

DETAILED TECTONIC EVOLUTION OF THE REYKJANES RIDGE
DURING THE PAST 15 MA USING MAGELLAN, A NEW TOOL FOR
MODELING MAGNETIC ANOMALIES

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAII IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN
GEOLOGY AND GEOPHYSICS

MAY 2011

By
Ásdís Benediktsdóttir

Thesis Committee:

Richard Hey, Chairperson
Fernando Martinez
Paul Wessel

Keywords: *Reykjanes Ridge, rift propagation, plate tectonics*

Acknowledgements

This thesis is dedicated to Tryggvi Björgvinsson. Without him I would not have taken the challenge to come to Hawaii in the first place. Thanks to my sisters Auður and Berglind and my parents for always being ready to talk to me on skype. I want to thank my advisor Richard Hey for his excellent guidance, support, and contagious enthusiasm. Special thanks to Paul Wessel whose courses I very much enjoyed taking and for the awesome GMT skills I acquired during my stay at the University of Hawaii. Also, special thanks to Fernando Martinez whose insight I was lucky to benefit from through great discussions and explanations. Fernando, thanks for the awk-lessons. Thanks to the crew of R/V Knorr in the summer of 2007 when we collected the data the thesis is built on. Thanks to Ármann Höskuldsson for teaching me the geology of Iceland. Thanks to John Sinton, Clint Conrad, Garret Ito, Janet Becker, Greg Moore for discussions I had with them and remarkable courses I took with them. Very special thanks to Michaela Conley with whom I spent countless hours in the office with the vibrating walls; also thanks for all the lunches, walks, and the other morale building activities. Nancy Niklis, thanks for being the perfect roommate and for all the good moments on the couch. Mahalo to all my hotspot friends around the world for support. Financial support was given by a National Science Foundation grant OCE-045213 and a fellowship from the Nordic Volcanological center in Iceland.

Abstract

This Master's thesis is in two parts. The first chapter is a paper describing the new forward magnetic modeling program, Magellan. The second chapter is a detailed study on the evolution of the Reykjanes Ridge based on the Magellan modeling program. Both chapters will be submitted as stand-alone publications. An abstract is included in each.

Contents

Acknowledgements	ii
Abstract	iii
List of Tables	vi
List of Figures	viii
1 Magellan: A new magnetic anomaly modeling program	1
1.1 Abstract	1
1.2 Introduction	1
1.3 Previous work	2
1.3.1 Other programs	2
1.4 Magellan	5
1.4.1 The code	5
1.4.2 Functions	6
1.5 Case Studies	15
1.5.1 No data: Teaching purposes	15
1.5.2 Case example: The Reykjanes Ridge	15
1.6 Conclusions	17
2 Detailed Tectonic Evolution of the Reykjanes Ridge During the Past	
15 Ma	18
2.1 Abstract	18
2.2 Introduction	18
2.3 Data	20
2.4 Methods	20

2.4.1	Modeling oblique spreading centers	20
2.4.2	Outward displacement	22
2.4.3	Picking the axis location	23
2.5	Reykjanes Ridge spreading rates	23
2.6	Reykjanes Ridge accretion asymmetry	25
2.6.1	Asymmetric spreading	25
2.7	Propagating rift magnetic modeling	27
2.7.1	How propagating rifts and ridge jumps relate	27
2.7.2	Prior results of <i>Hey et al.</i> [2010]	27
2.7.3	Assumptions	28
2.7.4	Magnetic models	28
2.8	Results	30
2.8.1	Reconstruction movies	31
2.8.2	Profile 19 and south	32
2.8.3	Transitional profiles 17-18	34
2.9	Discussion	36
2.9.1	Unresolved puzzles	38
2.10	Speculations	40
2.11	Conclusions	41
	References	76
	Appendices	82
	A Magellan	82
A.1	magellan	82
A.2	calc.py	88
A.3	plot.py	100
A.4	data.py	104

