part of the turns (not the desired sailing lines), so we would expect the distribution to be even tighter if these sections were excluded.

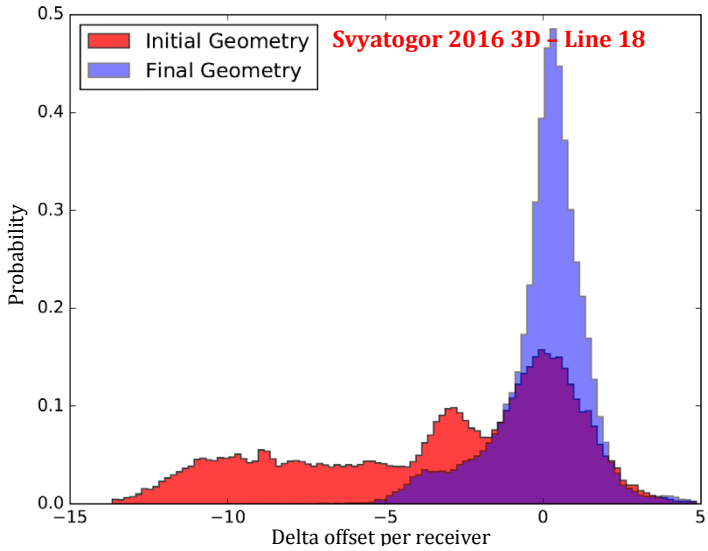


Figure 27 – histogram plot displaying the Δoffsets before and after geometry adjustment for sailing line 18. The trends shown in this example were similarly observed for the other sailing lines.

3.5.4 Before/after comparison of geometry

The motivation for adjusting the geometry is to avoid positional errors or uncertainties causing traces to be placed in the wrong CDP bins. By applying misfit thresholds, such that traces with misfit exceeding the threshold value are discarded, the improvement in the accuracy of the geometry was visualised using CDP fold maps (see Figure 28). At a given misfit threshold the CDP fold is substantially improved by the geometry adjustment. This translates into a decreased probability that traces will fall into incorrect CDP bins due to incorrect geometry, a situation that most affected traces belonging to the outermost streamers (visible as gaps in coverage on the fold maps). There was also some evidence that the process of adjusting the geometry was most effective for the first 15-16 sailing lines (the survey was acquired from the middle outwards) where the fold is maintained very well even with a threshold of 3ms allowed misfit between observed and modelled direct wave arrivals (see Figure 28). The reason for this inconsistency is unclear but it could hypothetically be related to a physical mechanism such as variable elastic properties of the cross cable after prolonged immersion leading to stretching.

In general, the qualitative differences between simple brute stack datasets using initial and final (adjusted) geometry were very subtle. On the other hand, trace-by-trace subtraction of the datasets showed that quantitative differences are extensive (Figure 30) but typically small in magnitude. Such differences could still be significant if the survey was intended for 4D processing, though this falls beyond the scope of the current project. Some improvement in reflector continuity was observed in the crossline direction along with a reduction in linear artefacts running parallel to the sailing direction, that were observed in variance time slices (Figure 29). An improvement in reflector continuity and signal to noise ratio was also observed on some inlines, typically around structurally complex areas and diffractions (Figure 30).

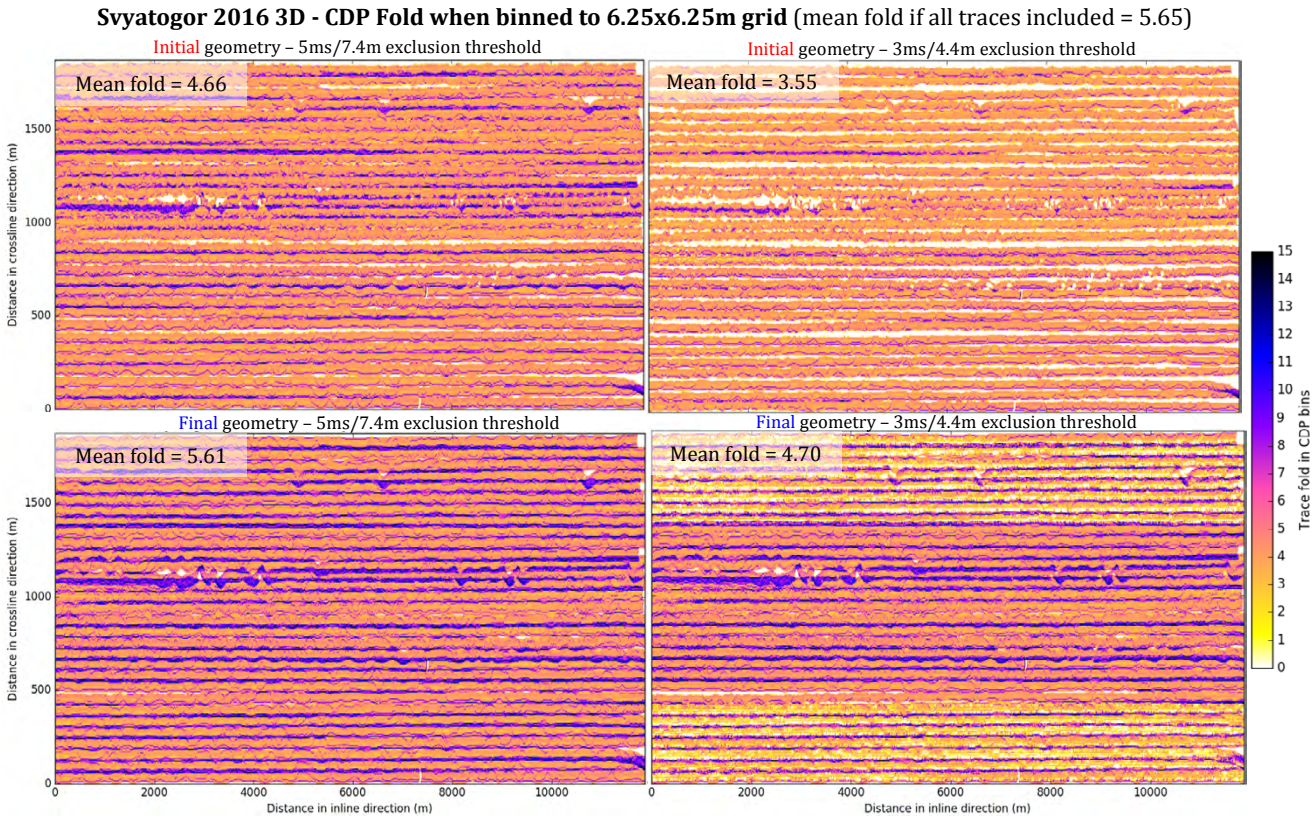


Figure 28 – Comparison of CDP fold when binned to 6.25x6.25m grids for the initial (top) and final/adjusted (bottom) geometries. Traces exceeding a misfit between calculated and picked direct wave arrivals of 5ms or 7.4m Δoffset (left) and 3ms or 4.4m Δoffset (right) have been excluded to highlight the differences.

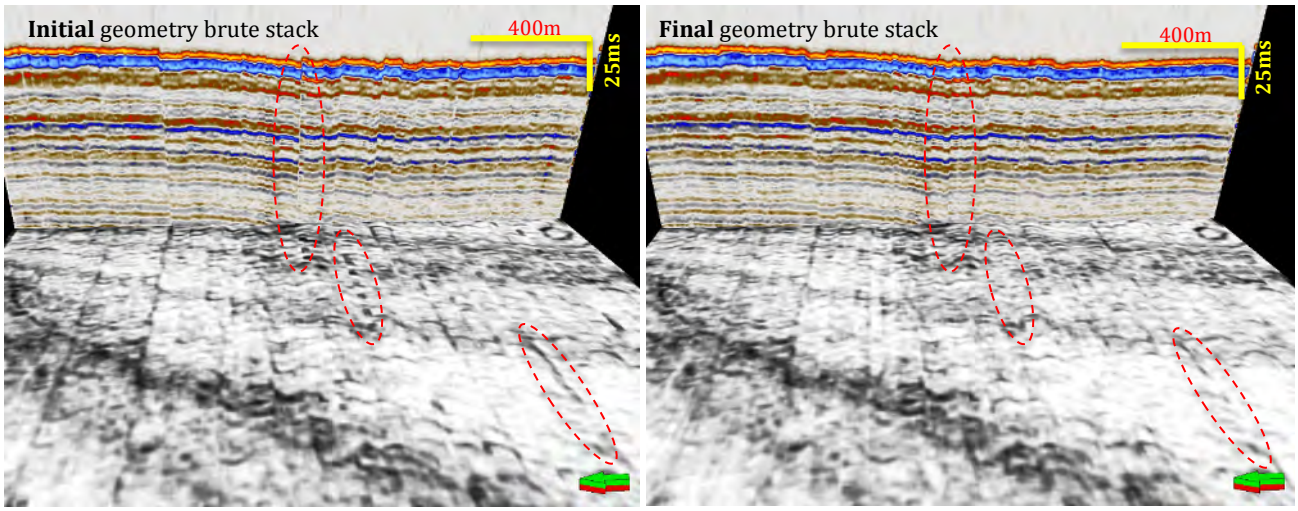


Figure 29 – crossline and variance time slice intersections comparing initial and final geometries (simple brute stack datasets). Some improvement in crossline reflector continuity was observed with the final geometry and the linear artefacts observed in variance slices were reduced (several examples have been circled in red).

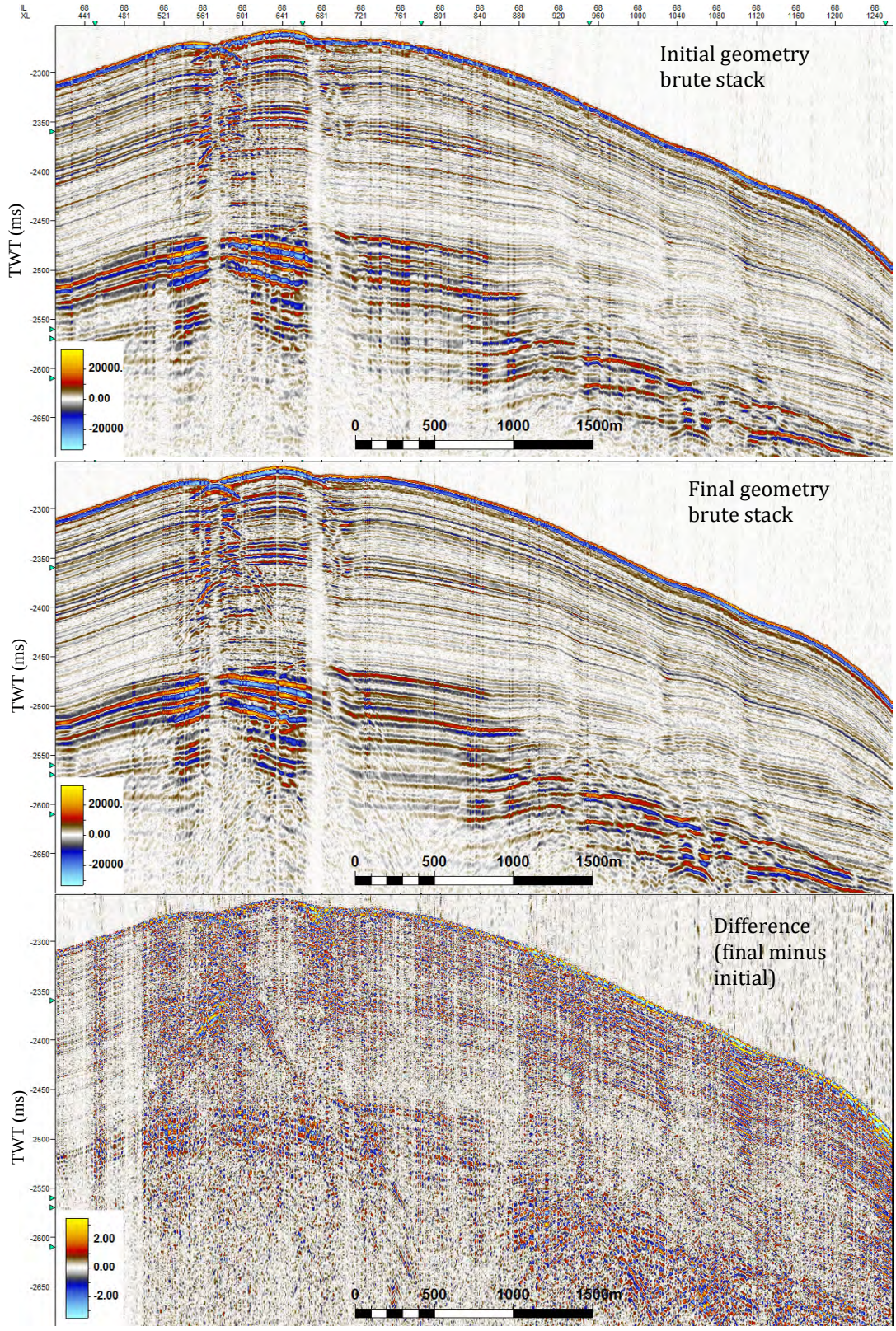


Figure 30 – Brute stack comparison initial geometry, final geometry and difference (final minus initial) for inline 68 of the Svyatogor 2016 3D dataset. The qualitative differences were typically subtle but some improvement in reflector continuity and signal to noise ratio was observed, particularly in structurally complex areas.

3.6 Broadband processing steps

For marine seismic data acquired using conventional flat streamers (as in the Svyatogor 2016 3D dataset), the bandwidth that can be achieved is strongly influenced by destructive interference of reflections from the sea surface (ghost waves). Traditional processing is only suited to recovering frequencies below the first ghost notch (Willis, et al., 2015). Broadband processing provides a cost-effective means (compared with broadband acquisition strategies like variable tow-depth streamers) to recover lost bandwidth, leading to enhanced seismic resolution and improved structural and stratigraphic imaging (Willis, et al., 2015).

The broadband processing scheme that was applied to the Svyatogor 2016 3D dataset is illustrated in Figure 31 and the important processes are described in further detail in the following sections. It was convenient to pick the seafloor during this stage as a pre-requisite for subsequent static correction because of the noise that the deghosting process tended to introduce into the section just above the seafloor. The combined effect of the complete processing flow on a single channel gather is illustrated in Figure 32.

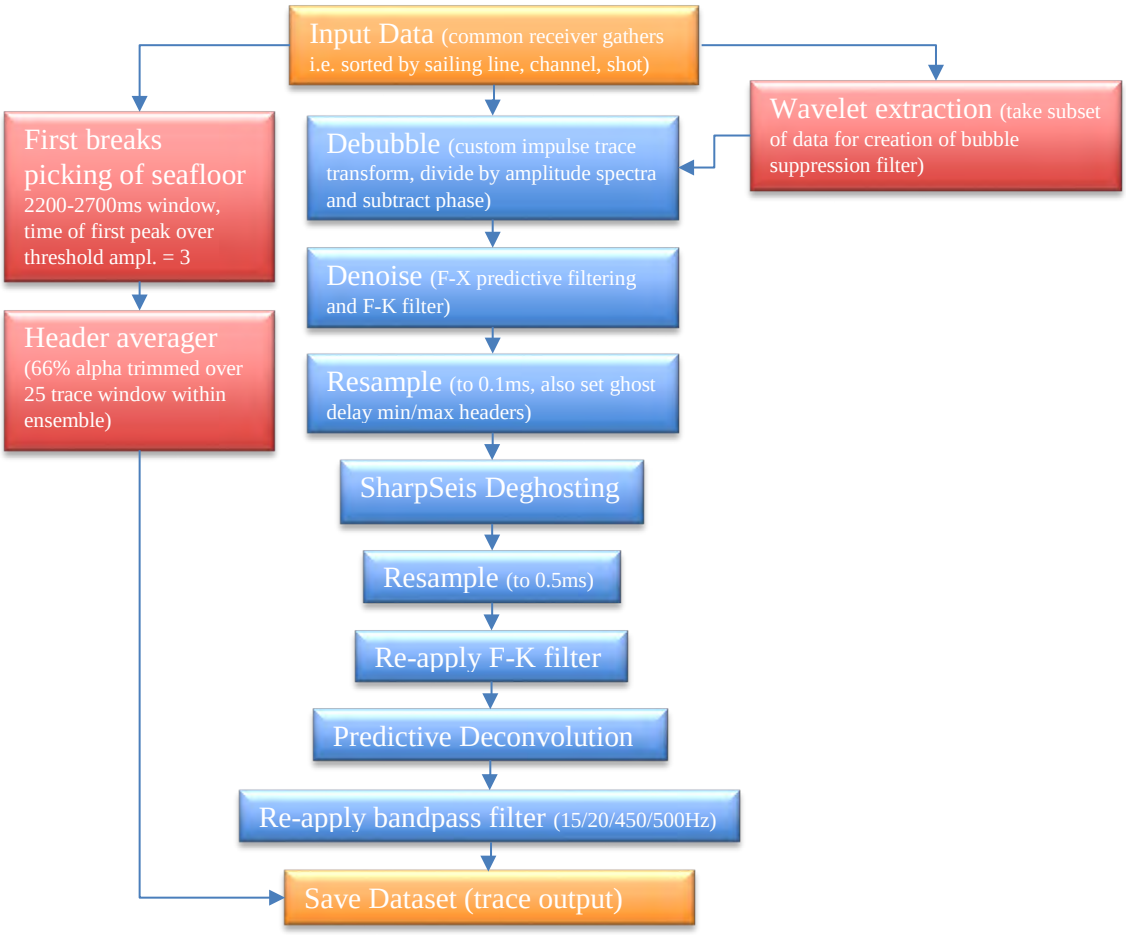


Figure 31 – overview of broadband processing steps applied to the Svyatogor 2016 3D dataset. The wavelet extraction process was conducted prior to running the main flow and the first breaks picking of the seafloor was included in the flow but is external to the main broadband processing goal.

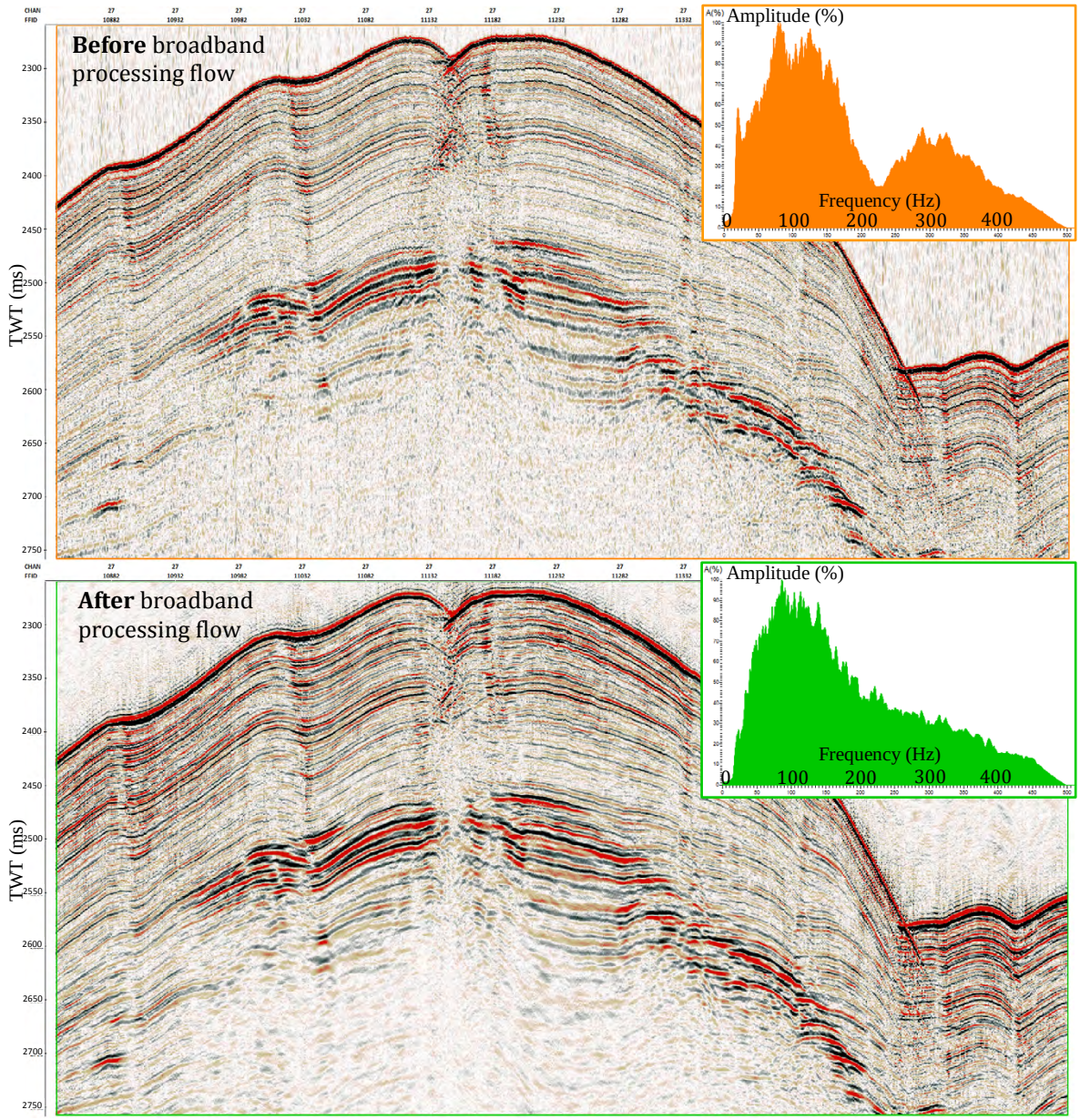


Figure 32 – illustration of the effect of the complete prestack broadband processing flow on a single channel-gather.

3.6.1 Debubble processing

Debubbling is a type of signature deconvolution and refers to the attenuation, during seismic processing, of acoustic oscillations caused by successive expansions and contractions of gas bubbles initiated by the seismic source (Wood, et al., 1978). Vertical seismic resolution is inversely proportional to the period of the source pulse so attenuating the bubble oscillations can lead to improved resolution by shortening the effective source pulse. A characteristic of the GI airguns that were used to acquire the Svyatogor 2016 3D dataset is that they use a two-phase firing sequence to reduce bubble oscillations (see section 2.3 pg.11) so the bubble effect should be small compared with data acquired using conventional sources.

The wavelet extraction module in RadexPro was used to extract the zero-phase equivalent wavelet over the window from 10ms above to 50ms below the seafloor reflector (Figure 33-a). The wavelet extraction was averaged across many shots and channels and the extracted wavelet was very consistent between sailing lines indicating that the source signature remained stable over the course of the survey. The Kolmogorov spectral factorisation module was used to convert the extracted zero phase wavelet to its minimum phase equivalent, which represents the far-field source signal (Figure 33-b). By zeroing energy with lag <20ms, corresponding to the primary pulse, the bubble signal was isolated (Figure 33-c). The bubble filter was applied using the 'custom impulse trace transforms' module in RadexPro, dividing by the amplitude spectra of the bubble pulse and subtracting the phase. The bubble filter had a very subtle effect on the data, likely because the GI airguns firing in harmonic mode are already doing a good job of supressing bubble oscillations during the acquisition.

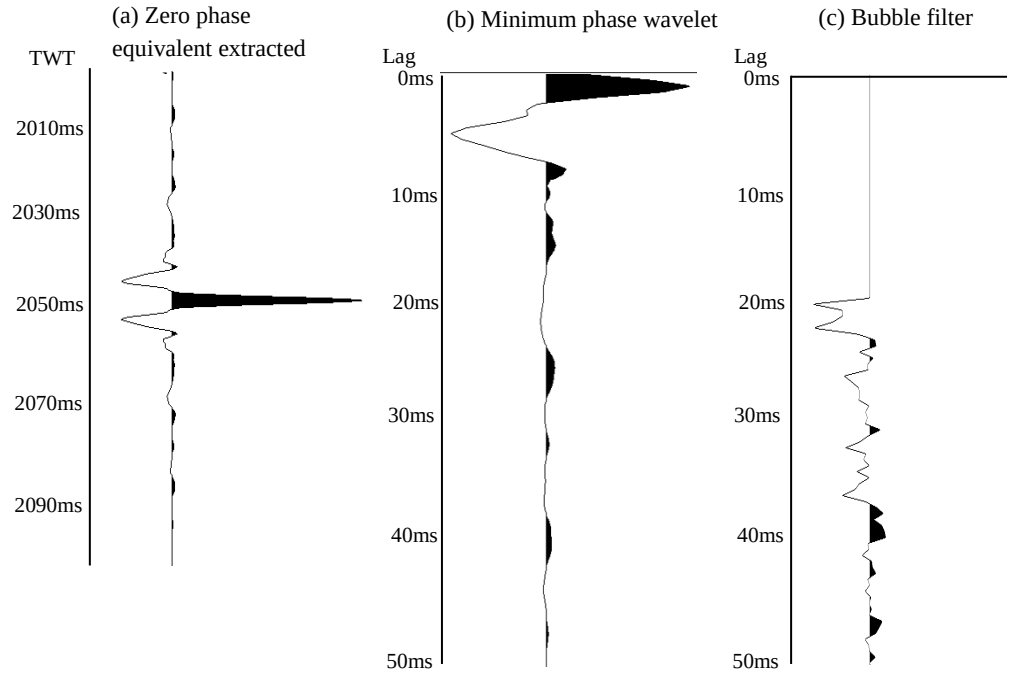


Figure 33 – isolating the bubble oscillation signal to create a bubble suppression filter, note that different gain settings apply for each trace to allow a legible comparison.

3.6.2 Noise attenuation

In the context of this broadband processing workflow the attenuation of noise can be considered a secondary process because it primarily serves to improve the effectiveness of the primary process of deconvolution (here consisting of debubble, deghost and predictive deconvolution) as discussed by Yilmaz & Doherty (2001). Deconvolution typically assumes a stationary, vertically incident, minimum-phase source wavelet and white reflectivity series that is free of noise (Yilmaz & Doherty, 2001). However, a common measure of the usable bandwidth in the data is the area of the amplitude spectrum where the signal is greater than the noise i.e. signal to noise ratio > 1 (Bancroft, et al., 2012). In this sense, reducing the level of noise in the data also leads directly to an improvement in bandwidth.

3.6.2.1 F-X Predictive Filtering

The F-X predictive filtering approach is based on the prediction of linear events in the frequency-space domain and is used to suppress random noise. The basic principle of the method is that reflections in the t-x domain map to a superposition of harmonics in the f-x domain that are perfectly predictable using an autoregressive filter. When the data contains random noise, the signal is considered to be the superposition of harmonics that can be predicted by the autoregressive filter and the noise that remains is discarded (Bekara & van der Baan, 2009). This predictability breaks down for events that are nonlinear or nonstationary such as a hyperbolic moveout or a linear event with an amplitude that varies with distance. This problem can be mitigated by applying the filter using small, overlapping spatial windows over which the data can be safely assumed to be stationary and linear. In addition, some detail in the seismic data can be smeared out laterally to an extent that depends on the prediction operator length (Galbraith & Yao, 2012).

For the Svyatogor 2016 3D dataset the best performance of F-X predictive filtering was achieved with the parameters listed in Table 3. Even though a very short filter length (3 traces) was used to maintain best-possible definition of faults a significant reduction in random noise was achieved (see Figure 34).

Table 3 – parameters used for random noise attenuation with F-X predictive filtering

Filter length (operator)	3 traces
White noise level	1%
Horizontal window length	5 traces
Time window length/overlap	200/100ms
Frequency range	15-500Hz

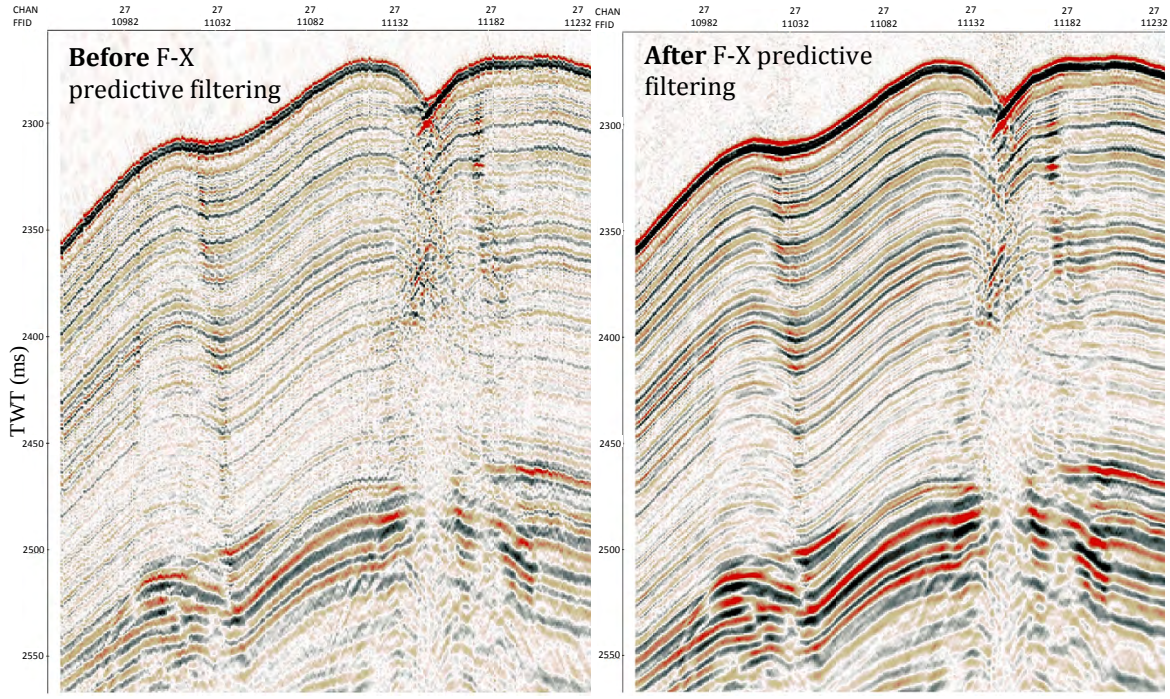


Figure 34 – example illustrating the attenuation of random noise achieved using F-X predictive filtering with the parameters listed in Table 3.

3.6.2.2 F-K Filtering

F-K filtering was applied to remove a class of coherent noise with linear moveout (see Figure 35). This class of noise can be produced by waves which are trapped in a water layer or low-velocity layer, streamer motion caused by the tugging action of the boat or scattered energy caused by irregularities in the seafloor or sub-bottom (Larner, et al., 1983). Because the majority of this noise has much steeper dip than the true reflections it can be separated effectively from the signal and removed (Figure 35). Larner et. al. (1983) also found that moveout or dip filtering was the best method for suppressing noise while preserving signal and observed that the removal of this noise from the prestack data is an important precursor to predictive deconvolution (it also contributed to better deghosting results in the present study).

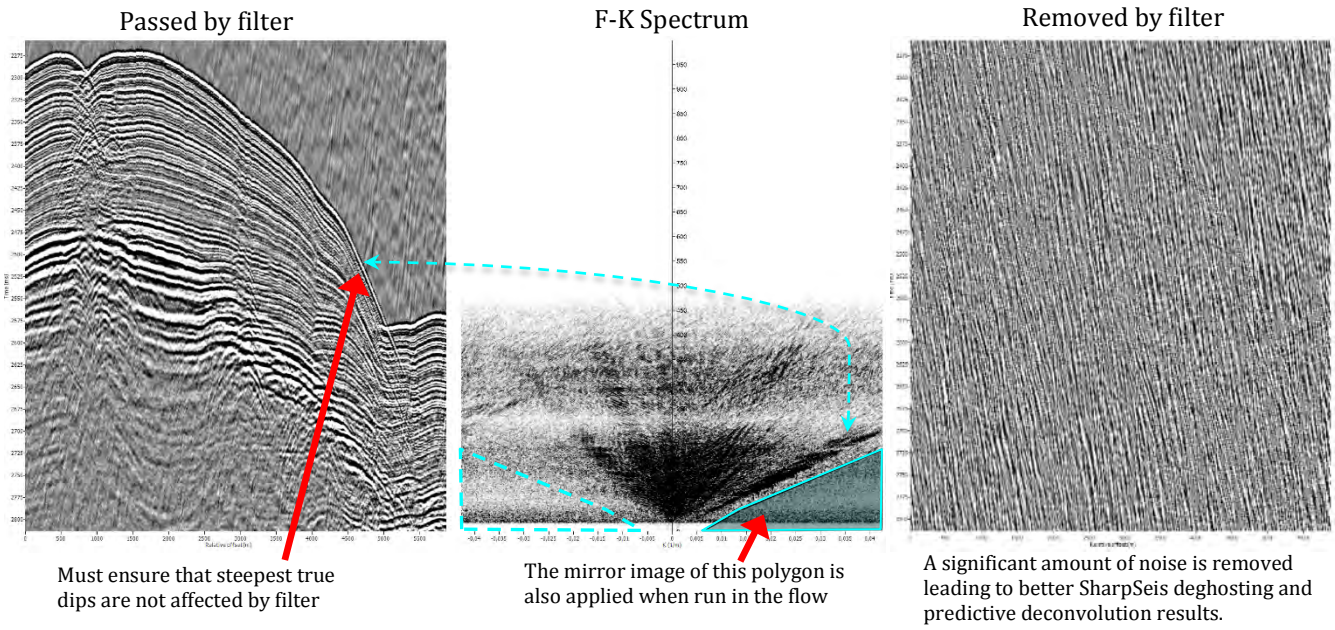


Figure 35 – illustration of F-K filtering to remove steeply-dipping coherent linear noise from common receiver gather. Note that in this example only the noise dipping to the right has been removed (shaded polygon) but when run in the processing flow the noise dipping to the left is also removed (dashed polygon). The polygon representing the part of the spectrum that will be removed was carefully defined to avoid attenuation of the steepest true dips in the data.

3.6.3 Deghosting

Ghost reflections are short-path multiples where an additional reflection occurs at the sea surface, either at the source or receiver (see illustration of source ghost in Figure 36) and are one of the main limiting factors on the resolution of marine seismic data (Yilmaz & Baysal, 2015). The ghost waves arrive with a certain delay that depends on the depth of source/receivers so that all reflections appear as double images. At certain frequencies, the ghost wave arrives at multiples of one half wavelength behind the primary (see Figure 36), leading to destructive interference and distinctive notches in the power spectrum. In the Svyatogor 2016 3D dataset, the most significant ghost is the source ghost around 225Hz corresponding to a source depth of 3.3m. While we do not have good control on the actual tow depth of the streamers we expect them to be around 1.5m, which would lead to a receiver ghost at 490Hz that is above the main signal range and is therefore considered of minor importance.

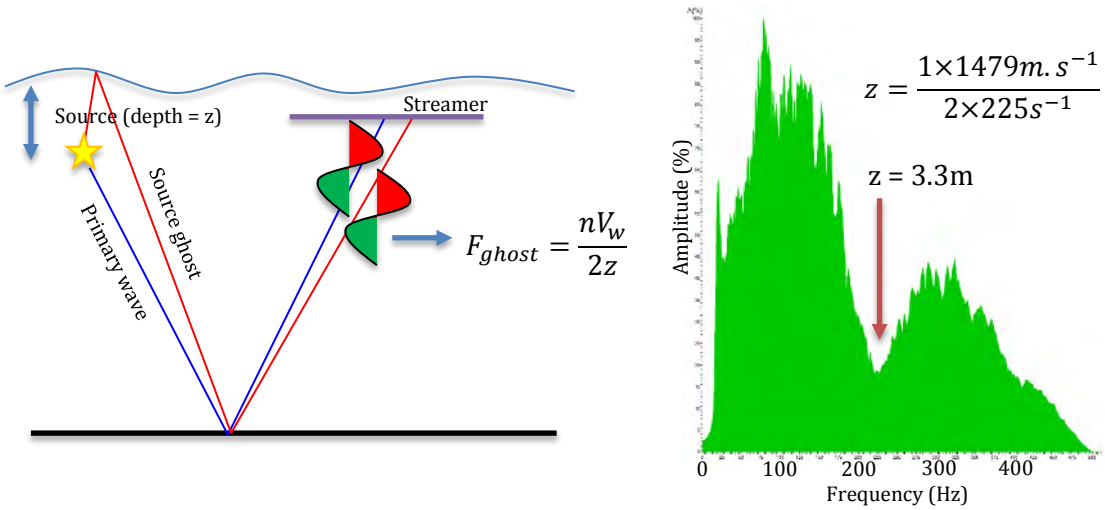


Figure 36 – Illustration of the source ghost and the destructive interference that produces the notch at 225Hz observed in a typical amplitude spectrum for the Svyatogor 2016 3D dataset, corresponding to a source depth of 3.3m.

3.6.3.1 SharpSeis Deghosting

SharpSeis is a dedicated routine for removing ghost wavefields from marine seismic data in RadexPro. It uses a stabilized approximate recursive filter solution (Vakulenko, et al., 2014) and separates each trace into two components: the primary wavefield without the ghost, and the ghost wavefield without the primary. The ghost is then effectively removed via a non-linear combination of these two components designed to maximise the signal to noise ratio of the result. The optimum ghost delay, within user specified bounds, is determined for small sliding windows of traces so that the ghost model is tuned adaptively along the length of the profile to account for variations in source/receiver depths.

In practice, the SharpSeis process performed well if the signal to noise ratio of the input data was sufficiently high but could not effectively separate and remove the ghost wavefield if the noise level was too high. This was overcome using noise removal filtering (see section 3.6.2 pg.36) prior to SharpSeis deghosting. The deghosting was applied prestack and pre-NMO on channel/common receiver gathers (gathers) and best performance was achieved using the parameters listed in Table 4.

The SharpSeis deghosting process successfully decreased the notch corresponding to the source ghost and produced a better-defined section with improved temporal resolution (Figure 37). However, the process also tended to introduce some high frequency noise into the section, particularly just above the seafloor.

Table 4 – parameters used for RadexPro SharpSeis deghosting.

Minimum-Maximum ghost time-delay (ms)	4-5ms (corresponding to source depth of 3.0-3.7m)
Time window length/step (ms)	50ms/40ms
Trace window length/step (num traces)	25/12
Minimum-maximum ghost amplitude	0.75-0.85
Time window length for forward and reverse trace combination (ms)	10ms
Sample rate during SharpSeis processing (data resampled to this rate prior to SharpSeis and resampled back to original sample rate immediately after)	0.1ms
Top muting	Muted down to 2200ms to remove direct wave.

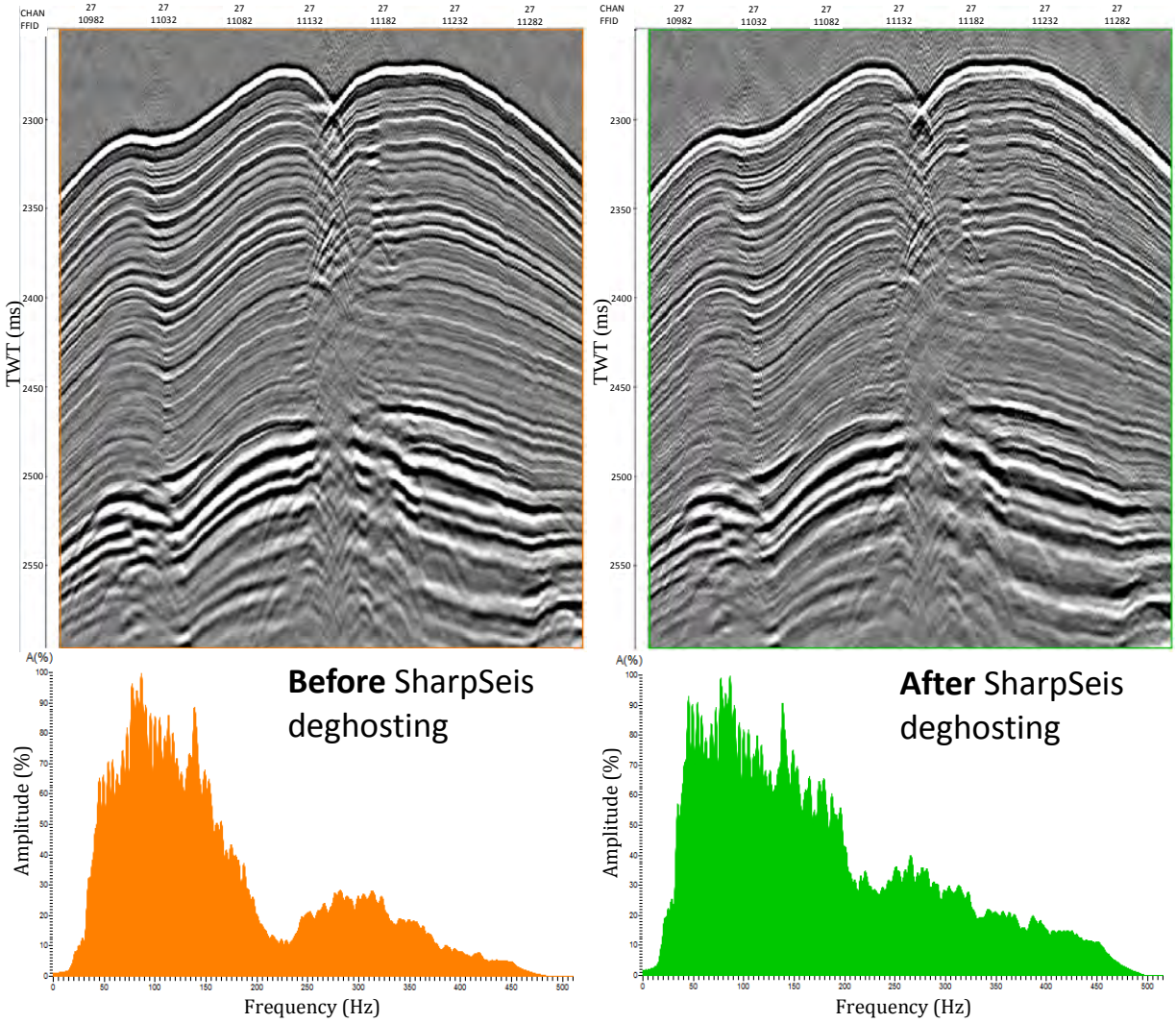


Figure 37 – Example of a single channel (common receiver) gather before and after SharpSeis deghosting using the parameters in Table 4. The deghosting has flattened the amplitude spectrum considerably, decreasing the notch corresponding to the source ghost. The seismic reflectors appear sharper i.e. temporal resolution increased, although some high frequency noise is also more apparent (particularly just above the seafloor).

3.6.3.2 Predictive Deconvolution

Predictive deconvolution involves the use of information from the earlier part of a seismic trace to predict and deconvolve the latter part of the trace (Egbai, et al., 2012). This means that it can be used to attenuate multiples, which create predictable disturbances along the trace, by designing an operator that identifies and removes the predictable part of the wavelet (the multiple), leaving only its non-predictable part. For the Svyatogor 2016 3D dataset, predictive deconvolution using a prediction gap of 8ms (TWTT), a short autocorrelation window of 12ms and a preliminary whitening of 0.01% was used to attenuate residual short path multiples that were still present following SharpSeis deghosting. As shown in Figure 38, the residual notch in the amplitude spectrum has been eliminated following predictive deconvolution. The reflections also appear subtly sharper indicating the slight increase in temporal resolution that has been gained.

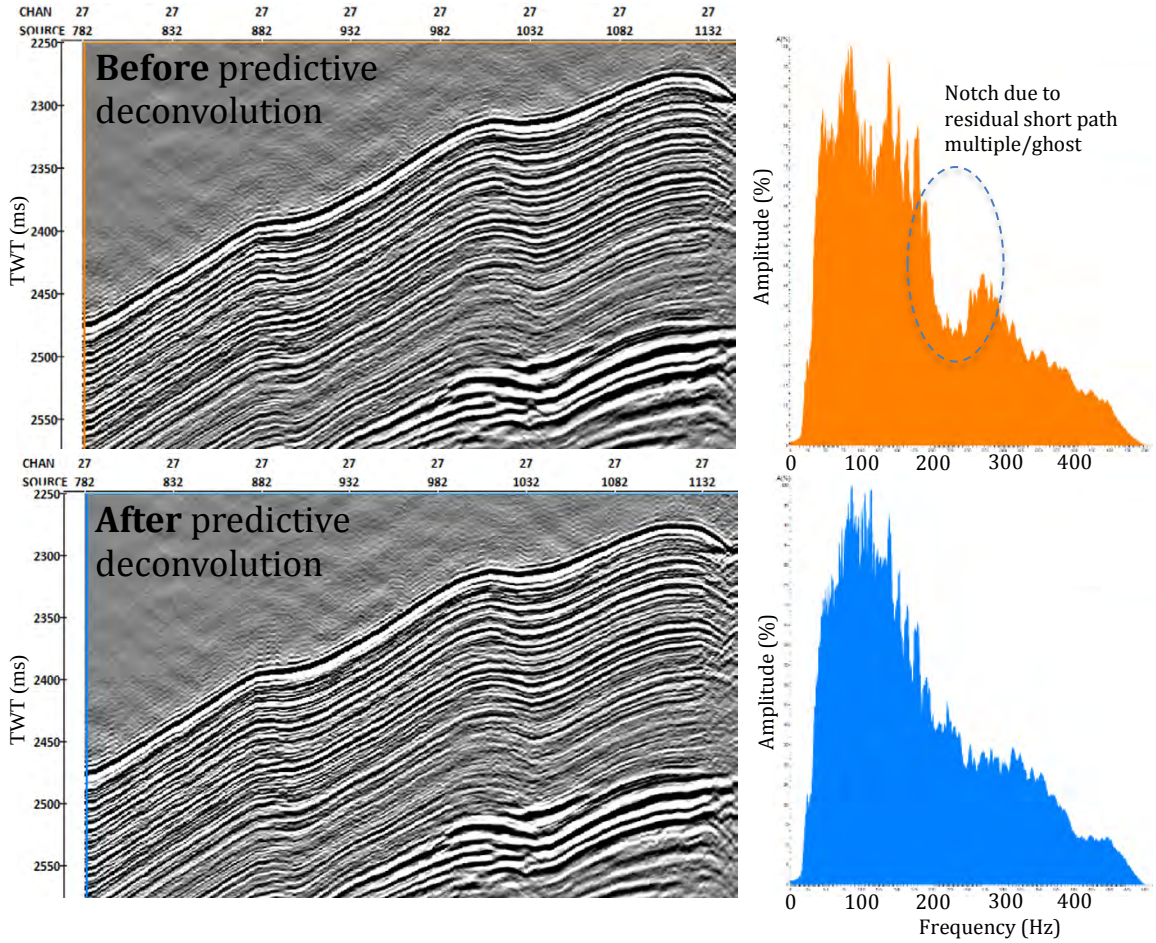


Figure 38 – illustrative common receiver sections and corresponding amplitude spectra (taken from a larger window than shown) before and after predictive deconvolution (SharpSeis deghosting has already been applied in both cases). Before predictive deconvolution a shallow notch is observed in the frequency spectrum due to residual short path multiples/ghost, which is no longer observed after the deconvolution and reflections appear slightly sharper i.e. temporal resolution is increased.

3.7 Static corrections

High-resolution seismic data is very sensitive to the vertical movement of source and receiver because unlike in conventional seismic, these movements occur at a vertically resolvable scale (Lee, et al., 2004). Such vertical movements can be caused by a combination of tides, swell, variations in the buoyancy of streamers etc. and are collectively referred to as statics.

For the Svyatogor 2016 3D dataset, tidal statics were calculated based on inverse modelling of barotropic ocean tides (Egbert & Erofeeva, 2002). Residual tidal statics and statics due to variations in receiver depth were then calculated using the HighRes Statics module in RadexPro. This module requires a seafloor pick, which was picked automatically on common receiver gathers following denoising (see Figure 13). The water velocity was set to 1479m/s. The HighRes statics calculation takes alpha trimmed averages of the seafloor picks and assumes that the minimum mean value between adjacent sailing lines should be zero and any residual difference is due to tidal statics. In this manner, a single tidal correction is derived for each sailing line. Once the tidal correction is applied, receiver statics are calculated in the same manner by assuming that the minimum mean value between channels is zero and the residual is due to statics so that one correction is generated per channel. The seafloor relief is separated from the statics by finding the average of the seafloor picks over an area (25 inlines by 25 crosslines in this case) and subtracting the resulting trend from the calculated statics.

This process proved to be an effective means of correcting statics for the Svyatogor 2016 3D dataset leading to significantly improved reflector continuity, most notably in the crossline direction (Figure 39), and fewer artefacts on extracted surfaces (Figure 40).

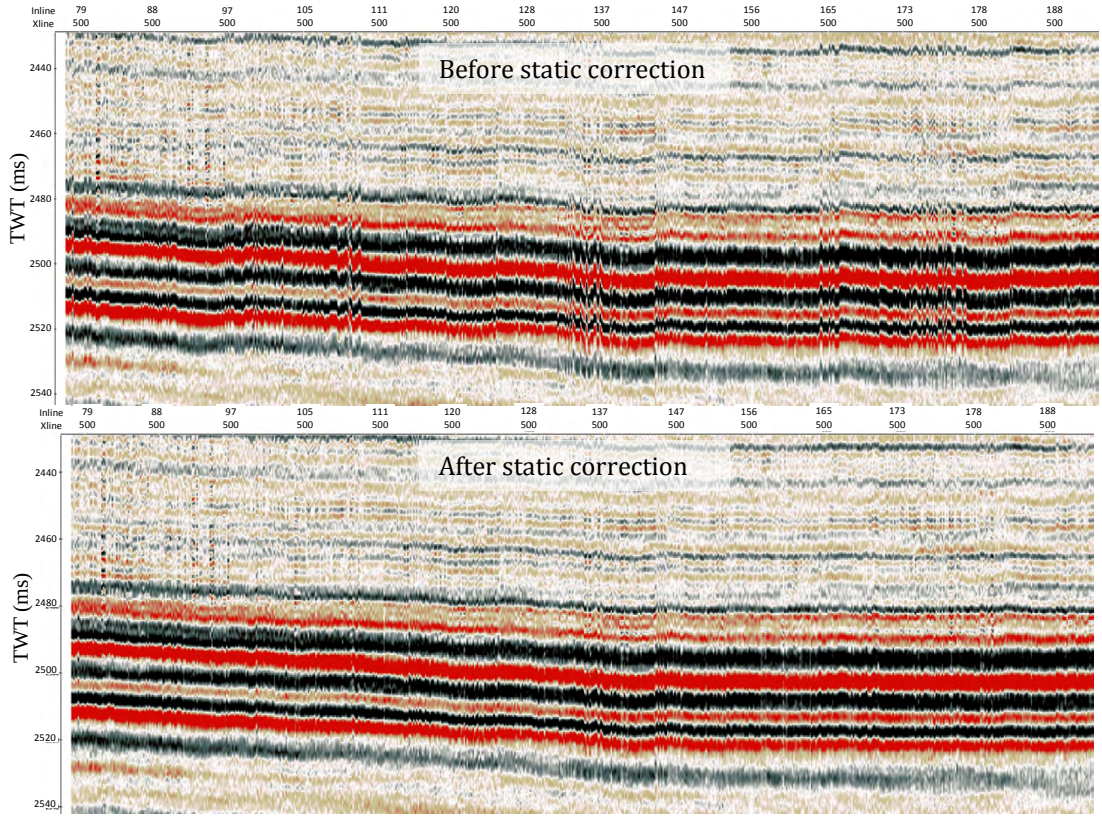


Figure 39 – Example crossline (stacked but unmigrated) demonstrating the improvement in reflector continuity achieved using the HighRes Statics module in RadexPro to correct for source/receiver statics.

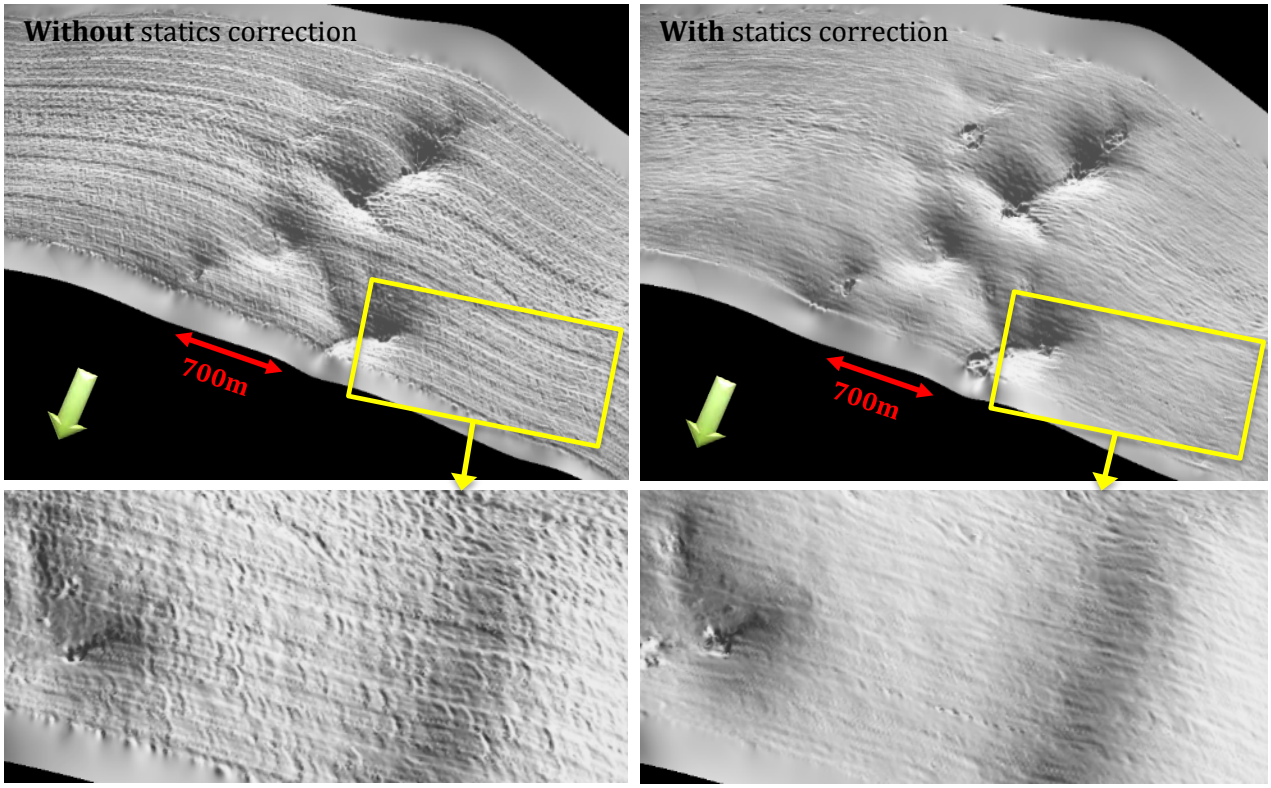


Figure 40 – Comparison of water bottom surface picked on stacked, unmigrated seismic volumes with low angled artificial illumination to highlight surface irregularities. Without static correction inline striping and a catenary shaped footprint reflecting the receiver geometry is observed. These artefacts were effectively removed with the HighRes Statics module in RadexPro.

3.8 3D Regularization (Interpolation)

Prior to migration the dataset was interpolated to a regular grid using RadexPro's 3D Regularization module (newly introduced in version 2016.2), which uses F-Kx-Ky reconstruction to perform the interpolation. Seismic data sets are commonly irregularly or sparsely sampled in the spatial domain (Zhang & Lu, 2014) leading to an uneven distribution of fold and offset. The uneven fold and offset distribution of the Svyatogor 2016 3D dataset prior to interpolation is shown in Figure 42. Irregular sampling or missing offsets can generate artefacts and cause aliasing in prestack seismic data and can degrade the results of multichannel processes including velocity analysis, stacking and migration (Cao, et al., 2011). The goal of 3D regularization is to produce a uniform distribution of offsets by interpolating common offset bin volumes.

3.8.1 CDP binning and spatial aliasing

Spatial aliasing is a common problem in seismic data that stems from the difference in vertical and horizontal resolution and whereby it becomes difficult to accurately resolve the dip of high frequency, steeply dipping events (Spitz, 1991). For a given trace spacing, the steeper the dip of the reflector, the lower the frequency at which spatial aliasing occurs (Yilmaz & Doherty, 2001). As such, one must strike a compromise between the maximum dip angle to be imaged, the desired frequency content, the trace spacing (defined by the source and receiver intervals and relating to acquisition time/cost) and the amount of aliasing related noise that can be tolerated (Bancroft, 1995). Spatial aliasing can also negatively affect the performance of processes like f-k filtering and migration, which are reliant on the accurate resolution of dip. For example, migration moves the spatially aliased frequency components

in the wrong direction and generates a dispersive noise that degrades the quality of the migrated section (Yilmaz & Doherty, 2001).

Since there are significant physical constraints on shot and receiver intervals (e.g. 4 sec record length, shots cannot overlap, compressor capacity to recharge guns etc.), it can be necessary to deal with spatial aliasing via different methods. One approach would be to apply a low-pass filter to remove the high frequencies that are spatially aliased for steeply dipping reflectors. However, this is not a good solution because the bandwidth is reduced, both for the steeply dipping events that were impacted by spatial aliasing and the flatter events that were not. One of the best approaches to this problem is to use interpolation to decrease the trace spacing (Lines, et al., 2001), although the additional computational cost of increasing the number of traces through interpolation to finer and finer grids must also be kept in mind. Interpolation is an effective means of reducing spatial aliasing because the dip can be detected and measured for the unaliased frequencies and then used to guide the interpolation of both the unaliased and aliased frequencies (Yilmaz & Doherty, 2001; Spitz, 1991).

To avoid aliasing the trace spacing, Δx, must satisfy the following condition:

$$\Delta x \leq \frac{v}{4f \sin\theta}$$

where v is the velocity, f is the maximum frequency and θ is the maximum dip (measured in ms/trace)

This expression can also be arranged to solve for the maximum frequency that will be unaffected by spatial aliasing:

$$f_{max} = \frac{v}{4\Delta x \sin\theta}$$

This expression gives the frequency thresholds shown in Table 5. The maximum dip observed in the Svyatogor 2016 3D dataset was around 12° as illustrated in Figure 41. After trialling several different possibilities, a trace spacing/CDP bin size of 6.25m was selected as the best compromise between:

- Avoiding spatial aliasing for steep dips - frequencies up to ~300Hz will be unaliased for largest observed dip and the majority of the usable bandwidth in the dataset lies below this frequency.
- Size of the interpolated data volume – which affects the speed of subsequent processing, critically migration, which is a computationally intensive and time-consuming operation.
- Interpolation limits – data was acquired with a ship speed of 4kn and shot interval of 6s, giving a shooting interval of approx. 12m. Interpolation to smaller bin sizes (3.125m & 4m) was attempted but the results were poorer than those at 6.25m spacing because the initial data traces were not dense enough for the 3D regularization to produce optimal results. This result is consistent with Lines et. al. (2001) who find that the practical limit for successful interpolation is to produce data with effectively half the original group spacing.

Table 5 – Frequency threshold for spatial aliasing, red numbers indicate aliasing for frequencies within the useful bandwidth of P-Cable data and the best-compromise trace spacing is highlighted with a blue box.

Max frequency without spatial aliasing (Hz, v=1500m/s)					
Dip Angle (°)	Trace spacing (m)				
	3.125	4	6.25	12.5	25
5	1377	1076	688	344	172
7.5	919	718	460	230	115
10	691	540	346	173	86
12.5	554	433	277	139	69
15	464	362	232	116	58
20	351	274	175	88	44
30	240	188	120	60	30

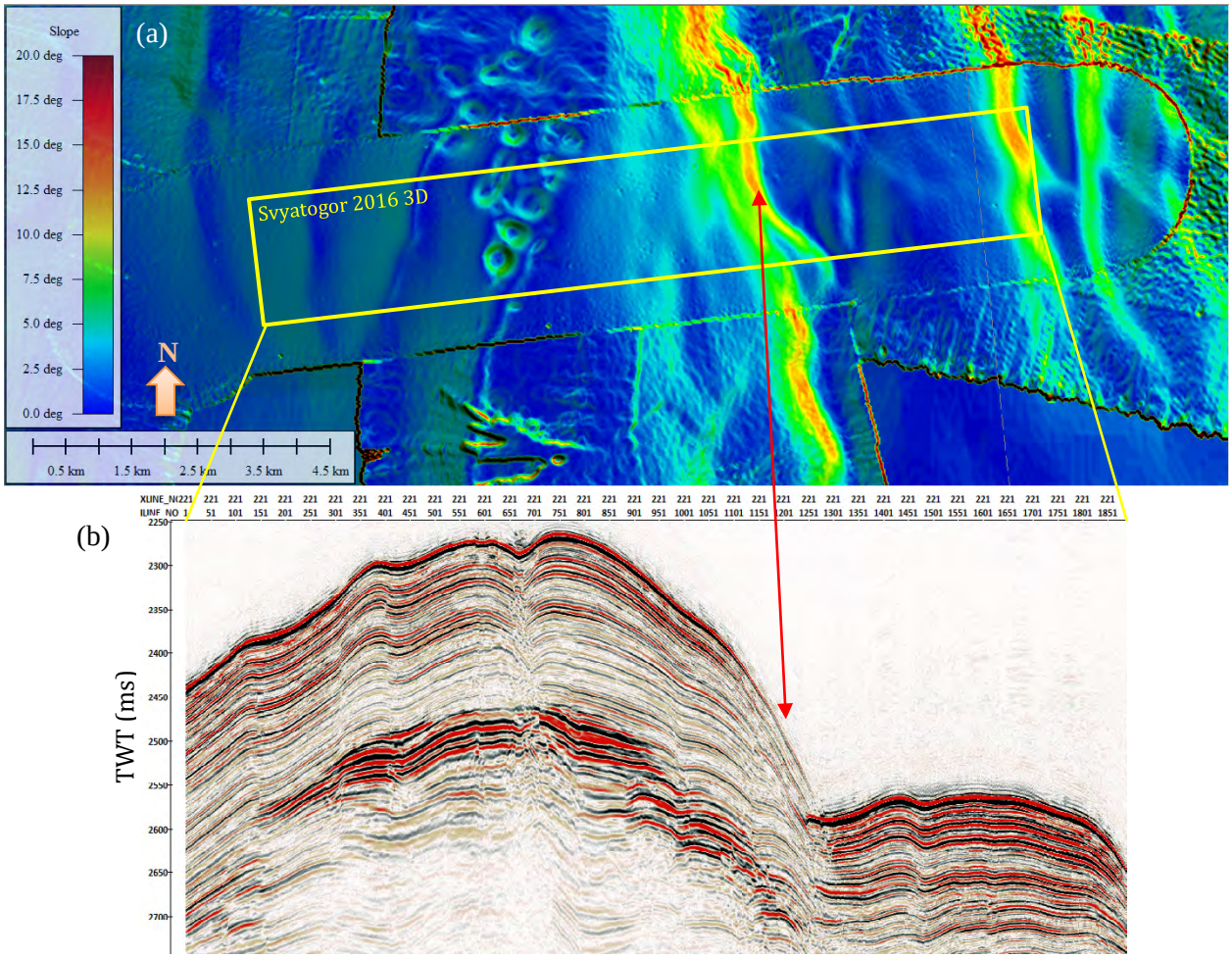


Figure 41 – (a) Multibeam bathymetry coloured by slope in the region of the Svyatogor 2016 3D seismic survey (yellow rectangle) and (b) an example inline (unmigrated) illustrating the steeply dipping area of the seafloor that is the area most prone to spatial aliasing in the dataset resulting in a loss of amplitude and coherence.

3.8.2 Offset binning

3D regularization involves the interpolation of common offset volumes so it is necessary to apply suitable offset binning prior to running the regularization. The offset range for the Svyatogor 2016 3D survey is relatively limited, spanning 103-175m, because of the short (25m) streamers that were used to acquire it. Consequently, the best 3D regularization results were achieved by grouping the traces into just two offset bins; near traces spanning 103-140m offsets and far traces spanning 140-175m offsets. The fold distribution when binned into near and far offset volumes is shown in Figure 42. This appeared to give better definition compared to grouping all traces into a single offset bin and attempting to divide the traces into three or more offset bins resulted in an unacceptably high level of sparseness within each offset bin. Even with two offset bins some gaps in coverage were observed in the crossline direction (Figure 42) but these were minor enough to be effectively interpolated across by the 3D regularization operator. It is common that trace interpolation in the crossline direction is required for 3D seismic data prior to 3D migration (Yilmaz & Doherty, 2001).

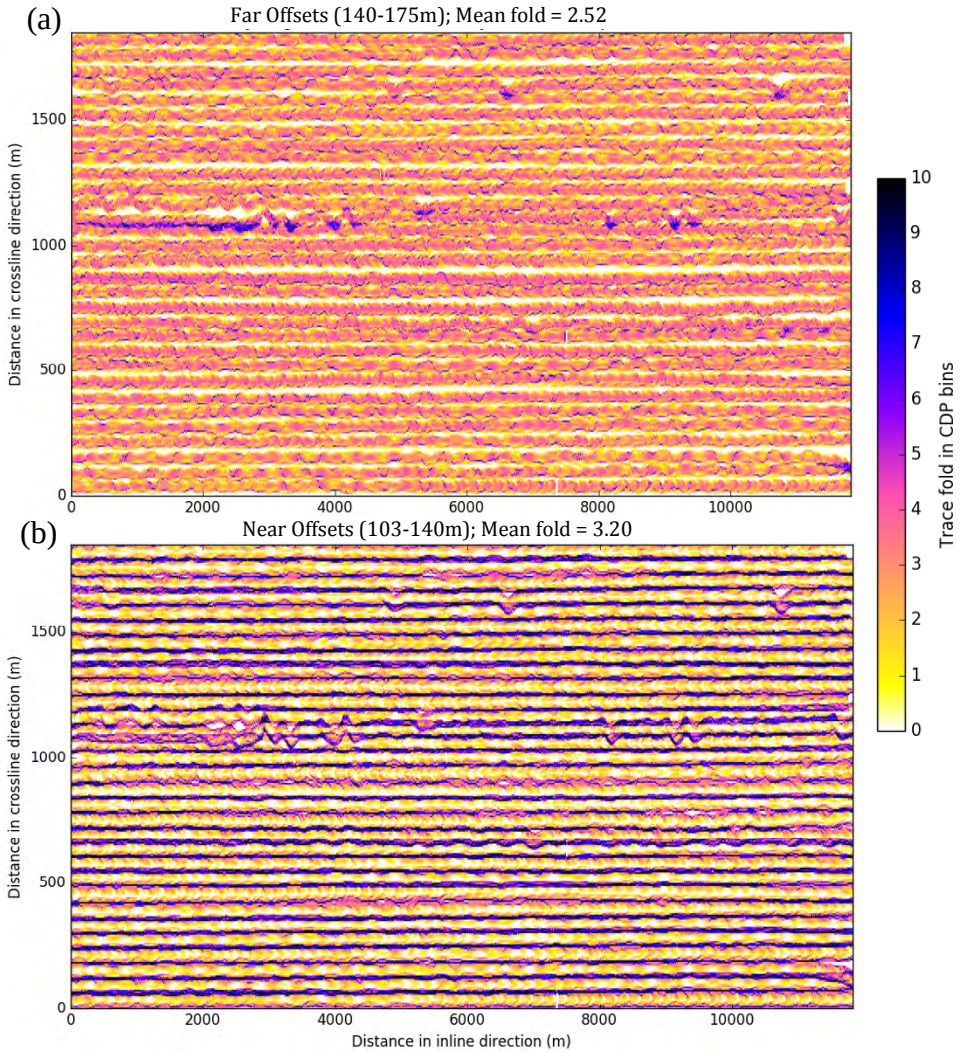


Figure 42 – CDP fold maps for the common offset bins assigned prior to 3D Regularization. Because of the short streamers and small offset range the traces were simply divided into (a) far and (b) near offset bins, which already contain some gaps in coverage in the crossline direction. Note that the crossline scale is exaggerated.

3.8.3 3D regularization parameters

The 3D regularization module in RadexPro uses F-Kx-Ky reconstruction to interpolate traces to a regular grid. While the module includes a ‘sparse’ algorithm capable of handling nonuniform input data by finding the sparse solution in F-Kx-Ky domain for each time frequency (F) slice, the solution may not precisely fit the input data and dim events may be wiped out. As a result, it is necessary to fine-tune the parameters to manage the balance between the ‘sparseness’ and smoothness of solution and the fit to the input data. Since the parameters are interrelated, it was necessary to test various combinations on a small portion of the dataset to achieve an acceptable balance. The parameters that were selected are shown in Figure 43. A comparison of the data before and after 3D regularization (Figure 44) shows that the distribution of traces in the offset bins has been made uniform, while missing traces have been interpolated and the bandwidth of the data has been maintained.

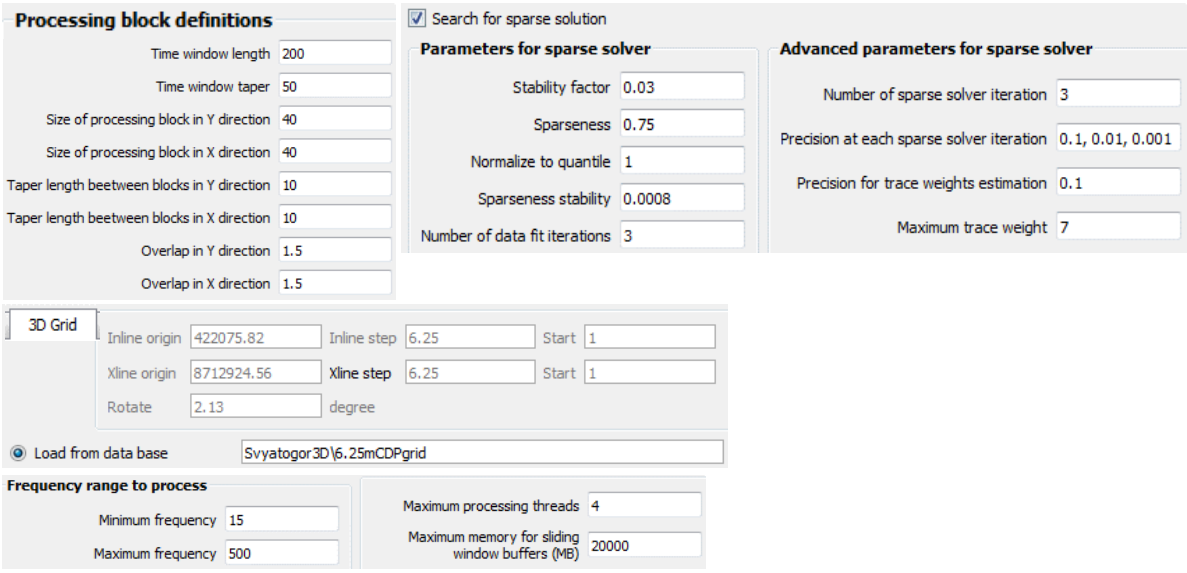


Figure 43 – Important parameters used in RadexPro’s 3D regularization module.

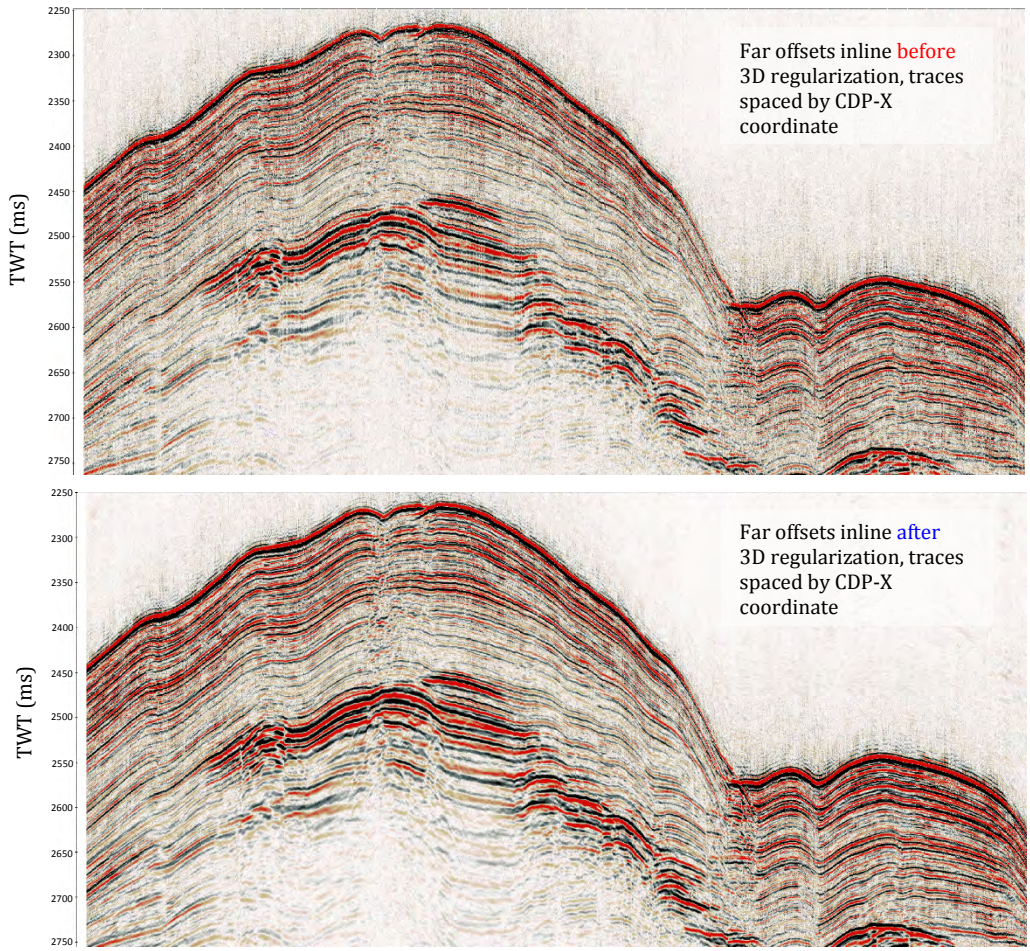


Figure 44 – Typical before/after 3D regularization comparison for inline 120 composed of only far offset traces; the distribution of traces has been made uniform, missing traces have been interpolated and the bandwidth of the data has been maintained.

3.9 Migration

Migration aims to move dipping reflectors to their true positions and to collapse diffracted energy to its source point resulting in increased spatial resolution and a seismic 'image' that more accurately reflects the true subsurface acoustic reflectivity (Yilmaz & Doherty, 2001).

The Svyatogor 2016 3D dataset was migrated using prestack 3D Kirchhoff time migration. The basic concept of Kirchhoff time migration is that each seismic event is associated with a travel time corresponding to the time taken for the energy to propagate from the source to the reflection point and then back to the receiver. During migration, the energy of the seismic event is smeared across all possible points whose reflection travel times are equal to the observed travel time of the event (Sun, 2001) and which must necessarily include the true reflection point. Such constant travel time surfaces follow ellipsoidal to quasi-ellipsoidal geometries depending on the velocity variation of the medium. When many traces are migrated and stacked, constructive interference tends to strengthen true reflectors and destructive interference tends to attenuate the spuriously smeared energy i.e. migration artefacts.

In practice, artificial events may remain at far-field positions (Sun, 2001) due to the number of traces being insufficient to attenuate energy that has been smeared a long way from the true reflector. This problem is addressed by limiting the migration aperture i.e. the event is smeared only to a small region surrounding the initial position of the trace rather than the complete ellipsoid. The migration aperture is an important parameter in Kirchhoff migration and should be large enough that it contains the true reflection points for the full range of dips to be imaged (see section 3.9.2 pg.49). Using a limited migration aperture also decreases the computational cost of the migration, which can be significant for Kirchhoff migration (Sun, 2001).

3.9.1 Defining a velocity model

Seismic migration requires some knowledge of the subsurface velocity distribution and the more accurately the velocity model can represent the true velocity distribution, the better the migration result. The very short (25m) streamers and limited moveout of the Svyatogor 2016 3D dataset render conventional stacking velocity estimation by semblance picking ineffective. An initial velocity model was estimated by measuring the shapes of diffraction hyperbola. The initial model was then updated iteratively by assessing successive migrations for signs of over- or under-migration until an acceptable result was achieved. The control points used to define the final iteration of the velocity model are shown in Table 6, velocities were interpolated along the time axis and spatially away from control points during the migration.

Table 6 – control points from final velocity model used in prestack Kirchhoff time migration of the Svyatogor 2016 3D dataset, velocities were linearly interpolated along the time axis and spatially away from control points.

Xline	Inline	TWT interval (ms): velocity (km/s)
100	150	0-2520:1.48, 2570-2800:1.65, 2870-4000:1.7
500	150	0-2450:1.48, 2520-2700:1.6, 2770-4000:1.7
600	150	0-2400:1.48, 2450-2575:1.5, 2625-4000:1.65
800	150	0-2450:1.47, 2520-2600:1.6, 2670-4000:1.65
1200	150	0-2550:1.47, 2620-3000:1.5, 3050-4000:1.6
1300	150	0-2550:1.47, 2620-3000:1.5, 3050-4000:1.6
1400	150	0-2650:1.47, 2720-2950:1.5, 3000-4000:1.6
1550	150	0-2550:1.4, 2620-2870:1.5, 2940-4000:1.6
1700	150	0-2650:1.4, 2720-2950:1.5, 3000-4000:1.6

3.9.2 Migration Aperture

The migration aperture width can be thought of as the maximum horizontal distance that reflected energy is displaced during migration. Migration aperture is a critical parameter in Kirchhoff migration. Small apertures cause steeply dipping events to be attenuated (Yilmaz & Doherty, 2001). If the migration aperture is too large, random noise that tends to increase with depth in a seismic section can be relocated shallower in the section during migration where it can degrade the signal to noise ratio. This leads to a trade-off whereby it may be necessary to compromise on migration aperture deeper in the section to avoid this problem (Yilmaz & Doherty, 2001).

A common method for selecting an appropriate migration aperture is to consider the maximum horizontal displacement in migration for the steepest dip of interest in the input section (Yilmaz & Doherty, 2001):

$$d_x = \frac{v^2 t}{4}\frac{\Delta t}{\Delta x}$$

Where d_x is the horizontal displacement, v is the velocity (m/s), t is the travel time (s), and $\Delta t/\Delta x$ (s/m) is the apparent dip as measured on the unmigrated time section.

This horizontal displacement is essentially a radius and the migration aperture will be given by twice the horizontal displacement. One of the steepest dipping areas observed in the Svyatogor 3D 2016 dataset is shown in Figure 45 and we can calculate an appropriate migration aperture as follows:

$$d_x = \frac{(1500m.s^{-1})^2.(2.6s)}{4}\frac{(0.172s)}{(962.5m)} = 261m$$

$$Migration\ aperture = 2d_x = 2{\times}261m = 522m$$

In practice, the migration aperture was varied as a function of TWT as follows: 320m from 0-2300ms, 475m at 2400ms, 500m at 2500ms and 520m from 2700-4000ms and a taper of 30% was applied to smoothly attenuate the ends of the migration operator and avoid edge effects.

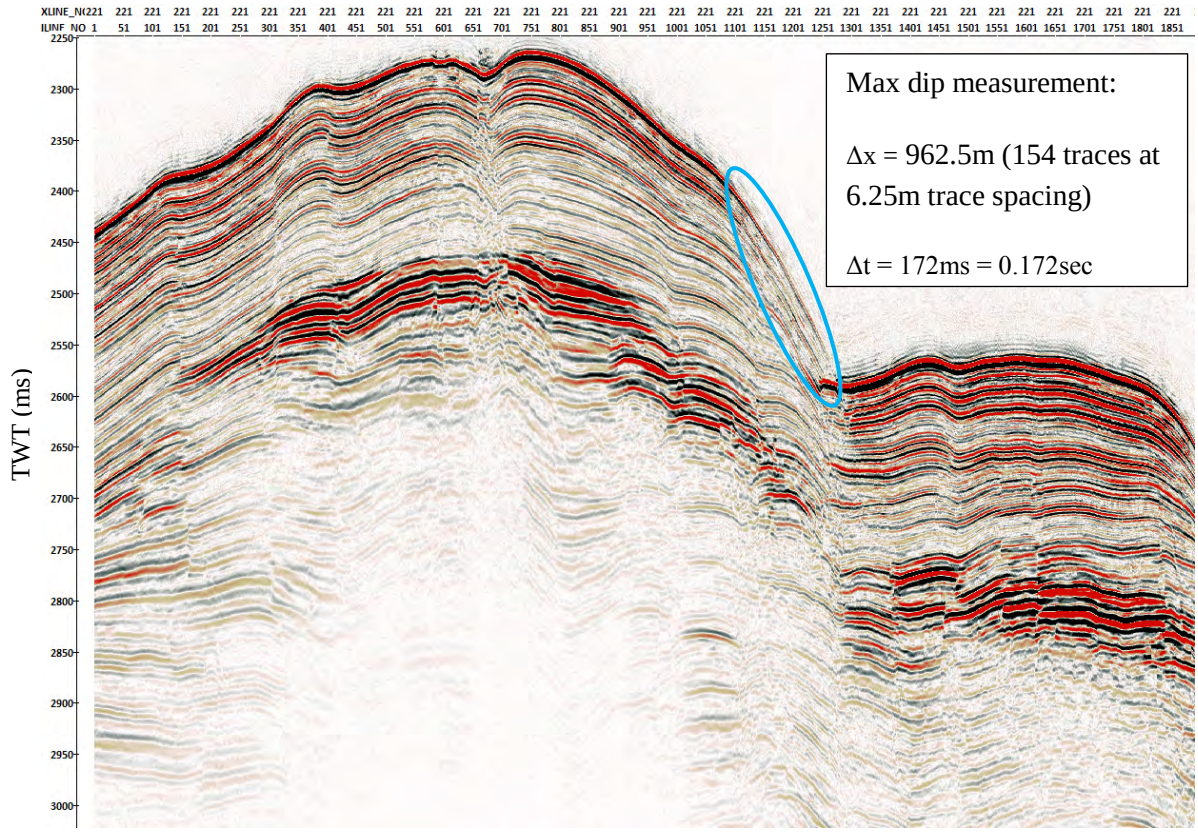


Figure 45 – Example showing steeply dipping seafloor that was used to define the required migration aperture width, the maximum migration aperture was 520m.

3.9.3 Migration result

The result of the prestack (using two offset bins) Kirchhoff time migration is illustrated in Figure 46. The migration was considered largely successful after ~4 iterations of velocity model refinement.

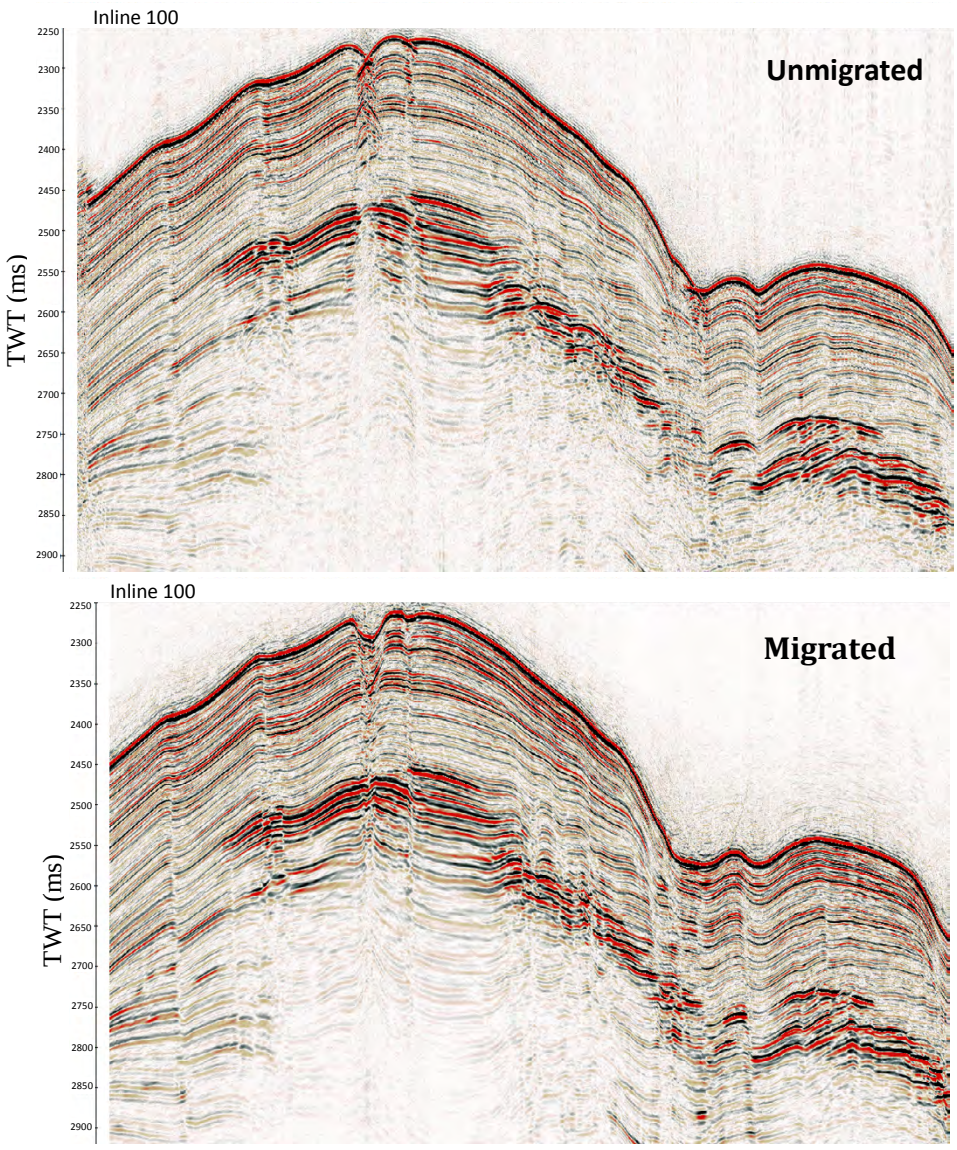


Figure 46 – Comparison before and after prestack Kirchhoff time migration (two offset bins). Migration was largely successful – faults are sharpened and prominent bowties are relocated. Prominent upward bending of reflectors is observed beneath the ridge-crest pockmarks and may be due to migration induced distortion due to an inability to accurately model the velocity structure at this fine lateral scale.

Faults appear significantly sharper following migration and prominent bowties reflections from pockmarks and other synform features have been relocated to their true subsurface positions. Some subtle under- and over-migration can still be observed along fault planes, particularly in the deeper section. However, the migration result was considered sufficient given the time costs associated with continued trial and error refinement of the velocity model. In addition, prominent upward bending of reflectors is observed beneath the ridge-crest pockmarks, which may be partly due to migration induced distortion caused by an inability to accurately model the velocity structure at this fine lateral scale (discussed further in section 5.1 pg.86).

3.10 Post Migration Denoise and time-varying gain

All migration algorithms will tend to create some component of noise in the output image, because they are not perfect solutions of the wave equation (Jones, 2011). For the example of a Kirchhoff migration, energy within the Fresnel zone of the true reflection point should interfere constructively while energy spread across the remainder of the migration aperture should interfere destructively and attenuate. In practice, the number of traces is usually insufficient to completely attenuate this energy and some of it remains in the output image, usually as a form of steeply dipping and sometimes aliased noise (Jones, 2011).

F-X-Y predictive filtering was applied to attenuate this migration noise and other remaining random noise. F-X-Y deconvolution is the 3D extension of F-X predictive filtering (see section 3.6.2.1 pg.37). Spatial prediction filtering in 3D gives superior performance for weak, curved or faulted events resulting in less distortion of geological structure when compared to 2D predictive filtering (Chase, 1992). The filter parameters are shown in Table 7. The removal of random noise by F-X-Y predictive filtering enhanced reflector continuity without excessive edge smearing effects such that fault definition was maintained (see Figure 47).

A time varying bandpass filter was also applied because the deep section of the data did not contain any useful signal in the high frequency range (as high frequencies are preferentially attenuated with depth). This filter limited the bandwidth to 15/20/150/200Hz (low-cut/low-pass/high-pass/high-cut) below 2900ms (with a tapering length of 100ms). In addition, an exponential gain correction of 15dB/sec was applied to compensate for intrinsic attenuation of the wavefield and improve the amplitude balance down the section (see Figure 47).

Table 7 – parameters used for F-X-Y predictive filtering

Filter Type	Multi-pass
Algorithm	Adaptive (rate=0.9)
Filter size (inline x crossline)	5 x 5 traces
Horizontal window length/overlap	28x28 / 8x8 traces
Time window length/overlap	200/50ms
Frequency range	10-500Hz

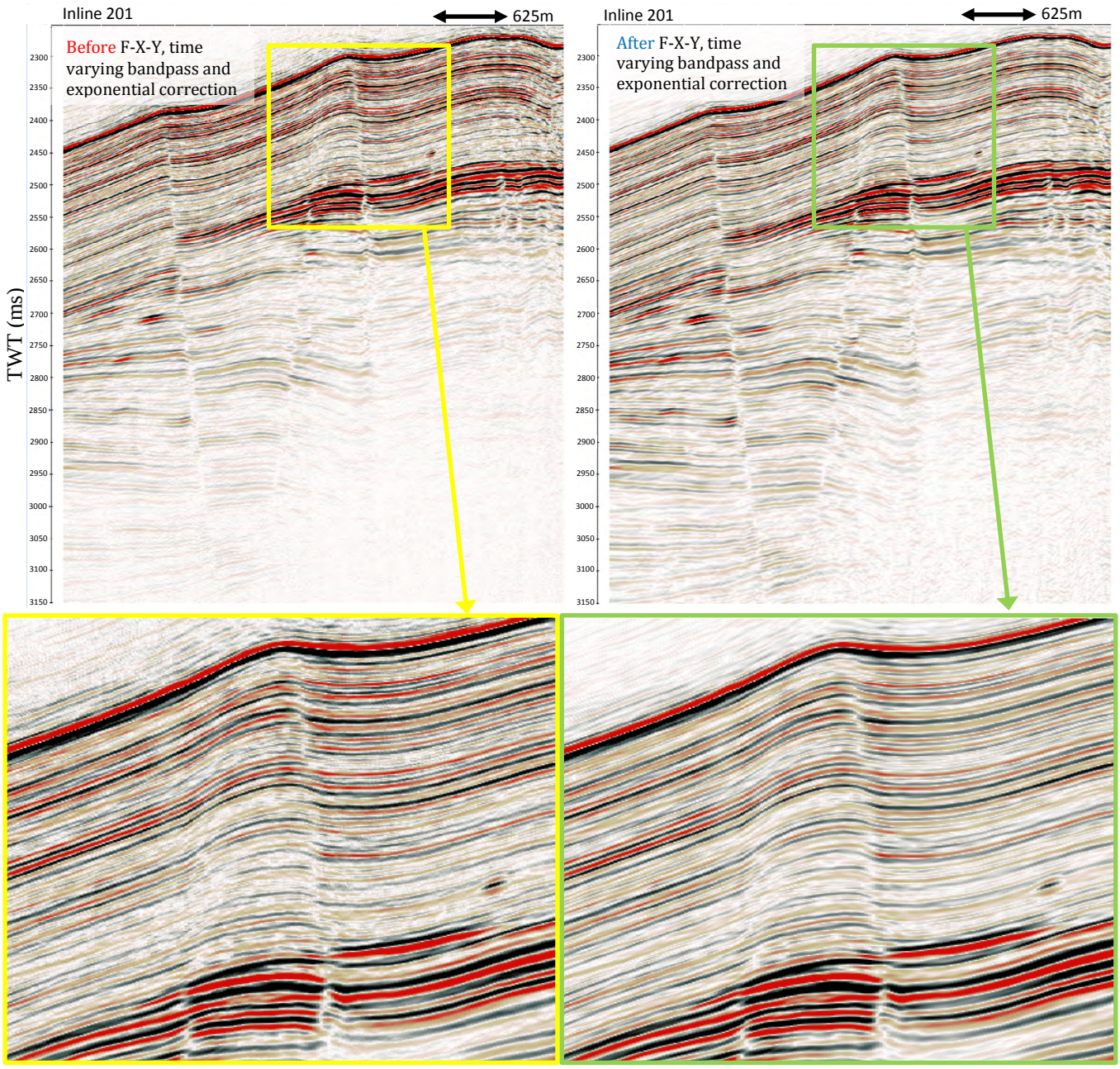


Figure 47 – Comparison before and after post stack denoise (F-X-Y predictive filtering and time-varying bandpass) and time-varying gain (exponential correction). The amplitudes show improved balancing down the section (upper) and the removal of random noise has improved reflector continuity while maintaining fault definition (lower).