List of Tables

Chapter 1: Magellan

1.1 Galapagos modeling parameters	3
1.2 Reykjanes Ridge modeling parameters	11

Chapter 2: Reykjanes Ridge evolution

2.1 Location of ridge axis and central anomaly misfit	67
2.2 Stage poles of rotation for Eu-Na (Eurasia fixed).	68
2.3 Distance of anomalies from the ridge in kilometers.	69
2.4 Full spreading rates (km/Myr)	70
2.5 Asymmetry of profile 20	71
2.6 Parameters used for the forward magnetic modeling.	72
2.7 Magnetic modeling misfit values from this study and <i>Hey et al.</i> [2010]	75

List of Figures

Chapter 1: Magellan

1.1	Comparison between MODMAG and Magellan	4
1.2	The coordinate system used in Magellan	7
1.3	Geometry of variables in Magellan	8
1.4	Output of Magellan with data	12
1.5	Explanation of a propagating rift	13
1.6	Default output of Magellan	16

Chapter 2: Reykjanes Ridge evolution

2.1	Tectonic setting of the Reykjanes Ridge	43
2.2	Magnetic data survey area	44
2.3	Lithospheric accretion vs. distance from pole of rotation	45
2.4	Spreading rates used in the magnetic modeling vs. distance away from pole of rotation	46
2.5	Profile 20 modeled using asymmetric spreading	47
2.6	Sense of asymmetric accretion on Profile 20	47
2.7	Schematic comparison between two asymmetry-producing mechanisms, a propagating rift and asymmetric spreading	48
2.8	Profile 17 modeled using ridge jumps	49
2.9	Profile 18 modeled using ridge jumps	50
2.10	Profile 19 modeled using ridge jumps	51
2.11	Profile 20 modeled using ridge jumps	52
2.12	Profile 21 modeled using ridge jumps	53
2.13	Profile 22 modeled using ridge jumps	54

2.14 Profile 23 modeled using ridge jumps	55
2.15 Profile 24 modeled using ridge jumps	56
2.16 Profile 25 modeled using ridge jumps	57
2.17 Time of jumps vs. distance from the Reykjanes Peninsula	58
2.18 Schematic evolution of the Reykjanes Ridge	59
2.19 Evolution of the Reykjanes Ridge based on satellite gravity	62
2.20 Propagation of Loki	65
2.21 Propagation pattern in the survey area	66

Chapter 1

Magellan: A new magnetic anomaly modeling program

1.1 Abstract

Critical to the theories of seafloor spreading and plate tectonics are seafloor ages and spreading rates determined from modeling marine magnetic anomalies. We have developed a new marine magnetic modeling program, Magellan, which retains the advantages of existing programs and overcomes many of their limitations. Magellan uses open-source and platform-independent code to model two dimensional magnetic fields produced by magnetic polarity intervals created by seafloor spreading. The block model can include seafloor bathymetry and variable magnetization strength and thickness of the source layer. Other variables include an arbitrary number of ridge jumps and spreading rates. The model also treats oblique spreading, variable profile azimuth, extinct spreading centers, and can apply a "contamination coefficient" to simulate broad polarity reversals sometimes observed at slow spreading rates. Suggestions and comments from users have and will contribute to further development of Magellan, allowing them to add new features and fixes which can be included in future releases of the software. Magellan and its manual can be downloaded at www.magellan-project.net.

1.2 Introduction

As new oceanic crust is generated it cools through the Curie temperature and becomes magnetized in the direction of Earth's current magnetic field. Combined with reversals

of the Earth’s magnetic field and seafloor spreading, this produces a series of normal and reversed magnetized stripes on the ocean floor (*Vine and Matthews* [1963]).

Correlations of the magnetic anomaly pattern with the magnetic reversal timescale (e.g. *Cande and Kent* [1995]; *Lourens et al.* [2004]) are typically done by two-dimensional forward modeling of magnetic anomalies (e.g. *Mendel et al.* [2005]), which reveal seafloor ages and rates of plate motion. In this paper we introduce a new platform-independent program for this purpose named Magellan. Magellan builds on older marine magnetic modeling programs and improves their capabilities, including the modeling of phenomena such as ridge relocations, extinct spreading centers, and asymmetric and oblique spreading. Magellan is free and open source software which gives the user the freedom to use and change Magellan so it fits the users’ needs. It also gives the user the freedom to adapt Magellan to more specific tasks, such as interfacing with existing software (such as the Generic Mapping Tools (*Wessel and Smith* [1995]) and Matlab) for graphical output that can be used for teaching and research purposes.