4 Seismic Interpretation

The primary focus of interpretation for this study was the newly acquired Svyatogor 2016 3D P-Cable seismic survey and the additional insights it provides in understanding the Svyatogor gas-hydrate, free-gas system. This chapter incorporates both an explanation of the key methodologies that were applied during the investigation of the Svyatogor 2016 3D dataset and a description of the observations that underpin the interpretation. A correlation of the new dataset with the existing stratigraphic framework is also provided to place the interpretation in context.

4.1 Stratigraphy

4.1.1 Regional stratigraphic context

Three regional seismostratigraphic units: YP-1, YP-2 & YP-3 have previously been described on the southern Yermak Plateau i.e. to the north of the Molloy transform fault (Eiken & Hinz, 1993; Geissler & Jokat, 2004). Age control for these units is based on correlation with ODP Leg 151 sites 910- 912 on the Yermak plateau and sites 908 & 909 located in the Fram Strait to the south of the Molloy Transform Fault (MTF) and west of Svyatogor Ridge. The base of the youngest YP-3 unit has been correlated with an age of ~2.7Ma (Knies, et al., 2009; Mattingsdal, et al., 2014) and corresponds to a distinct unconformity on the southern Yermak Plateau reflecting a major increase in sedimentation rate and the onset of glacial deposits. The base of the YP-2 unit is estimated to be ~11-14.6 Ma (Geissler, et al., 2011; Mattingsdal, et al., 2014). This is older than the relatively young oceanic crust that underlies Svyatogor ridge, constrained by magnetic anomaly chrons 5 & 2A to an age in the range of 9.8-2.8 Ma (Engen, et al., 2008), so the base of YP-2 is not expected to be present in the study area.

The boundary between the YP-3 and YP-2 units has been mapped on regional 2D seismic profiles, tied back to a correlation with ODP site 909, in the Svyatogor Ridge region (Westvig, 2015). While this correlation has been reviewed and revised over the course of the present study (K Waghorn 2017 pers. comm., 27 Mar), the correlation of the YP-3/YP-2 boundary across the study area remains challenging with presently available data and is subject to significant uncertainty.

A correlation of the Svyatogor 2016 3D dataset with a subset of the regional 2D seismic profiles and the interpreted YP-3/YP-2 boundary is illustrated in Figure 48 & Figure 49. The correlation of this boundary with the set of horizons that were picked locally across the 3D dataset to investigate the stratigraphic development is shown in Figure 52. The YP-3/YP-2 boundary correlates to a position between horizons H5 & H4, indicating that the stratigraphic development discussed in section 4.1.3 (pg. 56) mostly concerns the past 2.7Ma.

The correlation of the YP-2/YP-3 boundary across the Svyatogor 2016 3D dataset suggests the deposition of a ~270-315ms thick sedimentary sequence has occurred over the past 2.7Ma in the study area, corresponding to 210-230m using migration velocities ranging from 1400-1650m/s. The thick package of sedimentary reflections observed below the YP-3/YP-2 boundary (e.g. Figure 48 & Figure 52) indicates that a significant interval of YP-2 sediments are present and that sedimentation began prior to 2.7Ma.

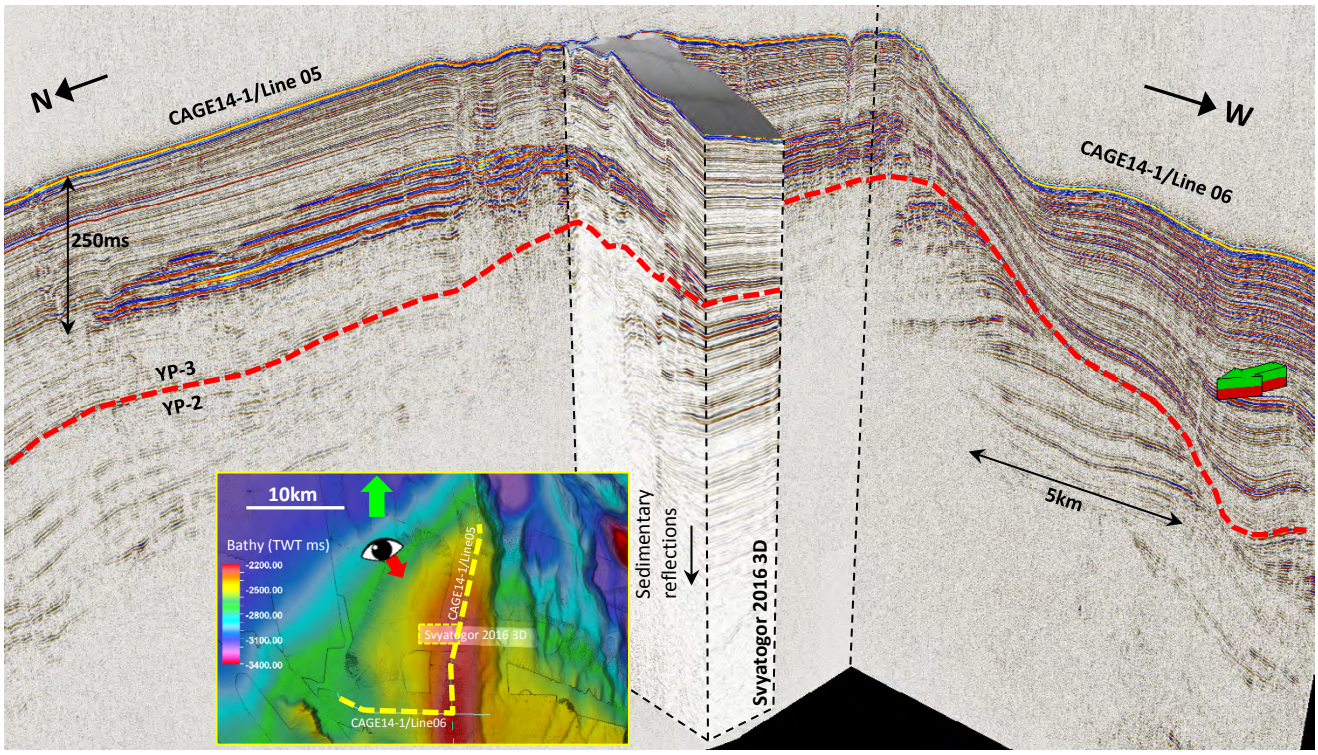


Figure 48 – Northerly and westerly stratigraphic context of the Svyatogor 2016 3D dataset provided by high-resolution 2014 vintage 2D P-Cable seismic profiles. Red dashed line marks the YP-2 to YP-3 boundary corresponding to ~2.7Ma (Mattingsdal, et al., 2014). Inset location map highlights the visible parts of the seismic profiles.

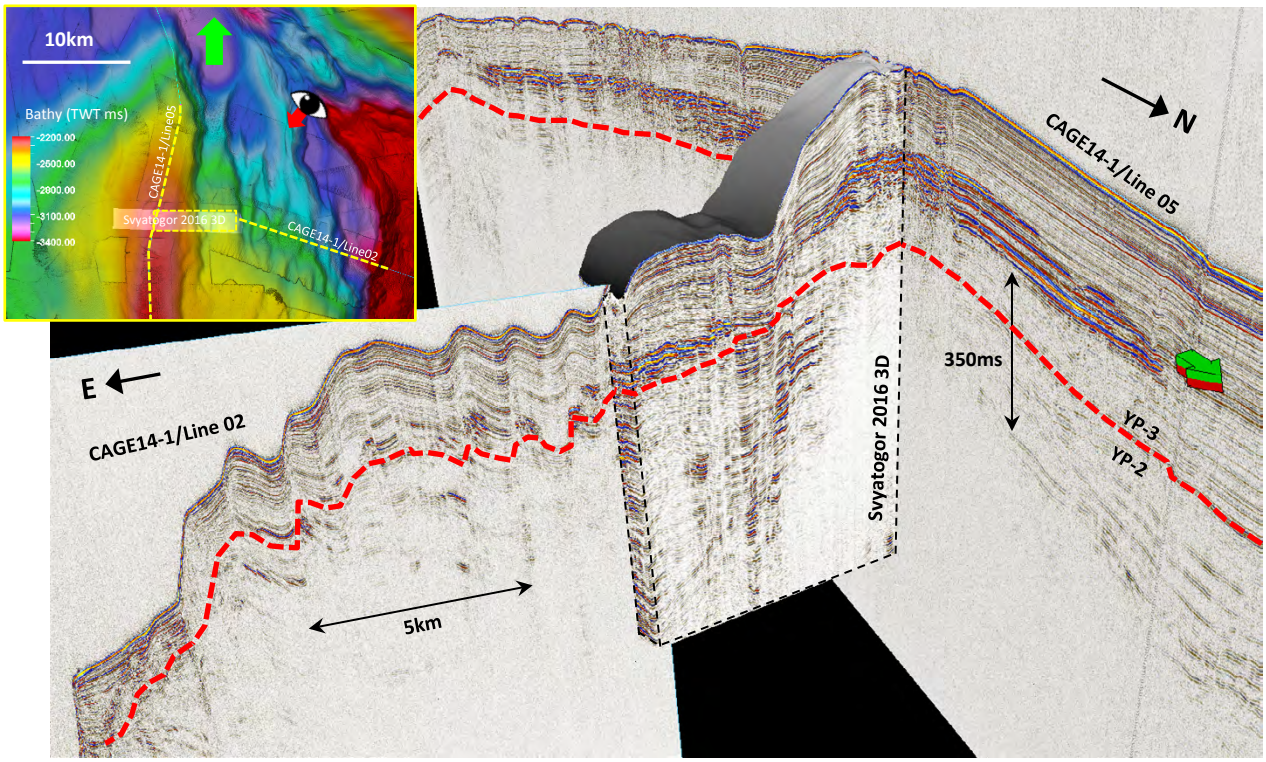


Figure 49 – Stratigraphic context of the eastern side of Svyatogor Ridge provided by high-resolution 2014 vintage 2D P-Cable seismic profiles. Red dashed line marks the YP-2 to YP-3 boundary corresponding to ~2.7Ma (Mattingsdal, et al., 2014). Inset location map highlights the visible parts of the seismic profiles.

4.1.2 Bottom Simulating Reflector (BSR)

A series of sub-bottom reflections with enhanced amplitudes were observed in the Svyatogor 2016 3D dataset and do not follow a consistent stratigraphic interval but rather follow the topography of the seafloor (Figure 50). The topmost of these reflections has reversed polarity (trough-peak) compared with the seafloor reflector (peak-trough), which indicates a negative acoustic impedance contrast in this minimum phase dataset (see Figure 50). The enhanced amplitude reflections topped by a negative acoustic impedance contrast is consistent with the presence of free-gas (Brown, 2011). The enhanced amplitudes terminate abruptly laterally and the pattern of these terminations also follows the topography of the seafloor. These characteristics are consistent with a gas-hydrate bottom-simulating reflector (BSR), where the top of the zone of enhanced amplitudes indicates the base of the gas hydrate stability zone (BGHSZ) as described in section 1.2 (pg. 4). The sequence of reflections with enhanced amplitude beneath the BSR indicates the presence of a well-developed free-gas zone. While not directly imaged, the free-gas zone is inferred to be trapped by an impermeable layer of gas-hydrate cemented sediment where the trap is enhanced by anticlinal structural closure as the BSR parallels the seafloor ridge morphology.

The BSR and underlying free-gas zone are referred to in the subsequent descriptions of stratigraphic development, faulting and interpreted fluid-flow features. This is because the BSR and free-gas zone are a prominent feature of the dataset and the attenuation through the free-gas zone has a strong impact on the seismic imaging and interpretability of the dataset. In addition, the position of the BSR and free-gas zone is important when describing faulting and fluid-flow because these may all be interrelated.

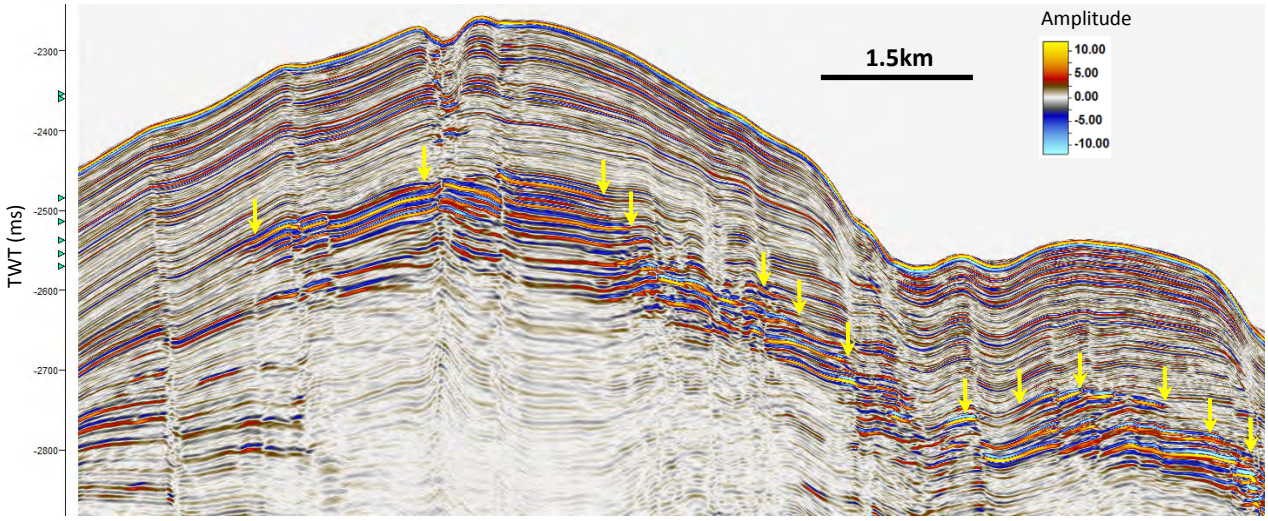


Figure 50 – Example of a bottom-simulating reflector (BSR) that corresponds to the base of the gas hydrate stability zone (BGHSZ). The level of the BSR is indicated with yellow arrows, enhanced amplitude seismic reflections below the BSR are due to the presence of free-gas. Example from the Svyatogor 2016 3D dataset (Inline 80).

4.1.3 Stratigraphic Development

Sedimentary reflections were imaged to a maximum depth of ~710-725m in the Svyatogor 2016 3D dataset (see Figure 51), using migration velocities (see section 3.9.1 pg.49) for depth conversion. While the loss of signal due to attenuation makes it difficult to characterise the strata below this level, further south on the ridge a defined basement contact was imaged with ~700m thickness of sediments

overlying oceanic crust (Johnson, et al., 2015). Thus, the loss of signal below ~710-725m may also correspond to the basement contact. Signal penetration along the crest of the ridge was poorer due to increased attenuation through a thicker free-gas interval and possible scattering effects associated with the large pockmarks that are concentrated along the ridge crest. As a result, the potential basement contact could not be followed across the whole length of the dataset.

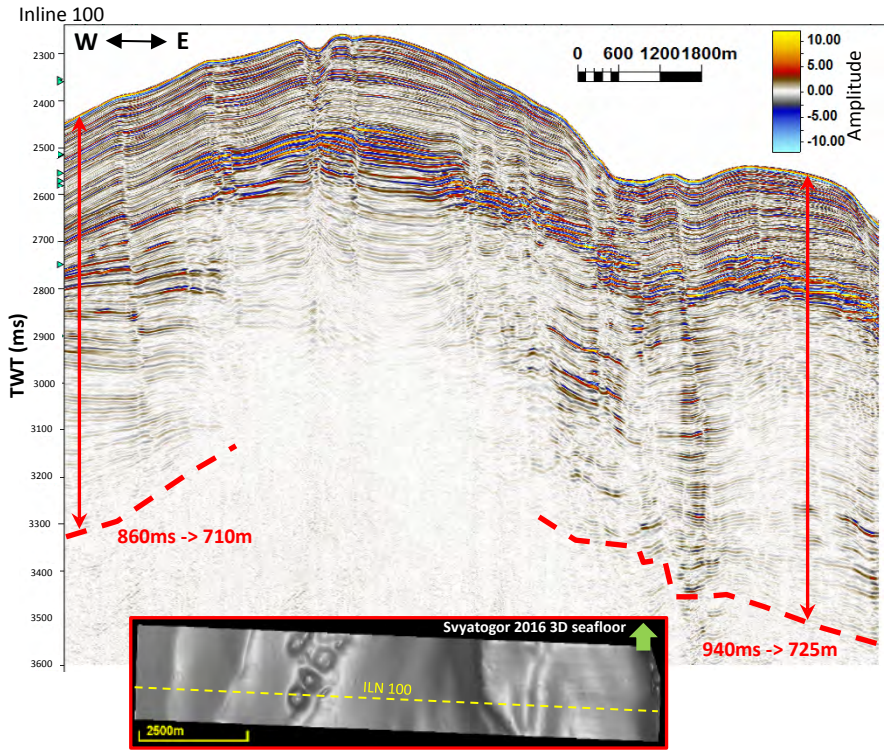


Figure 51 – the maximum depth of observed sedimentary reflections (red dashed line) indicates a sedimentary thickness of 710-725m, while the seismic imaging is not good enough to directly characterise this as a basement contact (sediments on oceanic crust) the thickness does coincide with the total sediment drift thickness of ~700m reported by Johnson et. al. (2015). Thicknesses were converted from TWT to depth (annotated) using migration velocities (ranging from 1400-1700m/s).

4.1.3.1 3D Horizon mapping

After establishing a structural framework, a series of seismic horizons were interpreted to investigate the stratigraphic development across the Svyatogor 2016 3D dataset (see Figure 52). The reduced penetration beneath the ridge crest was the main constraint on the deepest horizon that could be correlated across the extent of the dataset with a reasonable level of confidence. Reduced reflection continuity across the steeply dipping eastern flank of Svyatogor Ridge, where some slumping may have occurred, made it difficult to carry interpretation directly through this region. This problem was overcome by correlating vertical seismic sections from undisturbed areas on either side of this zone, using Petrel's 'ghost curve' tool, to ensure horizons were picked at a consistent level. In addition to the seafloor, five interpreted horizons (see Figure 52) form the basis for the investigation of the stratigraphic development of Svyatogor Ridge over the study area.

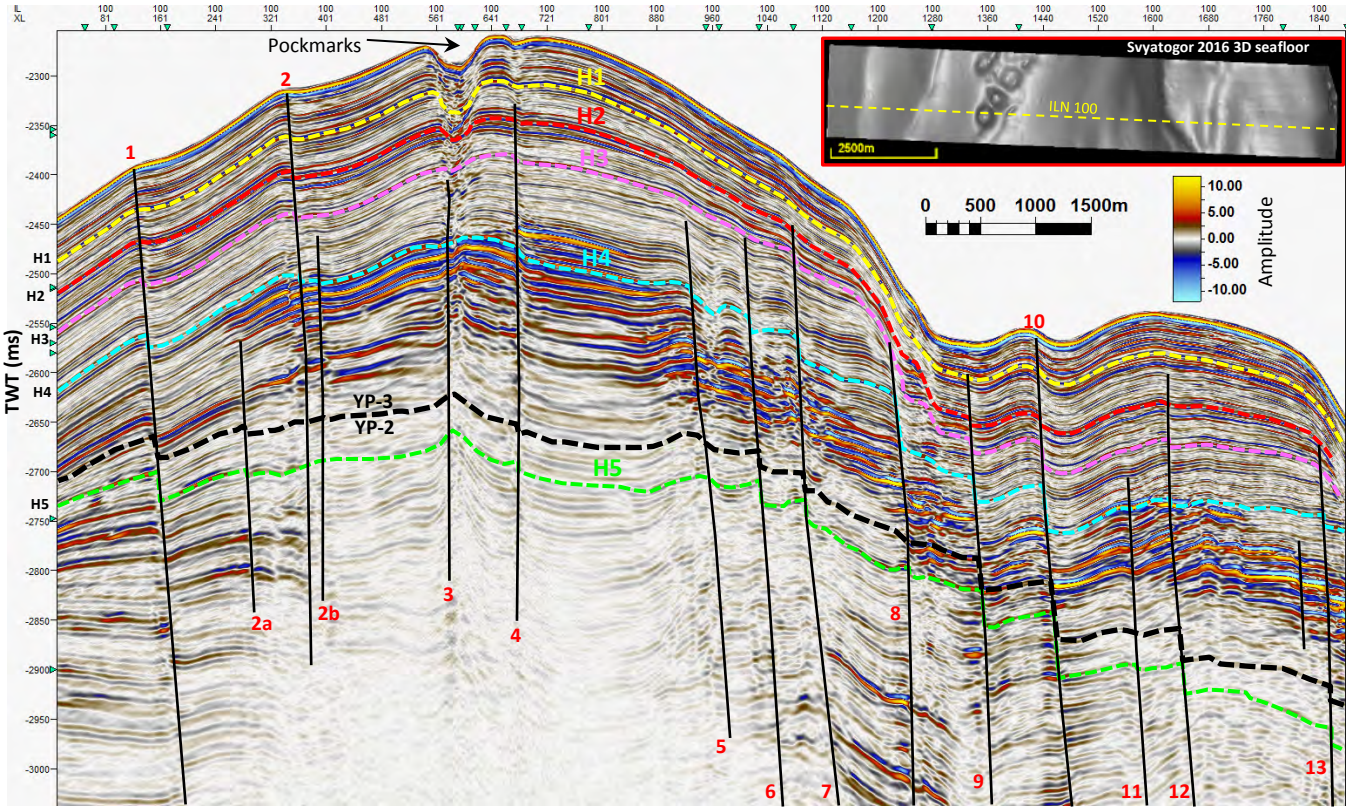


Figure 52 – Overview of interpreted subsurface horizons H1-H5. Position of YP-2 to YP-3 boundary corresponding to ~2.7Ma (Mattingsdal, et al., 2014) is marked with black dashed line as stratigraphic age control. Faults are indicated with black lines and follow the same numbering as in Figure 54*. The location of the seismic profile relative to seafloor features is indicated on the inset map.*

The stratigraphic development over the study area can be summarized using a series of isochore maps that illustrate changes in vertical thickness between the interpreted horizons (see Figure 53). The variation in thickness illustrated by these maps leads to the following interpretation, from most recent sedimentary interval to the oldest:

Seafloor to H1 – Thickening associated with pockmarks is an indication that the pockmarks were inactive with net deposition into the accommodation space created by an earlier active pockmark phase. Fault 2 is associated with the greatest growth during this period as indicated by the thickening of footwall strata. Slight thinning is observed on the steep eastern flank of the ridge and may be associated with local sediment slumping (seafloor slope is ~12° in this area).

H1 to H2 – Thinning associated with pockmarks indicate that the pockmarks were active during this interval leading to net erosion. Pronounced thickening of the footwall of Fault 10 indicates that it was the most active fault during this interval. Thinning on the steep eastern flank of the ridge indicates that sediment slumping may have been more significant during this interval than above H1.

H2 to H3 – Slight thinning is associated with most of the pockmarks indicating they may have been active, although some pockmarks appear to have been inactive and infilling during this interval. The strongest growth is recorded across Fault 1, although Faults 3 & 10 also appear to have been quite active.

H3 to H4 – Fault 3 records the strongest growth (footwall thickening) during this interval and in contrast to the younger intervals we see deposition focussed more along the axis of Svyatogor Ridge.

H4 to H5 – During this interval we also see deposition focussed along the axis of Svyatogor Ridge and surmise that the broad scale topography of the ridge may have developed during this interval (reflectors from H5 and below are relatively flat across the ridge axis). Thinning across the steep ridge flank area is interpreted to result from the onset of slumping due to this period of ridge growth and flank steepening. Above H5 it is interpreted that Fault 8 (see Figure 52) is influenced by slumping and mass wastage rather than simply tectonic movement.

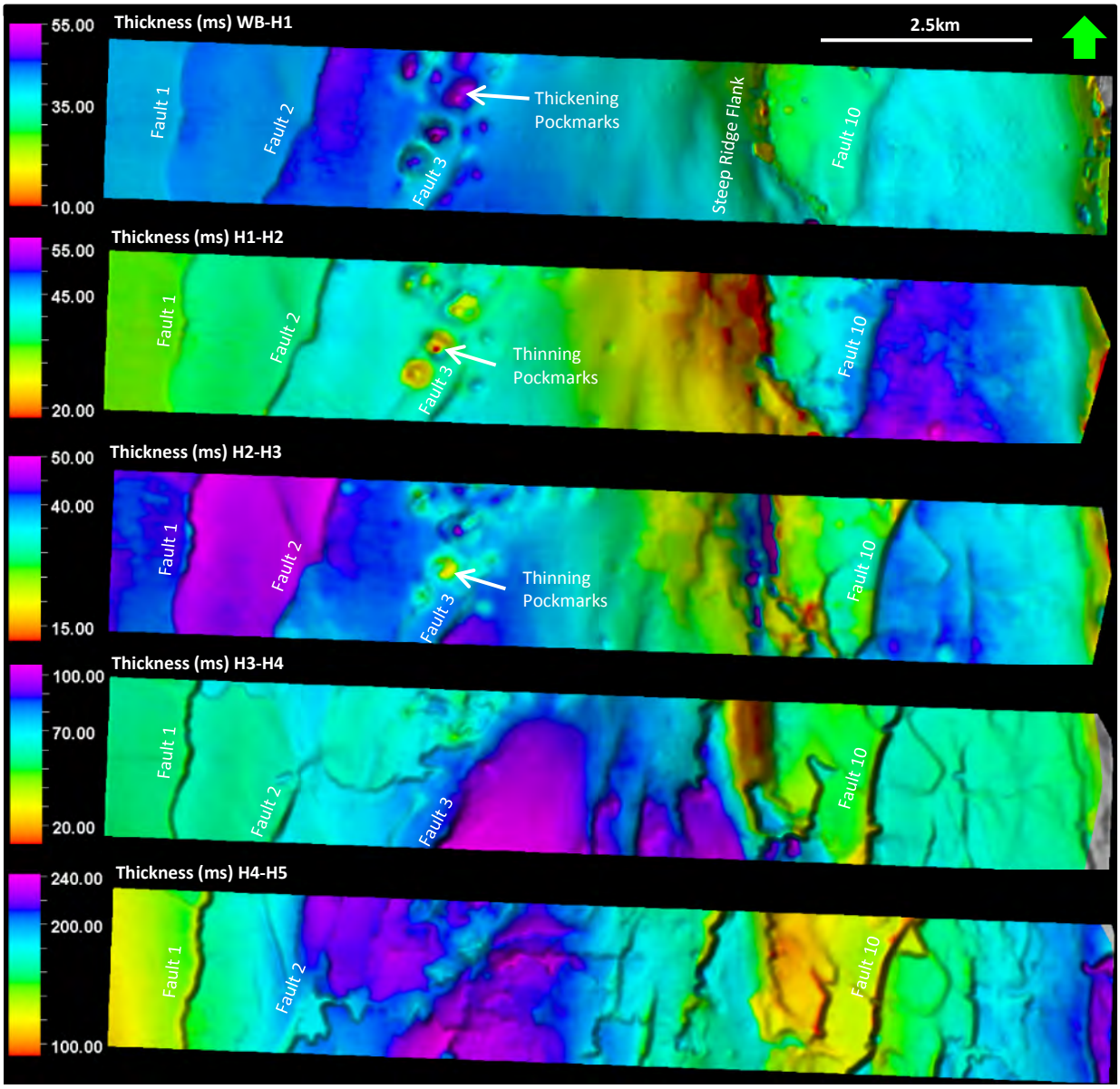


Figure 53 – Svyatogor 2016 3D isochore time thickness maps for interpreted horizons WB-H5 (see Figure 52*) illustrate the stratigraphic development of Svyatogor ridge. The most important faults are labelled as in* Figure 52 *for the uppermost stratigraphic interval.*

4.2 Faulting

4.2.1 Description of fault style

The 'variance' seismic attribute (which is the inverse of coherence) was used to give an overview of the faulting imaged in the Svyatogor 2016 3D dataset. Variance volumes with 5x5 trace filter-size and 15 sample vertical range were computed and then RMS averages were taken across key intervals above and below the BSR (see Figure 54b-d), which acted as a means of noise suppression and enhanced the appearance of faults. The variance volumes were calculated both horizontally (Figure 54b-c) and along structural dip (Figure 54d).

The dominant style of faulting imaged in the Svyatogor 2016 3D dataset consists of NNE striking synsedimentary normal faults that dip eastward i.e. towards Knipovich ridge (see Figure 54). The mean strike of this group of faults is ~014°, which is consistent with the northern Knipovich fault population described by Curewitz, et al., (2010). These faults can be classified as striking sub-perpendicular to the plate motion vector of 307° following the Curewitz, et al., (2010) classification scheme and indicates that the dominant stress regime in this area is related to crustal extension at Knipovich ridge. The faults typically dip at 50-60° but dips as high as 81° were measured on the upper sections of faults connected to fluid flow features (pockmarks and gas chimneys) and dips as low as 33° were measured on the deepest parts of some fault planes. Fault dips were calculated assuming average interval velocities of 1500-1700m/s.

A transition from discrete fault offset to fold-like deflections of reflections with a normal-drag geometry was observed along the upper portions of several of the normal faults (see Figure 60). The effect is also illustrated by comparing maps of variance attributes calculated horizontally with variance calculated along structural dip (see Figure 54c-d) and demonstrates that these fold-like features are common across the study area. These observations are consistent with an interpretation of fault propagation folding (Baudon & Cartwright, 2008; Sharp, et al., 2000) that is likely synsedimentary and related to accommodation of fault displacement by ductile processes (Baudon, 2007).

The pattern of faulting becomes more complicated towards the eastern end of the dataset (Figure 54), particularly below the level of the BSR, with fault strike becoming more variable and some occurrences of faults trending roughly E-W and dipping towards the north. In contrast to the dominant group of faults, none of these smaller faults with variable orientation were expressed at the seafloor, indicating that they are likely shorter-lived features. In map view, the strike of these faults ranges from oblique to parallel to the plate motion vector of 307° following the Curewitz, et al., (2010) classification scheme. Some faults striking along a more NE-SW trend than the main set of normal faults were also observed at the western end of the 3D dataset (see Figure 54) and are characterised by relatively small offsets compared to the main set of normal faults.

A significant number of circular variance anomalies appear in variance maps (Figure 54) and are typically spatially correlated with faults. These anomalies correspond to pockmark and gas chimney features that are described further in Section 4.5.2 (pg.79).

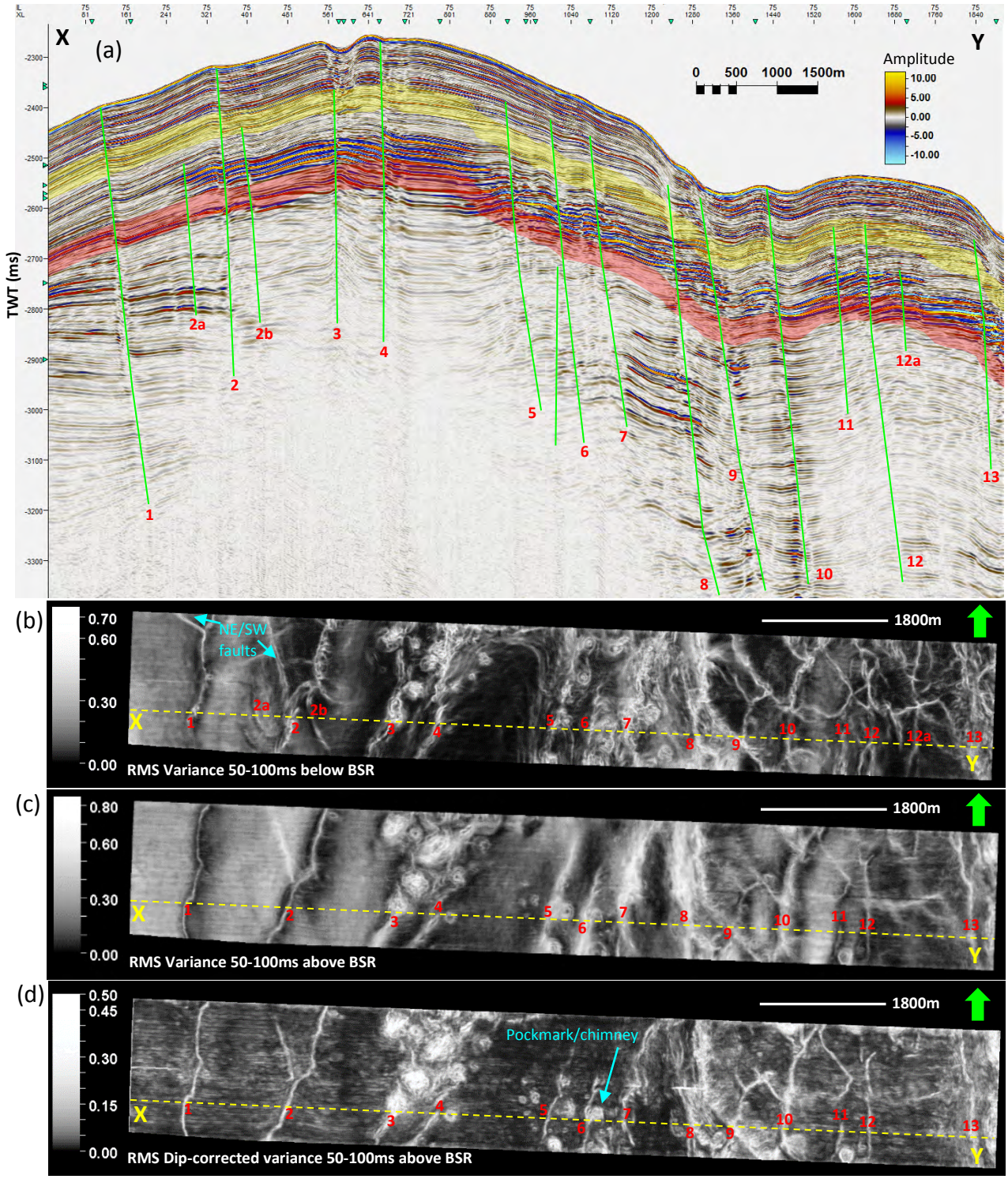


Figure 54 – Overview of the faulting observed in the Svyatogor 2016 3D dataset. (a) The vertical seismic section has been correlated with RMS variance maps (described in section 4.2.1 pg.60) using windows from (b) 50-100ms below the regional BSR (yellow highlighted zone on vertical section) and (c) 50-100ms above the regional BSR (red highlighted zone). (d) RMS variance extracted from a dip-corrected variance volume for the upper zone reduces the influence of fault-related folding. Circular variance anomalies correspond to pockmark/gas chimney features and red numbers denote the correlation of faults from vertical section to variance maps.

4.2.2 Fault throw analysis

A detailed analysis of throw distribution was conducted for a subset of the normal faults imaged in the Svyatogor 2016 3D dataset (see Figure 56). Vertical throw distribution (T-z) profiles were constructed by correlating reflections and reflector packages across faults for finely spaced marker horizons (see Figure 55) using iterative interpretation and horizon flattening to QC that correlations across the fault were valid. The thickness of the intervals between marker horizons on the hanging wall (T_{HW}) and footwall (T_{FW}) of the fault were compared via the growth index relation (Childs, et al., 2003):

$$Growth\ index = \frac{(T_{HW} - T_{FW})}{T_{FW}}$$

For a synsedimentary fault, the growth index is equivalent to the ratio between the relative rates of throw and footwall sedimentation. For example, if the sedimentation rate only slightly exceeds the displacement rate, the hanging wall stratigraphic thickness will be significantly greater than the footwall thickness, growth indices will be high and fault throw will decrease rapidly upwards within the growth sequence. Conversely, if the sedimentation rate greatly exceeds the fault displacement rate, growth indices will be low and fault throws will typically decrease slowly upwards (Childs, et al., 2003).

Growth faults record the interactions between sedimentation and fault-slip history, and can be useful in high-resolution kinematic analyses (Mansfield & Cartwright, 1996). For example, it has been established that intervals of zero gradient on T-z plots i.e. constant throw with depth, represent periods of fault inactivity, while intervals with positive slopes i.e. increasing throw with depth, typically represent periods of active growth faulting (Cartwright, et al., 1998).

It is also important to recognize that a fault may have both synsedimentary and post-sedimentary throw components (Baudon & Cartwright, 2008) because the movement of all or part of a fault surface may post-date a particular interval in a syn-faulting sequence. As a result, the terms syn- and post-sedimentary may not only distinguish one fault from another but may also distinguish between parts of the same fault surface active at different times (Childs, et al., 2003).

While fault history can be complex and lithologies with different mechanical properties, interval velocity uncertainties or compactional effects can further increase the level of complexity and cloud the interpretation, the most important aspect of the fault throw analysis is the qualitative evolution of slopes on the T-z plots. The correlation of similar slope changes across several T-z plots inside a basin can be viewed as robust features (Castelltort, et al., 2004).

A detailed study of fault throw was performed for a subset of six faults (denoted A-F, see Figure 56) across the Svyatogor 2016 3D dataset. These faults belong to the dominant class of NNE striking, eastward-dipping normal faults.

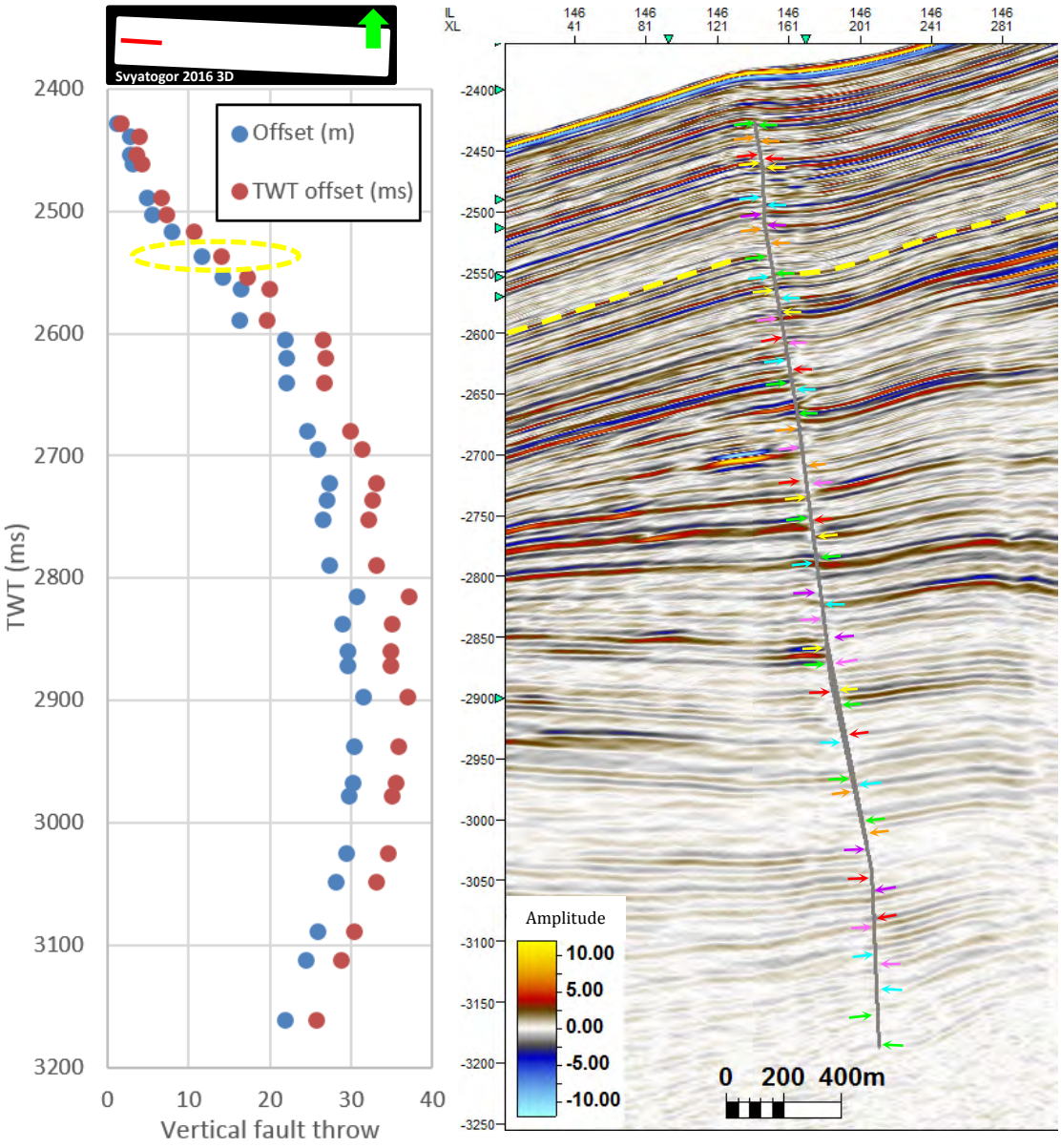


Figure 55 – Variation in fault throw by vertical TWT offset and depth converted offset using interval velocities from migration velocity model (velocities range from 1480-1700m/s), throws are plotted according to hanging wall TWT. Inset (top left) shows the location of the seismic profile (Inline 146 – Svyatogor 2016 3D) and coloured arrows along the seismic profile mark the horizon correlations across the fault that were used to determine the fault throw. The yellow dashed line marks the horizon referred to in Figure 59*. This fault is denoted Fault 1 (see Figure 52) or Fault A amongst the subset of faults in the throw analysis (see Figure 56).*

4.2.2.1 Observations of trends in vertical fault throw (T-z)

The faults in the Svyatogor 2016 3D dataset show a step-like trend in T-z plots characterised by an upper interval with high, positive throw gradient (throw increasing with depth) overlying an interval of very low, positive throw gradient and a lower interval of gradually decreasing throw (see Figure 55). This pattern was consistent across the study area (Figure 56), although the lower interval with negative throw gradient was only observed for the deepest imaged faults (i.e. faults A, E & F, which were imaged below ~3000ms). The boundary between the upper strong, positive throw-gradient and the underlying interval with very weak throw gradient does not follow a particular stratigraphic interval but rather follows the approximate trend of the BSR, although it typically occurs somewhat lower than the present BSR within the free-gas zone (see Figure 56).

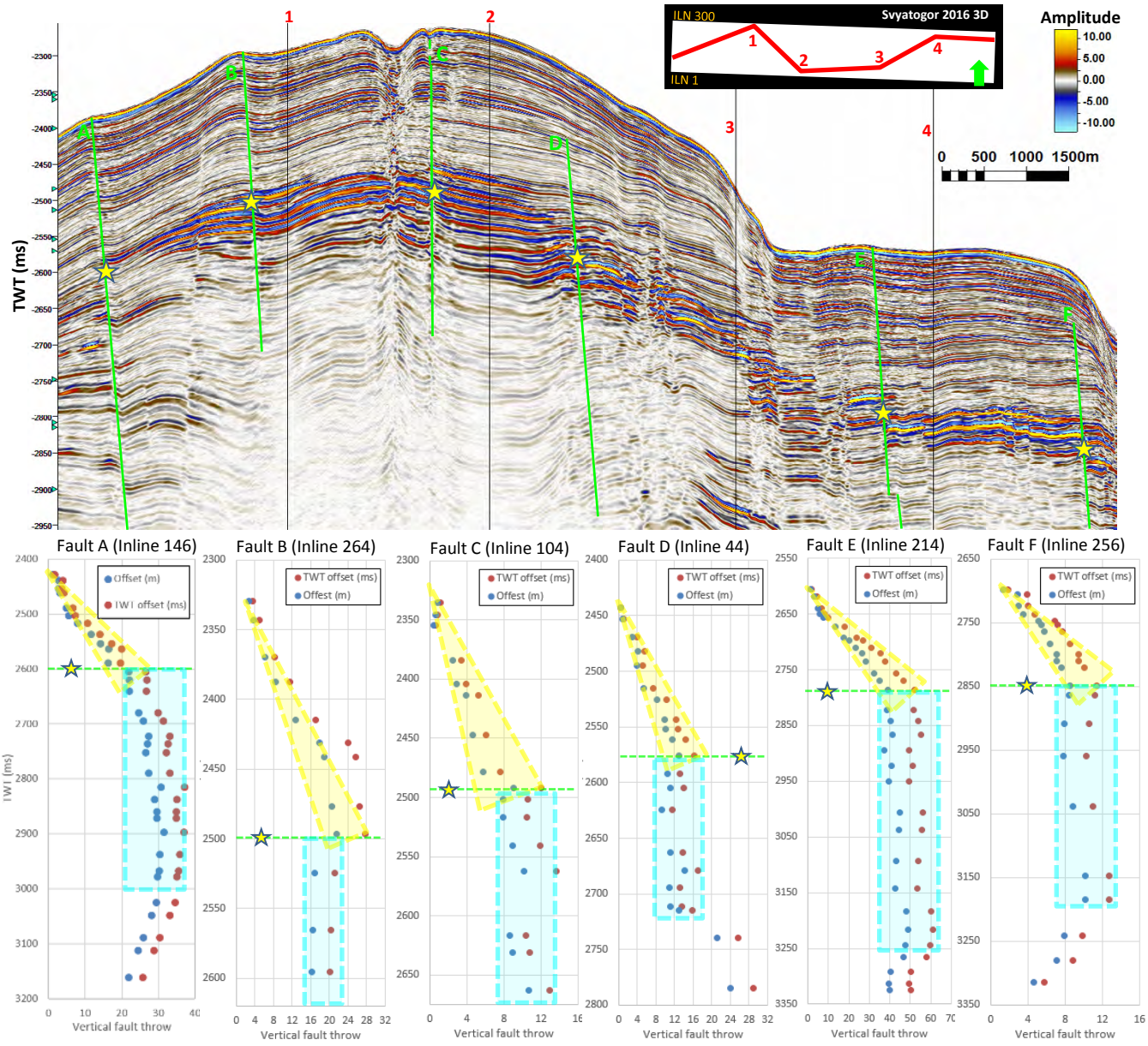


Figure 56 – Correlation of vertical fault throws (T-z profiles) across the Svyatogor 2016 3D dataset. Throws were interpreted on six faults (A-F) and picked on inlines (that were near perpendicular to strike) where offsets could be most easily tracked by successive flattening on interpreted horizon segments. Seismic profile is a cross section through locations where fault throws were picked. In all cases an upper zone characterised by increasing throw with depth (yellow highlight) is separated from a zone of relatively constant throw with depth (blue highlight) by a marked horizon (green line/yellow star) that follows the topography of the BSR more than a particular stratigraphic horizon when plotted on vertical seismic section (yellow stars).

The upper interval that is characterised by strong positive throw-gradient is also associated with high growth indices (>0.1) indicating fault interaction with the free-surface and active growth faulting (see Figure 57). The measured growth indices were quite consistent when plotted by sub-bottom depth, with high growth indices typically observed in an interval extending 200m below the seabed, and lower growth indices scattered around zero below this depth (Figure 57).

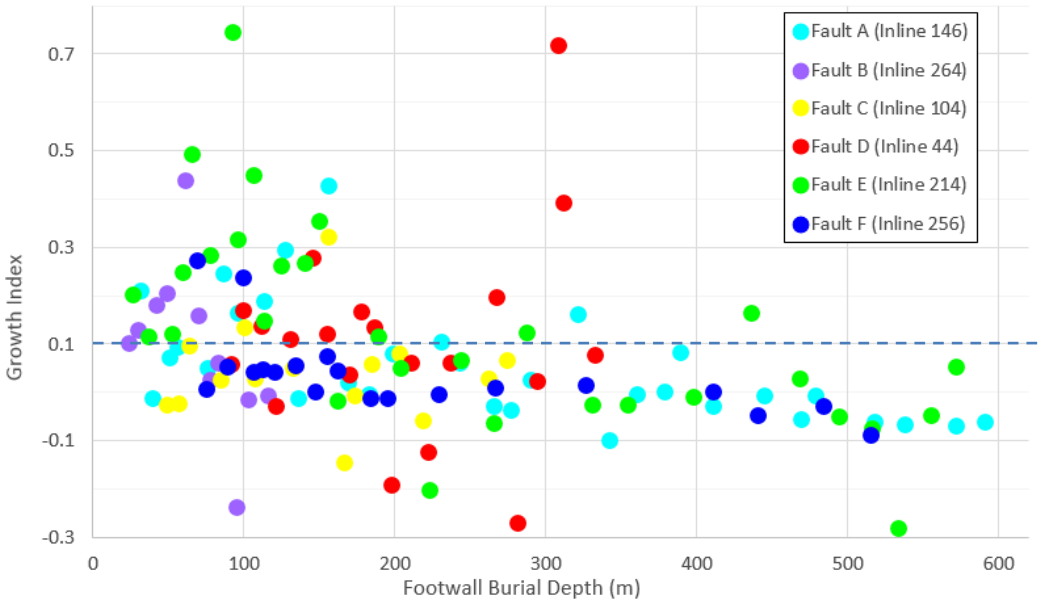


Figure 57 – Growth index for Faults A-F (see Figure 56) versus burial depth. Burial depth was calculated by subtracting the seafloor and using the average interval velocity between the seafloor and the footwall marker horizons for depth conversion.

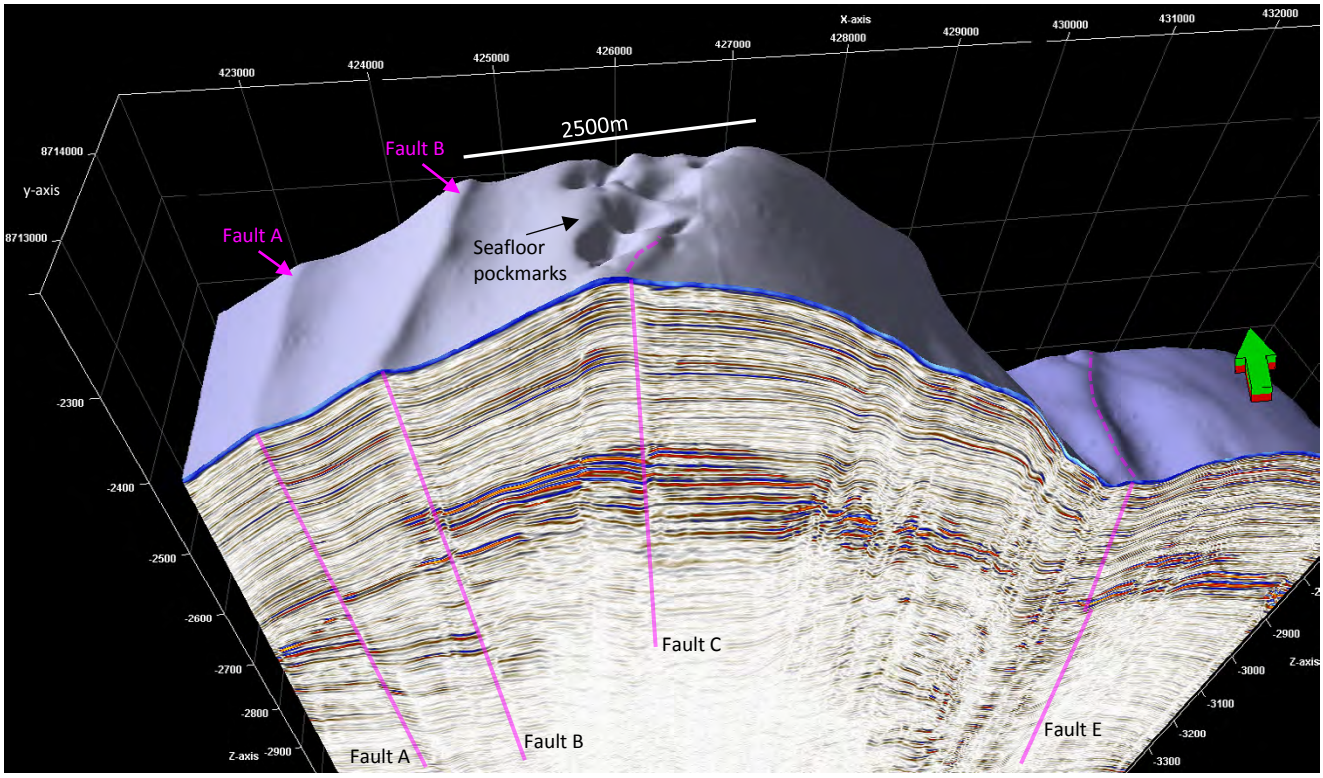


Figure 58 – 3D projection showing the seafloor expression of faults A, B, C & E (see Figure 56*). Fault C is associated with a fault scarp at the seafloor transitioning into an elongated pockmark above the lateral fault tip (see* Figure 61*). Z-scale is 10X with illumination from West at 30°.*

Faults A, B, C & E all show clear association with seafloor topography, indicating that they have been recently active (Figure 58). The sharpest seafloor scarp is associated with Fault B indicating that it may have been active most recently, while Fault A shows a smoother scarp indicating less recent or weaker activity relative to the rate of sedimentation (this is also consistent with the trend shown by

isochore thickness maps in Figure 53). By contrast, Fault D terminates well below the seafloor and transitions from discrete fault offset of reflections to monoclinal fold-like deflections with a normal-drag geometry (Figure 60) that is consistent with an interpretation of fault propagation folding that is likely related to accommodation of fault displacement by ductile processes (Baudon, 2007) possibly enhanced by upward migration of fluids along pockmark conduits leading to reduced sediment strength.

4.2.2.2 Observations of along strike fault throw (T-x)

Throw-distance (T-x) plots were constructed by extracting the throw along a fault at a fine spatial sampling for selected horizons (e.g. Figure 59). Horizons were selected based on strong amplitudes and good reflector continuity that facilitated precise mapping. The observation of varying fault displacement along strike for Fault A, with distinct intervals separated by local throw minima (Figure 59) is consistent with fault growth by segment linkage (Cartwright, et al., 1995). The hydrocarbon industry has played a leading role in understanding the importance of fault segmentation where it has been shown that breaks in fault continuity between fault segments provide potential flow zones through which hydrocarbons can migrate across a faulted region (Kattenhorn & Pollard, 2001).

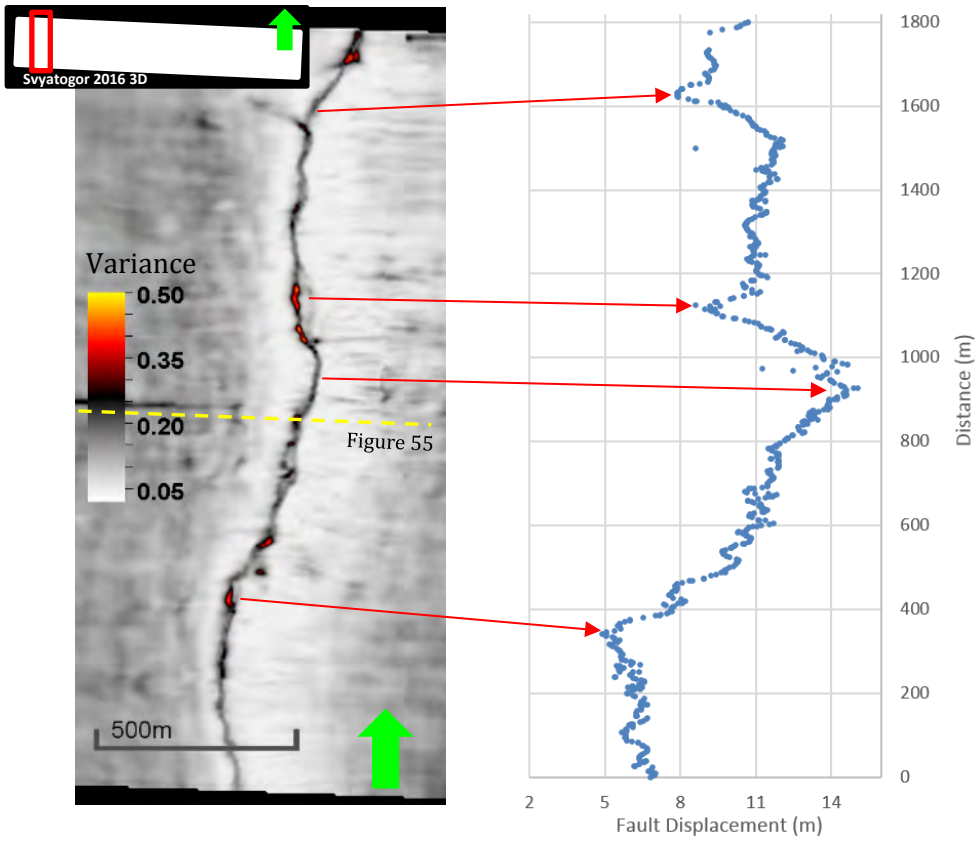


Figure 59 – Fault A throw variation along strike (T-x).Variance extracted along horizon highlighted in Figure 55 *has been correlated with a strike profile of fault displacements (computed for an average interval velocity of 1650m/s across the fault zone). Yellow dashed line marks the position of the profile shown in* Figure 55*.*

Along Fault D, several gas chimney features are observed, indicative of focussed fluid flow (Figure 60 & Figure 71) and marked by circular zones of high seismic variance. Significant variation in throw is observed along the length of the fault and local throw minima tend to be associated with variance anomalies related to gas chimneys. This could be an example of fluid flow being localised along the damage zones linking fault-segments, although the disturbance to seismic imaging created by the gas chimneys makes this assessment uncertain.

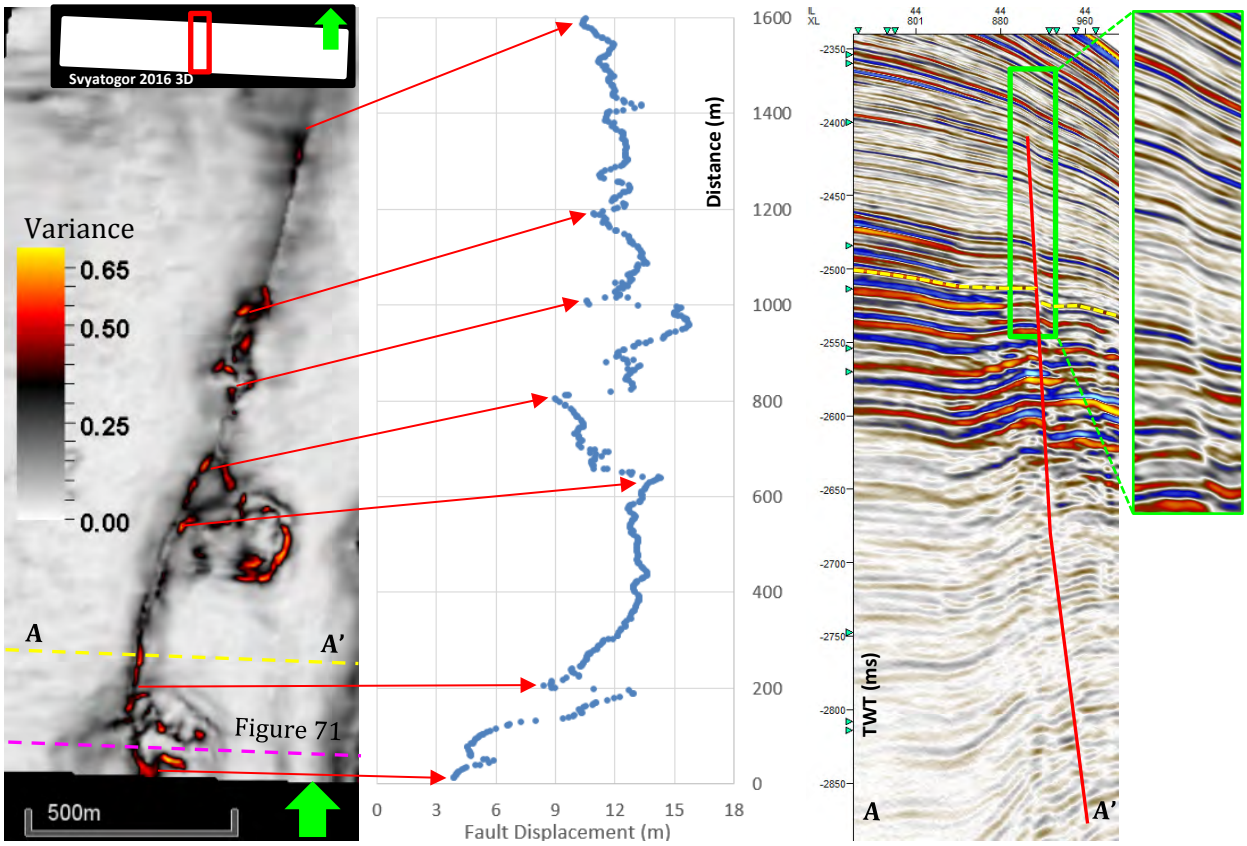


Figure 60 – Fault D (see Figure 56) that is associated with gas chimney/pockmark structures (position of transect from Figure 71 *that illustrates a gas chimney is marked), corresponding throw-distance (T-x) plot and vertical seismic section are also shown. Enlarged view of the upper fault tip shows transition from discrete fault offset of reflections to fold-like deflections with a normal-drag geometry. Upper part of fault dips at 59° and lower part (below 2680ms) dips at 33° (assuming 1600m/s interval velocity), and may indicate dip-linkage with a deeper pre-existing fault.*

Fault C also shows evidence of fault controlled, localised fluid migration where an elongated pockmark is observed directly above the lateral fault tip (see Figure 61). The pockmark is smaller and steeper sided than the nearby pockmarks, indicating that it may have developed more recently, and the long axis of the pockmark is aligned directly with the strike of the fault. The lateral variation in throw along the associated fault is generally consistent with the gradual decline from fault centre to tip associated with a radially propagating fault segment, although a small local increase in throw was observed directly beneath the pockmark (Figure 61).

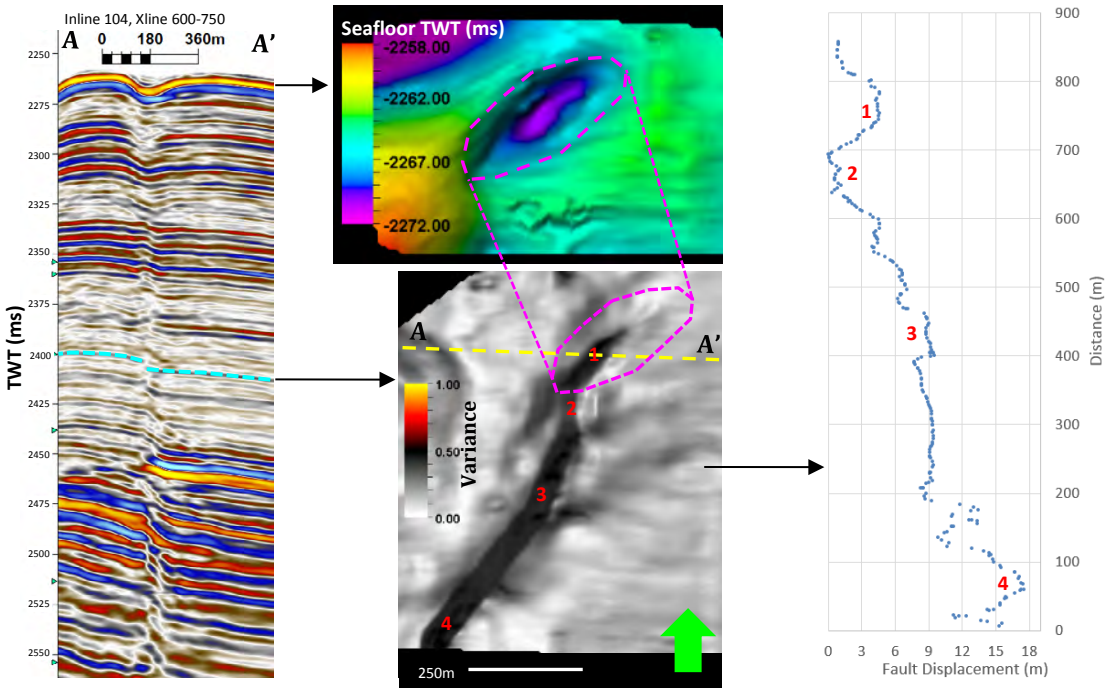


Figure 61 – Elongated seafloor pockmark (approx. 315m long, 150m wide and 4.5m deep) located above the lateral tip (illustrated by variance map) of Fault C (see Figure 56*) at the crest of Svyatogor ridge. Throw-distance plot (T-x) indicates that the fault displacement decreases steadily towards the lateral fault tip but a local increase in displacement occurs beneath the pockmark*

4.2.2.3 Interpretation of fault kinematics

All of the interpreted faults show a strong contrast in throw gradient on T-z profiles where an upper (recent) section with strong positive throw gradient is distinguished from a lower (older) zone with much weaker throw gradient (Figure 56). This change in throw gradient does not appear to correlate with a consistent stratigraphic horizon and rather follows the topography of the BSR, but shifted downwards by 25-50ms to within the present free-gas zone.

The strong throw gradients and high growth indices associated with this recent period of fault activity indicates that fault movement was synsedimentary and that interaction with the free surface occurred. A large proportion of the throw recorded on the faults accumulated during this period. The synsedimentary movement of faults A, B, C & E is further highlighted by their association with present day seafloor topography indicating that interaction with the free surface is ongoing. However, it is important to recall that the movement of all or part of a fault surface may post-date a particular interval in a syn-faulting sequence i.e. it is still possible that post-sedimentary slip has occurred on faults that began as synsedimentary growth faults (Childs, et al., 2003; Baudon & Cartwright, 2008).

Fault D is also interpreted to have developed as a synsedimentary growth fault. The fact that the upper tip line of the fault does not reach the seafloor, nor do the pockmark/fluid conduit features associated with this fault, indicates that this fault became inactive while faults A, B, C & E continued to develop. The change in dip and sudden change in throw observed on the lower part of the fault plane (below ~2680ms) indicate that the growth of fault D may have been controlled by dip-linkage with a deeper pre-existing fault.

The decrease in throw on the lowermost sections of Fault A, E & F could indicate that initial fault nucleation occurred below the free surface and the lowermost section therefore reflects downward propagation in a blind manner. Similar fault behaviour was described by Baudon & Cartwright (2008b) who argue that fault nucleation at the free surface is unlikely due to the low confining stress, combined with the typically high porosity and low shear strengths of near-surface sediments. Unless nucleation occurs at the free surface, synsedimentary faults will almost invariably experience a degree of propagation as a blind fault, since propagation of the upper tip to the free surface necessarily proceeds while the tip line propagates laterally and downward in a blind manner (Childs, et al., 2003; Baudon & Cartwright, 2008b). The amount of displacement added after the transition from blind fault to growth fault decreases towards the lower fault tip so that a low negative throw gradient is preserved (Baudon, 2007). An alternative explanation would be that the negative throw gradients on the lowest imaged sections of the faults could be caused by dip-linkage with deeper faults (Mansfield & Cartwright, 1996) in the underlying oceanic crust that have been previously described (Johnson, et al., 2015; Waghorn, et al., 2015) but were not imaged by the Svyatogor 2016 3D survey.

The gas chimney and pockmark features observed along faults C & D are consistent with a study of hydrothermal systems by Curewitz & Karson (1997) who found that hydrothermal outflow occurs most commonly at fault tips and in the zones where multiple faults interact such as at the junctions between fault segments. In the case of fault C an elongated pockmark appears to be associated with the lateral fault tip while gas chimney features along fault D may be associated with damage zones linking fault-segments (see section 4.2.2.2 pg.66).

4.3 Investigation of the bottom-simulating reflector (BSR)

4.3.1 Detailed mapping of the BSR

As described in section 4.1.2 (pg. 56), a bottom-simulating reflector (BSR) representing the base of the gas hydrate stability zone (GHSZ) was observed across the Svyatogor 2016 3D dataset. The BSR surface was mapped in detail using a combination of manual interpretation and auto-tracking on the seismic attribute “sweetness”. Manual interpretation was based on the recognition of abruptly terminating or cross cutting reflections with anomalously high amplitude and reversed-polarity. The seismic attribute “sweetness” is calculated as the reflection strength (also called “envelope”) divided by the square-root of the instantaneous frequency. This attribute highlights the free-gas zone due to the strong impedance contrast and rapid attenuation of high frequencies associated with the presence of free-gas. Autotracking along the top of the anomalously high sweetness response provided an additional constraint for the detailed mapping of the BSR (Figure 62).

Consistent with previous mapping (Westvig, 2015), the Svyatogor 2016 3D datasets shows a BSR that is asymmetrically distributed along the eastern side of the ridge (e.g. Figure 51, Figure 63, Figure 65). This trend is further highlighted when viewed in the context of regional 2D profiles (Figure 48 & Figure 49). This trend could imply some degree of migration up the eastern flank of the ridge is occurring i.e. from the direction of Knipovich ridge and focussed along the axial detachment faults.

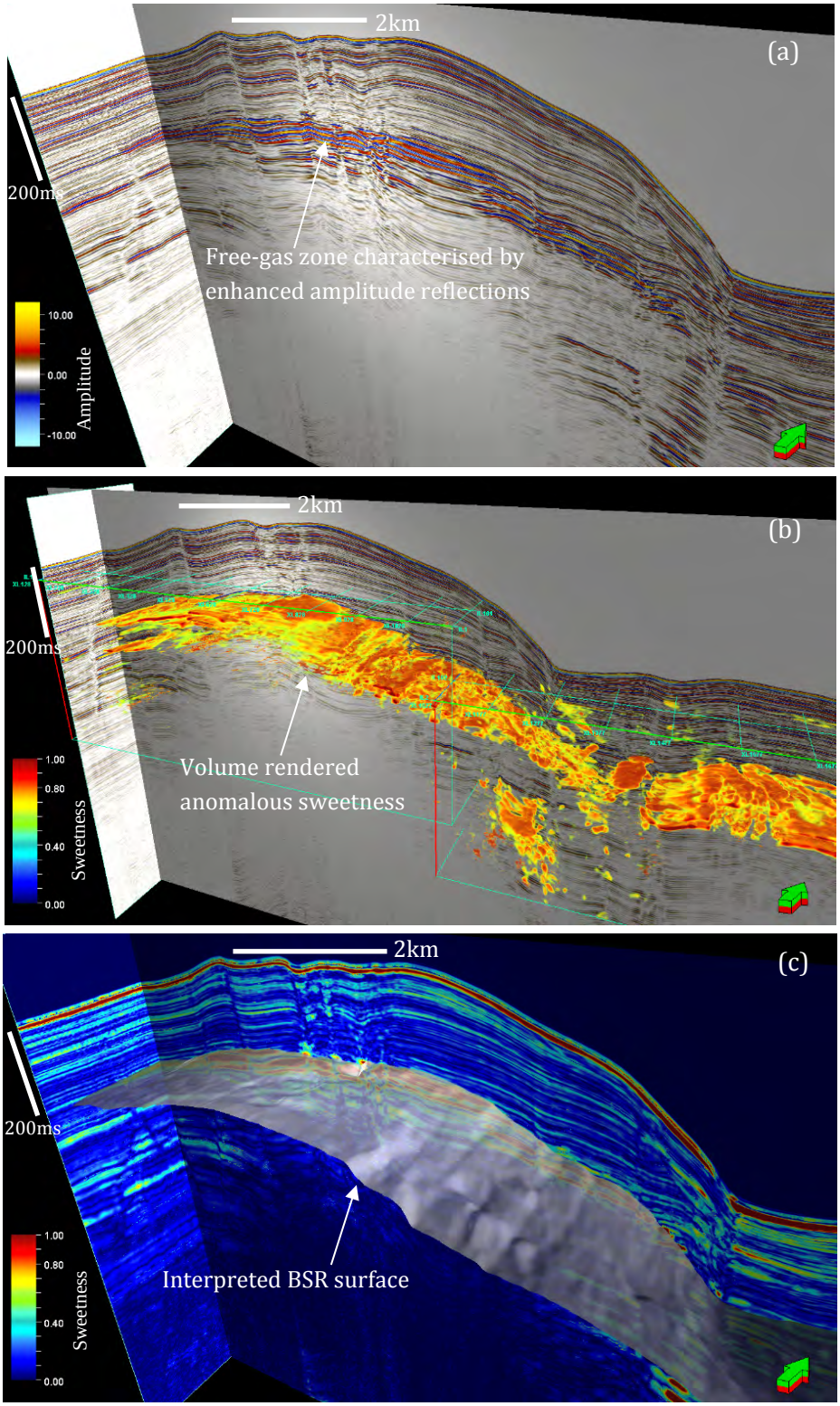


Figure 62 – the free-gas zone is characterised by high amplitude reflections (a) with an anomalous loss of high frequencies, giving a strong sweetness attribute response (b). The top of the sweetness anomaly was used to assist tracking of (c) the interpreted BSR surface.

4.3.2 Forward modelling of the BSR

To investigate the controls on the BGHSZ in the study area, the depth of the BSR was forward modelled using standard gas hydrate phase boundary methods (Sloan & Koh, 2008). The primary model input was the interpreted seafloor horizon, which defines an isotherm equal to the bottom water temperature, assumed constant at -1°C across the study area, consistent with World Ocean Database measurements (Boyer, et al., 2013). A hydrostatic pressure regime (which is reasonable for contouritic drift sediments at BSR depths) was assumed. A pore water salinity of 35‰ was used for pressure calculations. The short-offset P-Cable data acquired over the study area provides limited velocity information so the average seismic velocity between seafloor and BSR was simply assumed to be 1500m/s, consistent with migration velocities used during seismic processing. In the absence of geochemical sampling of the gas at Svyatogor Ridge, the gas composition was simply assumed to be 100% methane.

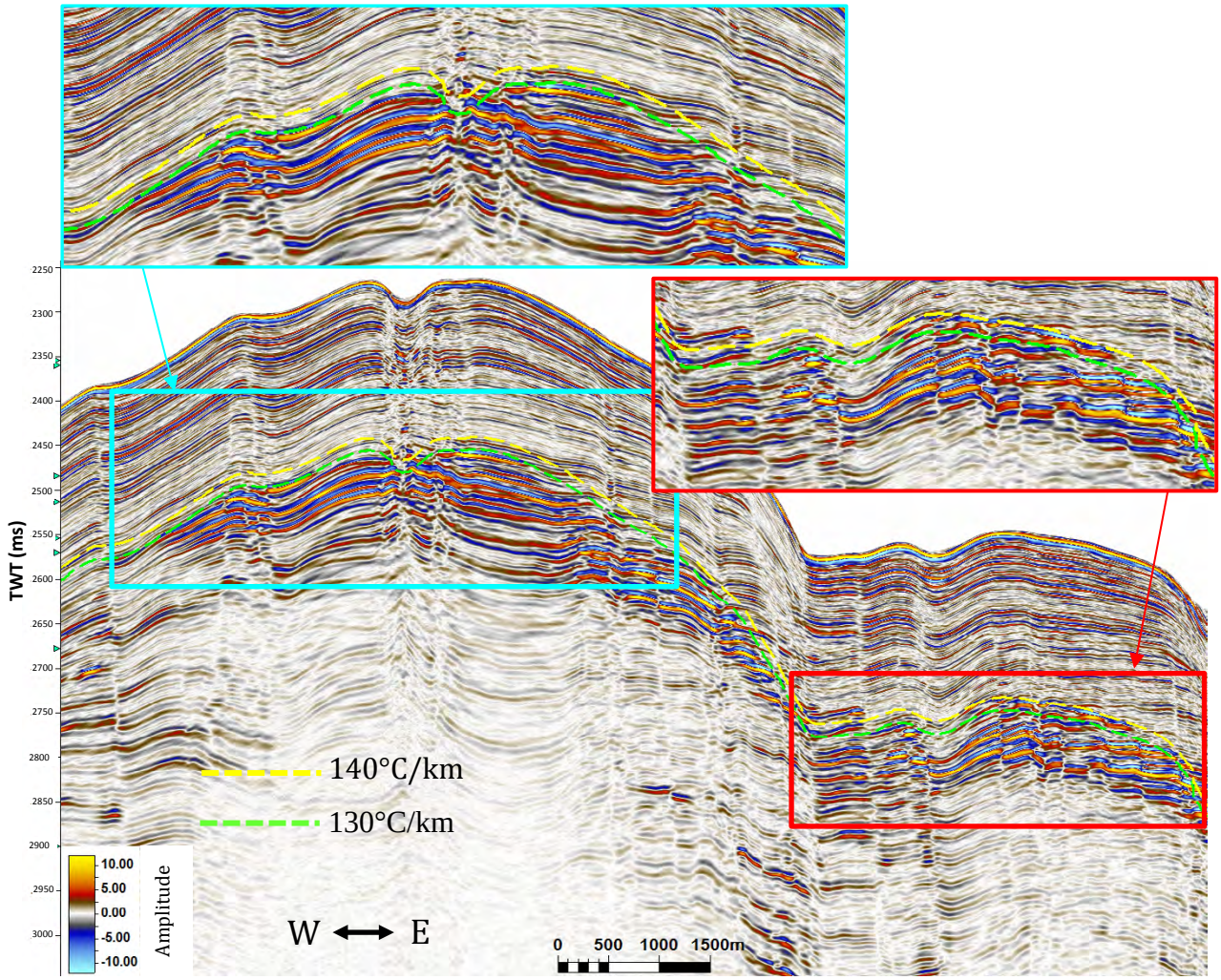


Figure 63 – Forward models of BSR position based on standard gas hydrate phase boundary conditions (Sloan & Koh, 2008), assuming 100% methane, seafloor temp. -1°C, hydrostatic pressure and pore-water salinity of 35‰, 1500m/s average velocity of overlying sediments and geothermal gradients of 130°C/km (green dashed line) and 140°C/km (yellow dashed line).

The forward models showed good fit with the observed BSR with geothermal gradients of 130-140°C/km (Figure 63). A geothermal gradient of 130°C/km showed reasonable fit across the majority of the study area and is consistent with a borehole measurement of geothermal gradient from Crane, et

al., (1991) that coincides with the Svyatogor 2016 3D dataset. A notable exception was an area with an anomalously shallow BSR at the eastern end of the study area, where a modelled geothermal gradient of 140°C/km was required to fit the observed BSR. Anomalously shallow BSRs may be caused by either high concentrations of hydrate above the BSR leading to anomalously high velocity, or by upward advection of heat and fluids (Hornbach, et al., 2012). This is discussed further in section 5.4.3 (pg. 92).

4.3.3 Inversion of BSR for geothermal gradient

To further investigate the anomalies in the depth of the observed BSR, the interpreted BSR surface was inverted to estimate geothermal gradient. This inversion is reasonable because the BSR can, to a first order approximation, be thought to mark an isotherm, due to the strong influence of temperature on hydrate stability (Shankar, et al., 2010). BSR depth variations can consequently give an indication of local variations in heat flow, which can result from recent fluid flow or faulting (Shankar, et al., 2010).

In this study, the BSR inversion was carried out using the Python script provided as Appendix 4. To invert the BSR for geothermal gradient it is assumed that the position of the BSR corresponds to the base of the gas hydrate stability zone (BGHSZ). The pressure at the BSR was assumed hydrostatic and was calculated for a seawater density of 1027kg.m^{-3} and a seismic velocity of 1479m.s^{-1}. The temperature at the BGHSZ was calculated assuming 100% methane and 3.5% NaCl by interpolating along the Sloan & Koh (2008) limit of hydrate stability curve (see Figure 64).

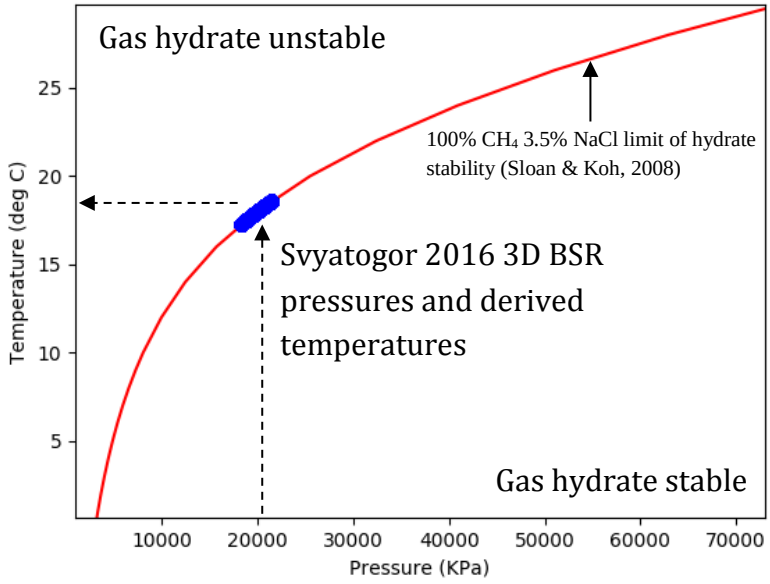


Figure 64 – BSR pressures (assumed hydrostatic with density of 1027kg.m^{-3}) and derived temperatures from limit of hydrate stability relation for 100% CH_4, 3.5% NaCl (Sloan & Koh, 2008).

The temperature at the water bottom is assumed to be -1°C across the study area, consistent with World Ocean Database measurements (Boyer, et al., 2013). The thickness of the hydrate stability zone was calculated by depth conversion and subtraction of the seafloor and BSR interpreted surfaces using a velocity of 1479m.s^{-1} and ranges from 127-182m across the study area. The geothermal gradient is then simply given by:

$$GTG = \frac{Temp_{BSR} - Temp_{Seafloor}}{Depth_{BSR} - Depth_{Seafloor}}$$

Geothermal gradients ranging from 102-145°C/km were derived by inversion of the BSR and the spatial variation in geothermal gradient is illustrated in Figure 65. High geothermal gradients were associated with structural highs, both along the main ridge and a small topographic high towards the eastern end of the dataset where the highest values were observed. A relatively cooler zone was observed on the ridge flank, although with geothermal gradient in the 102-110°C/km range this zone may be more representative of a regional mean level of heat flow, rather than a truly anomalously cool zone. The BSR derived geothermal gradients compare well with the Crane, et al., (1991) data point of 131°C/km that coincides with the 3D survey but are systematically somewhat higher than the recent heat flow probe measurements of Bohrmann, et al., (2017). Such a systematic overestimation of geothermal gradient may indicate that the seismic velocity of 1479m.s^{-1} was too low for correct depth conversion of the BSR surface.

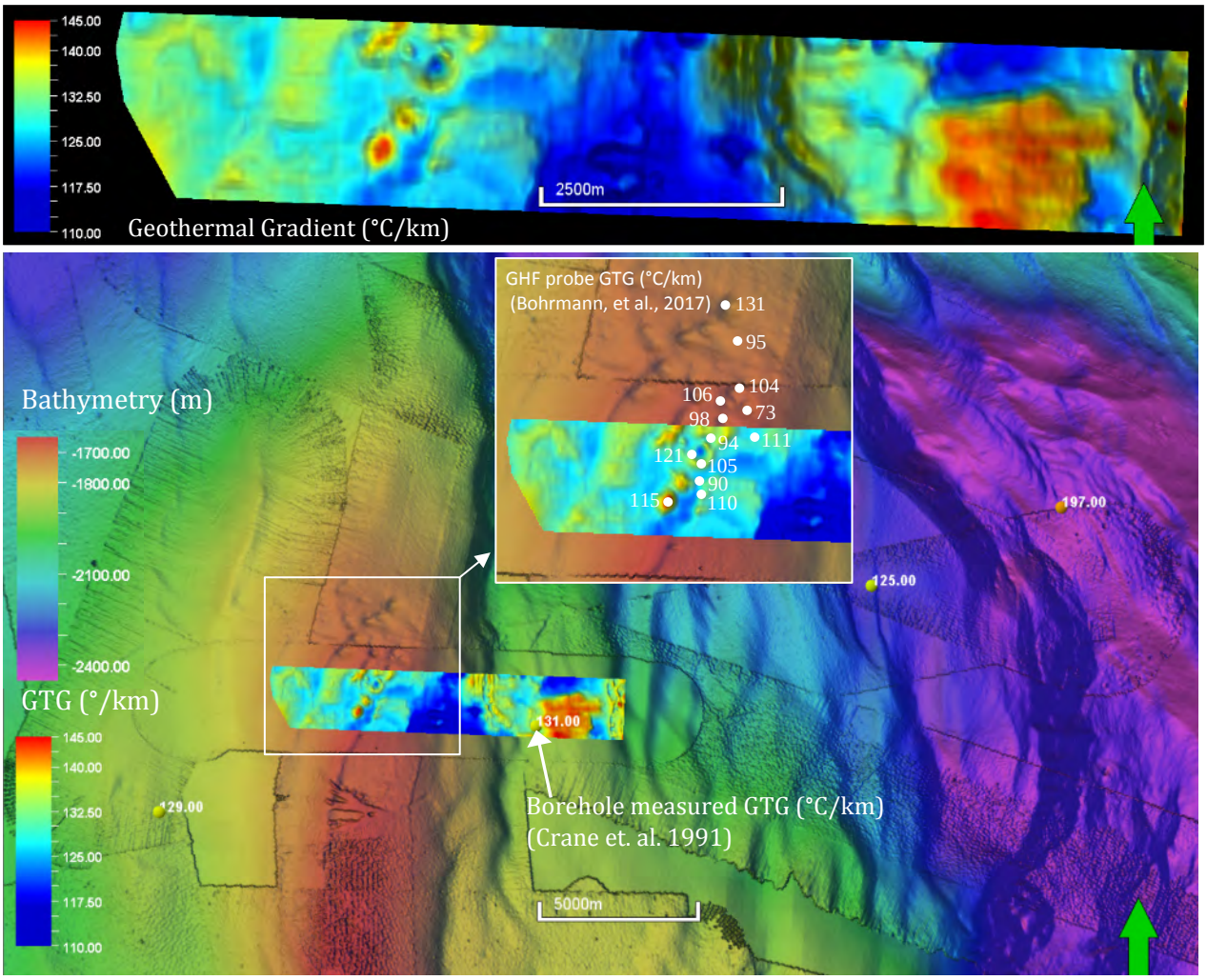


Figure 65 – Geothermal gradient derived from inversion of BSR (see section 4.3.3 pg.72), circles are labelled with borehole measured geothermal gradients (Crane, et al., 1991) and an enlargement of the pockmark cluster shows the Giant Heat Flow Probe (GHF) geothermal gradient measurements of Bohrmann, et al., (2017). A local zone of elevated geothermal gradient occurs towards the Eastern end of the Svyatogor 2016 3D dataset and a zone of decreased heat flow occurs centrally in the dataset along the flank of Svyatogor Ridge.

4.4 Size of the free-gas system

As described in section 4.1.2 (pg. 56), a well-developed free-gas zone is trapped beneath the BSR in the study area. To estimate the size of the free-gas system it was assumed that the signal penetration was sufficient to allow the entire vertical extent of the gas zone to be imaged. Sedimentary reflections without the enhanced amplitudes associated with the presence of gas were imaged across the Svyatogor 2016 3D dataset beneath the free-gas zone. However, it is still possible that strong seismic attenuation through the free-gas zone may have prevented the base of the free-gas system from being imaged. The vertical extent of the free-gas zone could therefore be greater than the imaged extent.

The imaged free-gas zone was mapped using geobody extraction (at a voxel size of 6.25m x 6.25m x 0.5ms) based on anomalously high sweetness (Figure 66a). The extracted geobody consists of a single connected geoblob and suggests that a continuous, interconnected free-gas phase of significant thickness could be present. However, it is expected that the thickness of a continuous gas-phase would only be a fraction of the total gas-zone thickness (Flemings, et al., 2003).

The geobody was interpolated onto a 3D grid with resolution of 6.25m x 6.25m x 2.5ms (~64 million total grid cells) and horizon parallel layering that conforms to the top of free-gas surface. The interpreted free-gas zone was mapped to ~2.5 million grid cells (cell volume of 6.25m x6.25m x1.875m = 73.24m^3) with a total bulk volume of 0.187km^3 (Figure 66a). The vertical thickness of the free-gas zone was then calculated (Figure 66b-c) and is taken as the upper-bound of the possible column height of an interconnected free-gas phase after Hornbach et. al. (2004).

The dominant lithology recorded in ODP boreholes 908-912 for sub-bottom depths corresponding to the depth of the free-gas zone on Svyatogor ridge is silty-clay with variation in coarse fraction controlled by the varying influence of the West-Spitsbergen current over Pleistocene glacial-interglacial cycles (Ocean Drilling Program, 2005; Wolf-Welling, et al., 1993). It was assumed that the sediment physical properties from these boreholes should give at least a broad indication of the expected physical properties within the Svyatogor 2016 3D dataset for similar sub-bottom depths. Based on this assumption the porosity within the free-gas zone was estimated from the 100-200mbsf average of ODP sites 908-912 as 0.47±0.07 (std. dev.).

The average gas saturation within the free-gas zone is difficult to constrain from seismic reflection data alone. While seismic reflectivity can be used to discriminate between fully water saturated sediments and sediments containing gas (which tend to show anomalously high-amplitude, phase-reversed reflections) the reflectivity is insensitive to the degree of partial gas saturation and there is still no generally accepted seismic reflection method to discriminate commercial scale gas saturations from low gas saturation (Gomez & Tatham, 2007). Based on estimates of free-gas saturations below gas hydrates from around the world, generally with the additional constraints of drilling and well log data, it appears unlikely that the average free-gas saturation would exceed ~10% (Lu & McMechan, 2002; Carcione, et al., 2005; Zillmer, 2006). For the present study, we simply consider a range of possible free-gas saturations from 0.5-15% to estimate the gas volume in the free-gas zone for the Svyatogor 2016 3D dataset (see Table 8).