1.3 Previous work

Magellan was written to combine the useful features of two previous magnetic modeling programs, Magbath and Modmag. Below we review the features and limitations of these programs.

1.3.1 Other programs

Magbath (e.g. *Hey et al.* [1980]; *Caress et al.* [1988]; *Naar and Hey* [1989]; *Fernandez and Hey* [1991]; *Wilson and Hey* [1995]) was developed in the 1970’s by Richard Hey and a series of programmers and students by adding seafloor bathymetry to earlier synthetic block models used by Fred Vine and Jason Morgan in their demonstrations

Table 1.1: Parameters for the modeling of the Galapagos data shown in Figure 1.1.

Spreading rates		Asymmetry		Jumps		Magnetization	
Period (Ma)	Value (km/Myr)	Period (Ma)	Value (%)	Time of Jump (Ma)	Distance (km)	Period (Ma)	Value (A/m)
0–1.5	60.8	0–1.5	-3	1.25	22	0–0.78	13
1.5–4.1	57.1	1.5–3.0	5			0.78–118	10
4.1–5.0	39.4	3.0–5.0	-7				
5.0–118.0	40.0	5.0–118.0	20				

of seafloor spreading and plate tectonics, respectively. It was the first forward modeling program to accurately simulate ridge jumps and show the pseudofaults and failed rifts associated with this process (*Hey et al.* [1980]). The graphical interface that had been incorporated into Magbath and made it so useful was platform dependent, written for long-obsolete graphics terminals and is the main limitation of Magbath.

MODMAG (*Mendel et al.* [2005]) is a more recent forward modeling program. It is easy to learn to use and has a user-friendly graphical interface, which however depends on Matlab, a proprietary software package. The graphical user interface allows for a set number of parameters which the user can input to the program. The graphical representation of the magnetic model and bathymetry in Magellan is based on MODMAG as is the option of modeling extinct spreading centers. One problem with MODMAG is that pseudofaults and failed rifts are represented incorrectly. Figure 1.1A shows a data set from the Galapagos spreading center at $88^{\circ}W/0.8^{\circ}N$ (*Wilson and Hey* [1995]) modeled with the MODMAG program with parameters shown in Table 1.1, including one 22 km ridge jump occurring at 1.25 Ma. This data was first modeled with Magbath in *Wilson and Hey* [1995] which produced essentially identical fits to those with MODMAG and Magellan (Figure 1.1). One MODMAG pseudofault is shown in the same color as the failed rift and the other pseudofault has a different color (pseudofaults and failed rifts are vertical lines) (1.1A). Furthermore, there is one additional vertical line that is not a tectonic boundary, but shows the width