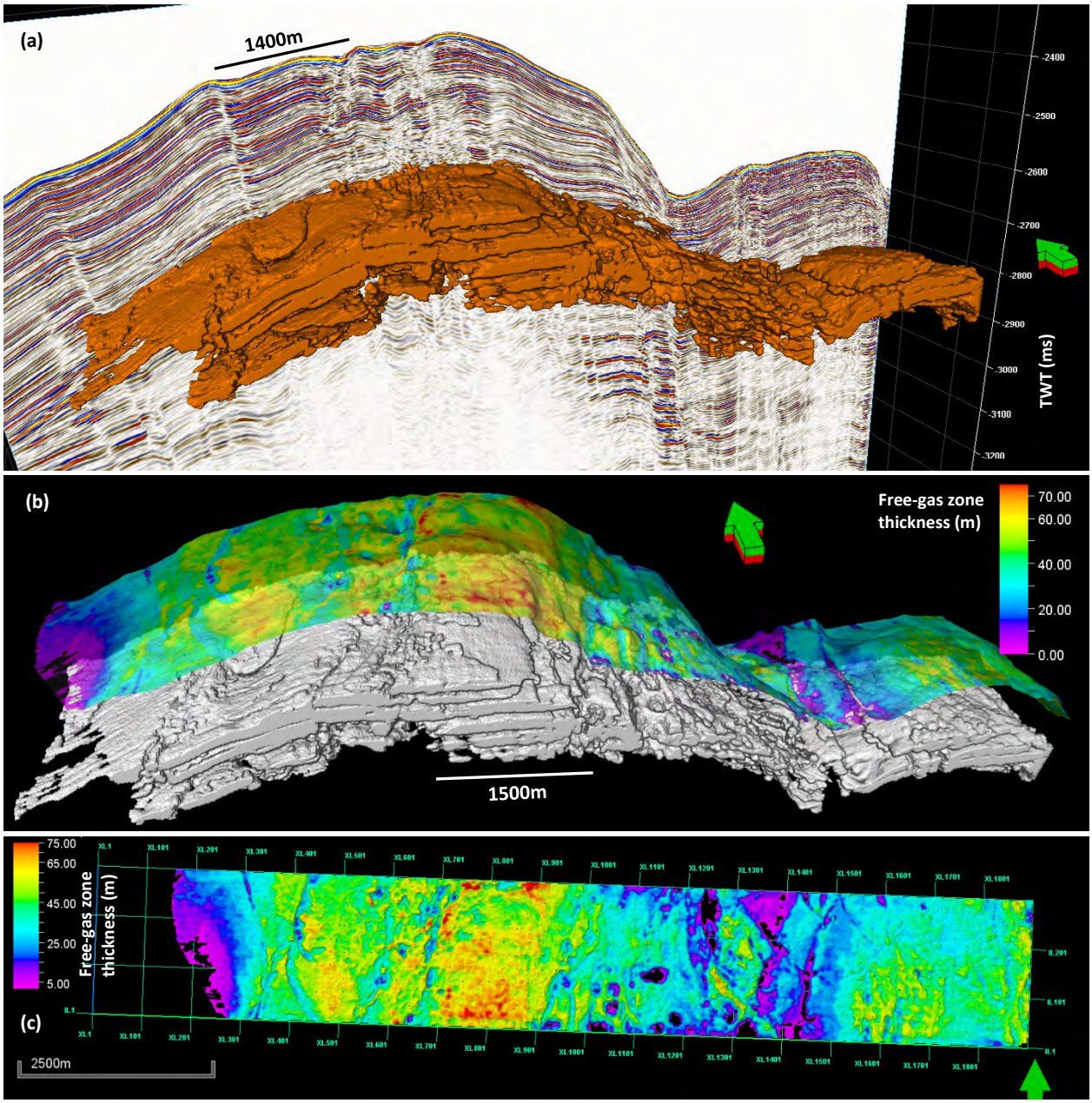


Figure 66 – (a) geobody extraction of the free-gas zone based on anomalous sweetness, the geobody consists of a single connected geoblob, (b) 3D draped vertical thickness map of the geobody following 3D gridding and depth conversion, arbitrary offset added to surface to aid visual comparison with the initial geobody (c) map view of the geobody vertical thickness map, the greatest accumulation of free-gas follows the crest of Svyatogor Ridge.

The amount of gas held in the free-gas system (see Table 8) was estimated based on the deviated ideal gas law; $n = PV/ZRT$, where n is the number of moles of gas, PVT is pressure, volume and temperature, R is the gas constant (8.314 $J.K^{-1}.mol^{-1}$) and Z is the compression factor that accounts for the deviation of a real gas from ideal gas behaviour. The compression factor for 100% methane at 291.15K, 19MPa (i.e. approx. condition at BSR) is ~0.8 (Starling & Savidge, 1992).

Table 8 – estimation of gas volume and amount for the free-gas zone imaged by the Svyatogor 2016 3D survey for a range of assumed porosities and average gas saturations.

Bulk volume	Porosity (Φ)	Gas Saturation (%)	Gas Volume (m^3)	Gas amount (moles)
0.187	0.47±0.07	0.5	$3.7x10^5 – 5.0x10^5$	$3.7x10^9 – 5.0x10^9$
		1	$7.5x10^5 – 1.0x10^6$	$7.3x10^9 – 9.9x10^9$
		2	$1.5x10^6 – 2.0x10^6$	$1.5x10^{10} – 2.0x10^{10}$
		5	$3.7x10^6 – 5.0x10^6$	$3.7x10^{10} – 5.0x10^{10}$
		10	$7.5x10^6 – 1.0x10^7$	$7.3x10^{10} – 9.9x10^{10}$
		15	$1.1x10^7 – 1.5x10^7$	$1.1x10^{11} – 1.5x10^{11}$

4.4.1 Critical free-gas pressure

If a deep methane source is feeding a gas-hydrate free-gas system, such as the migration of abiotic methane into the developing Svyatogor sediment drift along basement detachment faults proposed by Johnson, et al., (2015), then the size of the free-gas zone is likely to increase over time unless it is able to leak into the overlying gas-hydrate zone. Previous studies have found that if the column height of gas contained in the underlying free-gas zone exceeds a critical threshold, faults in the overlying hydrate-bearing sediments may reactivate allowing gas to migrate out of the free-gas zone and potentially through the hydrate stability zone (Hornbach, et al., 2004). At Blake Ridge, a contourite drift located off the south-eastern US coast, it has been reported that the gas pressure may in some cases exceed the lithostatic effective stress leading to tensile failure of the sealing gas hydrate cemented layer, dilating fractures and allowing upward gas migration into the regional methane hydrate stability zone (Flemings, et al., 2003).

The critical pressure required to cause mechanical failure of the hydrate cemented seal was calculated for Svyatogor Ridge assuming two possible failure mechanisms 1) slip on existing faults and 2) formation of dilating fractures when gas pressure exceeds lithostatic effective stress subjecting the seal to tensional failure.

The total vertical stress was calculated at the BSR as the weight of the overlying strata and water column:

$$\sigma_v = g(\rho_{water} h_1 + \rho_{bulk\ sediment} h_2) \quad (1)$$

where h_1 is the water depth and h_2 is the thickness of sediment above the BSR.

The total horizontal stress was calculated assuming a poroelastic medium under uniaxial strain imposed by the vertical lithostatic stress (Engelder & Fischer, 1994; Hornbach, et al., 2004):

$$\sigma_h = \frac{\nu}{1-\nu}\sigma_v + \alpha\frac{1-2\nu}{1-\nu}P_p \quad (2)$$

Where: ν is Poisson's ratio, α is Biot's coefficient and P_p is the pore pressure.

4.4.1.1 Estimating Biot's Coefficient - α

Biot's coefficient allows the coupling of the fluid pressure and solid stress and represents the ratio of the volume of water squeezed out of the formation to the total volume change of the formation during deformation at constant fluid pressure. For poorly consolidated sediments the bulk compressibility is much larger than the grain compressibility and Biot's coefficient tends towards 1. When the grain framework becomes more rigid, the pores are not compressed so readily under applied load and the Biot coefficient decreases. In this case Biot's coefficient was estimated using the Lee (2002) functional relationship for unconsolidated, hydrate bearing sediments:

$$\alpha = \frac{-184.0468}{1+e^{(\phi+0.56468)/0.10817}} + 0.99494 \quad (3)$$

Where φ is the water saturated porosity.

The water saturated porosity was estimated from the 100-200mbsf average of ODP sites 908-912 (see Figure 2) as 0.47±0.07 (std. dev.). The corresponding Biot's coefficients (α) are 0.98 (no gas hydrate) and 0.92 (40% of pore space occupied by gas hydrate).

4.4.1.2 Estimating Poisson's ratio – v

Poisson's ratio indicates the amount of strain produced perpendicular to a normal stress and can be thought of as the fraction of lateral deformation produced by a compressive stress. Soft sediments lacking hydrates have a low resistance to lateral deformation under applied lithostatic stress and will tend towards a Poisson's ratio around 0.5 while high concentrations of hydrate can stiffen the sediments and lower the Poisson's ratio (Lee, et al., 2010). The Poisson's ratio may be expected to vary from about 0.44 for deep marine sediments without gas hydrate (Hamilton, 1979), while values of 0.41 and 0.38 have been reported as characteristic of weakly and strongly hydrated sediments respectively (Katzman, et al., 1994; Tinivella & Accaino, 2000).

4.4.1.3 Fault reactivation

Fault reactivation is controlled by the shear and normal stresses on the fault plane. For normal faults with a dip angle of θ the shear (τ) and normal (σ_n) stresses can be resolved from the vertical and horizontal stresses as:

$$\tau = \frac{1}{2}(\sigma_v - \sigma_h)\sin(2\theta) \quad (4)$$

$$\sigma_n = \frac{1}{2}(\sigma_v + \sigma_h) + \frac{1}{2}(\sigma_v - \sigma_h)\cos(2\theta) \quad (5)$$

The shear slip tendency of the fault (Nacht, et al., 2010) was assessed based on the Mohr-Coulomb criterion:

$$\tau_{slip} = c + \mu(\sigma_n - \alpha P_p) \quad (6)$$

Where c is the fault cohesion (here assumed equal to zero i.e. cohesionless), μ is the coefficient of friction of the fault (assumed equal to 0.6) and $\sigma_n' = \sigma_n - \alpha P_p$ (7) is the effective normal stress on the fault plane.

The fault can reactivate if the shear stress reaches the Mohr-Coulomb failure envelope i.e. if the ratio $\frac{\tau_{slip}}{\tau} > 1$ and/or dilate if the effective normal stress $\sigma_n{}'$ is reduced to zero.

4.4.1.4 Critical gas pressure

The pore pressure at the top of the free-gas zone is increased by the buoyant pressure exerted by the interconnected free-gas phase. The buoyant pressure depends on the thickness of this interconnected gas zone, hereafter simply referred to as the gas column height. At the 'critical' gas column height, the increased pore pressure can reactivate faults or initiate fractures allowing gas to migrate upwards (Daigle & Dugan, 2010). The pore pressure at the top of the free-gas zone is considered to be the sum of the hydrostatic pressure and the buoyant pressure due to the interconnected gas column:

$$P_p = (h_1 + h_2)g\rho_{water} + h_{gas}g(\rho_{water} - \rho_{gas}) \quad (8)$$

Where: h_1 is the water depth, h_2 is the thickness of sediment above the BSR and h_{gas} is the interconnected gas column height.

Table 9 – Parameters used in calculation of seal failure due to free-gas zone overpressure.

	Range of values	**Notes**
Water depth	1670m	Depth at crest of ridge.
Gas hydrate stability zone thickness	128-152m	Thickness at crest of ridge. Water bottom to interpreted BSR pick depth converted using 1500m/s velocity. Range of values mainly due to pockmark topography.
Water density	1027kg/m^{-3}	
Sediment density	1924±156 (std. dev.) kg/m^{-3}	Average overburden bulk density calculated as average of 0-150mbsf measurements from ODP holes 908-912 (Figure 2)
Gas density	157kg/m^{-3}	Calculated for 100% CH_4 at crest of ridge, BSR depth (~18°C, 19MPa) (Starling & Savidge, 1992)
Fault Angle	60°	Normal faults with dips of 50-60° measured in Svyatogor 2016 3D dataset, those at ~60° are the most prone to reactivation.
Cohesion strength	0	Assumed cohesionless
Coefficient of friction	0.6	Lowest friction coefficient typical for hydraulically active faults (Nacht, et al., 2010)
Poisson's ratio	0.38-0.44	strongly hydrated sediment and no hydrate respectively (see section 4.4.1.2 pg.77)
Biot's coefficient	0.92-0.98	40% of pore space occupied by gas hydrate and no gas hydrate respectively (see section 4.4.1.1 pg.77)

By solving equations 1-8 using the parameters shown in Table 9, the gas column height required to make $\frac{\tau_{slip}}{\tau} > 1$ and/or $\sigma_n{}' \leq 0$ was determined. A gas column height of 99-140m is predicted to cause shear slip fault reactivation for low gas-hydrate concentration (Poisson's ratio=0.41-0.44, Biot's coefficient=0.98). For high gas-hydrate saturation (Poisson's ratio=0.38, Biot's coefficient=0.92) a Gas column height of 142-154m is predicted to cause fault reactivation, also via shear slip. The minimum gas column height predicted to cause fault reactivation is 77m, corresponding to a Poisson's ratio of 0.38, Biot's coefficient of 0.98 and gas hydrate stability zone thickness of 128m. A lowered Poisson's ratio without affecting Biot's coefficient may occur if the gas hydrate grows along grain contacts, however it is more probable that hydrate nucleates on grain surfaces and initially grows into

pores before becoming interconnected at higher hydrate saturations (Lee, et al., 2010) so this minimum gas column height scenario is considered unlikely.

4.5 Indications of fluid migration

4.5.1 Free-gas migration

In addition to the free-gas zone beneath the BSR some amplitude anomalies that show reversed polarity compared to the seafloor reflection were also observed (Figure 67) and are interpreted as bright spots indicating pockets of free-gas. These anomalies were mostly observed at the westernmost end of the Svyatogor 2016 3D dataset and appear to reflect gas migration up the western flank of the sediment drift along permeable beds (see Figure 67). The anomalies terminate at faults where the gas is presumably able to migrate vertically along the fault planes before accumulating in the main sub-BSR free-gas zone. Similar amplitude anomalies were notably absent along faults above the BSR, which indicates that leakage is currently inactive and the free-gas that may have previously migrated into the hydrate stability zone has either escaped into the water column or formed hydrates.

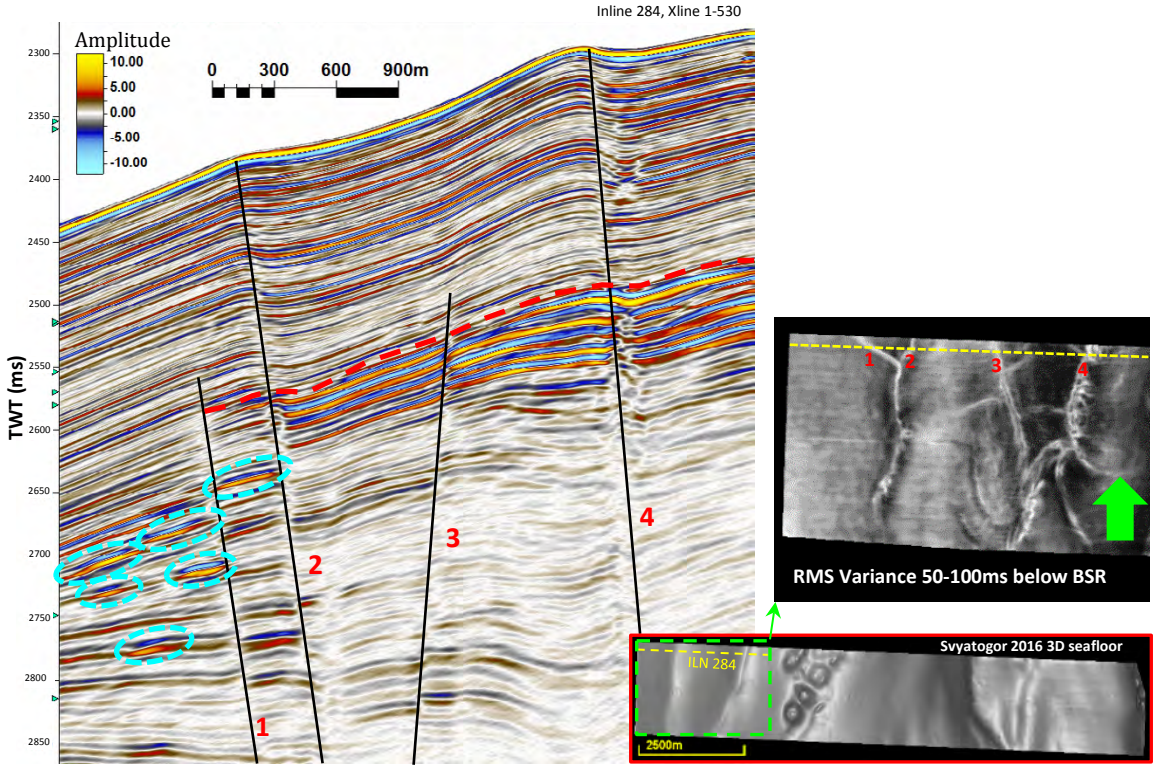


Figure 67 – Amplitude anomalies (blue circles) with reversed polarity compared with the seafloor reflection at the far Western end of the Svyatogor 2016 3D survey and occurring significantly below the BSR(red dashed line) are interpreted as pockets of free-gas migrating up-dip along permeable layers and then directed by faults. The inset variance map illustrates the fault geometry and the position and extent of the vertical seismic profile (yellow dashed line).

4.5.2 Gas chimneys and pockmarks

Pockmarks are depressions, typically circular to elliptical, that form where fluids escape upward through fine-grained seafloor sediments (Cathles, et al., 2010). Pockmarks are commonly associated with acoustically transparent or disturbed gas chimneys that represent zones of focussed vertical fluid migration (Plaza-Faverola, et al., 2010). When fluids discharge rapidly enough through these focussed

zones, the near surface sediments become liquefied as grain-to-grain contact is lost and the grains are suspended dynamically by the upward flow (Cathles, et al., 2010). This liquefied sediment is then highly prone to erosion by ocean bottom currents leading to the formation of a pockmark.

Due to their spatial correspondence, gas chimney and pockmark features identified in the Svyatogor 2016 3D survey were mapped together as fluid-flow conduits. The mapping of these conduits was achieved by scanning through variance time-slices and interpreting zones of reduced seismic coherence (high variance) with a circular/elliptic cross-section and some indication of vertical extension (see Figure 68). The spatial distribution of the interpreted fluid flow conduits is illustrated in Figure 69. The lowermost extent of these fluid flow zones typically occurs within the free-gas zone.

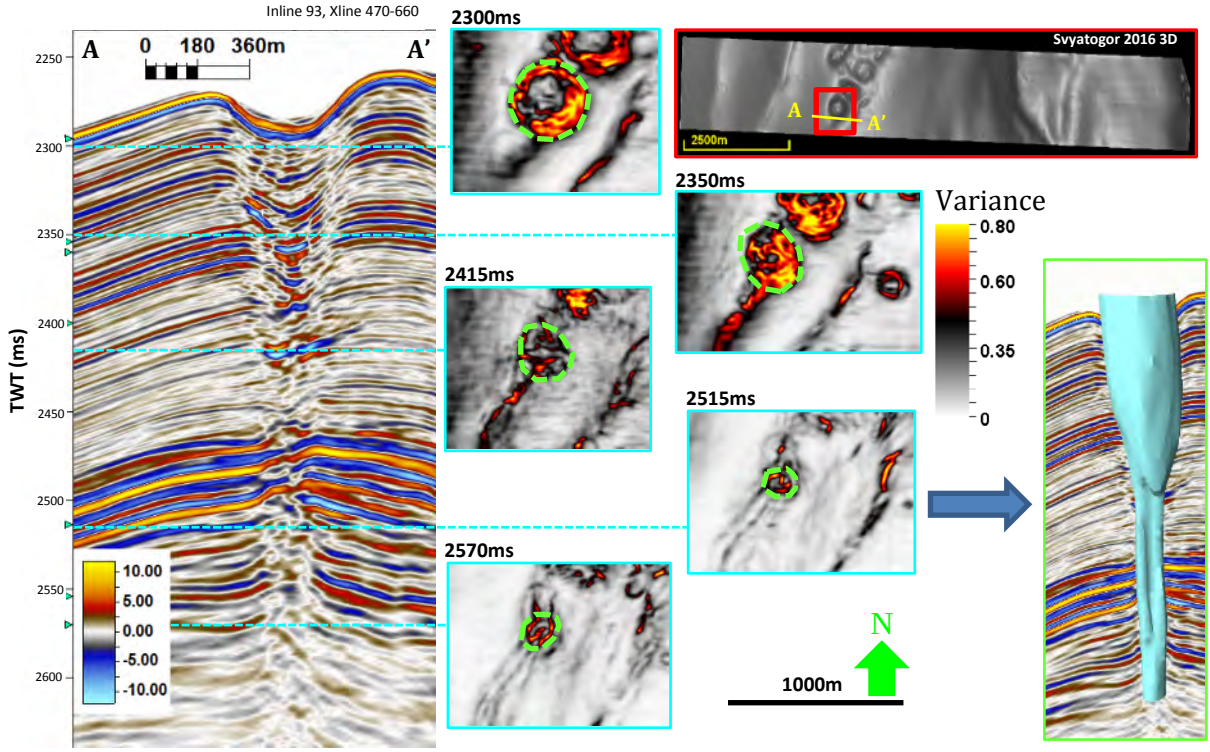


Figure 68 – illustration of the interpretation of potential fluid conduits (gas chimney/pockmark/paleo-pockmark structures) based on vertically linked, rounded time slice variance anomalies (blue highlighted insets). The location of the feature within the 3D dataset on a seafloor map (red highlighted inset) and a multi-Z triangular mesh shows the final interpreted fluid conduit (green highlighted inset).

4.5.2.1 Identification of buried 'paleo-pockmarks'

Pockmark depressions can also be observed buried beneath the seabed, reflecting episodes of localized fluid flow to the paleo-seafloor and can therefore be often referred to as paleo-pockmarks (Andresen & Huuse, 2011; Moss, et al., 2012; Waghorn, et al., 2017). Once formed, fluid conduits feeding pockmarks tend to remain embedded within otherwise low-permeability successions as a structure of significant permeability for long periods of time (Cartwright, 2007; Moss, et al., 2012). Sequences of vertically stacked pockmarks provide strong evidence of this long-lived, episodic behaviour as vertical fluid flow conduits (Cartwright, 2007; Andresen & Huuse, 2011). To understand the episodic activity of these buried pockmark conduits it is important to differentiate between true paleo-pockmarks representing active fluid flow and inactive intervals of draping and infilling sedimentation (Andresen & Huuse, 2011). In this thesis, paleo-pockmarks reflecting active fluid flow episodes were recognised on the basis of 1) stratigraphic thinning representing net erosion (with reference to Figure 53) and 2)

steep cross-sectional profiles and downcutting erosional truncation of underlying reflections. By contrast, periods of draping or infilling sedimentation were recognised by 1) stratigraphic thickening representing net deposition (Figure 53) and 2) broadening and smoothing of cross-sectional profiles and interpreted surfaces.

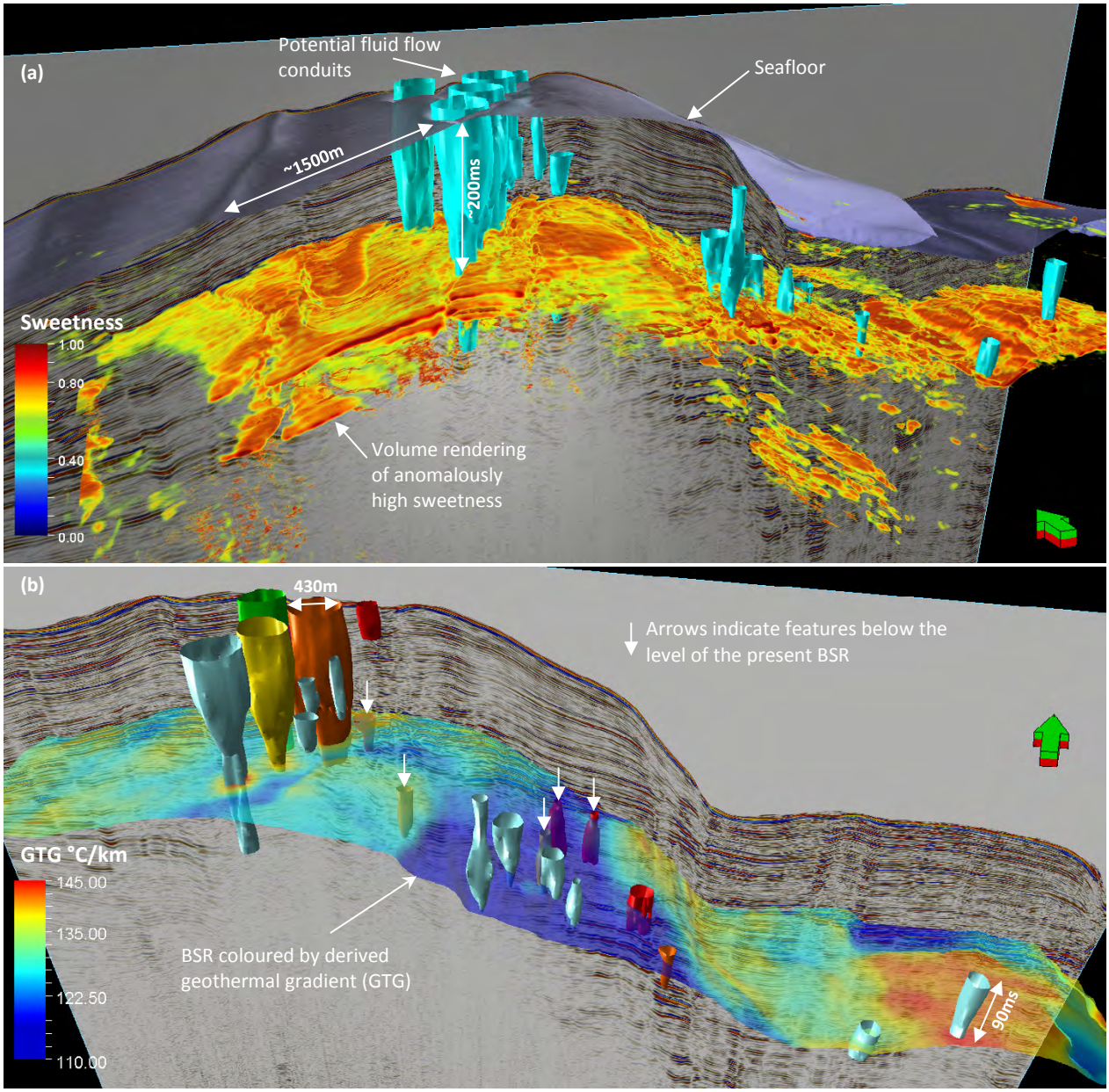


Figure 69 – (a) relationship between the seafloor, potential fluid flow conduits (pockmark/gas chimney structures) and the free-gas zone visualized by volume rendering anomalously high sweetness values. (b) The same potential flow conduits coloured for visual clarity, structures below the present BSR are indicated with downward pointing arrows (some features extend from below to above the BSR and are not marked with arrows), the BSR is coloured by derived geothermal gradient. Note that chimney/pockmark structures along the crest of the ridge are larger and many reach the seafloor.

The interpreted fluid-flow conduits can be broadly grouped into three categories:

1. The largest pockmarks, typically with significant expression at the seabed, occur along the crest of Svyatogor Ridge and are vertically stacked above buried paleo-pockmark features.
2. Smaller gas chimney/vertically stacked paleo-pockmark features that show very clear association with faults and are always observed on the hanging-wall side of the faults. These occur near the eastern flank of the sediment drift in a region with relatively low geothermal gradient (see Figure 69).
3. Several small gas chimney/pockmark features that occur towards the easternmost end of the 3D dataset and are associated with an area of shoaling BSR (Figure 69).

Each of these categories is presented in greater detail in the following sections to further illustrate their characteristics.

4.5.2.2 Ridge crest pockmarks

A cluster of large pockmarks, stacked above buried paleo-pockmarks, are concentrated along the crest of Svyatogor ridge within the 3D dataset. The pockmarks are up to 430m diameter, 17m deep at the modern seabed and 26m deep within the sub-seabed (assuming v=1500m/s). There seems to be some association with underlying faults (see Figure 54 & Figure 68) but it is relatively difficult to discern due to the disturbed seismic imaging caused by the closely-spaced pockmark cluster. The mapped flow conduits (Figure 69) tend to broaden around H3, which also corresponds to the base of a nearby group of smaller conduits indicating that an intensification of fluid flow may have occurred around this interval. These pockmarks are interpreted to be currently inactive and undergoing a phase of infilling that is causing their cross-sectional profiles to gradually flatten out (see Figure 70).

Paleo-pockmarks were identified by their relatively steep cross-sectional profiles and truncation of underlying reflectors. For the example shown in Figure 70, the most recently active paleo-pockmark corresponds to locally picked surface number 3, which appears to cut down and truncate the underlying reflectors. Surfaces 1, 2 and WB follow the same basic morphology as surfaces 3, but become progressively smoother and lower relief towards the seabed, reflecting the phase of inactivity and infilling. The largest vertical pockmark relief was observed along surface 3 (between H3 & H2), which is also an indicator that this was an active period of fluid flow. This interpretation of active and inactive periods of fluid flow is further supported by observations of stratigraphic thickness maps (see Figure 53 & section 4.1.3.1, pg.57). While the pockmarks largely overprint it, there is also an associated fault that is visible on TWT maps where the eastern side is upthrown relative to the western side (Figure 70) and is also visible in variance maps (Figure 54 & Figure 68). Significant upward bending of reflectors was observed beneath the ridge-crest pockmarks (Figure 70) and could be related to migration artefacts, velocity anomalies or sedimentary deformation.

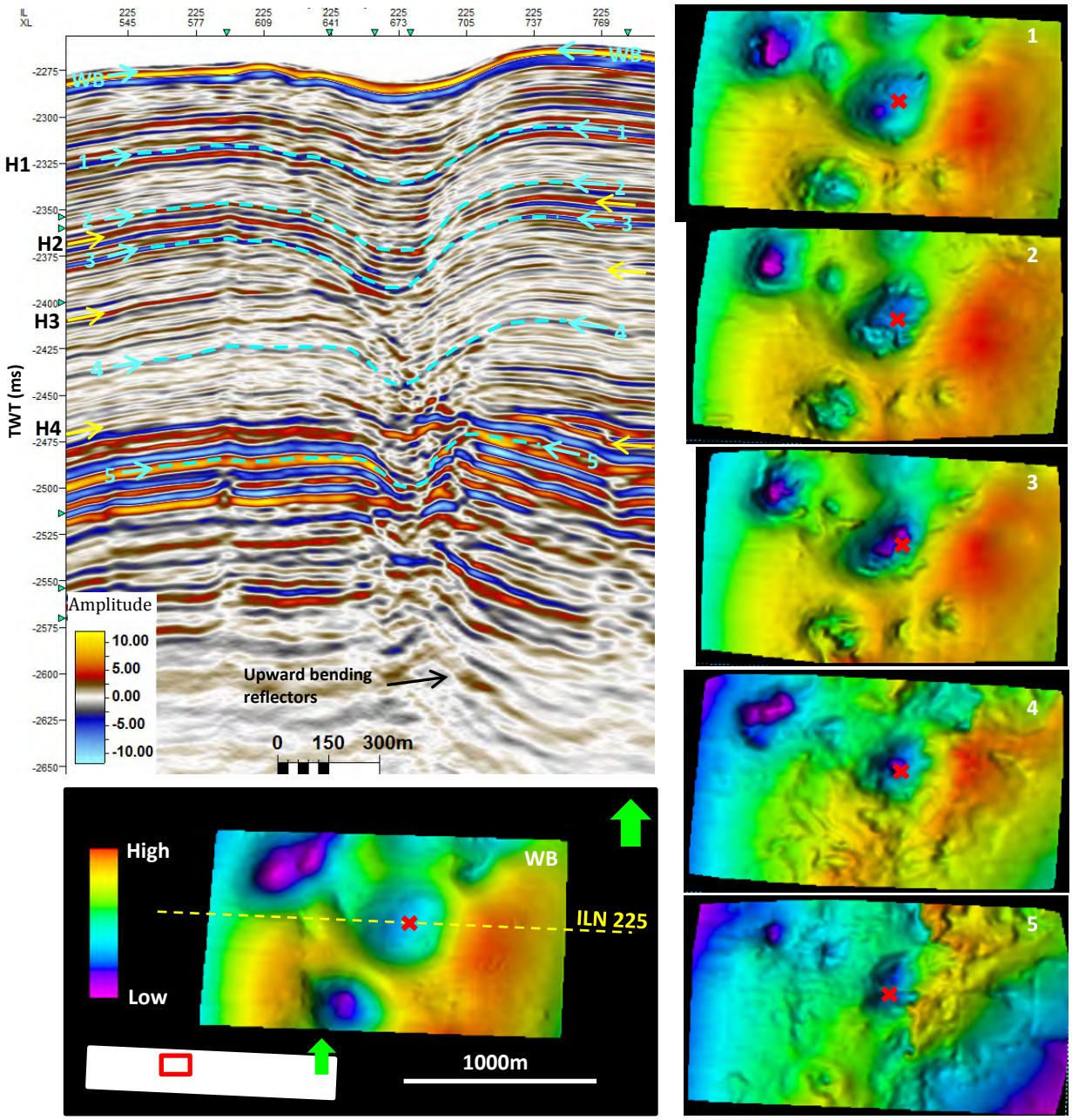


Figure 70 – Vertically stacked pockmarks/paleo-pockmarks located on the crest of Svyatogor Ridge illustrated on a vertical seismic profile and key interpreted surfaces whose stratigraphic position is marked by blue arrows and dashed lines in the vertical profile. The red cross marks the same position on each of the surfaces and the orientation and scale is constant in all cases, surfaces are illuminated from the West at 80° with 10x vertical exaggeration. Position of horizons H1-H4 that were mapped across the 3D dataset (see Figure 52*) are indicated with yellow arrows (except H1 which corresponds directly to the locally interpreted surface number 1).*

4.5.2.3 Fault controlled fluid migration in area of depressed BSR

An example of an interpreted fluid flow conduit, consisting of stacked paleo-pockmarks and clearly associated with a fault, is shown in Figure 71 and the fault is further illustrated in Figure 60. In common with the other fluid flow conduits, this feature appears to have been inactive for some time and there is no expression at the present seabed. The steepest cross-sectional profiles are observed between H4 & H3 (Figure 71) and may reflect the timing of the most vigorous interval of fluid flow and paleo-pockmark formation. Pockmark activity appears to cease around H2, which is somewhat earlier than the flow conduits in crestal positions. In common with all of the fluid flow conduits

directly associated with faults, this example occurs on the hanging wall side of the fault. Significant amplitude blanking was observed within the underlying free-gas zone. One possible explanation is that leakage along the fluid flow conduit has caused the free-gas zone to become locally depleted. Since the uppermost pockmark is well below the seabed the amplitude blanking cannot be attributed to this episode of leakage since the gas-hydrate stability zone (GHSZ) would have been deeper when this paleo-pockmark formed. One possibility is that the blanking could be due to a more recent episode of leakage caused by a reactivation of the fluid conduit associated with fault slip. The leakage was presumably small-scale and the gas did not reach the seafloor and instead formed hydrate within the GHSZ. The loss of signal due to wavefield scattering caused by pockets of high gas-hydrate concentration (Singhroha, et al., 2015) along the fluid conduits could further exaggerate the amplitude blanking.

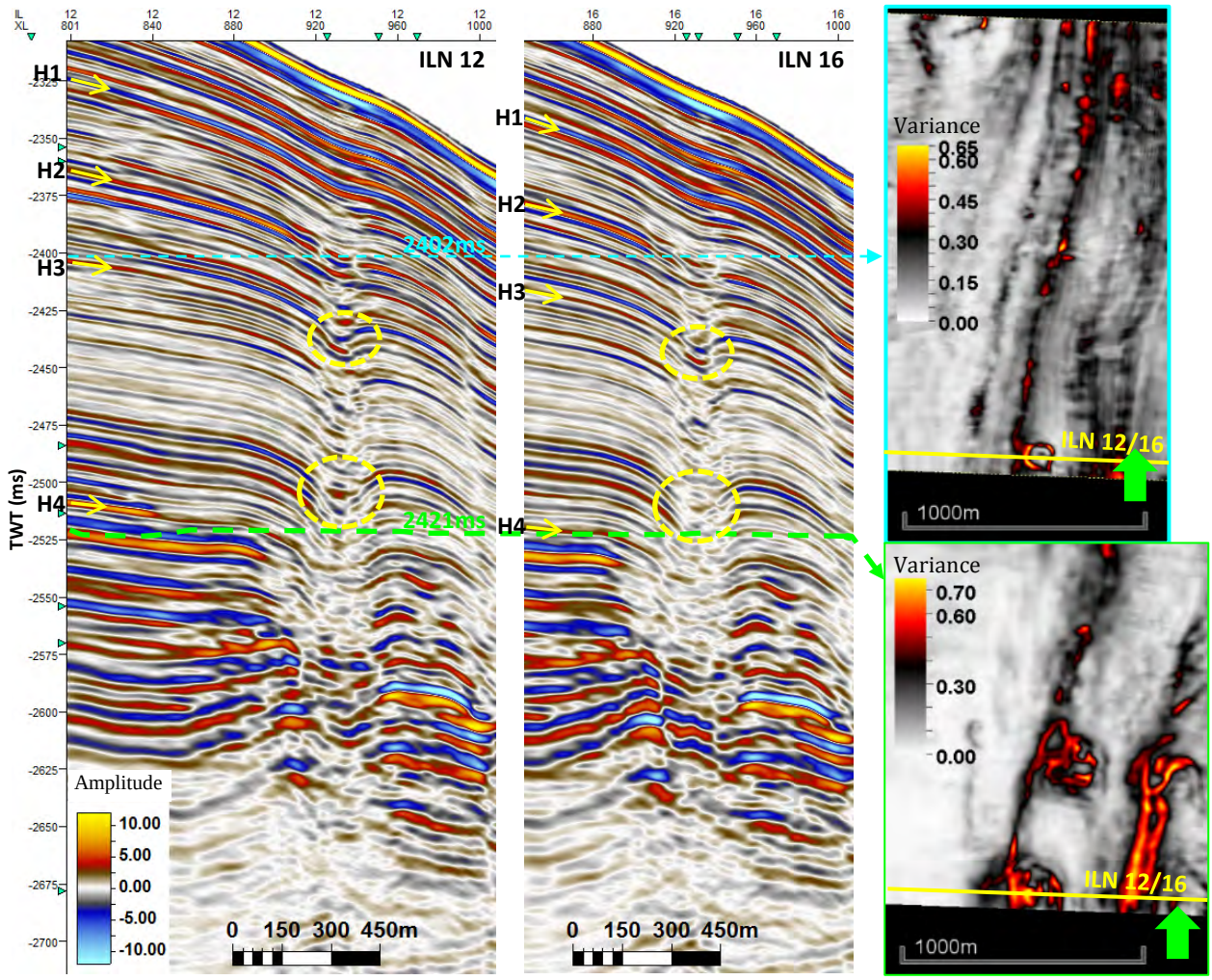


Figure 71 – Example of a pockmark stack flow conduit/gas chimney associated with Fault D (see Figure 56*) that marks a potential zone of vertical fluid migration from the free-gas zone (high-amplitude reflections below ~2525ms) into the regional hydrate stability zone. Some pockmark-like deflections of reflectors are observed (circled on vertical profiles) but the significant loss of reflector continuity over short length scales (i.e. 25m) from inline 12 to inline 16 complicates the interpretation. Position of horizons H1-H4 that were mapped across the 3D dataset (see* Figure 52*) are indicated with yellow arrows.*

4.5.2.4 Fluid migration above shoaling BSR

A nice example of a potential fluid flow conduit was associated with a zone of shoaling BSR that may be a product of the advection of heat by warm fluids (see Figure 72). It is notable that this feature does not appear to be directly associated with a fault, although small scale faults are observed to either side. It does however, appear to be associated with the crest of the shoaling BSR structure. It is also notable that while significant down-bending of reflectors is observed, there is significantly less acoustic blanking than typically associated with gas chimneys. This may be because the system has been inactive for some time e.g. between H1 and the seabed represents a phase of quiescence and infilling and there is no bathymetric expression at the seabed. It is therefore expected that the sediments above the BSR do not currently contain free-gas and the down-bending of reflectors represents a stacked succession of paleo-pockmarks rather than low velocity associated pushdowns. The absence of free-gas may partly explain the relatively good quality of the seismic imaging across the feature. The balance between erosion and deposition along the stacked succession of paleo-pockmarks may be related to the level of fluid flow through the system, with more erosive features corresponding to periods of more vigorous flow. Some of the most erosive features have been marked in Figure 72 and may correspond to the most dynamic periods of fluid flow. From the position of the feature at the crest of the shoaling BSR, it is interpreted that it may have formed by migration of deep, warm fluids driving gas zone overpressure resulting in capillary failure of the gas-hydrate bearing seal, followed by gas chimney and pockmark development as described by Cathles, et al. (2010).

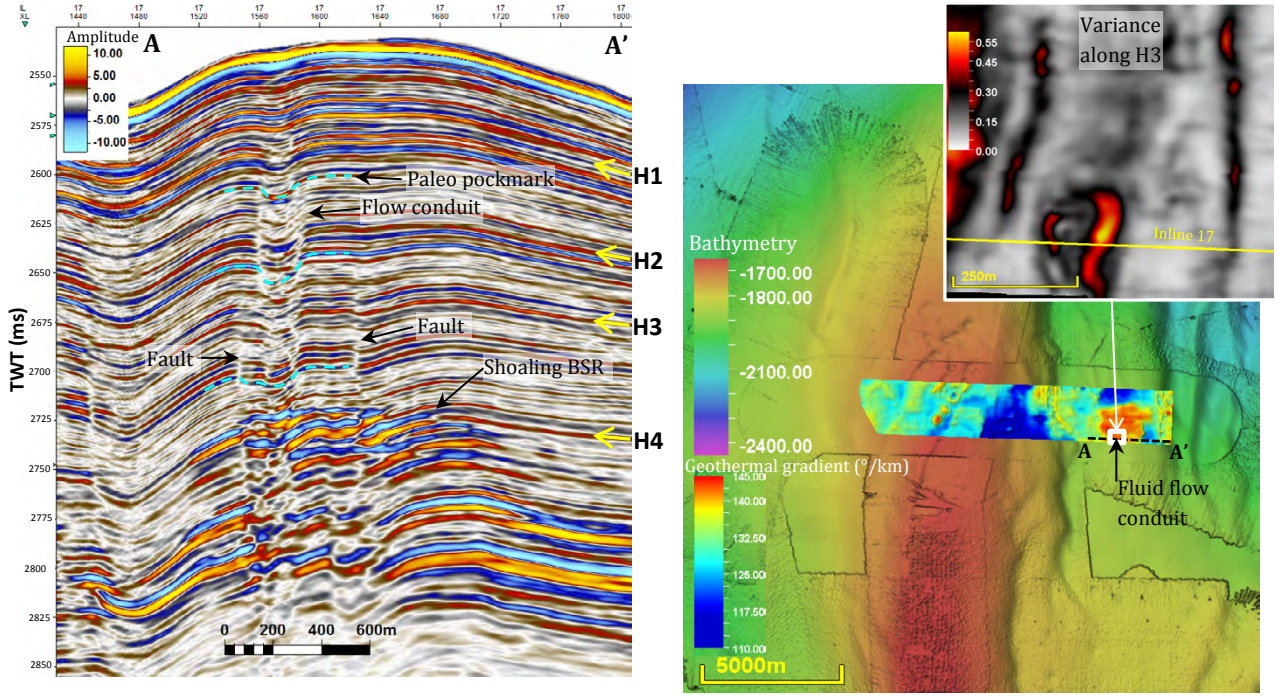


Figure 72 – Potential fluid flow conduit above an area with shoaling BSR that may be associated with increased geothermal gradient and heat flow as shown on map to the right. Position of horizons H1-H4 that were mapped across the 3D dataset (see Figure 52*) are indicated with yellow arrows. Inset map shows variance extracted along H3 and shows the small-scale faults that occur either side of the fluid conduit and the circular variance profile displayed by the fluid conduit. Light-blue dashed lines mark interpreted paleo-pockmarks that appear to truncate underlying reflectors.*

5 Discussion

The primary focus of this thesis was the acquisition, processing and interpretation of a high-resolution 3D P-Cable seismic survey targeting a pockmark cluster on Svyatogor Ridge. This chapter will discuss the progress that this dataset has allowed towards an improved understanding of the structural and stratigraphic setting of the study area and the dynamics of the gas-hydrate, free-gas system hosted within the Svyatogor sediment drift.

5.1 Critical processing steps

The processing flow that was applied to the Svyatogor 2016 3D dataset was largely effective in enhancing the recorded signal component of the dataset while suppressing noise and minimizing artefacts (see section 3 pg.13). In particular, burst-noise filtering allowed useful signal to be extracted from channels that would otherwise have been discarded, improving the overall trace density of the dataset. The broadband processing sequence consisting of bubble, ghost and noise suppression operations proved an effective means of maximising the usable bandwidth of the data. High-resolution static correction proved an important means of improving reflector continuity and suppressing acquisition footprint noise caused by streamer depth variations etc. Considerable effort was also spent on improving the receiver geometry based on a least squares type inversion of direct wave arrivals (see section 3.5 pg.21). While the improved geometry produced a noticeable, if subtle, improvement to the reflector continuity of the processed dataset, the benefits of such processing could become more apparent in a 4D processing scheme and this is something that should be attempted in the future given an appropriate time-lapse seismic dataset.