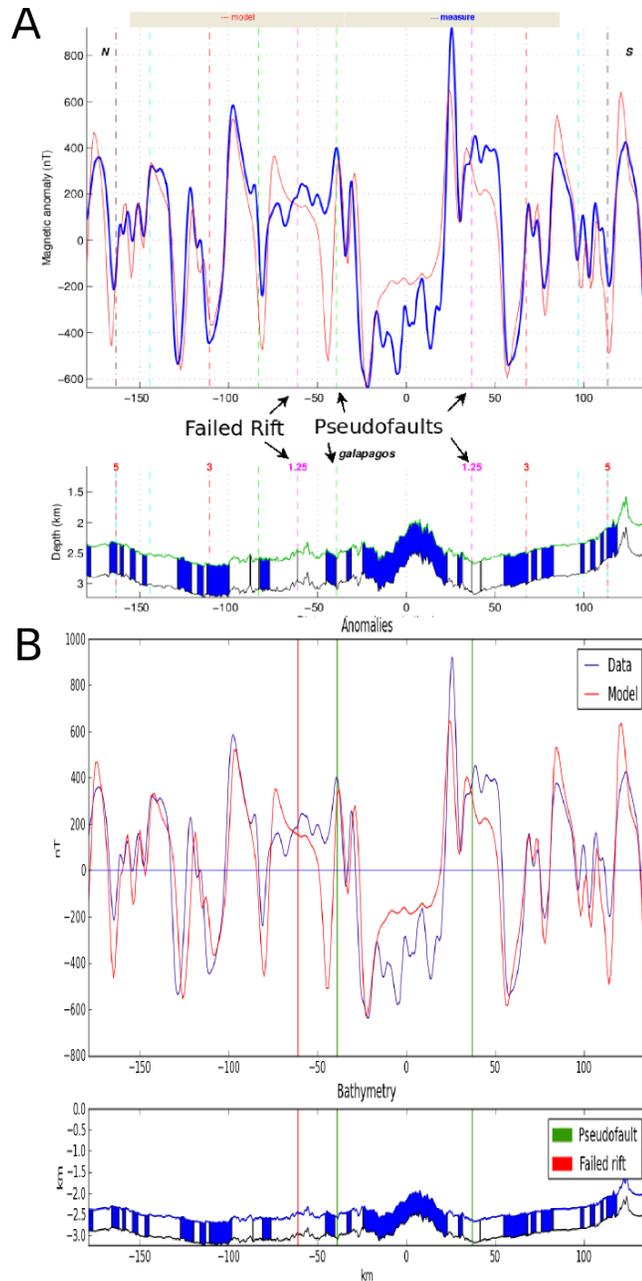


Figure 1.1: Data from the Galapagos area ($88^{\circ}W/0.8^{\circ}N$) modeled with **A** MODMAG (*Mendel et al.* [2005]) and **B** Magellan using parameters from Table 1.1. Blue and white boxes follow the bathymetry and represent normal and reversely magnetized blocks, respectively. **A** The red line is the model and the blue line is the data. In MODMAG the two pseudofaults and failed rift associated with the ridge jump are not represented quite correctly (see text). **B** Red line is the model and blue line is the data. Here the pseudofaults (green vertical lines) and failed rift (red vertical line) are represented correctly.

of lithosphere created symmetrically on the failed rift. The use of a contamination coefficient (*Tisseau and Patriat* [1981]) to smooth the model when modeling slow spreading ridges was used in MODMAG and is incorporated in Magellan. The model and the data are plotted together in Magellan as in MODMAG, rather than offset as in Magbath. To model magnetic anomalies on the Reykjanes Ridge Benediktsdottir et al. (in prep.) needed more than 8 jumps which is not possible in MODMAG (the user interface allows for no more than 8 jumps). Also, some magnetic models in *Hey et al.* [2010] needed jumps younger than the youngest reversal (0.78 Ma), which crashed MODMAG. These were some of the main motivations for creating Magellan.

1.4 Magellan

1.4.1 The code

Magellan is a free and open source software and is released under the GNU General Public Licence (GPL) version 3. It gives the user the freedom to use, study, adapt, improve, and redistribute the software. This will lead to a better and constantly evolving Magellan because it allows the software to be critically examined by others and thus improved.

Magellan is written in the programming language Python, which runs on all common platforms (Linux, Unix, Os X, Windows). Although Python is not optimized for number-crunching processes, it is chosen because it was designed with emphasis on code readability. This makes it easy for the user to read and modify the program, thus promoting its maintainability (*Sanner* [1999]). Python also comes with extensive standard libraries, which make the graphical representation of the data easy to control. Also, calculations in Magellan use the Python Numpy module which supports matrix operations and therefore makes the code run faster. Magellan is a command-line based program, which gives the scientist the freedom to work with the data in an

efficient way, unlike 'many-button' graphical user interfaces, which have the tendency to constrain the options presented to the user. Users can, however, always write and share their own user interfaces tailored to their research. This leaves a huge window open for the program to expand in directions led by the needs of the scientists doing magnetic modeling. Although Python is not the ideal language for heavy and complex calculations, it has the ability to easily read functions written in C or Fortran, leaving the option open to link extensions to Magellan.