It is unfortunate that the underlying basement and detachment faults were not well imaged in the Svyatogor 2016 3D dataset. This may be partly attributed to the well-developed free-gas zone observed beneath the BSR in the study area, which is highly reflective and causes rapid attenuation and decreased penetration of the seismic signal. Scattering at pockmarks due to the possible presence of hydrates and/or carbonates may have also impeded deep imaging as in Singhroha, et al., (2015). Finally, spatial aliasing over the steeply dipping seafloor on the flank of the main sediment drift and in vicinity of a basement detachment fault (see section 3.8.1 pg.43 & Figure 5) could have also impeded the ability to accurately resolve the detachment faulting in this area. While some of these issues could be addressed if it were possible to acquire a dataset with increased trace density, it could be more beneficial to target a future acquisition either north or south ~5-10km along the axis of the Svyatogor sediment drift (see Figure 74). To the north, an area on the edge of the well-developed free-gas zone could be targeted, although the issues of spatial aliasing associated with steeply dipping strata over the detachment fault would remain. The problem of steep dips and spatial aliasing could be avoided to the south, but there would likely still be a significant free-gas zone to image through.

5.2 Development of Svyatogor Ridge

It has been proposed that Svyatogor Ridge developed rapidly as a southern extension of Vestnesa ridge and was then offset along the Molloy Transform Fault (MTF) at a rate of 14.5mm/yr, reaching its present position ~30km offset from Vestnesa in ~2Ma (Johnson, et al., 2015). As noted by the authors of this model, if the Svyatogor drift were much older than 2 Ma the offset distance between the faulted and pockmarked ridge crests north and south of the MTF would be larger than the observed 30km present offset (at a full plate slip rate of 14.5mm/yr on the MTF). However, a thick sedimentary

sequence is observed below the ~2.7Ma YP-2/YP-3 boundary in the Svyatogor 2016 3D dataset (see section 4.1.1 pg.54), which implies that sedimentation has been ongoing for significantly longer than the past 2Ma.

It should be noted that the correlation of the YP-2/YP-3 boundary south of the MTF and east of ODP site 909 is difficult with presently available seismic data and remains subject to significant uncertainty. However, the 210-230m thickness of YP-3 sediments interpreted in the present study (see section 4.1.1 pg.54) is consistent with the 500ms thickness (~375-425m with assumed interval velocities of 1500-1700m/s), interpreted at the SE segment of Vestnesa Ridge by Plaza-Faverola, et al., (2015), since we expect lower sedimentation rates at Svyatogor due to its more distal position with respect to the West Spitsbergen Current (WSC). It is considered improbable that the ~700m thick sedimentary sequence at Svyatogor (see section 4.1.3 pg.56) could belong entirely to the YP-3 sequence, which would imply a sedimentation rate almost double that observed at the SE segment of Vestnesa ridge by Plaza-Faverola, et al., (2015). Therefore, while the exact position of the YP-2/YP-3 boundary is remains uncertain it is highly probable that a significant interval of YP-2 sediments older than 2.7Ma have been deposited along Svyatogor ridge and alternatives to the Johnson, et al., (2015) model for the development of Svyatogor Ridge should therefore be considered.

5.2.1 Proposed model for growth of Svyatogor sediment drift

An alternate model for the growth of the Svyatogor sediment drift is that it grew independently of the Vestnesa drift over a longer period of time. The age of the Svyatogor drift must still be younger than the Vestnesa drift as the age of the underlying oceanic crust is constrained to within the range 9.8-2.8 Ma by magnetic anomaly chrons 5 & 2A (Engen, et al., 2008). The age of the underlying crust at Svyatogor may be further constrained by considering that it lies ~17km closer to the spreading axis (younger) than magnetic anomaly chron 5 and ~26km further from the spreading axis (older) than chron 2A. If the current half spreading rate at Knipovich ridge of 8mm/yr (Ehlers & Jokat, 2009) is used to linearly interpolate from chrons 2A & 5 to the position of Svyatogor ridge, the underlying crust is estimated at 6.1-7.7Ma.

We have no direct means to constrain the onset of sedimentation at Svyatogor, but assuming that the ~470-490m thick sequence below the YP-2/YP-3 boundary was deposited at a similar rate to the <2.7Ma sequence at Vestnesa (because Svyatogor would have been in a similar position with respect to the West Spitsbergen Current when it began to grow), sedimentation would have begun around 5.7-6.2Ma. It is therefore inferred that sedimentation began shortly after the underlying oceanic crust was formed.

With this alternate model, it seems plausible that the Svyatogor sediment drift developed as a basement/fault controlled drift (see Figure 73) as described by Rebesco & Stow (2001) and observed in the Weddell Sea (Maldonado, et al., 2005). Such drifts are the result of disruptions of bottom current flow caused by basement or fault structure and develop parallel to the structural trend (Maldonado, et al., 2005). These tectonic drifts are also expected to prograde parallel to the bottom current direction (Rebesco & Stow, 2001; Maldonado, et al., 2005), which would be towards the north at Svyatogor under the influence of the WSC and is consistent with the observed pattern of northward prograding clinoform reflection packages (see Figure 74). An additional characteristic of fault-controlled drifts may be (subsequently reactivated) syn-depositional faulting affecting the relatively

steeper side of the drifts (Rebesco, 2005), which is also consistent with observed behaviour at Svyatogor ridge.

It therefore seems most likely that the Svyatogor sediment drift grew along the axial, basement detachment faults that it remains associated with to the present day (Waghorn, et al., 2015), with sedimentation initiating shortly after significant seafloor surface expression of these faults developed (possibly up to ~5.7Ma). The growth of the sediment drift is therefore independent of Vestnesa ridge, even though deposition is still controlled by the WSC that also controls deposition at Vestnesa. The Svyatogor sediment drift has continued to grow since 2.7Ma but at a progressively lower sedimentation rate compared to Vestnesa due to the increasing distance from the WSC.

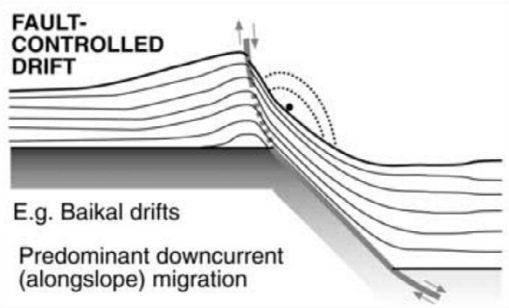


Figure 73 – Illustration of a fault-controlled drift from Rebesco & Stow (2001) that may represent a plausible model for the formation of the Svyatogor sediment drift.

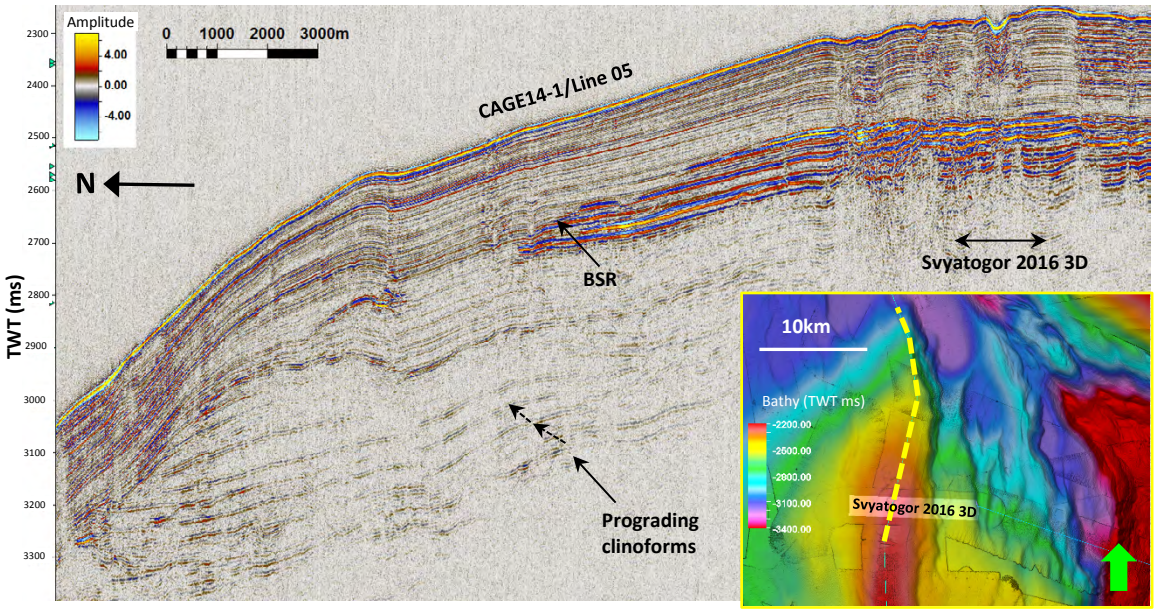


Figure 74 – High-resolution 2D P-Cable seismic line illustrating the northern extent of the Svyatogor sediment drift. A series of northward prograding clinoform reflection packages are observed. This line also shows the northern extent of the BSR and free-gas system that has been imaged by the Svyatogor 2016 3D survey. Inset map shows location of the 2D profile with respect to seafloor bathymetry and the location of the 3D survey.

5.3 Structural and stratigraphic setting

5.3.1 Style of faulting

The dominant mode of faulting observed in the Svyatogor 2016 3D dataset are normal faults with a mean strike of ~014° and dip of 50-60° towards the east. This is consistent with the fault population for the northern part of Knipovich ridge described by Curewitz, et al., (2010), based on structural analysis of side-scan sonar images collected along the ridge axis. These faults can be classified as sub-perpendicular to the plate motion vector of ~307° that is predicted by the NUVEL-1A and REVEL global plate-tectonic models (DeMets, et al., 1990; DeMets, et al., 1994; Sella, et al., 2002; Curewitz, et al., 2010).

Faults that are somewhat oblique to both plate motion and ridge trend are typical of tectonically driven faults in oblique extensional environments based on laboratory experiments and field observations (Clifton, et al., 2000; Clifton & Schlische, 2003). Clay models are particularly useful as analogues of slow-ultraslow spreading ridges where magma is limited and faulting contributes significantly to extension. These models predict that short faults trending slightly oblique to rift-parallel will form on the margins of the deformed zone (as opposed to along axis in the centre of the rift) (Clifton, et al., 2000). This obliquity is hypothesized to occur due to reorientation of the principal stress due to plate motion by a secondary stress field related to changes in topography and crustal rheology at the rift valley walls (Clifton, et al., 2000). The dominant group of faults imaged by the Svyatogor 2016 3D dataset occur on the western margin of the deformed zone associated with Knipovich ridge and are slightly oblique to the Knipovich axial trend. It is therefore assumed that these faults are tectonically driven and controlled by the mechanics of oblique rifting.

A group of faults with small throws and highly variable orientations (in some cases oriented E-W and dipping towards the north) was also imaged in the Svyatogor 2016 3D survey, particularly towards the eastern end of the dataset. No seafloor expression of these faults was observed and they were mostly confined to the interval below the present BSR, indicating that they are shorter-lived features than the dominant group of faults. A population of short faults, also with highly variable orientations was observed at the seafloor along the Knipovich ridge axis by Curewitz, et al., (2010) using side-scan sonar data.

A possible explanation for these variably oriented faults could be stress field perturbation caused by shearing along the Molloy Transform Fault (MTF) driving oblique normal faulting (Behn, et al., 2002). However, the apparently short-lived nature of these faults compared to the main set of tectonically driven faults makes such an explanation unlikely. Since these faults are laterally linked with the larger NNE striking normal faults they may be simply relay structures related to the interaction between the larger faults (Fossen & Rotevatn, 2016). Another possibility is that the vertical migration of fluids could lead to the build-up of overpressure beneath the BSR (see section 4.4.1 pg.76), that might lead to complex local distortion of the stress field if the fluid migration were confined to isolated compartments by a limited number of presumably fault-controlled migration pathways (Cartwright, 1994). Such a distortion of the stress field could lead to hydrofracturing, which may explain the almost random strike distribution of the faulting (Cartwright, 1994) and why it appears to be short-lived.

5.3.2 Fault throw analysis

The throw distribution on the main set of normal faults indicates that they are synsedimentary in nature (see section 4.2.2.1 pg.63) and have greatly influenced the creation of accommodation space and sediment accumulation at Svyatogor ridge (see section 4.1.3.1 pg.57). The marked difference in thickness between adjacent fault blocks indicates the strong influence of faulting on creating and modifying accommodation space and influencing sediment deposition. Depocenters tend to switch spatial location during the evolution of Svyatogor ridge and this can largely be explained by the relative activity of different faults. Thinning over the steep eastern ridge flank (seafloor slope ~12°) is likely caused by small-scale sediment slumping or sliding, which is prone to occur in synsedimentary tectonically active areas for slopes of this magnitude (Kuenen, 1956; McAdoo, et al., 2000). We also see variation in the balance between erosion and deposition associated with pockmarks. Where pockmarks are active, net erosion and thinning occurs whereas during periods of inactivity net deposition tends to fill in the depressions resulting in thickening. That the most recent sedimentary interval (seafloor to H1) is associated with infilling of pockmarks is consistent with the lack of evidence of active leakage at Svyatogor Ridge.

A pronounced change in throw gradient was observed for the investigated subset of normal faults, separating an upper section with strong positive throw gradient from a lower zone with much weaker throw gradient (see section 4.2.2.1 pg.63). This reflects a transition from a period of low fault activity to a more recent period of increased fault activity (Cartwright, et al., 1998). It appears that the change in throw gradient follows the trend of the BSR (see Figure 56) rather than a single stratigraphic horizon. This may be an indication that the period of increased fault activity was initiated by post-depositional fluid overpressure along a paleo-BSR running approximately parallel to the present BSR and driving reactivation of faults extending to the seafloor (i.e. interacting with the free surface and behaving as growth faults). This interpretation is consistent with the idea of increasing accumulation of free-gas below the BSR causing overpressure that eventually leads to fault slip when a critical pressure is reached (Flemings, et al., 2003; Hornbach, et al., 2004). Furthermore, it has been reported that growth faults may only act as fluid flow conduits during active slipping while tending to restrict flow when inactive (Hooper, 1991). This provides a mechanism for repeated, episodic fluid flow and accumulation of slip on growth faults. It is assumed that these faults reached the seafloor from the beginning of the period of increased fault activity so the paleo-BSR would lie ~120-150m below the marker indicating the change in throw gradient (assuming similar temperature gradient, gas composition etc. to present conditions).

However, it is also possible to infer that the change in gradient on the faults towards the western end of the dataset (faults A-C see Figure 56) occurs along a consistent stratigraphic horizon. The faults at the eastern end of the dataset (faults E-F) also occur along a roughly consistent stratigraphic horizon, but one that does not correlate with the western group of faults. It is therefore possible to interpret that the change in fault activity is stratigraphically linked but with a different chronology at the opposing ends of the dataset. Since the two sets of faults occur on opposite sides of an underlying basement detachment faults (see Figure 5) it is reasonable to expect asynchronous slip history. In this case, the faults at the eastern end of the dataset (faults E-F) would appear to have become more active at an earlier time than the faults at the western end of the dataset (faults A-C).

5.4 Dynamics of fluid-flow system

5.4.1 Evidence from pockmarks

The seafloor pockmarks along the crest of Svyatogor Ridge were the primary motivation for acquiring the Svyatogor 2016 3D dataset in this area. The ridge-crest seafloor pockmarks stack above paleo-pockmarks (as described in section 4.5.2 pg.79) forming long-lived fluid flow conduits. The episodic activity of these conduits is reflected by the varying degree of net erosion and deposition associated with these features. Erosive and therefore active periods of fluid leakage and paleo-pockmark formation were identified based on truncation of underlying reflectors and thinning of sedimentary packages between interpreted horizons (see section 4.5.2.1 pg.79). Depositional and therefore inactive periods were recognized by a gradual flattening of pockmark profiles and thickening of sedimentary packages. Following this approach, the seafloor pockmarks appear to have been inactive over the most recent geological past. This is consistent with the lack of shallow-gas amplitude anomalies and lack of water-column gas flares in the area. One example of a small, elongated seafloor pockmark was observed above the lateral tip of a fault, overprints the larger and broader pockmarks and probably reflects the most recent episode of seafloor fluid leakage.

The upward bending of reflectors beneath the ridge-crest pockmarks (see Figure 56 & Figure 70) could be a velocity pull-up caused by anomalously high concentrations of gas hydrate around the BGHSZ along these zones of focussed fluid leakage (Westbrook, et al., 2008; Plaza-Faverola, et al., 2010). The appearance of the upward bending reflectors may have been further distorted during migration due to the inability to resolve these velocity anomalies. Sedimentary deformation caused by upward migrating fluids has also been suggested as a cause for upward bending strata beneath pockmarks (Westbrook, et al., 2008), but the bending pattern observed in the Svyatogor 2016 3D dataset appears more characteristic of a velocity and/or migration effect.

5.4.2 Fault controlled fluid migration

There is clearly an association between the faulting observed in the Svyatogor 2016 3D dataset and zones of fluid leakage (see Figure 71). Faults C & D were particularly interesting in this respect; an elongated pockmark was directly associated with the lateral tip of fault C while pockmark and potential fluid conduits along fault D may be associated with damage zones linking fault-segments (Figure 61 & Figure 71). These observations are consistent with the idea that stress concentration occurs at actively propagating fault tips or at zones where multiple faults interact, causing active fracturing and continued re-opening of fluid-flow conduits. In the hydrothermal systems described by Curewitz & Karson (1997), this permits long-lived hydrothermal flow despite potential clogging of fractures due to mineral precipitation. The Svyatogor gas-hydrate system may behave in an analogous way, in that leakage of free-gas into the hydrate stability zone along faults may be expected to precipitate gas-hydrate, assuming availability of water and appropriate salinity conditions, leading to a reduction in permeability. Continued re-opening of fractures at fault tips or segment junctions may provide a kinetic mechanism whereby free-gas could migrate to the seafloor faster than hydrate precipitation can inhibit.

5.4.3 Thermal evidence of fluid flow

It has been suggested that BSR shoaling can be caused by either: high concentrations of hydrate above the BSR leading to anomalously high velocity, or by upward advection of heat and fluids (Hornbach, et al., 2012). Drilling results at Hydrate Ridge, offshore Oregon suggest that shallow BSRs generally occur where both elevated heat flow and elevated hydrate concentrations exist (Hornbach, et al., 2012). However, the main driver of BSR shoaling appears to be the advection of deep fluids bringing heat and possibly higher salinity from depth, while the anomalous velocity effect is thought to be minor (Vadakkepuliyambatta, et al., 2015).The BSR shoaling that was observed in the Svyatogor 2016 3D dataset is therefore attributed to the fault-controlled advection of deep fluids and driving in-situ hydrate dissociation on a local scale (Hustoft, et al., 2006).

Geothermal gradients calculated by inversion of the interpreted BSR are relatively high (~130°C/km) along the ridge-crest area. Recent heat flow probe (22 thermistors over an active length of 5.2m) measurements over a ~3.5km section of ridge axis covering the Northern Svyatogor pockmark cluster record geothermal gradients ranging from 73°C/km to 131°C/km (Bohrmann, et al., 2017). The thermal gradients were systematically higher inside pockmarks (on average 113°C/km) compared to outside pockmarks (on average 94°C/km). A similar trend was observed at Vestnesa Ridge although the Svyatogor temperature gradients were systematically higher due to the younger underlying crust (Bohrmann, et al., 2017).

The Bohrmann, et al., (2017) measurements typically report lower thermal gradients than those obtained by BSR inversion, which probably reflects an incorrect velocity was used to depth convert the interpreted BSR surface. However, they do support the idea that focussed fluid flow can lead to local scale variation in heat flow similar to that described by Hustoft, et al., (2006). Interestingly, these thermal anomalies persist even though the northern Svyatogor pockmarks appear to have been inactive for some time. This may reflect the episodic nature of the fluid migration i.e. during leakage advection of fluids could drive relatively rapid fluxes of heat while during periods of quiescence the system may become diffusion dominated causing the return to thermal equilibrium to be very slow.

The particularly strong BSR shoaling at the eastern end of the dataset (BSR derived geothermal gradient of ~140°C/km) might indicate that migration of fluids from depth is locally concentrated in this region and more active compared to the main ridge-crest area. Hornbach, et al., (2012) made similar observations of pronounced BSR shoaling in an area of fractured sediments but with no evidence for seafloor venting. These observations support the possibility the small-throw faults with variable orientations that were observed at the eastern end of the dataset could be a result of hydrofracturing. An interpreted fluid conduit associated with this zone of pronounced BSR shoaling appears to have been inactive for some time, with no bathymetric expression at the seabed. An alternate interpretation is that the flow conduit could be a result of ongoing fluid migration and is still growing toward the seabed. Cathles, et al., (2010) predict that seafloor pockmark formation is likely to initiate when a gas chimney reaches half way from the base of its source gas pocket to the seafloor. Given that this flow conduit extends more than half way from the free-gas zone to the seafloor, the lack of a seafloor pockmark in this area is inconsistent with the interpretation of an upward-propagating gas chimney. While fluid leakage therefore appears to be inactive, if the BSR shoaling indeed reflects preferential fault-controlled migration of deep fluids into this area, it remains possible that this flow conduit could be reactivated at some point in the future.

The relatively low BSR derived geothermal gradients along the flank of the ridge may reflect more of a 'background' level of the heat flow for the study area i.e. in the absence of recent fault-controlled advection of fluids from depth. Buried fluid-flow conduits were observed in this area, associated with the hanging wall of faults. However, these flow conduits are made up of much smaller pockmark features than observed along the ridge crest. They do not reach the seafloor and appear to have become inactive around horizon H2, which is earlier than the flow conduits associated with higher geothermal gradients. The longer period of inactivity with respect to fluid leakage in this area is consistent with heat flow having sufficient time to re-equilibrate to a background level. The position of these flow conduits away from the crest of a free-gas zone structural closure would also tend to decrease their chance of becoming active if free-gas zone overpressure plays a role in promoting fault slip.

5.4.4 Fluid migration: dissolved methane or free-gas?

Free-gas is not typically stable within the hydrate stability zone, and will tend to form hydrate. However, modelling supported by field observations from ODP drilling of methane venting through the hydrate stability zone at southern Hydrate Ridge, offshore Oregon, suggests that free-gas supplied from below forms hydrate, depletes water, and elevates salinity until the pore water becomes too saline for further hydrate formation (Liu & Flemings, 2006). This generates a local three-phase equilibrium where free-gas, gas-hydrate and water can coexist and allows free-gas migration to the seafloor (Liu & Flemings, 2006). However, this three-phase equilibrium may only occur when the methane is supplied dominantly in a free-gas phase (Daigle & Dugan, 2010). If methane supply is dominantly in the dissolved phase, free-gas is unable to enter the regional hydrate stability zone because water flux removes excess salt before it reaches three-phase equilibrium (Daigle & Dugan, 2010).

The lack of amplitude anomalies indicative of free-gas within the hydrate stability zone at Svyatogor Ridge, in addition to the proximity of an active spreading ridge (with presumably active hydrothermal circulation), imply that Svyatogor is likely a water-dominated system with gas supplied dominantly in the dissolved phase. Therefore, following Daigle & Dugan (2010), hydrate is expected to form throughout the regional hydrate stability zone, with the highest concentration at the BSR. Free-gas will only initiate fractures at the BSR when the column of gas that develops beneath the BSR reaches the critical pressure required for failure, leading to episodic, focused flow, and migration of gas through the regional hydrate stability zone controlled mainly by pressure fluctuations (Daigle & Dugan, 2010). This dynamic is consistent with the observation of abundant pockmark and gas chimney features in the Svyatogor 2016 3D dataset indicating that rapid, focussed fluid flow has occurred in the past even though evidence of present day leakage is lacking.

5.4.5 Potential of the free-gas zone to promote fault slip

The maximum thickness of the free-gas zone within the Svyatogor 2016 3D survey was ~75m along the ridge crest (see section 4.4 pg.74). The thickness of the free-gas zone is an upper bound on the interconnected gas column height, which could be significantly smaller if only the upper part of the gas zone forms an interconnected phase and the lower part is just a residual gas phase (Flemings, et al., 2003). However, if we simply assume the free-gas zone thickness corresponds to the interconnected column height, it is estimated to be 49-76% of the column height required to induce fault reactivation (see section 4.4.1.4 pg.78). In the unlikely situation of hydrate growth along grain

contacts rather than within pores allowing a Poisson's ratio of 0.38 and Biot's coefficient of 0.98 (see section 4.4.1.4 pg.78), the column height may be as high as high as 97% of the critical threshold. However, it is anticipated that hydrate is more likely to grow within pores before becoming interconnected at higher hydrate saturations (Lee, et al., 2010), so the 49-76% of critical pressure range is considered most probable. This is a plausible result because while observed gas chimney/pockmark structures indicate that leakage has occurred in the past, the seafloor pockmarks appear to have been inactive for some time and no gas flares have been observed in the water column above Svyatogor ridge to indicate leakage is currently active.

Svyatogor ridge developed in an active tectonic setting, whose activity continues to the present day as illustrated by the distribution of modern earthquakes recorded over the period 1953-2016 (see Figure 3). As a result, it is considered unlikely that the free-gas system would grow to a size where it was capable of initiating fault reactivation and leakage into the overlying gas-hydrate stability zone due to gas-buoyancy related overpressure alone. Rather, it is anticipated that free-gas zone overpressures may significantly increase the slip tendency of faults at Svyatogor ridge making them more prone to reactivation when tectonic events related to the active spreading Knipovich ridge produce dynamic stresses triggering fault slip.

5.5 Size of the free-gas system – implications for methane source

It is estimated that the free-gas system imaged by the Svyatogor 2016 3D survey may hold somewhere in the vicinity of $3.7x10^{9}$-$1.5x10^{11}$ moles of methane (see section 4.4 pg.74). The methane held in this system may be derived by two classes of processes: in-situ production or advection from a deeper source (Paull, et al., 1994; Milkov, 2005; Bhatnagar, et al., 2007). This section presents a discussion of the likelihood of methane being dominantly supplied by in-situ methane production due to bacterial degradation of organic carbon during burial, or by advection of methane from a deeper source.

5.5.1 Consideration of in-situ methane production

In-situ methanogenesis in a closed system (i.e. no external advection) is ultimately constrained by the amount of solid organic carbon that is buried with sediments (Paull, et al., 1994; Bhatnagar, et al., 2007). However, not all of the TOC (total organic carbon) in the sediments is available for conversion to methane by methanogenic bacteria. For example, methane production commences below the aerobic oxidation and sulphate reduction zones. If sediments are buried too slowly, the utilizable organic carbon may be consumed in the aerobic oxidation and sulphate reduction zones before reaching the methanogenesis zone (Waseda, 1998). Comparing oxygen to carbon ratios in fresh sediment with late diagenetic stage sediment, to estimate the amount of carbon consumed in the methanogenesis zone, Clayton et. al. (1992) derive that under optimal conditions about 10% of the TOC can be converted to methane.

During methanogenesis, microbial methane production causes pore-fluid methane concentrations to increase. However, for gas hydrate to form the pore-fluid methane concentration must equal or exceed saturation before the base of the gas hydrate stability zone (BGHSZ) is reached (Paull, et al., 1994; Bhatnagar, et al., 2007). As burial continues, the precipitated gas-hydrate will typically reach the

BGHSZ where it will dissociate and form a free-gas layer immediately below gas hydrate (Davie & Buffett, 2001).

The methane solubility at the Svyatogor ridge BGHSZ with conditions of ~290K, ~20MPa, seawater salinity is estimated as 144mM.L^{-1} (mole fraction ~0.0026) using the Bhatnagar et. al. (2007) solubility curves. Taking 10% as the amount of TOC available to methane producing bacteria and assuming a porosity of 47%, bulk density of 1.952g.cm^{-3} (average of 100-200mbsf measurements from ODP holes 908-912) and pore water density of 1.027g.cm^{-3} it is estimated that the TOC must exceed 1.38% of dry sediment weight for pore-water methane to reach saturation and gas-hydrate to form.

The average TOC of sediments <2.8Mya from ODP holes 908-912 is ~0.73%, derived from O'Connel et. al. (1996) and the maximum TOC is 1.58%. This suggests that it has been a rare occurrence for sediments to be deposited in this region containing sufficient organic matter (>1.38% TOC) to allow in-situ produced methane to saturate pore fluids and precipitate gas hydrate. Furthermore, taking the maximum observed TOC (1.58%) for sediments <2.8Mya, it would require an average thickness of at least 20m across the entire Svyatogor 2016 3D survey to generate the volume of gas estimated to be held in the free-gas zone at the lowest assumed gas saturation of 0.5% (see Table 10). Such a thickness is not supported by the O'Connel et. al. (1996) TOC data. Lateral migration from further down the ridge flanks than imaged by the 3D survey could reduce this required thickness but then a more laterally extensive BSR than the observed would be expected. Thus, based on the O'Connel et. al. (1996) TOC values, it is considered unlikely the bulk of gas-hydrate at Svyatogor ridge could have formed by in-situ methane production and highly unlikely that in-situ methane production alone could have produced enough free-gas to explain the scale of the observed free-gas system.

Table 10 – Average 1.58% TOC sediment thickness required across Svyatogor 2016 3D survey area (~2.2x$10^7$$m^2$) to produce observed quantity of free-gas (at different assumed average gas saturation) by in-situ methane generation alone.

Bulk volume (km^3)	Porosity (Φ)	Gas Saturation (%)	Gas amount (moles)	Average thickness of sediment with 1.58% TOC required across 3D survey area (m)
0.187	0.47	0.5	4.35x10^9	20
		1	8.60x10^9	40
		2	1.75x10^{10}	81
		5	4.35x10^{10}	202
		10	8.60x10^{10}	399
		15	1.30x10^{11}	603

5.5.2 Consideration of a potential thermogenic source

Advection from a deeper thermogenic methane source is considered relatively unlikely because the sedimentary thickness of only ~700m at Svyatogor ridge is not sufficient to reach the thermogenic gas generation window (100-200°C) even with the relatively high geothermal gradient of 130-140°C/km that was estimated for the study area. However, geochemical analyses have shown that the methane at Vestnesa Ridge may be dominantly thermogenic (Smith, et al., 2014). In addition, 1D petroleum systems modelling has indicated that gas generation potential from Miocene source rocks exists along a transect (see Figure 75) extending from ODP site 909 towards the NE side of the Molloy Transform Fault (MTF), although the amount of hydrocarbons produced SE of the MTF was two orders of

magnitude lower than that to the NE (Dumke, et al., 2016). Therefore, a possible contribution from a thermogenic source located to the north of Svyatogor Ridge with lateral gas migration up the ridge flank and focussed along the crest cannot be conclusively ruled out.

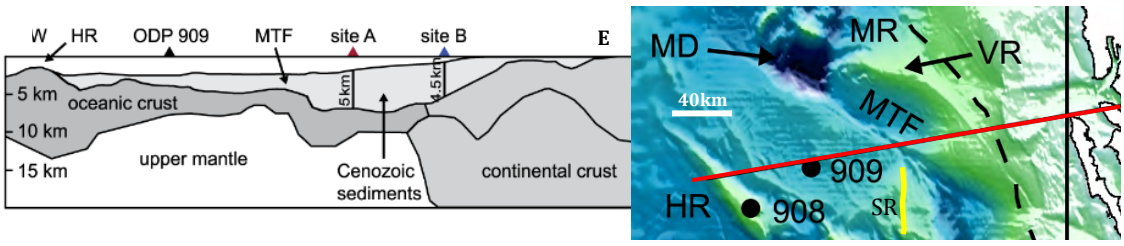


Figure 75 – Modified from Dumke, et al. (2016) who conducted 1D petroleum systems modelling and report thermogenic gas generation is possible from Miocene source rocks at ODP site 909 and sites A & B (generated mass of hydrocarbons was two orders of magnitude lower at ODP 909 than at sites A & B). The position of the geological model is shown on the inset map with a red line and the axis of Svyatogor Ridge is marked in yellow. HR – Hovgaard Ridge, MTF – Molloy Transform Fault, MD – Molloy Deep, MR – Molloy Ridge, VR – Vestnesa Ridge, SR – Svyatogor Ridge.

5.5.3 Consideration of an abiotic methane source

Johnson et. al. (2015) proposed that abiotic methane produced by the serpentinization of ultramafic rocks could contribute a significant amount of methane to the Svyatogor gas-hydrate system. The flux of serpentinization derived methane per km of ridge axis has been estimated by Cannat et. al. (2010) for the Mid-Atlantic Ridge, with the closest site being the Rainbow hydrothermal field (~36°N).

The spreading rate at Rainbow is classed as slow, with an exhumation rate of 10-11mm/yr and the hydrothermal vents occur above the exhumed footwall of axial detachment faults (Cannat, et al., 2010). The exhumation rate at Knipovich spreading ridge adjacent to the study area is similar at 8mm/yr (Ehlers & Jokat, 2009) and the association with axial detachment faults is also similar at Svyatogor ridge. Therefore, while a large degree of uncertainty remains due to the geographic separation, we may expect some degree of similarity between the methane flux estimated at Rainbow hydrothermal field and the potential methane flux from the Knipovich spreading ridge adjacent to the study area.

Table 11 – Time required for the amount of gas estimated to be held in the observed free-gas zone (see section 4.4 pg.74) to accumulate for different average gas saturations based on the flux of serpentinization derived methane estimated by (Cannat, et al., 2010) at Rainbow hydrothermal field.

Bulk volume (km^3)	Porosity (Φ)	Gas Saturation (%)	Gas amount (moles)	Accumulation time (kyrs) at $7.2x10^5$ $moles.yr^{-1}$ flux
0.187	0.47±0.07	0.5	$3.7x10^9 – 5.0x10^9$	5 - 7
		1	$7.3x10^9 – 9.9x10^9$	10 - 14
		2	$1.5x10^{10} – 2.0x10^{10}$	21 - 28
		5	$3.7x10^{10} – 5.0x10^{10}$	51 - 69
		10	$7.3x10^{10} – 9.9x10^{10}$	101 - 138
		15	$1.1x10^{11} – 1.5x10^{11}$	153 - 208

The methane flux per km of spreading ridge at Rainbow hydrothermal field was estimated at $4x10^5$ $moles.km^{-1}.yr^{-1}$ (Cannat, et al., 2010). For simplicity, we assume that 1km of spreading ridge feeds

laterally into 1km of Svyatogor ridge and ignore possible topographic focussing effects and/or loss of methane during migration from source to reservoir. With these assumptions, the methane flux into the section of Svyatogor ridge imaged by the Svyatogor 2016 3D dataset would be $7.2x10^5$ moles.yr^{-1} and it would take approximately 5-200kyr (see Table 11) to charge the observed free-gas zone with serpentinization derived (abiotic) methane. Given the age of Svyatogor ridge is at least 2Ma (Johnson, et al., 2015), if the assumptions associated with these calculations hold, there has been sufficient time for the observed free-gas zone to be charged by abiotic methane. From the observed pockmark and gas chimney structures we infer that significant leakage episodes have occurred in the past, implying the total volume of methane fed into the system over time must be significantly larger than the current size of the reservoir. However, given the timescales involved, it still appears plausible that an abiotic source could largely sustain the system.

6 Conclusions

The stratigraphic correlation of the YP-2/YP-3 boundary indicates that the Svyatogor sediment drift must be older than the ~2 Mya proposed by Johnson, et al., (2015). An alternate model is proposed whereby the Svyatogor sediment drift grew as a basement/fault controlled drift, controlled by the axial basement detachment faults that it remains associated with to the present day. Sedimentation likely began shortly after significant seafloor surface expression of these faults developed (possibly up to ~5.7Ma).

The dominant mode of faulting observed in the Svyatogor 2016 3D dataset are normal faults with a mean strike of ~014° and dip of 50-60° towards the east and are likely tectonically driven and controlled by the mechanics of the oblique rifting of Knipovich spreading ridge. The normal faults are predominantly sedimentary and have greatly influenced the creation of accommodation space and sediment accumulation at Svyatogor ridge. At the local scale, depocenters tend to switch spatial location during the evolution of Svyatogor ridge controlled by the relative activity of different faults. Vertical throw gradients on T-z profiles record a transition from an earlier period of lower fault activity to a more recent period of enhanced fault activity.

The degree to which free-gas zone overpressure or external tectonic stresses control fault-slip is difficult to differentiate, but it is most likely that both play a role i.e. free-gas zone overpressures may significantly increase the slip tendency of faults at Svyatogor ridge making them more prone to slip or reactivation when earthquakes related to the active spreading Knipovich ridge produce dynamic stresses triggering fault slip. Continued re-opening of fractures at fault tips or fault-segment junctions may provide a kinetic mechanism whereby free-gas could migrate to the seafloor faster than hydrate precipitation can inhibit.

A summary of the interpreted kinematics of the Svyatogor gas-hydrate free-gas system is that 1) gas migrates into the system along faults, probably as a dissolved phase 2) gas-hydrate formation at the BGHSZ produces a hydrate-cemented seal that results in a structurally enhanced trap 3) continued gas migration from depth and recycling of hydrate at the BGHSZ leads to accumulation of a free-gas

phase contained beneath the structurally enhanced hydrate-cemented trap 4) overpressure builds beneath the BGHSZ as gas charge continues eventually resulting in episodic gas release triggered in combination with dynamic stresses from earthquakes. The system appears to be relatively stable at present – continued leaking is considered unlikely due to the lack of shallow gas amplitude anomalies or water column gas flares and the observed inactivity and infilling of the large seafloor pockmarks along the crest of the ridge. Shoaling of the BSR at the eastern end of the Svyatogor 2016 3D dataset and associated anomalously high thermal gradient may indicate migration of fluids from depth is more active in this eastern region than in the main ridge-crest area.

Based on estimates of in-situ methane production it appears highly unlikely that the Svyatogor gas-hydrate/free-gas system could have developed in the absence of advection from an external/deep methane source. It remains difficult to conclusively rule out the contribution of a thermogenic source, but this would likely be dependent on lateral migration from the north as the relatively thin sedimentary section at Svyatogor lies above the gas generation window. However, it does appear plausible that the observed free-gas zone could have been charged by abiotic methane migrating along axial detachment faults during the period of active sedimentation on Svyatogor ridge assuming a similar flux rate to that reported by Cannat et. al. (2010) for the Rainbow hydrothermal field. In light of this, with presently available data we are left to conclude that a dominantly abiotic methane source as proposed by Johnson et. al. (2015) is indeed the most likely scenario.

Definitively resolving the issue of methane source is likely to require geochemical sampling and isotopic measurement of the methane at some time in the future. Since the system appears to have been relatively inactive with respect to leakage during the most recent geological past, it has been difficult to obtain gas/hydrate samples from shallow sediment cores. However, a small elongated pockmark located above the lateral tip of an underlying fault is likely a younger and more recently active feature than the larger pockmarks and appears to be the most promising target for future sediment coring aiming to recover gas or hydrate bearing samples for geochemical analysis.

References

Andresen, K. & Huuse, M., 2011. 'Bulls-eye'pockmarks and polygonal faulting in the Lower Congo Basin: relative timing and implications for fluid expulsion during shallow burial. *Marine Geology,* 279(1), pp. 111-127.

Bancroft, J. C., 1995. Aliasing in prestack migration. *Society of exploration geophysicists,* Volume 7, p. 16.

Bancroft, J. C., Guirigay, T. & Isaac, H., 2012. Deconvolution After Migration. *CSEG RECORDER,* 37(10), pp. 28-31.

Baudon, C., 2007. *Propagation and early growth of normal faults.* s.l.:PhD diss., Cardiff University.

Baudon, C. & Cartwright, J., 2008. 3D seismic characterisation of an array of blind normal faults in the Levant Basin, Eastern Mediterranean. *Journal of Structural Geology,* 30(6), pp. 746-760.

Baudon, C. & Cartwright, J., 2008b. Early stage evolution of growth faults: 3D seismic insights from the Levant Basin, Eastern Mediterranean.. *Journal of Structural Geology,* 30(7), pp. 888-898.

Behn, M., Lin, J. & Zuber, M., 2002. Evidence for weak oceanic transform faults. *Geophysical Research Letters,* 29(24).

Bekara, M. & van der Baan, M., 2009. Random and coherent noise attenuation by empirical mode decomposition. *Geophysics,* 74(5), pp. v89-v98.

Bhatnagar, G. et al., 2007. Generalization of gas hydrate distribution and saturation in marine sediments by scaling of thermodynamic and transport processes.. *American Journal of Science,* 307(6), pp. 861-900.

Bohrmann, G. et al., 2017. *R/V MARIA S. MERIAN Cruise Report MSM57, Gas Hydrate Dynamics at the Continental Margin of Svalbard, Reykjavik – Longyearbyen – Reykjavik, 29 July – 07 September 2016.,* Bremen ISSN 2195-7894.: Berichte, MARUM – Zentrum für Marine Umweltwissenschaften, Fachbereich Geowissenschaften, Universität Bremen, No. 314.

Boyer, T. et al., 2013. *World Ocean Database 2013, NOAA Atlas NESDIS 72.* Silver Spring(MD): s.n.

Brown, A. R., 2011. *Interpretation of Three-Dimensional Seismic Data.* 7th ed. Tulsa, Oklahoma, U.S.A.: AAPG & SEG.

Buck, W., Lavier, L. & Poliakov, A., 2005. Modes of faulting at mid-ocean ridges.. *Nature,* 434(7034), pp. 719-723.

Bünz, S. et al., 2012. Active gas venting through hydrate-bearing sediments on the Vestnesa Ridge, offshore W-Svalbard. *Marine Geology,* Volume 332, pp. 189-197.

Cannat, M., Fontaine, F. & Escartin, J., 2010. Serpentinization and associated hydrogen and methane fluxes at slow spreading ridges. In: P. Rona, C. Devey, J. Dyment & B. Murton, eds. *Diversity of hydrothermal systems on slow spreading ocean ridges .* Washington DC: American Geophysical Union Geophysical Monograph Series, 188., pp. 241-264.

Cao, J., Wang, Y., Zhao, J. & Yang, C., 2011. A review on restoration of seismic wavefields based on regularization and compressive sensing. *Inverse Problems in Science and Engineering,* 19(5), pp. 679-704.

Carcione, J., Gei, D., Rossi, G. & Madrussani, G., 2005. Estimation of gas-hydrate concentration and free-gas saturation at the Norwegian-Svalbard continental margin.. *Geophysical Prospecting,* 53(6), pp. 803-810.

Cartwright, J., 1994. Episodic basin-wide hydrofracturing of overpressured Early Cenozoic mudrock sequences in the North Sea Basin. *Marine and Petroleum Geology,* 11(5), pp. 587-607.

Cartwright, J., 2007. The impact of 3D seismic data on the understanding of compaction, fluid flow and diagenesis in sedimentary basins. *Journal of the Geological Society,* 164(5), pp. 881-893.

Cartwright, J., Bouroullec, R., James, D. & Johnson, H., 1998. Polycyclic motion history of some Gulf Coast growth faults from high-resolution displacement analysis.. *Geology,* 26(9), pp. 819-822.

Cartwright, J., Trudgill, B. & Mansfield, C., 1995. Fault growth by segment linkage: an explanation for scatter in maximum displacement and trace length data from the Canyonlands Grabens of SE Utah. *Journal of Structural Geology,* 17(9), pp. 1319-1326.

Castelltort, S., Pochat, S. & Van Den Driessche, J., 2004. Using T–Z plots as a graphical method to infer lithological variations from growth strata. *Journal of Structural Geology,* 26(8), pp. 1425-1432.

Cathles, L., Su, Z. & Chen, D., 2010. The physics of gas chimney and pockmark formation, with implications for assessment of seafloor hazards and gas sequestration. *Marine and Petroleum Geology,* 27(1), pp. 82-91.

Chamov, N. et al., 2010. Structure and composition of the sedimentary cover in the Knipovich Rift valley and Molloy Deep (Norwegian-Greenland basin). *Lithology and Mineral Resources,* 45(6), pp. 532-554.

Chase, M., 1992. Random noise reduction by FXY prediction filtering. *Exploration Geophysics,* 23(2).

Childs, C., Nicol, A., Walsh, J. & Watterson, J., 2003. The growth and propagation of synsedimentary faults. *Journal of Structural Geology,* 25(4), pp. 633-648.

Clayton, C., 1992. Source volumetrics of biogenic gas generation. In: R. Vially, ed. *Bacterial Gas.* Paris: Technip, pp. 191-204.

Clifton, A. & Schlische, R., 2003. Fracture populations on the Reykjanes Peninsula, Iceland: Comparison with experimental clay models of oblique rifting. *Journal of Geophysical Research: Solid Earth,* 108(B2).

Clifton, A., Schlische, R., Withjack, M. & Ackermann, R., 2000. Influence of rift obliquity on fault-population systematics: results of experimental clay models. *Journal of Structural Geology,* 22(10), pp. 1491-1509.

Crane, K., Sundvor, E., Buck, R. & Martinez, F., 1991. Rifting in the northern Norwegian-Greenland Sea: Thermal tests of asymmetric spreading. *Journal of Geophysical Research,* 96(14), pp. 529-550.

Curewitz, D. & Karson, J., 1997. Structural settings of hydrothermal outflow: Fracture permeability maintained by fault propagation and interaction. *Journal of Volcanology and Geothermal Research,* 79(3), pp. 149-168.

Curewitz, D. et al., 2010. Structural analysis of fault populations along the oblique, ultra-slow spreading Knipovich Ridge, North Atlantic Ocean, 74° 30′ N-77° 50′ N. *Journal of Structural Geology,* 32(6), pp. 727-740.

Daigle, H. & Dugan, B., 2010. Effects of multiphase methane supply on hydrate accumulation and fracture generation. *Geophysical Research Letters,* 37(20).

Davie, M. & Buffett, B., 2001. A numerical model for the formation of gas hydrate below the seafloor. *Journal of Geophysical Research: Solid Earth,* 106(B1), pp. 497-514.

DeMets, C., Gordon, R., Argus, D. & Stein, S., 1990. Current plate motions. *Geophysical journal international,* 101(2), pp. 425-478.

DeMets, C., Gordon, R., Argus, D. & Stein, S., 1994. Effect of recent revisions to the geomagnetic reversal time scale on estimates of current plate motions. *Geophysical research letters,* 21(20), pp. 2191-2194.

Dickens, G., 2011. Down the Rabbit Hole: toward appropriate discussion of methane release from gas hydrate systems during the Paleocene-Eocene thermal maximum and other past hyperthermal events. *Climate of the Past,* 7(3), pp. 831-846.

Dumke, I. et al., 2016. Gas hydrate distribution and hydrocarbon maturation north of the Knipovich Ridge, western Svalbard margin. *Journal of Geophysical Research: Solid Earth,* 121(3), pp. 1405-1424.

Egbai, J., Atakpo, E. & Aigbogun, C. O., 2012. Predictive deconvolution in seismic data processing in Atala prospect of rivers State, Nigeria. *Advances in Applied Science Research,* 3(1), pp. 520-529.

Egbert, G. D. & Erofeeva, S. Y., 2002. Efficient Inverse Modeling of Barotropic Ocean Tides. *Journal of Atmospheric and Oceanic Technology,* Volume 19, pp. 183-204.

Ehlers, B. & Jokat, W., 2009. Subsidence and crustal roughness of ultra-slow spreading ridges in the northern North Atlantic and the Arctic Ocean. *Geophysical Journal International,* 177(2), pp. 451-462.

Eiken, O. & Hinz, K., 1993. Contourites in the Fram Strait. *Sedimentary Geology,* 82(1-4), pp. 15-32.

Elboth, T., Geoteam, F. & Hermansen, D., 2009. *Attenuation of noise in marine seismic data.* SEG Annual Meeting, Society of Exploration Geophysicists.

Engelder, T. & Fischer, M. P., 1994. Influence of poroelastic behavior on the magnitude of minimum horizontal stress Sh in overpressured parts of sedimentary basins. *Geology,* Volume 22, pp. 949-952.

Engen, Ø., Faleide, J. & Dyreng, T., 2008. Opening of the Fram Strait gateway: A review of plate tectonic constraints. *Tectonophysics,* 450(1), pp. 51-69.

Flemings, P. B., Xiaoli, L. & Winters., W. J., 2003. Critical pressure and multiphase flow in Blake Ridge gas hydrates. *Geology,* 31(12), pp. 1057-1060.

Fossen, H. & Rotevatn, A., 2016. Fault linkage and relay structures in extensional settings—A review. *Earth-Science Reviews,* Volume 154, pp. 14-28.

Galbraith, M. & Yao, Z., 2012. FX plus deconvolution for seismic data noise reduction. *SEG Annual Meeting. Society of Exploration Geophysicists*.

Gebhardt, A., Geissler, W., Matthiessen, J. & Jokat, W., 2014. Changes in current patterns in the Fram Strait at the Pliocene/Pleistocene boundary. *Quaternary Science Reviews,* Volume 92, pp. 179-189.

Geissler, W. & Jokat, W., 2004. A geophysical study of the northern Svalbard continental margin. *Geophysical Journal International,* 158(1), pp. 50-66.

Geissler, W., Jokat, W. & Brekke, H., 2011. The Yermak Plateau in the Arctic Ocean in the light of reflection seismic data-implication for its tectonic and sedimentary evolution. *Geophysical Journal International,* 187(3), pp. 1334-1362.

Gomez, C. & Tatham, R., 2007. Sensitivity analysis of seismic reflectivity to partial gas saturation.. *Geophysics,* 72(3), pp. C45-C57.

Hamilton, E., 1979. V p/V s and Poisson's ratios in marine sediments and rocks. *The Journal of the Acoustical Society of America,* 66(4), pp. 1093-1101.

Holbrook, W. et al., 1996. Methane hydrate and free-gas on the Blake Ridge from vertical seismic profiling. *Science,* 273(5283), p. 1840.

Hooper, E., 1991. Fluid migration along growth faults in compacting sediments. *Journal of Petroleum Geology,* 14(S1), pp. 161-180.

Hornbach, M., Bangs, N. & Berndt, C., 2012. Detecting hydrate and fluid flow from bottom simulating reflector depth anomalies. *Geology,* 40(3), pp. 227-230.

Hornbach, M. J., Saffer, D. M. & Steven Holbrook, W., 2004. Critically pressured free-gas reservoirs below gas-hydrate provinces. *Nature,* 427(6970), pp. 142-144.

Howe, J., Shimmield, T., Harland, R. & Eyles, N., 2008. Late Quaternary contourites and glaciomarine sedimentation in the Fram Strait. *Sedimentology,* 55(1), pp. 179-200.

Hustoft, S., Bünz, S., Mienert, J. & Chand, S., 2009. Gas hydrate reservoir and active methane-venting province in sediments on< 20 Ma young oceanic crust in the Fram Strait, offshore NW-Svalbard. *Earth and Planetary Science Letters,* 284(1), pp. 12-24.

Hustoft, S., Mienert, J., Bünz, S. & Vanneste, M., 2006. Heat-, fluid flow, and geological variations controlling gas hydrate systems on the western Svalbard continental shear margin. *Geophysical Research Abstracts,* Volume 8, p. 06655.

Johnson, J. E. et al., 2015. Abiotic methane from ultraslow-spreading ridges can charge Arctic gas hydrates. *Geology,* 43(5), pp. 371-374.

Jones, I. F., 2011. A Practical Review of Migration Issues and Solutions. *Search and Discovery Article #40723.*

Kattenhorn, S. & Pollard, D., 2001. Integrating 3-D seismic data, field analogs, and mechanical models in the analysis of segmented normal faults in the Wytch Farm oil field, southern England, United Kingdom. *AAPG bulletin,* 85(7), pp. 1183-1210.

Katzman, R., Holbrook, W. & Paull, C., 1994. Combined vertical-incidence and wide-angle seismic study of a gas hydrate zone, Blake Ridge. *Journal of Geophysical Research: Solid Earth,* 99(B9), pp. 17975-17995.

Knies, J. et al., 2009. The Plio-Pleistocene glaciation of the Barents Sea–Svalbard region: a new model based on revised chronostratigraphy. *Quaternary Science Reviews,* 28(9), pp. 812-829.

Kuenen, P., 1956. The difference between sliding and turbidity flow.. *Deep Sea Research (1953),* 3(2), pp. 134-139.

Kvenvolden, K., 1993. Gas hydrates-geological perspective and global change. *Reviews of Geophysics,* 31(2), pp. 173-187.

Kvenvolden, K., 1995. A review of the geochemistry of methane in natural gas hydrate. *Organic Geochemistry,* 23(11-12), pp. 997-1008.

Kvenvolden, K., 1998. A primer on the geological occurrence of gas hydrate. *Geological Society, London, Special Publications,* 137(1), pp. 9-30.

Landrø, M., 1992. Modelling of GI gun signatures. *Geophysical Prospecting,* 40(7), pp. 721-747.

Larner, K. et al., 1983. Coherent noise in marine seismic data. *Geophysics,* 48(7), pp. 854-886.

Lee, H.-Y.et al., 2004. High-resolution shallow marine seismic surveys off Busan and Pohang, Korea, using a small-scale multichannel system. *Journal of Applied Geophysics,* 56(1), pp. 1-15.

Lee, J., Francisca, F., Santamarina, J. & Ruppel, C., 2010. Parametric study of the physical properties of hydrate-bearing sand, silt, and clay sediments: 2. Small-strain mechanical properties. *Journal of Geophysical Research: Solid Earth,* 115(B11105).

Lee, M. W., 2002. Biot–Gassmann theory for velocities of gas hydrate-bearing sediments. *Geophysics,* 67(6), pp. 1711-1719.

Lines, L., Brittle, K., Watson, I. & Cary, P., 2001. Aliasing for the Layperson. *CSEG Recorder Online,* April.26(4).

Liu, X. & Flemings, P. B., 2006. Passing gas through the hydrate stability zone at southern Hydrate Ridge, offshore Oregon.. *Earth and Planetary Science Letters,* 241(1-2), pp. 211-226.

Lu, S. & McMechan, G., 2002. Estimation of gas hydrate and free-gas saturation, concentration, and distribution from seismic data.. *Geophysics,* 67(2), pp. 582-593.

MacDonald, I. et al., 1994. Gas hydrate that breaches the sea floor on the continental slope of the Gulf of Mexico. *Geology,* 22(8), pp. 699-702.

Maldonado, A. et al., 2005. Miocene to recent contourite drifts development in the northern Weddell Sea (Antarctica). *Global and Planetary Change,* 45(1), pp. 99-129.

Mansfield, C. & Cartwright, J., 1996. High resolution fault displacement mapping from three-dimensional seismic data: evidence for dip linkage during fault growth. *Journal of Structural Geology,* 18(2-3), pp. 249-263.

Mattingsdal, R. et al., 2014. A new 6 Myr stratigraphic framework for the Atlantic–Arctic Gateway. *Quaternary Science Reviews,* Volume 92, pp. 170-178.

McAdoo, B., Pratson, L. & Orange, D., 2000. Submarine landslide geomorphology, US continental slope. *Marine Geology,* 169(1), pp. 103-136.

Milkov, A., 2004. Global estimates of hydrate-bound gas in marine sediments: how much is really out there?. *Earth-Science Reviews,* 66(3), pp. 183-197.

Milkov, A., 2005. Molecular and stable isotope compositions of natural gas hydrates: A revised global dataset and basic interpretations in the context of geological settings.. *Organic Geochemistry,* 36(5), pp. 681-702.

Mosher, D. C. & Simpkin, P. G., 1999. Environmental Marine Geoscience 1. Status and Trends of Marine High-Resolution Seismic Reflection Profiling: Data Acquistion.. *Geoscience Canada,* 26(4).

Moss, J., Cartwright, J. & Moore, R., 2012. Evidence for fluid migration following pockmark formation: Examples from the Nile Deep Sea Fan. *Marine Geology,* Volume 303, pp. 1-13.

Nacht, P., De Oliveira, M., Roehla, D. & Costa, A., 2010. *Investigation of geological fault reactivation and opening..* Buenos Aires, Aergentina.: Association Argentina de Mecanica Computacional.

O'Connel, S., Cremer, M., Stein, R. & Wolf-Welling, T., 1996. Neogene paleoceanography and paleoclimate history from FramStrait: changes in accumulation rates (ODP Leg 151). In: J. Thiede, A. M. Myhre & J. Firth, eds. *oceedings of the OceanDrilling Program, Scientific results, Vol. 151.* College Station, TX: s.n., pp. 569-582.

Ocean Drilling Program, 2005. *Archive of Core and Site/Hole Data and Photographs from the Ocean Drilling Program (ODP).,* s.l.: NOAA National Centers for Environmental Information..

Paull, C., Ussle, W. & Borowski, W., 1994. Sources of Biogenic Methane to Form Marine Gas Hydrates In Situ Production or Upward Migration? a.. *Annals of the New York Academy of Sciences,* 715(1), pp. 392-409.

Petersen, C. J. et al., 2010. High-resolution P-Cable 3D seismic imaging of gas chimney structures in gas hydrated sediments of an Arctic sediment drift. *Marine and Petroleum Geology,* Volume 27, pp. 1981-1994.

Planke, S. et al., 2009. P-Cable high-resolution seismic. *Oceanography,* 22(1), p. 85.

Plaza-Faverola, A. et al., 2015. Role of tectonic stress in seepage evolution along the gas hydrate-charged Vestnesa Ridge, Fram Strait. *Geophysical Research Letters,* 42(3), pp. 733-742.

Plaza-Faverola, A. et al., 2010. Evidence from three-dimensional seismic tomography for a substantial accumulation of gas hydrate in a fluid-escape chimney in the Nyegga pockmark field, offshore Norway.. *Journal of Geophysical Research: Solid Earth,* 115(B8).

Portnov, A., Vadakkepuliyambatta, S., Mienert, J. & Hubbard, A., 2016. Ice-sheet-driven methane storage and release in the Arctic. *Nature Communications,* Volume 7.

Proskurowski, G. et al., 2008. Abiogenic Hydrocarbon Production at Lost City Hydrothermal Field. *Science,* 319(5863), pp. 604-607.

Rebesco, M., 2005. SEDIMENTARY ENVIRONMENTS | Contourites. In: R. C. Selley, L. Robin, M. Cocks & I. R. Plimer, eds. *Encyclopedia of Geology.* s.l.:Elsevier, pp. 513-527.

Rebesco, M. & Stow, D., 2001. Seismic expression of contourites and related deposits: a preface. *Marine Geophysical Researches,* 22(5-6), pp. 303-308.

Sella, G., Dixon, T. & Mao, A., 2002. REVEL: A model for recent plate velocities from space geodesy. *Journal of Geophysical Research: Solid Earth,* 107(B4).

Sercel, 2006. *Marine Sources.* Printed in France © Sercel 09/06: s.n.

Shankar, U., Riedel, M. & Sathe, A., 2010. Geothermal modeling of the gas hydrate stability zone along the Krishna Godavari Basin. *Marine Geophysical Research,* Volume 31, pp. 17-28.

Sharp, I., Gawthorpe, R., Underhill, J. & Gupta, S., 2000. Fault-propagation folding in extensional settings: Examples of structural style and synrift sedimentary response from the Suez rift, Sinai, Egypt.. *Geological Society of America Bulletin,* 112(12), pp. 1877-1899.

Singhroha, S., Bünz, S., Plaza-Faverola, A. & Chand, S., 2015. Gas hydrate and free-gas detection using seismic quality factor estimates from high-resolution P-cable 3D seismic data. *Interpretation,* 4(1), pp. SA39-SA54.

Sloan, E. J., 1998. *Clathrate hydrates of natural gases.* 2nd ed. s.l.:Marcel Dekker Inc..

Sloan, E. & Koh, C., 2008. *Clathrate Hydrates of Natural Gases.* 3rd ed. Boca Raton, FL: CRC Press.

Smith, A. J., Mienert, J., Bünz, S. & Greinert, J., 2014. Thermogenic methane injection via bubble transport into the upper Arctic Ocean from the hydrate-charged Vestnesa Ridge, Svalbard.. *Geochemistry, Geophysics, Geosystems ,* 15(5), pp. 1945-1959.

Smith, D. et al., 2002. Hydroacoustic monitoring of seismicity at the slow-spreading Mid-Atlantic Ridge. *Geophysical Research Letters,* 29(11).

Spitz, S., 1991. Seismic trace interpolation in the F-X domain. *Geophysics,* 56(6), pp. 785-794.

Starling, K. E. & Savidge, J., 1992. *Compressibility Factors for Natural Gas and Other Related Hydrocarbon Gases, American Gas Association (AGA) Transmission Measurement Committee Report No. 8.* 2nd ed. s.l.:American Petroleum Institute (API) MPMS, .

Sun, H., 2001. *Wavepath Migration For Depth Imaging and Velocity Analysis (Doctoral dissertation).* Viewed online http://utam.gg.utah.edu/UTAMtheses/HongchuanSun/thesis.html 16/01/2017: Department of Geology and Geophysics, The University of Utah.

Tinivella, U. & Accaino, F., 2000. Compressional velocity structure and Poisson's ratio in marine sediments with gas hydrate and free-gas by inversion of reflected and refracted seismic data (South Shetland Islands, Antarctica).. *Marine Geology,* 164(1), pp. 13-27.

Tucholke, B. & Lin, J., 1994. A geological model for the structure of ridge segments in slow spreading ocean crust.. *Journal of Geophysical Research: Solid Earth,* 99(B6), pp. 11937-11958.

USGS, 2017. *Search Earthquake Catalog - ANSS Comprehensive Earthquake Catalog (ComCat).* [Online]
Available at: https://earthquake.usgs.gov/earthquakes/search/
[Accessed 1 March 2017].

Vadakkepuliyambatta, S., Hornbach, M., Bünz, S. & Phrampus, B., 2015. Controls on gas hydrate system evolution in a region of active fluid flow in the SW Barents Sea. *Marine and Petroleum Geology,* Volume 66, pp. 861-872.

Vakulenko, S. A., Buryak, S. V., Gofman, P. A. & Finikov, D. B., 2014. *Deghosting of High Resolution Marine Seismic Data by Adaptive Filtering Algorithm.* s.l., Near Surface Geoscience 2014-First Applied Shallow Marine Geophysics Conference.

Vanneste, M., Guidard, S. & Mienert, J., 2005. Bottom-simulating reflections and geothermal gradients across the western Svalbard margin. *Terra Nova,* 17(6), pp. 510-516.

Waghorn, K. A. et al., 2015. *Origins and Driving Mechanisms for Shallow Methane Accumulations on the Svyatogor Ridge, Fram Strait.* San Francisco, American Geophysical Union.

Waghorn, K. et al., 2017. Paleo-fluid expulsion and contouritic drift formation on the Chatham Rise, New Zealand. *Basin Research.*

Waseda, A., 1998. Organic carbon content, bacterial methanogenesis, and accumulation processes of gas hydrates in marine sediments.. *Geochemical Journal,* 32(3), pp. 143-157.

Westbrook, G. et al., 2008. High-resolution 3D seismic investigations of hydrate-bearing fluid-escape chimneys in the Nyegga region of the Vøring plateau, Norway.. *Proceedings of the 6th International Conference on Gas Hydrates (ICGH 2008). 6-10 July, Vancouver, Canada,* p. 12pp.

Westvig, I., 2015. *Structural and Stratigraphic Setting and Fluid Flow Features of the Svyatogor Ridge, a Sediment Drift South of the Molloy Transform.* s.l.:MS thesis. UiT The Arctic University of Norway.

Willis, A., Poche, S., Degel, T. & Clarke, R., 2015. *Enhanced imaging with broadband seismic data processing in "under-explored" areas.* [Online]
Available at: http://www.spectrumgeo.com/geological-resources/technical-paper-library/enhanced-imaging-with-broadband-seismic-data-processing-in-under-explored-areas
[Accessed 29 November 2016].

Wolfe, C., Purdy, G., Toomey, D. & Solomon, S., 1995. Microearthquake characteristics and crustal velocity structure at 29°N on the Mid-Atlantic Ridge: The architecture of a slow spreading segment. *Journal of Geophysical Research,* Volume 100, pp. 24449-24472.

Wolf-Welling, T. et al., 1993. Cenozoic Arctic gateway paleoclimate variability: Indications from changes in coarse-fraction composition. In: J. Thiede, et al. eds. *PROCEEDINGS-OCEAN DRILLING PROGRAM SCIENTIFIC RESULTS.* s.l.:NATIONAL SCIENCE FOUNDATION, pp. 515-568.

Wood, L. C., Heiser, R. C., Treitel, S. & Riley, P. L., 1978. The debubbling of marine source signatures. *Geophysics,* 43(4), pp. 715-729.

Yilmaz, O. & Baysal, E., 2015. *An Effective Ghost Removal Method for Marine Broadband Seismic Data Processing.* Madrid, 77th EAGE Conference and Exhibition 2015.

Yilmaz, O. & Doherty, S. M., 2001. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data.* Tulsa, OK: Society of Exploration Geophysicists.

Yun, T., Francisca, F., Santamarina, J. & Ruppel, C., 2005. Compressional and shear wave velocities in uncemented sediment containing gas hydrate. *Geophysical Research Letters,* 32(10).

Yu, Z. et al., 2013. Distribution of large-scale detachment faults on mid-ocean ridges in relation to spreading rates. *Acta Oceanologica Sinica,* 32(12), pp. 109-117.

Zhang, Y.-Q. & Lu, W.-K., 2014. 2D and 3D prestack seismic data regularization using an accelerated sparse time-invariant Radon transform. *Geophysics,* 79(5), pp. 165-177.

Zillmer, M., 2006. A method for determining gas-hydrate or free-gas saturation of porous media from seismic measurements. *Geophysics,* 71(3), pp. N21-N32.

Appendix 1 – Python script for adjusting receiver geometry based on direct wave picks

```
#-------------------------------------------------------------------------------
# Name:        PCable GeometryFit
# Purpose: Apply misfit guided distortion to P-Cable geometry assigned by RADEX
#
# Author:      Rowan Romeyn
#
# Created:     19/07/2016
#-------------------------------------------------------------------------------
# Import required modules
import pandas
import numpy as np
from scipy.optimize import leastsq,fsolve
from scipy.integrate import quad
from matplotlib.backends.backend_pdf import PdfPages
from matplotlib import pyplot as plt
import matplotlib.collections
import pp #Parallel Python
import sys, os, glob
import time
start_time = time.time()

#nonbuffered_stdout = os.fdopen(sys.stdout.fileno(), 'w', 0) #controls frequency of print statements
#sys.stdout = nonbuffered_stdout

### Input and output directories ###
InputDir = 'TestData'#'ExportedGeometryHeaders' #subdirectory or full directory path containing input ASCII header files
OutputDir = 'ProcessedNavHeadersTest' #subdirectory or full directory path to save processed ASCII header files
OutputPlotDir = 'QC_plotsTest' #subdirectory or full directory path to save QC plots

NavDir = os.path.abspath("PythonInterpretedSeatrack/")#python interpreted seatrack files with LineXX included in filename where XX specifies line number e.g. Line01

### User input parameters section ################################################
PlotN = 50 #Plot geometry of every Nth shot for QC
MaxPlotShots = None #maximum number of shots to plot
PlotThresh = 200 #plot shots with misfit exceeding specified value, OVERRIDES PlotN and MaxPlotShots
AutoscalePlotColours = False #colours adjust automatically to data range
ManualPlotColourLim = 6 #if AutoscalePlotColours is False colour scale will be symmetric about zero up to this limit (m deltaoffset)
PlotInTransformCoords = False #QC plot without inverse coordinate transformation applied
LockPlotAspectRatio = False #If true the X and Y scales are forced to stay equal
PlotDeltaOffsetCircles = True #If True circles of radius deltaoffset plus offset along streamer are plotted
#WaterVelocity = 1.479 #water velocity in m/ms
WaterVelocitySOS = 1.479 #water velocity in m/ms at the start of the survey
WaterVelocityEOS = 1.479 #water velocity in m/ms at the end of the survey

updateStreamerXpos = True #if False streamers will only be shifted forwards and backwards, not recommended
FitThroughParavanes = True #cross cable model is forced to fit through paravanes
UseShotsForHeading = False #If true the heading of the ship will be a linear fit to the trend of the N shots defined by CalculateHeadingFromN_Shots
CalculateHeadingFromN_Shots = 2 #must be 2 or more shots to extract linear trend as heading
```

```
49. CalculateHeadingFromCHAN =[49,50,51,52,53,54,55,56] #if UseShotsForHeading is False the headin
    g is calculated from the linear trend throught this group of receivers
50. SetOriginAtShot = False #if true each shot will be transformed to coordinate system with origi
    n at shotpoint
51. CalculateOriginFromChan = [49,50,51,52,53,54,55,56] #if SetOriginAtShot is False the origin wi
    ll be placed at the mean point of this group of receivers
52.
53. NumStreamers = 14
54. NumChanPerStreamer = 8
55. StreamerOffsetList = [6+0.3+1.5625+3.125*0,6+0.3+1.5625+3.125*1,6+0.3+1.5625+3.125*2,6+0.3+1.5
    625+3.125*3,6+0.3+1.5625+3.125*4,6+0.3+1.5625+3.125*5,6+0.3+1.5625+3.125*6,6+0.3+1.5625+3.125*
    7]
56. ParavaneXshift = 0 #offset from GPS on paravane to wire connection point (pos moves connection
     point inwards)
57. ParavaneYshift_STB = 2
58. ParavaneYshift_PRT = 2
59. ### END USER INPUT PARAMETERS ###
60.
61. ### Data Input ##########################################################################
62. infileList = glob.glob(os.path.join(os.getcwd(), InputDir, "*.*"))
63. filecounter = 0
64. WaterVelList = np.linspace(WaterVelocitySOS,WaterVelocityEOS,len(infileList))
65.
66. for infile in infileList:
67.
68.     print "Processing", infile
69.
70.     data = pandas.read_csv(infile,sep='\t')
71.
72.     ### read data from specified header fields ###
73.     ShotList = np.asarray((data.SOURCE),dtype='int32')
74.     ChanList = np.asarray((data.CHAN),dtype='int16')
75.     SourceX_list = np.asarray((data.SOU_X),dtype='float64')
76.     SourceY_list = np.asarray((data.SOU_Y),dtype='float64')
77.     ReceiverX_list = np.asarray((data.REC_X),dtype='float64')
78.     ReceiverY_list = np.asarray((data.REC_Y),dtype='float64')
79.     CurrentOffsetList = np.asarray((data.ORIGOFF),dtype='float64') #initial offset header
80.     ObsDirWavT = np.asarray((data.PICK3),dtype='float64')#Direct wave arrival header
81.     HourList = np.asarray((data.HOUR),dtype='int16')
82.     MinList = np.asarray((data.MINUTE),dtype='int16')
83.     SecList = np.asarray((data.SECOND),dtype='int16')
84.     #print ShotList
85.
86.     NumLines = len(ShotList)
87.     FirstShot = np.min(ShotList)
88.     LastShot = np.max(ShotList)
89.     ShotCount = (LastShot - FirstShot) + 1
90.     print NumLines, 'records found in file'
91.     print 'File contains', ShotCount, 'shots, from', FirstShot, 'to', LastShot
92.
93.     ### The maths ###########################################################################
94.     WaterVelocity = WaterVelList[filecounter]
95.     print "using water velocity of", WaterVelocity, "m/s"
96.     filecounter += 1
97.     TrueOffsetList = ObsDirWavT*WaterVelocity #np.asarray((ObsDirWavT*WaterVelocity),dtype='fl
    oat64')
98.     DeltaOffsetList = TrueOffsetList - CurrentOffsetList
99.     #print DeltaOffsetList
100.
101.    #Determine the sailing direction, used during coordinate transformation
102.    if SourceX_list[-1] - SourceX_list[0] > 0:
103.        SailingDirection = 'E'
104.    else:
105.        SailingDirection = 'W'
```

```
106.
107.     Count = 0
108.     ProcessedShots = [ShotList[0]] #list to store shot numbers that have been encountered
109.     ShotBreaks = [[ShotList[0],0]] #list to store indexs grouping common shot gathers
110.     for i in range(NumLines):
111.         if not ShotList[i] in ProcessedShots:
112.             ShotBreaks[Count].append(i)
113.             ShotBreaks.append([ShotList[i],i])
114.             ProcessedShots.append(ShotList[i])
115.             Count += 1
116.     ShotBreaks[-1].append(i+1)
117.     numProcessedShots = len(ProcessedShots)
118.     print numProcessedShots, "were sorted"
119.     #print ShotBreaks
120.
121.     ### Match to seatrack navigation data ########################################
122.     NavMatchStartTime = time.time()
123.     NavFileList = glob.glob(os.path.join(NavDir, "*.txt"))
124.     foundNavMatch = False
125.     NavData = []
126.     for NavFile in NavFileList:
127.         index = NavFile.lower().find('line')
128.         LineNameStub = NavFile[index:index+6]
129.         if LineNameStub in infile:
130.             NavData = pandas.read_csv(NavFile,sep='\t')
131.
132.         if foundNavMatch:
133.             break
134.
135.     numNavRecords = len(NavData.hour)
136.
137.     PRT_paravane_E = np.zeros(numProcessedShots)
138.     PRT_paravane_N = np.zeros(numProcessedShots)
139.     STB_paravane_E = np.zeros(numProcessedShots)
140.     STB_paravane_N = np.zeros(numProcessedShots)
141.     AFT_E = np.zeros(numProcessedShots)
142.     AFT_N = np.zeros(numProcessedShots)
143.
144.     PrevMatchIndex = 0
145.     CompleteSuccess = True
146.     for i in range(numProcessedShots):
147.         ShotStartIndex = ShotBreaks[i][1]
148.         ShotEndIndex = ShotBreaks[i][2]
149.         ShotHour = HourList[ShotStartIndex]
150.         ShotMin = MinList[ShotStartIndex]
151.         ShotSec = SecList[ShotStartIndex]
152.         #print ShotHour,ShotMin, ShotSec
153.         NavMatch = False
154.         for j in range(PrevMatchIndex, numNavRecords):
155.             #print NavData.hour[j], NavData.minute[j], NavData.second[j]
156.             if int(NavData.hour[j]) == ShotHour:
157.                 if int(NavData.minute[j]) == ShotMin:
158.                     if np.fabs(NavData.second[j] - ShotSec) <= 1.0 :
159.                         #print "we have a navigation match for shot", ShotBreaks[i][0]
160.                         NavMatch = True
161.                         PRT_paravane_E[i] = NavData.PRT_E[j]
162.                         PRT_paravane_N[i] = NavData.PRT_N[j]
163.                         STB_paravane_E[i] = NavData.STB_E[j]
164.                         STB_paravane_N[i] = NavData.STB_N[j]
165.                         AFT_E[i] = NavData.AFT_E[j]
166.                         AFT_N[i] = NavData.AFT_N[j]
167.                         PrevMatchIndex = j
168.                         break
169.         if not NavMatch:
```

```
170.            print "no navigation matched with shot",ShotBreaks[i][0]
171.            CompleteSuccess = False
172.    if CompleteSuccess:
173.        print "All shots successfully matched with SeaTrack navigation data; paravanes have be
    en located. Operation took", round(time.time()-NavMatchStartTime,2), "seconds"
174.    else:
175.        print "Bugger, some shots could not be matched with SeaTrack navigation"
176.
177.    #Transform coordinate system
178.    TotalMisfitList = np.zeros(numProcessedShots)
179.    OriginXCoordList = np.zeros(numProcessedShots)
180.    OriginYCoordList = np.zeros(numProcessedShots)
181.    TransformAngleList = np.zeros(numProcessedShots)
182.
183.    for i in range(numProcessedShots):
184.        ShotStartIndex = ShotBreaks[i][1]
185.        ShotEndIndex = ShotBreaks[i][2]
186.        TotalMisfitList[i] = sum(np.abs(DeltaOffsetList[ShotStartIndex:ShotEndIndex]))
187.        OriginXtemp = 0
188.        OriginYtemp = 0
189.        HeadingXtemp = []
190.        HeadingYtemp = []
191.        if SetOriginAtShot:
192.            OriginXCoordList[i]=SourceX_list[ShotBreaks[i][1]]
193.            OriginYCoordList[i]=SourceY_list[ShotBreaks[i][1]]
194.        else:
195.            OriginXtemp = 0
196.            OriginYtemp = 0
197.            NumFail = 0
198.            for originCHAN in CalculateOriginFromChan:
199.                #print originCHAN
200.                try:
201.                    OriginChanIndex = np.where(ChanList[ShotStartIndex:ShotEndIndex]==originCH
    AN)[0][0]
202.                    OriginXtemp += ReceiverX_list[ShotStartIndex+OriginChanIndex]
203.                    OriginYtemp += ReceiverY_list[ShotStartIndex+OriginChanIndex]
204.                except Exception:
205.                    print "channel", originCHAN, "not found"
206.                    NumFail += 1
207.            if NumFail < len(CalculateOriginFromChan):
208.                OriginXCoordList[i]=OriginXtemp/(len(CalculateOriginFromChan) - NumFail)
209.                OriginYCoordList[i]=OriginYtemp/(len(CalculateOriginFromChan) - NumFail)
210.            else:
211.                print "no origin channels found, using shot as origin instead"
212.                OriginXCoordList[i]=SourceX_list[ShotBreaks[i][1]]
213.                OriginYCoordList[i]=SourceY_list[ShotBreaks[i][1]]
214.
215.        if UseShotsForHeading:
216.            if i - (CalculateHeadingFromN_Shots-1)/2 < 0 or i + (CalculateHeadingFromN_Shots-
    1)/2 > numProcessedShots -1:
217.                for headingCHAN in CalculateHeadingFromCHAN:
218.                    try:
219.                        HeadingChanIndex = np.where(ChanList[ShotStartIndex:ShotEndIndex]==hea
    dingCHAN)[0][0]
220.                        HeadingXtemp.append(ReceiverX_list[ShotStartIndex+HeadingChanIndex])
221.                        HeadingYtemp.append(ReceiverY_list[ShotStartIndex+HeadingChanIndex])
222.                    except Exception:
223.                        pass
224.            else:
225.                for k in range(-1*(CalculateHeadingFromN_Shots-
    1)/2,(CalculateHeadingFromN_Shots-1)/2+1):
226.                    #print k
227.                    HeadingXtemp.append(SourceX_list[ShotBreaks[i+k][1]])
228.                    HeadingYtemp.append(SourceY_list[ShotBreaks[i+k][1]])
```

```
        else:
            for headingCHAN in CalculateHeadingFromCHAN:
                try:
                    HeadingChanIndex = np.where(ChanList[ShotStartIndex:ShotEndIndex]==heading
    CHAN)[0][0]
                    HeadingXtemp.append(ReceiverX_list[ShotStartIndex+HeadingChanIndex])
                    HeadingYtemp.append(ReceiverY_list[ShotStartIndex+HeadingChanIndex])
                except Exception:
                    pass

        if not HeadingXtemp:
            HeadingXtemp = []
            HeadingYtemp = []
            for k in range(-1*(CalculateHeadingFromN_Shots-1)/2,(CalculateHeadingFromN_Shots-
    1)/2+1):
                #print k
                HeadingXtemp.append(SourceX_list[ShotBreaks[i+k][1]])
                HeadingYtemp.append(SourceY_list[ShotBreaks[i+k][1]])

        LineFit = np.polyfit(HeadingXtemp, HeadingYtemp, 1)
        if SailingDirection == 'E':
            angle = np.arctan2(LineFit[0],1) + 3*np.pi/2
        elif SailingDirection == 'W':
            angle = np.arctan2(LineFit[0],1) + np.pi/2
        TransformAngleList[i] = angle
        #print LineFit[0], np.rad2deg(angle)

    # Transform To New Coordinate System
    ReceiverXstar_list = np.zeros(NumLines)
    ReceiverYstar_list = np.zeros(NumLines)
    SourceXstar_list = np.zeros(NumLines)
    SourceYstar_list = np.zeros(NumLines)
    PRT_paravane_Estar = np.zeros(numProcessedShots)
    PRT_paravane_Nstar = np.zeros(numProcessedShots)
    STB_paravane_Estar = np.zeros(numProcessedShots)
    STB_paravane_Nstar = np.zeros(numProcessedShots)
    STB_paravane_Estar = np.zeros(numProcessedShots)
    STB_paravane_Nstar = np.zeros(numProcessedShots)
    AFT_Estar = np.zeros(numProcessedShots)
    AFT_Nstar = np.zeros(numProcessedShots)

    #delta = np.deg2rad(1)
    for i in range(numProcessedShots):
        #print 'transforming shot', ShotBreaks[i][0]
        STB_Xshifted = STB_paravane_E[i] - OriginXCoordList[i]
        STB_Yshifted = STB_paravane_N[i] - OriginYCoordList[i]
        PRT_Xshifted = PRT_paravane_E[i] - OriginXCoordList[i]
        PRT_Yshifted = PRT_paravane_N[i] - OriginYCoordList[i]
        AFT_Xshifted = AFT_E[i] - OriginXCoordList[i]
        AFT_Yshifted = AFT_N[i] - OriginYCoordList[i]
        S_Xshifted = SourceX_list[ShotBreaks[i][1]:ShotBreaks[i][2]] - OriginXCoordList[i]
        S_Yshifted = SourceY_list[ShotBreaks[i][1]:ShotBreaks[i][2]] - OriginYCoordList[i]
        Xshifted = ReceiverX_list[ShotBreaks[i][1]:ShotBreaks[i][2]] - OriginXCoordList[i]
        Yshifted = ReceiverY_list[ShotBreaks[i][1]:ShotBreaks[i][2]] - OriginYCoordList[i]
        AFT_Estar[i] = 1.0*AFT_Xshifted*np.cos(TransformAngleList[i])+1.0*AFT_Yshifted*np.sin(
    TransformAngleList[i])
        AFT_Nstar[i] = -
    1.0*AFT_Xshifted*np.sin(TransformAngleList[i])+1.0*AFT_Yshifted*np.cos(TransformAngleList[i])

        PRT_paravane_Estar[i] = 1.0*PRT_Xshifted*np.cos(TransformAngleList[i])+1.0*PRT_Yshifte
    d*np.sin(TransformAngleList[i])
        PRT_paravane_Nstar[i] = -
    1.0*PRT_Xshifted*np.sin(TransformAngleList[i])+1.0*PRT_Yshifted*np.cos(TransformAngleList[i])
```

```
        STB_paravane_Estar[i] = 1.0*STB_Xshifted*np.cos(TransformAngleList[i])+1.0*STB_Yshifted*np.sin(TransformAngleList[i])
        STB_paravane_Nstar[i] = -1.0*STB_Xshifted*np.sin(TransformAngleList[i])+1.0*STB_Yshifted*np.cos(TransformAngleList[i])

        SourceXstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]] = 1.0*S_Xshifted*np.cos(TransformAngleList[i])+1.0*S_Yshifted*np.sin(TransformAngleList[i])
        SourceYstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]] = -1.0*S_Xshifted*np.sin(TransformAngleList[i])+1.0*S_Yshifted*np.cos(TransformAngleList[i])
        ReceiverXstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]] = 1.0*Xshifted*np.cos(TransformAngleList[i])+1.0*Yshifted*np.sin(TransformAngleList[i])
        ReceiverYstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]] = -1.0*Xshifted*np.sin(TransformAngleList[i])+1.0*Yshifted*np.cos(TransformAngleList[i])

    #Save Initial Values
    InitialRecXstar = 1.0*ReceiverXstar_list
    InitialRecYstar = 1.0*ReceiverYstar_list
    InitialDeltaOffsets = 1.0*DeltaOffsetList
    InitialTotalMisfit = 1.0*TotalMisfitList

    #Add offsets from paravane GPS to cross cable
    STB_paravane_Nstar += ParavaneYshift_STB
    PRT_paravane_Nstar += ParavaneYshift_PRT
    if np.mean(STB_paravane_Estar) > np.mean(PRT_paravane_Estar): #starboard +x in transformed coords
        STB_paravane_Estar -= ParavaneXshift
        PRT_paravane_Estar += ParavaneXshift
    else:
        raise ValueError('Transformation of coordinates failed, starboard not pos x')
    if not np.mean(STB_paravane_Estar) > np.mean(PRT_paravane_Estar): #starboard +x in transformed coords
        raise ValueError('Transformation of coordinates failed, starboard not pos x')

        #Find cross cable position by collapsing delta offset shifted receivers by known along streamer positions
    def FindPoly(i,CurrentShot,CurrentChannels,NumChanPerStreamer,StreamerOffsetList,DeltaOffsets,CurrentReceiverXstar,CurrentReceiverYstar,FitThroughParavanes,ParavaneXs,ParavaneYs,GlobNumStreamers):
        #print ParavaneXs,ParavaneYs
        #print 'working on shot', CurrentShot

        #offset by channel position along streamer to find position of cross cable
        numChan = len(CurrentChannels)
        ChanOffsets = np.zeros(numChan)
        currentStreamer = 1
        Count = 0
        StreamerBreaks = [[currentStreamer,0]]
        for k in range(numChan):
            channel = CurrentChannels[k]
            streamer = np.ceil(float(channel)/NumChanPerStreamer)
            #print channel,streamer
            if channel > NumChanPerStreamer:
                PosOnStreamer = np.remainder(channel-1, NumChanPerStreamer) + 1
            else:
                PosOnStreamer = channel
            channel_offset = StreamerOffsetList[PosOnStreamer-1]
            #print channel, PosOnStreamer, channel_offset
            ChanOffsets[k] = channel_offset
            if not streamer == currentStreamer:
                #print 'analysing streamer', streamer, 'with first channel', channel
                currentStreamer = streamer
                StreamerBreaks[Count].append(k-1)
                StreamerBreaks.append([currentStreamer,k])
                Count += 1
```

```
        StreamerBreaks[-1].append(k)

        #print StreamerBreaks
        #ShiftToCrossCable = DeltaOffsetShifted_ReceiverYstar + ChanOffsets
        Displacement = np.abs(ChanOffsets - np.round(DeltaOffsets,decimals=1))
        #print Displacement
        NumStreamers = len(StreamerBreaks)

        def cost_fn(coord,args):
            x,y,r = args
            # notice that we're returning a vector of dimension 3
            return ((coord[0]-x)**2) + ((coord[1] - y)**2) - (r**2)

        def cost_fn_minimize(coord,args):
            x,y,r = args
            vec = ((coord[0]-x)**2) + ((coord[1] - y)**2) - (r**2)
            return np.sum(vec**2)

        StreamerJuncX = np.zeros(NumStreamers)
        StreamerJuncY = np.zeros(NumStreamers)
        InitStreamerJuncX = np.zeros(NumStreamers)
        InitStreamerJuncY = np.zeros(NumStreamers)
        StreamerSigmaRes = np.zeros(NumStreamers)
        UpdateReceiverXstar = np.zeros(numChan)
        UpdateReceiverYstar = np.zeros(numChan)
        for j in range(NumStreamers):

            x = CurrentReceiverXstar[StreamerBreaks[j][1]:StreamerBreaks[j][2]]
            y = CurrentReceiverYstar[StreamerBreaks[j][1]:StreamerBreaks[j][2]]
            r = Displacement[StreamerBreaks[j][1]:StreamerBreaks[j][2]]

            x0 = np.mean(x) #+ np.random.uniform(low=-100, high=100)
            y0 = np.mean(y) + np.mean(r)#np.mean(y - DeltaOffsets[StreamerBreaks[j][1]:Streame
    rBreaks[j][2]] + ChanOffsets[StreamerBreaks[j][1]:StreamerBreaks[j][2]])
            initial_guess = np.array([x0,y0])
            residuals = cost_fn(initial_guess,args = [x,y,r])
            numNegResiduals = (residuals < 0).sum()
            PerRecResidual = np.abs(np.median(residuals))#np.sum(np.fabs(residuals))/numRecCur
    rentStreamer

            MaxAttempts = 5000
            AttemptCount = 0
            bestResidual = 1.0*PerRecResidual
            Best_x0 = x0
            Best_y0 = y0
            progress = True
            iterSinceUpdate = 0
            inside = True
            switchCount = 0
            while AttemptCount <  MaxAttempts and bestResidual > 0.01 and progress: #retry fit
    ting with different initial condition
                #factor = np.floor(AttemptCount)/(MaxAttempts/10) + 1
                #print 0.1*bestResidual
                new_x0 = Best_x0 + np.random.uniform(low=-
    1,high=1)#0.02*bestResidual**0.5,high=0.02*bestResidual**0.5)#5/factor, high=5/factor)
                new_y0 = Best_y0 + np.random.uniform(low=-
    5,high=5)#0.1*bestResidual**0.5,high=0.1*bestResidual**0.5)#20/factor, high=20/factor)
                if numNegResiduals >= 4:
                    new_y0 += 0.99*bestResidual**0.5#/(switchCount**0.9+1)
                    #print 10*bestResidual**0.5/(switchCount**0.9+1)
                    if not inside:
                        inside = True
                        switchCount += 1
                    #print "inside circles"
```

```
                else:
                    new_y0 -= 0.99*bestResidual**0.5#/(switchCount**0.9+1)
                    #print 10*bestResidual**0.5/(switchCount**0.9+1)
                    if inside:
                        inside = False
                        switchCount += 1
                    #print "outside circles"
                initial_guess = np.array([new_x0,new_y0])
                residuals = cost_fn(initial_guess,args = [x,y,r])
                numNegResiduals = (residuals < 0).sum()
                PerRecResidual = np.abs(np.median(residuals))#np.sum(np.fabs(residuals))/numRe
    cCurrentStreamer
                iterSinceUpdate += 1
                if PerRecResidual < bestResidual:
                    bestResidual = PerRecResidual
                    Best_x0 = new_x0
                    Best_y0 = new_y0
                    iterSinceUpdate = 0
                if bestResidual > 1.0:
                    if iterSinceUpdate > 1000:
                        progress = False
                else:
                    if iterSinceUpdate > 50:
                        progress = False
                    #print "terminating because iterations lack progress"
                AttemptCount += 1
            #if AttemptCount > 0:
            #print i, "Residual improved from", initialResidual, 'to', bestResidual, 'in', Att
    emptCount, 'iterations and ', switchCount, 'inside outside circles switches'#20/factor, 5/fact
    or
            #print x0, Best_x0, y0, Best_y0
            StreamerJuncX[j] = Best_x0
            StreamerJuncY[j] = Best_y0
            InitStreamerJuncX[j] = Best_x0
            InitStreamerJuncY[j] = Best_y0
            StreamerSigmaRes[j] = bestResidual

        def curve_length(Xpos,x0,S, length,quadtol):
            return quad(S, Xpos, x0,epsabs=quadtol,epsrel=quadtol)[0] - length

        def solve_t(curve_diff, length, x0,opttol=1.e-5,quadtol=1e-6):
            return fsolve(curve_length, 0.0, (x0, curve_diff, length,quadtol),xtol = opttol)[0
    ]

        StreamerCablePositions = [24.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,
    12.5,12.5] #from starbord to port
        CableLen2PrtPara = 25.25
        PortParavaneDistAlongX_Cable = sum(StreamerCablePositions)+CableLen2PrtPara
        weights = np.ones(NumStreamers)
        biasFactor = 0.75
        for ww in range(int(np.ceil(float(NumStreamers)/2))):
            weights[ww] += ww*biasFactor
            weights[-(ww+1)] += ww*biasFactor
        #print weights
        paravaneweight = 0.75*np.max(weights)

        def func1(x, a0, a1, a2, a3, a4):
            return a0*x**4 + a1*x**3 + a2*x**2 + a3*x**1 + a4
        def func2(x, a0, a1, a2, a3, a4, a5, a6):
            return a0*x**6 + a1*x**5 + a2*x**4 + a3*x**3 + a4*x**2 + a5*x**1 + a6
        def ArcLength(x,p):
            gradientFn = np.polyder(p)
            return np.sqrt(1+(np.polyval(gradientFn, x)**2))
        def ResidualFn(p,x,y,fitweights,ParavanesXpos,quadtol=1e-6):
```

```
            integral = quad( ArcLength, ParavanesXpos[0], ParavanesXpos[1], args=(p),epsabs=quadtol,epsrel=quadtol)[0]
            penalization = abs(PortParavaneDistAlongX_Cable - integral)
            return 2*fitweights*(y - func1(x, p[0],p[1],p[2],p[3],p[4]))+penalization
        def ResidualFn2(p,x,y,fitweights,ParavanesXpos,quadtol=1e-6):
            integral = quad( ArcLength, ParavanesXpos[0], ParavanesXpos[1], args=(p),epsabs=quadtol,epsrel=quadtol)[0]
            penalization = abs(PortParavaneDistAlongX_Cable - integral)
            return 2*fitweights*(y - func2(x, p[0],p[1],p[2],p[3],p[4],p[5],p[6]))+penalization

        if FitThroughParavanes:

            ForPolyFitY = np.append(StreamerJuncY,[ParavaneYs[0],ParavaneYs[1]])
            ForPolyFitX = np.append(StreamerJuncX,[ParavaneXs[0] ,ParavaneXs[1]])

            PolyFitWeights = np.append(weights,[paravaneweight,paravaneweight])
            #print PolyFitWeights
            if NumStreamers >= 3:
                result = leastsq(func=ResidualFn, x0=(1.,1.,1.,1.,1.), args=(ForPolyFitX,ForPolyFitY,PolyFitWeights,ParavaneXs),full_output=1)
                poly = result[0]
                poly_residuals = result[2]['fvec']
            else:
                poly,poly_residuals, rank, singular_values, rcond = np.polyfit(ForPolyFitX,ForPolyFitY,4,full=True,w=PolyFitWeights)
            PolyMisfit = np.sum(np.fabs(poly_residuals))/NumStreamers
            #print PolyMisfit
            if PolyMisfit > 2:#1.0:
                if NumStreamers >= 5:
                    new_result = leastsq(func=ResidualFn2, x0=(1.,1.,1.,1.,1.,1.,1.), args=(ForPolyFitX,ForPolyFitY,PolyFitWeights,ParavaneXs),full_output=1)
                    new_poly = new_result[0]
                    new_poly_residuals = new_result[2]['fvec']
                else:
                    new_poly,new_poly_residuals, rank, singular_values, rcond = np.polyfit(ForPolyFitX,ForPolyFitY,6,full=True,w=PolyFitWeights)
                NewPolyMisfit = np.sum(np.fabs(new_poly_residuals))/NumStreamers
                if NewPolyMisfit*9 < PolyMisfit*5:
                    poly = new_poly
                    #print i, "polynomial misfit improved from", PolyMisfit, "to", NewPolyMisfit

        else:
            poly,residuals, rank, singular_values, rcond = np.polyfit(StreamerJuncX,StreamerJuncY,4,full=True,w=weights)
        #print "residual from polynomial fitting is", residuals[0]#1 - residuals2
        #print poly
        def DiffArcLength2(x):
            gradientFn = np.polyder(poly)
            return np.sqrt(1+(np.polyval(gradientFn, x)**2))

        ApparentXcableLen = quad(ArcLength, ParavaneXs[0], ParavaneXs[1], args=(poly),epsabs=1e-6,epsrel=1e-6)[0]
        CableLenDelta = ApparentXcableLen - PortParavaneDistAlongX_Cable
        #print ApparentXcableLen, PortParavaneDistAlongX_Cable, CableLenDelta
        StreamerCablePositions[0] += CableLenDelta/2
        CableLen2PrtPara += CableLenDelta/2
        #print "lead in lengths STB/PRT are", StreamerCablePositions[0], CableLen2PrtPara

        updateStreamerXpos = True
        prevFixedXpoint = ParavaneXs[1]
        prevDistAlongXcable = 0
        for k in range(NumStreamers):
```

```
            currentStreamerNum = int(StreamerBreaks[k][0])
            distAlongXcable = sum(StreamerCablePositions[0:currentStreamerNum])#sum(StreamerCablePositions[0:k+1])
            StreamerPos = solve_t(DiffArcLength2, distAlongXcable-prevDistAlongXcable, prevFixedXpoint)
            if updateStreamerXpos:
                StreamerJuncX[k] = StreamerPos
            prevFixedXpoint = StreamerPos
            prevDistAlongXcable = distAlongXcable
        PortParavaneDistFromLastStreamer = CableLen2PrtPara
        PortParavaneXpos = solve_t(DiffArcLength2, PortParavaneDistFromLastStreamer, prevFixedXpoint)
        PortParavaneYpos = np.polyval(poly,PortParavaneXpos)
        PrtParaDeltaX = ParavaneXs[0] - PortParavaneXpos
        PrtParaDeltaY = ParavaneYs[0] - PortParavaneYpos
        PrtParaError = np.sqrt(PrtParaDeltaX**2+PrtParaDeltaY**2)

        StreamerJuncY = np.polyval(poly,StreamerJuncX)

        for hooray in range(NumStreamers):
            UpdateReceiverXstar[StreamerBreaks[hooray][1]:StreamerBreaks[hooray][2]+1] = StreamerJuncX[hooray]

        UpdateReceiverYstar = np.polyval(poly,UpdateReceiverXstar) - ChanOffsets

        ShotIndex = i
        if NumStreamers < GlobNumStreamers:
            #print NumStreamers
            appendnan = np.full(GlobNumStreamers - NumStreamers, np.nan)
            #print len( appendnan)
            InitStreamerJuncX = np.append(InitStreamerJuncX, appendnan)
            #print len( StreamerJuncX )
            InitStreamerJuncY = np.append(InitStreamerJuncY, appendnan)
            #print len( StreamerJuncY )

        return UpdateReceiverXstar, UpdateReceiverYstar, InitStreamerJuncX, InitStreamerJuncY, poly, Displacement, PrtParaError, ShotIndex

    #setup parallel processing job
    ppservers = ()
    if len(sys.argv) > 1:
        ncpus = int(sys.argv[1])
        # Creates jobserver with ncpus workers
        job_server = pp.Server(ncpus, ppservers=ppservers)
    else:
        # Creates jobserver with automatically detected number of workers
        job_server = pp.Server(ppservers=ppservers)
    print "Starting parallel processing job with", job_server.get_ncpus(), "workers"

    jobs = []
    for i in range(numProcessedShots):
        CurrentShot = ShotBreaks[i][0]
        CurrentChannels = ChanList[ShotBreaks[i][1]:ShotBreaks[i][2]]
        DeltaOffsets = DeltaOffsetList[ShotBreaks[i][1]:ShotBreaks[i][2]]
        CurrentReceiverXstar = ReceiverXstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]]
        CurrentReceiverYstar = ReceiverYstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]]
        ParavaneXs = [PRT_paravane_Estar[i],STB_paravane_Estar[i]]
        ParavaneYs = [PRT_paravane_Nstar[i],STB_paravane_Nstar[i]]
        jobs.append(job_server.submit(FindPoly, (i,CurrentShot,CurrentChannels,NumChanPerStreamer,StreamerOffsetList,DeltaOffsets,CurrentReceiverXstar,CurrentReceiverYstar,FitThroughParavanes,ParavaneXs,ParavaneYs,NumStreamers), (), ("import numpy as np","from scipy.optimize import leastsq, fsolve", "from scipy.integrate import quad")))

    # Retrieve results of all submited jobs
```

```
    CrossCableXList = np.zeros(numProcessedShots*NumStreamers)
    CrossCableYList = np.zeros(numProcessedShots*NumStreamers)
    PrtParaErrorList = np.zeros(numProcessedShots)
    DisplacementList = np.zeros(NumLines)
    PolyPlotX = np.zeros(100*numProcessedShots)
    for bbb in range(numProcessedShots):
        PolyPlotX[bbb*100:(bbb+1)*100] = np.linspace(PRT_paravane_Estar[bbb], STB_paravane_Est
ar[bbb], num=100)
    PolyPlotY = np.zeros(100*numProcessedShots)
    for job in jobs:
        UpdateReceiverXstar, UpdateReceiverYstar, StreamerJuncX, StreamerJuncY, poly, Displace
ment, PrtParaError, ShotIndex = job()
        CrossCableXList[int(ShotIndex*NumStreamers):int(ShotIndex*NumStreamers)+NumStreamers]
= StreamerJuncX
        CrossCableYList[int(ShotIndex*NumStreamers):int(ShotIndex*NumStreamers)+NumStreamers]
= StreamerJuncY
        DisplacementList[ShotBreaks[ShotIndex][1]:ShotBreaks[ShotIndex][2]] = Displacement
        ReceiverXstar_list[ShotBreaks[ShotIndex][1]:ShotBreaks[ShotIndex][2]] = UpdateReceiver
Xstar
        ReceiverYstar_list[ShotBreaks[ShotIndex][1]:ShotBreaks[ShotIndex][2]] = UpdateReceiver
Ystar
        PolyPlotY[100*ShotIndex:100*(ShotIndex+1)] = np.polyval(poly,PolyPlotX[100*ShotIndex:1
00*(ShotIndex+1)])
        PrtParaErrorList[ShotIndex] = PrtParaError

    #update the DeltaOffsets and calculate final misfits
    CurrentOffsetList = np.sqrt((ReceiverXstar_list - SourceXstar_list)**2 + (ReceiverYstar_li
st - SourceYstar_list)**2)
    DeltaOffsetList = TrueOffsetList - CurrentOffsetList

    TotalMisfitList = np.zeros(numProcessedShots)
    for i in range(numProcessedShots):
        TotalMisfitList[i] = sum(np.abs(DeltaOffsetList[ShotBreaks[i][1]:ShotBreaks[i][2]]))

    #Transform back to original coordinates
    finReceiverX_list = np.zeros(NumLines)
    finReceiverY_list = np.zeros(NumLines)
    transCrossCableXList = np.zeros(numProcessedShots*NumStreamers)
    transCrossCableYList = np.zeros(numProcessedShots*NumStreamers)
    transPolyPlotX = np.zeros(100*numProcessedShots)
    transPolyPlotY = np.zeros(100*numProcessedShots)
    for i in range(numProcessedShots):
        finReceiverX_list[ShotBreaks[i][1]:ShotBreaks[i][2]] = ReceiverXstar_list[ShotBreaks[i
][1]:ShotBreaks[i][2]]*np.cos(TransformAngleList[i])-
ReceiverYstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]]*np.sin(TransformAngleList[i]) + OriginXC
oordList[i]
        finReceiverY_list[ShotBreaks[i][1]:ShotBreaks[i][2]] = ReceiverXstar_list[ShotBreaks[i
][1]:ShotBreaks[i][2]]*np.sin(TransformAngleList[i])+ReceiverYstar_list[ShotBreaks[i][1]:ShotB
reaks[i][2]]*np.cos(TransformAngleList[i]) + OriginYCoordList[i]
        transCrossCableXList[int(i*NumStreamers):int(i*NumStreamers)+NumStreamers] = CrossCabl
eXList[int(i*NumStreamers):int(i*NumStreamers)+NumStreamers]*np.cos(TransformAngleList[i])-
CrossCableYList[int(i*NumStreamers):int(i*NumStreamers)+NumStreamers]*np.sin(TransformAngleLis
t[i]) + OriginXCoordList[i]
        transCrossCableYList[int(i*NumStreamers):int(i*NumStreamers)+NumStreamers] = CrossCabl
eXList[int(i*NumStreamers):int(i*NumStreamers)+NumStreamers]*np.sin(TransformAngleList[i])+Cro
ssCableYList[int(i*NumStreamers):int(i*NumStreamers)+NumStreamers]*np.cos(TransformAngleList[i
]) + OriginYCoordList[i]
        transPolyPlotX[100*i:100*(i+1)] = PolyPlotX[100*i:100*(i+1)]*np.cos(TransformAngleList
[i])-PolyPlotY[100*i:100*(i+1)]*np.sin(TransformAngleList[i]) + OriginXCoordList[i]
        transPolyPlotY[100*i:100*(i+1)] = PolyPlotX[100*i:100*(i+1)]*np.sin(TransformAngleList
[i])+PolyPlotY[100*i:100*(i+1)]*np.cos(TransformAngleList[i]) + OriginYCoordList[i]
        PRT_paravane_E[i] = PRT_paravane_Estar[i]*np.cos(TransformAngleList[i])-
PRT_paravane_Nstar[i]*np.sin(TransformAngleList[i]) + OriginXCoordList[i]
```

```
606.         PRT_paravane_N[i] = PRT_paravane_Estar[i]*np.sin(TransformAngleList[i])+PRT_paravane_N
    star[i]*np.cos(TransformAngleList[i]) + OriginYCoordList[i]
607.         STB_paravane_E[i] = STB_paravane_Estar[i]*np.cos(TransformAngleList[i])-
    STB_paravane_Nstar[i]*np.sin(TransformAngleList[i]) + OriginXCoordList[i]
608.         STB_paravane_N[i] = STB_paravane_Estar[i]*np.sin(TransformAngleList[i])+STB_paravane_N
    star[i]*np.cos(TransformAngleList[i]) + OriginYCoordList[i]
609. 
610.     ### Save Results to tab delimited text file ###
611.     directory = os.path.join(os.getcwd(), OutputDir)
612.     if not os.path.exists(directory):
613.         os.makedirs(directory)
614.     outfilename = infile.split('\\')[-1].split('_')[0] + "_BestFitGeom.txt"
615.     headertxt = "SOURCE\t CHAN\t REC_X\t REC_Y\t"
616.     np.savetxt(os.path.join(directory,outfilename),np.c_[ShotList,ChanList,finReceiverX_list,f
    inReceiverY_list],fmt=['%d','%d','%.5f','%.5f'],delimiter='\t',header=headertxt,comments='',ne
    wline='\n')
617. 
618.     #Plotting
619.     #create directory to save QC plots as multipage PDF files
620.     directory = os.path.join(os.getcwd(), OutputPlotDir)
621.     if not os.path.exists(directory):
622.         os.makedirs(directory)
623.     outfilename = infile.split('\\')[-1].split('_')[0] + "_QCplots.pdf"
624.     with PdfPages(os.path.join(directory,outfilename)) as pdf:
625. 
626.         plt.figure()
627.         plt.title("Shot number vs Total Absolute Misfit")
628.         plt.plot(ProcessedShots,InitialTotalMisfit)
629.         plt.plot(ProcessedShots,TotalMisfitList)
630.         pdf.savefig()  # saves the current figure into a pdf page
631.         plt.close()
632. 
633.         plt.figure()
634.         AxLim = (0,3)
635.         plt.title("Port paravane position error after integration along fitted curve (m)")
636.         plt.hist(PrtParaErrorList,bins=100,color='b',range=AxLim,histtype='stepfilled',alpha=0
    .75,normed=True)
637.         plt.xlabel("position error (m)")
638.         plt.ylabel("Probability")
639.         pdf.savefig()  # saves the current figure into a pdf page
640.         plt.close()
641. 
642.         plt.figure()
643.         AxLim = (np.percentile(InitialDeltaOffsets,1),np.percentile(InitialDeltaOffsets,99))
644.         plt.title("Delta offset per receiver")
645.         plt.hist(InitialDeltaOffsets,bins=100,label='Initial Geometry',color='r',range=AxLim,h
    isttype='stepfilled',alpha=0.75,normed=True)
646.         plt.hist(DeltaOffsetList,bins=100,label='Final Geometry', color='b',range=AxLim,histty
    pe='stepfilled',alpha=0.5,normed=True)
647.         plt.xlabel("Delta Offset (m)")
648.         plt.ylabel("Probability")
649.         plt.legend(loc='upper left')
650.         pdf.savefig()  # saves the current figure into a pdf page
651.         plt.close()
652. 
653.         if PlotThresh:
654.             PlotShots = []
655.             for i in range(numProcessedShots):
656.                 if TotalMisfitList[i] > PlotThresh:
657.                     #print "shot", ProcessedShots[i], "exceeds misfit threshold and will be pl
    otted"
658.                     PlotShots.append(int(ProcessedShots[i]))
659.         else:
660.             PlotShots = np.arange(FirstShot,LastShot,PlotN)
```

```
            if MaxPlotShots:
                PlotShots = PlotShots[0:min(MaxPlotShots,len(PlotShots)-1)]

        #Initialise empty lists to hold values for shots that will be plotted
        InitX = []
        InitY = []
        InitC = []
        PlotXs = []
        PlotYs = []
        PlotCs = []
        CrossCableX = []
        CrossCableY = []
        SPX = []
        SPY = []
        PRT_X = []
        PRT_Y = []
        STB_X = []
        STB_Y = []
        BoatX = []
        BoatY = []
        DisplacementRadius = []
        ### Populate plot data lists ###
        for i in range(numProcessedShots):
            if ShotBreaks[i][0] in PlotShots:
                #print "plotting channel",ChanList[i], ", shot", ShotList[i]
                if PlotInTransformCoords:
                    InitX.extend(InitialRecXstar[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    InitY.extend(InitialRecYstar[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    InitC.extend(InitialDeltaOffsets[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PlotXs.extend(ReceiverXstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PlotYs.extend(ReceiverYstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PlotCs.extend(DeltaOffsetList[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    CrossCableX.extend(CrossCableXList[int(i*NumStreamers):int(i*NumStreamers)
    +NumStreamers])
                    CrossCableY.extend(CrossCableYList[int(i*NumStreamers):int(i*NumStreamers)
    +NumStreamers])
                    SPX.extend(SourceXstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    SPY.extend(SourceYstar_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PRT_X.append(PRT_paravane_Estar[i])
                    PRT_Y.append(PRT_paravane_Nstar[i])
                    STB_X.append(STB_paravane_Estar[i])
                    STB_Y.append(STB_paravane_Nstar[i])
                    BoatX.append(AFT_Estar[i])
                    BoatY.append(AFT_Nstar[i])
                    DisplacementRadius.extend(DisplacementList[ShotBreaks[i][1]:ShotBreaks[i][
    2]])
                else:
                    InitX.extend(ReceiverX_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    InitY.extend(ReceiverY_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    InitC.extend(InitialDeltaOffsets[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PlotXs.extend(finReceiverX_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PlotYs.extend(finReceiverY_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PlotCs.extend(DeltaOffsetList[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    CrossCableX.extend(transCrossCableXList[int(i*NumStreamers):int(i*NumStrea
    mers)+NumStreamers])
                    CrossCableY.extend(transCrossCableYList[int(i*NumStreamers):int(i*NumStrea
    mers)+NumStreamers])
                    SPX.extend(SourceX_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    SPY.extend(SourceY_list[ShotBreaks[i][1]:ShotBreaks[i][2]])
                    PRT_X.append(PRT_paravane_E[i])
                    PRT_Y.append(PRT_paravane_N[i])
                    STB_X.append(STB_paravane_E[i])
                    STB_Y.append(STB_paravane_N[i])
                    BoatX.append(AFT_E[i])
```

```
                    BoatY.append(AFT_N[i])
                    DisplacementRadius.extend(DisplacementList[ShotBreaks[i][1]:ShotBreaks[i][
2]])

        ### Make the plots ###
        fig = plt.figure(figsize=(13, 6), dpi=100, facecolor='w', edgecolor='k')
        ColorLim = ManualPlotColourLim
        if LockPlotAspectRatio:
            ax1 = fig.add_subplot(121, aspect='equal')
        else:
            ax1 = fig.add_subplot(121) #before geometry adjustment figure
        if AutoscalePlotColours:
            ColorLim = np.percentile(np.abs(InitC),90)

        ax1.scatter(CrossCableX,CrossCableY,s=25,c='b',marker='+',edgecolors=None) #plot strea
mer junctions
        ax1.scatter(SPX,SPY,s=250,c='r',marker='*',edgecolors=None) #plot shot points
        ax1.scatter(PRT_X,PRT_Y,s=100,c='g',marker='x',edgecolors=None) #plot port paravane po
sitions
        ax1.scatter(STB_X,STB_Y,s=100,c='g',marker='x',edgecolors=None) #plot starboard parava
ne positions
        ax1.scatter(BoatX,BoatY,s=100,c='m',marker='s',edgecolors=None) #plot aft of ship posi
tions
        initialGeom = ax1.scatter(InitX, InitY,s=50,c=InitC,vmin=-
1*ColorLim,vmax=ColorLim,edgecolors=None) #plot receivers
        if AutoscalePlotColours or LockPlotAspectRatio:
            plt.colorbar(initialGeom)

        if LockPlotAspectRatio:
            ax2 = fig.add_subplot(122, sharex=ax1, sharey=ax1, aspect='equal') #After geometry
 adjustment plot
        else:
            ax2 = fig.add_subplot(122, sharex=ax1, sharey=ax1)
        if AutoscalePlotColours:
            ColorLim = np.percentile(np.abs(PlotCs),90)
        ax2.scatter(CrossCableX,CrossCableY,s=25,c='b',marker='+',edgecolors=None)#plot stream
er junctions
        ax2.scatter(SPX,SPY,s=250,c='r',marker='*',edgecolors=None) #plot shot points
        ax2.scatter(PRT_X,PRT_Y,s=100,c='g',marker='x',edgecolors=None) #plot port paravane po
sitions
        ax2.scatter(STB_X,STB_Y,s=100,c='g',marker='x',edgecolors=None) #plot starboard parava
ne positions
        ax2.scatter(BoatX,BoatY,s=100,c='m',marker='s',edgecolors=None) #plot aft of ship posi
tions
        finalGeom = ax2.scatter(PlotXs,PlotYs,s=50,c=PlotCs,vmin=-
1*ColorLim,vmax=ColorLim,edgecolors=None) #plot receivers
        plt.colorbar(finalGeom)

        #plot fitted polylines that model the cross cable position
        for i in range(numProcessedShots):
            if ShotBreaks[i][0] in PlotShots:
                if PlotInTransformCoords:
                    ax1.plot(PolyPlotX[100*i:100*(i+1)],PolyPlotY[100*i:100*(i+1)], "--r")
                    ax2.plot(PolyPlotX[100*i:100*(i+1)],PolyPlotY[100*i:100*(i+1)], "--r")
                else:
                    ax1.plot(transPolyPlotX[100*i:100*(i+1)],transPolyPlotY[100*i:100*(i+1)],
"--r")
                    ax2.plot(transPolyPlotX[100*i:100*(i+1)],transPolyPlotY[100*i:100*(i+1)],
"--r")

        #plot circles with a radius of delta offset at each of the initial receiver locations

        if PlotDeltaOffsetCircles:
            xy = np.column_stack((InitX,InitY))
```

```
            sizes = DisplacementRadius
            patches = [plt.Circle(center, size) for center, size in zip(xy, sizes)]
            coll = matplotlib.collections.PatchCollection(patches, facecolors='none', edgecolo
    rs='k',alpha=0.2)
            ax1.add_collection(coll)

        elapsed_time = time.time()-start_time
        if elapsed_time < 60:
            print "Execution took", round((time.time()-start_time),2), "seconds"
        else:
            print "Execution took", round((time.time()-start_time)/60,2), "minutes"

        plt.tight_layout()
        pdf.savefig()  # saves the current figure into a pdf page
        plt.close()
        #plt.show()
```

Appendix 2 – Python script used to generate fold maps for different CDP bin sizes

```
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 30 13:40:30 2016

Generate fold maps to visualise the stacking fold of 3D seismic surveys

@author: Rowan Romeyn
"""
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt

FileName = "FinalGeometrySouRecHeaders.txt"
GridOriginX = 422075.82
GridOriginY = 8712924.56
GridRotation = 2.13 #degrees clockwise
NumBinsX = 1900#1820
NumBinsY = 288#300#288
BinSizeX = 6.25
BinSizeY = 1.0*BinSizeX #can also give directly as a number
RestrictDeltaPickRange = None#(-5,5)   ##traces outside range discarded for CDP fold map
RestrictOffsetRange = (140,300)#None   ##traces outside range discarded for CDP fold map
SurveyInlineLength = 11830 #length of inlines in meters
SurveyXlineLength = 1850 #length of crosslines in meters
UseLineLength = True #num bins will be calculated based on line lengths and bin sizes, NumBins
X & NumBinsY ignored

#Create the grid
def DoRotation(xvec, yvec, RotRad=0, origin=(0,0)):
    """Generate a meshgrid and rotate it by RotRad radians."""

    # Clockwise, 2D rotation matrix
    RotMatrix = np.array([[np.cos(RotRad),  np.sin(RotRad)],
                          [-np.sin(RotRad), np.cos(RotRad)]])

    xprime = xvec - (origin[0] - BinSizeX /2)
    yprime = yvec - (origin[1] - BinSizeY /2)
    array = np.column_stack((xprime,yprime))
    RotatedPoints = np.dot(array,RotMatrix)
    rot_xvec = RotatedPoints[:,0] + (origin[0] - BinSizeX /2)
    rot_yvec = RotatedPoints[:,1] + (origin[1] - BinSizeY /2)

    return rot_xvec, rot_yvec

if UseLineLength:
    Xvec = np.arange(GridOriginX,GridOriginX+SurveyInlineLength+0.5,BinSizeX)
    Yvec = np.arange(GridOriginY,GridOriginY+SurveyXlineLength+0.5,BinSizeY)
    print "grid has ", len(Xvec), "cells in inline direction and ", len(Yvec), "crossline cell
s"
else:
    Xvec = np.zeros(NumBinsX+1)
    Xvec[0] = GridOriginX
    for i in range(1,NumBinsX+1):
        Xvec[i] = Xvec[i-1] + BinSizeX

    Yvec = np.zeros(NumBinsY+1)
    Yvec[0] = GridOriginY
    for i in range(1,NumBinsY+1):
```

```
        Yvec[i] = Yvec[i-1] + BinSizeY

#Read in the trace header data
fp = open(FileName)
ColNames = fp.readline().strip().split() #names of columns are read from first line in file
fp.close()
#print ColNames
data = pd.read_csv(FileName,delim_whitespace=True,header=1,names=ColNames)

#Calculate midpoints based on source and receiver positions
Midpoint_X = (data.REC_X + data.SOU_X)/2
Midpoint_Y = (data.REC_Y + data.SOU_Y)/2

if RestrictDeltaPickRange:
    RotatedX, RotatedY = DoRotation(Midpoint_X[(RestrictDeltaPickRange[0]<data.DeltaPick)&(data.DeltaPick<RestrictDeltaPickRange[1])], Midpoint_Y[(RestrictDeltaPickRange[0]<data.DeltaPick)&(data.DeltaPick<RestrictDeltaPickRange[1])], RotRad=np.deg2rad(GridRotation), origin=(GridOriginX,GridOriginY))
elif RestrictOffsetRange:
    RotatedX, RotatedY = DoRotation(Midpoint_X[(RestrictOffsetRange[0]<data.OFFSET)&(data.OFFSET<RestrictOffsetRange[1])], Midpoint_Y[(RestrictOffsetRange[0]<data.OFFSET)&(data.OFFSET<RestrictOffsetRange[1])], RotRad=np.deg2rad(GridRotation), origin=(GridOriginX,GridOriginY))
else:
    RotatedX, RotatedY = DoRotation(Midpoint_X, Midpoint_Y, RotRad=np.deg2rad(GridRotation), origin=(GridOriginX,GridOriginY))

H, yedges, xedges = np.histogram2d(RotatedY, RotatedX, bins=(Yvec, Xvec))
print H.shape
print "Minimum CDP fold in dataset is ", np.amin(H)
print "Maximum CDP fold in dataset is ", np.amax(H)
print "Mean CDP fold is ", np.mean(H), "with standard deviation of", np.std(H)

flatArray = H.flatten('K')
### Make plots ###
fig = plt.figure(figsize=(3.5, 6.5), dpi=100, facecolor='w', edgecolor='k')
plt.hist(flatArray,bins=int(np.amax(H)),align='left') #Simple 1D histogram plot of CMP fold
plt.xlabel('Trace Fold in CDP bins')
plt.ylabel('Number of traces')
#plt.xlim([np.floor(np.percentile(flatArray,0.1))-0.5,np.ceil(np.percentile(flatArray,99.9))+0.5])
plt.xlim([-0.5,18.5])
plt.ylim([0,250000])
plt.tight_layout()

fig = plt.figure(figsize=(12.5, 6.5), dpi=100, facecolor='w', edgecolor='k') #now plot the 2D fold map

im = plt.imshow(H, interpolation='hanning', aspect='auto', origin='low',clim=(0, 10),cmap='gnuplot2_r',#'hsv_r',aspect='equal'
                extent=[Xvec[0]-GridOriginX, Xvec[-1]-GridOriginX, Yvec[0]-GridOriginY, Yvec[-1]-GridOriginY])
plt.colorbar(ticks=range(11),label='Trace fold in CDP bins')
plt.xlabel('Distance in inline direction (m)')
plt.ylabel('Distance in crossline direction (m)')
plt.title('Trace fold in CDP bins')
plt.tight_layout()
plt.show()
```

Appendix 3 – Python script for interpolating and converting SeaTrack navigation files

```
# -*- coding: utf-8 -*-
"""
Read Seatrack navigation data and convert to a more useful format

Rowan Romeyn
"""
import pandas as pd
import numpy as np
import datetime
from dateutil.parser import parse
import glob, os, sys
import json
#from argparse import ArgumentParser
from gooey import Gooey, GooeyParser

np.set_printoptions(threshold=np.inf)#edgeitems=20)
pd.set_option('display.max_seq_items', None)
nonbuffered_stdout = os.fdopen(sys.stdout.fileno(), 'w', 0)
sys.stdout = nonbuffered_stdout

### Build the GUI
@Gooey(program_name="Process Seatrack Navigation Files",
       default_size=(960,600),
        required_cols=1)
def parse_args():
    """ Use GooeyParser to build up the arguments we will use in our script
    Save the arguments in a default json file so that we can retrieve them
    every time we run the script.
    """
    stored_args = {}
    # get the script name without the extension & use it to build up
    # the json filename
    script_name = os.path.splitext(os.path.basename(__file__))[0]
    args_file = "{}-args.json".format(script_name)
    # Read in the prior arguments as a dictionary
    if os.path.isfile(args_file):
        if not os.stat(args_file).st_size==0:
            with open(args_file) as data_file:
                stored_args = json.load(data_file)
    parser = GooeyParser(description='Process SeaTrack Navigation files (which should be supplied including headers and footers) into a more friendly format and interpolate across data gaps')
    parser.add_argument('data_directory',
                        action='store',
                        default=stored_args.get('data_directory'),
                        widget='DirChooser',
                        help="Directory containing SeaTrack navigation files with .log file extension")
    parser.add_argument('navigation_flags',
                        action='store',
                        default=stored_args.get('navigation_flags'),
                        help="Navigation flags seperated by commas e.g. STB, GUN, PRT, AFT")
    parser.add_argument('num_header_and_footer_lines',
                        action='store',
                        default=stored_args.get('num_header_and_footer_lines'),
                        help="Number of lines before data begins and after it ends (seperated by comma)")
    parser.add_argument('computer_timeshift',
```

```
                    action='store',
                    default=stored_args.get('computer_timeshift'),
                    help="timeshift of datestamp header compared to data section e.g. +2 if 2hrs ahead")
    parser.add_argument('output_directory',
                        action='store',
                        default=stored_args.get('output_directory'),
                        widget='DirChooser',
                        help="Directory where the processed navigation files will be saved as .txt files")
    parser.add_argument('-Sensible_EastingRange',
                    #action='store',
                    default=stored_args.get('Sensible_EastingRange'),
                    help="Eastings outside range invalid e.g. 400000, 499999 or None to ignore")
    parser.add_argument('-Sensible_NorthingRange',
                    #action='store',
                    default=stored_args.get('Sensible_NorthingRange'),
                    help="Northings outside range invalid e.g. 8000000, 8999999 or None to ignore")

    args = parser.parse_args()
    # Store the values of the arguments so we have them next time we run
    if args:
        with open(args_file, 'w') as data_file:
            # Using vars(args) returns the data as a dictionary
            json.dump(vars(args), data_file)
    else:
        sys.exit(0)
    return args

### The body of the program #######################################################
if __name__ == '__main__':
    SuppliedArgs = parse_args()
    ##### Parameters ##################################################################
    InDir = SuppliedArgs.data_directory
    OutDir = SuppliedArgs.output_directory
    FileList = glob.glob(os.path.join(InDir, "*.log"))
    NavigationFlags = SuppliedArgs.navigation_flags.replace(' ', '').upper().split(',')
    #print NavigationFlags
    num_headerlines = int(SuppliedArgs.num_header_and_footer_lines.split(',')[0])
    num_footerlines = int(SuppliedArgs.num_header_and_footer_lines.split(',')[1])
    LocalTime2UTC_offset = -1*float(SuppliedArgs.computer_timeshift)
    if SuppliedArgs.Sensible_EastingRange and not 'none' in SuppliedArgs.Sensible_EastingRange.lower():
        SaneEastingRange = [int(i) for i in SuppliedArgs.Sensible_EastingRange.split(',')]
    else:
        SaneEastingRange = [np.nan,np.nan]
    if SuppliedArgs.Sensible_NorthingRange and not 'none' in SuppliedArgs.Sensible_EastingRange.lower():
        SaneNorthingRange = [int(i) for i in SuppliedArgs.Sensible_NorthingRange.split(',')]
    else:
        SaneNorthingRange = [np.nan,np.nan]

    for NavFile in FileList:
        print "Processing file:", NavFile.split('\\')[-1]
        CreateOutputFile = True
        ##################################################################################
        colnames = ['hour','minute','second']
        colspecs = [(0,2),(2,4),(4,9)]

        numFlags = len(NavigationFlags)
        fp = open(NavFile)
        FoundData = False
```

```
        for i, line in enumerate(fp):
            if 'Date:' in line:
                timestamp = parse(line.replace('Date: ','').rstrip('\n'),fuzzy=True) + datetim
    e.timedelta(hours=LocalTime2UTC_offset)
                print "Timestamp read and shifted to", timestamp
                DayOfYear = timestamp.timetuple().tm_yday
                print "Corresponding day of year:", DayOfYear
            if i >= num_headerlines: #next line after header since 0 indexed
                if line[0:9].replace(".", "", 1).isdigit():
                    print "First line of valid data found at line", i+1
                    num_headerlines = i
                    FoundData = True
                    for k in range(numFlags):
                        index = line.find(NavigationFlags[k])
                        print NavigationFlags[k], 'found at col', index
                        EastingIndex = (index+3,index+13)
                        NorthingIndex =(index+13,index+25)
                        colnames.append(NavigationFlags[k]+'_E')
                        colspecs.append(EastingIndex)
                        colnames.append(NavigationFlags[k]+'_N')
                        colspecs.append(NorthingIndex)
            if FoundData:
                break
        fp.close()

        NavData = pd.read_fwf(NavFile,colspecs=colspecs,names=colnames,skiprows=num_headerline
    s,skipfooter=num_footerlines)

        #Perform sanity checks on data and interpolate positions across gaps if necessary
        #sanity check first and replace erroneous data with NaNs NOT IMPLEMENTED
        #NaNs in navigation fields will be interpolated across
        #time fields are considered periodic coordinates and are interpolated accordingly
        numCol = len(colnames)
        for k in range(numCol):
            if '_E' in colnames[k]:
                if min(NavData[colnames[k]]) < SaneEastingRange[0] or max(NavData[colnames[k]]
    ) >SaneEastingRange[1]:
                    print 'Invalid data range detected for', colnames[k], min(NavData[colnames
    [k]]), max(NavData[colnames[k]])
                    print 'The offending values will be removed and interpolated across'
                    NavData[colnames[k]] = [item if str(item).replace(".", "", 1).isdigit() el
    se np.NaN for item in NavData[colnames[k]]]
                #print colnames[k]
                array= np.asarray((NavData[colnames[k]]),dtype='float64')
                array[np.where(array>SaneEastingRange[1])[0]] = np.NaN
                array[np.where(array<SaneEastingRange[0])[0]] = np.NaN
                nans, x= np.isnan(array), lambda z: z.nonzero()[0]
                array[nans]= np.interp(x(nans), x(~nans), array[~nans])
                NavData[colnames[k]] = array
            elif '_N' in colnames[k]:
                if min(NavData[colnames[k]]) < SaneNorthingRange[0] or max(NavData[colnames[k]
    ]) >SaneNorthingRange[1]:
                    print 'Invalid data range detected for', colnames[k], min(NavData[colnames
    [k]]), max(NavData[colnames[k]])
                    print 'The offending values will be removed and interpolated across'
                    NavData[colnames[k]] = [item if str(item).replace(".", "", 1).isdigit() el
    se np.NaN for item in NavData[colnames[k]]]
                array= np.asarray((NavData[colnames[k]]),dtype='float64')
                array[np.where(array>SaneNorthingRange[1])[0]] = np.NaN
                array[np.where(array<SaneNorthingRange[0])[0]] = np.NaN
                nans, x= np.isnan(array), lambda z: z.nonzero()[0]
                array[nans]= np.interp(x(nans), x(~nans), array[~nans])
                NavData[colnames[k]] = array

```

```
169.         if min(NavData['second']) < 0 or max(NavData['second']) >= 60:
170.             print 'Invalid data range detected for seconds', min(NavData['second']), max(NavDa
    ta['second'])
171.             print 'The offending values will be removed and interpolated across'
172.             NavData['second'] = [item if str(item).replace(".", "", 1).isdigit() else np.NaN f
    or item in NavData['second']]
173.         array= np.asarray((NavData['second']),dtype='float64')
174.         array[np.where(array>= 60)[0]] = np.NaN
175.         array[np.where(array< 0)[0]] = np.NaN
176.         nans, x= np.isnan(array), lambda z: z.nonzero()[0]
177.         array[nans]= np.interp(x(nans), x(~nans), array[~nans],period=60)
178.         NavData['second'] = array
179.
180.         if min(NavData['minute']) < 0 or max(NavData['minute']) >= 60:
181.             print 'Invalid data range detected for minutes', min(NavData['minute']), max(NavDa
    ta['minute'])
182.             print 'The offending values will be removed and interpolated across'
183.             NavData['minute'] = [item if str(item).replace(".", "", 1).isdigit() else np.NaN f
    or item in NavData['minute']]
184.             NavData['minute'] = NavData['minute'].astype('float64')
185.         #print NavData['minute']
186.         a= np.asarray((NavData['minute']),dtype='float64')
187.         a[np.where(a>= 60)[0]] = np.NaN
188.         a[np.where(a< 0)[0]] = np.NaN
189.         NaNindexList = np.where(np.isnan(a))[0]
190.         for ind in NaNindexList:
191.             if NavData['second'][ind] > 1.0:
192.                 a[ind] = a[ind-1]
193.             else:
194.                 a[ind] = a[ind-1] + 1
195.         NavData['minute'] = a
196.         #print NavData['minute']
197.         NavData['minute'] = NavData['minute'].astype('int8')
198.
199.         if min(NavData['hour']) < 0 or max(NavData['hour']) >= 24:
200.             print 'Invalid data range detected for hours', min(NavData['hour']), max(NavData['
    hour'])
201.             print 'The offending values will be removed and interpolated across'
202.             NavData['hour'] = [item if str(item).replace(".", "", 1).isdigit() else np.NaN for
     item in NavData['hour']]
203.             NavData['hour'] = NavData['hour'].astype('float64')
204.         a= np.asarray((NavData['hour']),dtype='float64')
205.         a[np.where(a>= 24)[0]] = np.NaN
206.         a[np.where(a< 0)[0]] = np.NaN
207.         NaNindexList = np.where(np.isnan(a))[0]
208.         for ind in NaNindexList:
209.             if NavData['minute'][ind] == 0 and NavData['second'][ind] < 1.0:
210.                 a[ind] = a[ind-1] + 1
211.             else:
212.                 a[ind] = a[ind-1]
213.         NavData['hour'] = a
214.         NavData['hour'] = NavData['hour'].astype('int8')
215.
216.         #Check if the nav file crosses into a new day
217.         numRows = len(a)
218.         DayOfYearCol = np.full(numRows,DayOfYear,dtype='int32')
219.         #print DayOfYearCol
220.         for i in range(numRows):
221.             if NavData['hour'][i] == 0:
222.                 if NavData['minute'][i] == 0:
223.                     if NavData['second'][i] <= 1.0:
224.                         print "A new day begins, updating day of year to", DayOfYearCol[i-
    1] + 1
225.                         DayOfYearCol[i::] = DayOfYearCol[i-1] + 1
```

```
226.
227.        NavData.insert(0, 'DayOfYear', DayOfYearCol, allow_duplicates=False)
228.
229.        #optionally write output file
230.        if CreateOutputFile:
231.            print "writing output file..."
232.            OutputFile = os.path.join(OutDir,NavFile.split('\\')[-
    1].replace('.log','')+'_InterpretedSeaTrack.txt')
233.            NavData.to_csv(OutputFile,sep='\t',float_format='%.3f',index=False)
```

Appendix 4 – Python script used to invert observed BSR for geothermal gradient

```
1.  #-------------------------------------------------------------------------------
2.  # Name:        BSR_GTG_Inversion
3.  # Purpose:      invert BSR for geothermal gradient determination
4.  #
5.  # Author:      Rowan Romeyn
6.  # Created:     22/01/2017
7.  #-------------------------------------------------------------------------------
8.
9.  from matplotlib import pyplot as plt
10. import pandas as pd
11. import scipy as sp
12. from scipy.interpolate import interp1d, griddata
13.
14. ### Parameters ###############################################################
15.
16. HydrateStabilityFileName = "PT_100M.txt"
17. BSR_FileName = "BSR_TWT_Surface"
18. WB_FileName = "WB_PrestackMig_Export_v2.txt"
19.
20. velocity = 1479 #velocity for depth conversion in m/s
21. SeawaterDensity = 1027 #for hydrostatic pressure calculation (kg/m3)
22. BottomWaterTemp = -1 #degrees celsius
23.
24. outfile = "BSR_xyzGTG.txt"
25. ###############################################################################
26.
27. BSR_Data = pd.read_table(BSR_FileName,sep='\s+',comment='#',header=None, usecols=[0,1,2],names
    =['x','y','twt'])
28. WB_Data = pd.read_table(WB_FileName,sep='\s+',comment='#',header=None, usecols=[0,1,2],names=[
    'x','y','twt'])
29.
30. WB_BSR_Grid = griddata((WB_Data.x,WB_Data.y),WB_Data.twt,(BSR_Data.x,BSR_Data.y),method='neare
    st')
31. WB_BSR_Grid.reset_index(drop=True,inplace=True)
32. BSR_thickness = velocity*(BSR_Data.twt - WB_BSR_Grid)/2000 #convert to depth using velocity in
     m/s and twt in ms
33.
34. HydrateStabilityPoints = pd.read_table(HydrateStabilityFileName,sep='\s+',skiprows=2,header=No
    ne, usecols=[0,1],names=['temp','pressure'])
35. Sorted_HydrateStabilityPoints = HydrateStabilityPoints.sort_values('pressure',ascending=True)

36. ##print Sorted_HydrateStabilityPoints
37. ##print WB_BSR_Grid.min(),WB_BSR_Grid.max()
38. ##print WB_Data.twt.min(),WB_Data.twt.max()
39.
40. BSR_Pressure = (BSR_Data.twt*velocity/2000)*SeawaterDensity*9.8/1000 #Pressure in KPa, 9.8 m/s
    2 is acceleration due to gravity
41.
```

```
42. BSR_Temp = sp.interpolate.interp1d(Sorted_HydrateStabilityPoints.pressure, Sorted_HydrateStabi
    lityPoints.temp, kind='cubic')(BSR_Pressure) #cubic spline interpolation (can give integer rep
    resenting spline order for interpolation)
43. ##print BSR_Temp.min(), BSR_Temp.max()
44. 
45. GeothermalGradient = 1000*(BSR_Temp-BottomWaterTemp)/BSR_thickness #degrees per km
46. print "Geothermal gradient range from", GeothermalGradient.min(),GeothermalGradient.max()
47. 
48. BSR_Data['gtg'] = GeothermalGradient
49. 
50. BSR_Data.to_csv(outfile,sep='\t',index=False)
51. 
52. plt.figure()
53. plt.scatter(BSR_Data.x,GeothermalGradient,c='r',marker='o',s=20,alpha=0.002,edgecolors='none')

54. plt.xlabel('X (UTM Zone 32N)')
55. plt.ylabel('Geothermal Gradient (C/km)')
56. ##plt.show()
57. 
58. plt.figure()
59. plt.plot(HydrateStabilityPoints.pressure,HydrateStabilityPoints.temp,'ro')
60. plt.plot(BSR_Pressure,BSR_Temp,'bo')
61. ##plt.tricontourf(BSR_Data.x,BSR_Data.y,GeothermalGradient)
62. ##plt.colorbar
```