Magellan is built up from one main executable module and several python modules which contain the methods used in the main module. The python modules are

- magellan - The main module which executes the methods defined in the following modules.
- calc.py - The magnetic anomaly is calculated from the source layer model in this module.
- plot.py - All configurations for the plot are kept in this module.
- data.py - The data are read and stored into appropriate data structures in this module.

1.4.2 Functions

Magellan's algorithm is based on a technique developed by *Talwani and Heirtzler* [1964] where the contribution from each side of a two dimensional magnetized polygon is calculated along the x-axis of a right-handed (positive) Cartesian coordinate system shown in Figure 1.2. The positive direction of the x-axis has an azimuth in the 0-360 degree range, measured clockwise from north. The azimuth of the ridge is oriented along the y-axis (which is always 90° counter-clockwise from the x-axis) and the positive z-axis is oriented down (the depth measured is positive). The parallelepipeds

represent the homogeneously magnetized bodies and their upper boundary is defined by the bathymetry along the profile along which data are acquired (Figure 1.3). The lower boundary of the parallelepipeds is the bathymetry with the model thickness added and the sides are defined by vertical boundaries given by the geomagnetic reversal sequence. A cross section of the parallelepipeds along the x-axis reveals the magnetized polygons used to calculate the model (Figure 1.1).

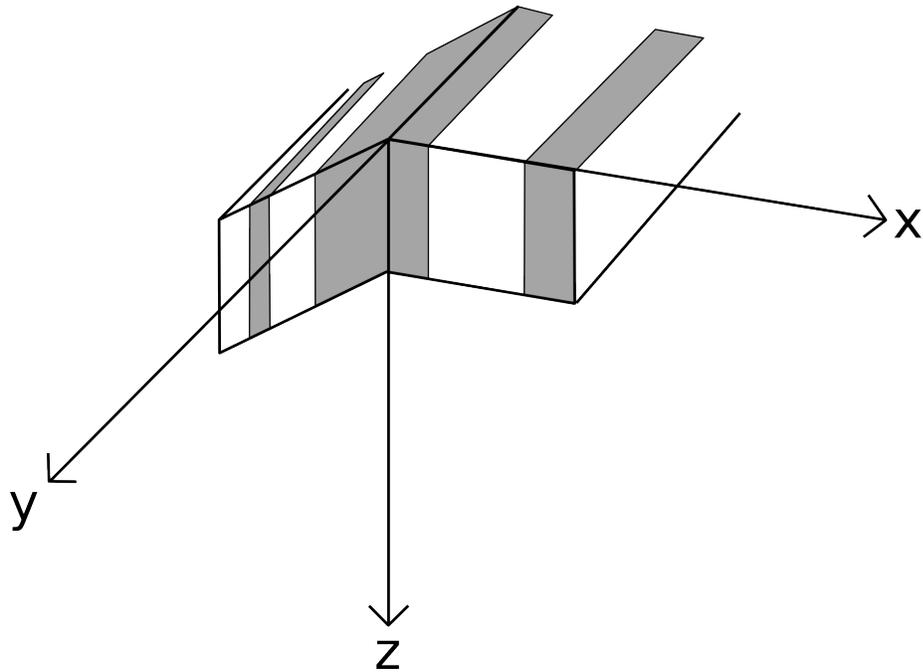


Figure 1.2: Right-handed coordinate system used in Magellan and a cross-section of the parallelepipeds, which extend infinitely along the ridge. The cross-section lies in the xz -plane so that a profile running along the x -axis is perpendicular to the strike of the parallelepipeds. Top and bottom boundaries of the magnetized blocks are defined by the bathymetry, here the grey and white boxes schematically show normal and reversely magnetized blocks.

The model that Magellan calculates arises only from the thermal remnant magnetization (TRM) of the basaltic basement so the International Geomagnetic Reference Field (IGRF, i.e. <http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html>) has to be removed from the data set to yield the magnetic anomaly which is compared to the model. The magnetized layer is often thought primarily to be the ~ 500 meter thick seismic layer 2A (*Keen and Tramontini [1970]; Schouten et al. [1999]*) while others

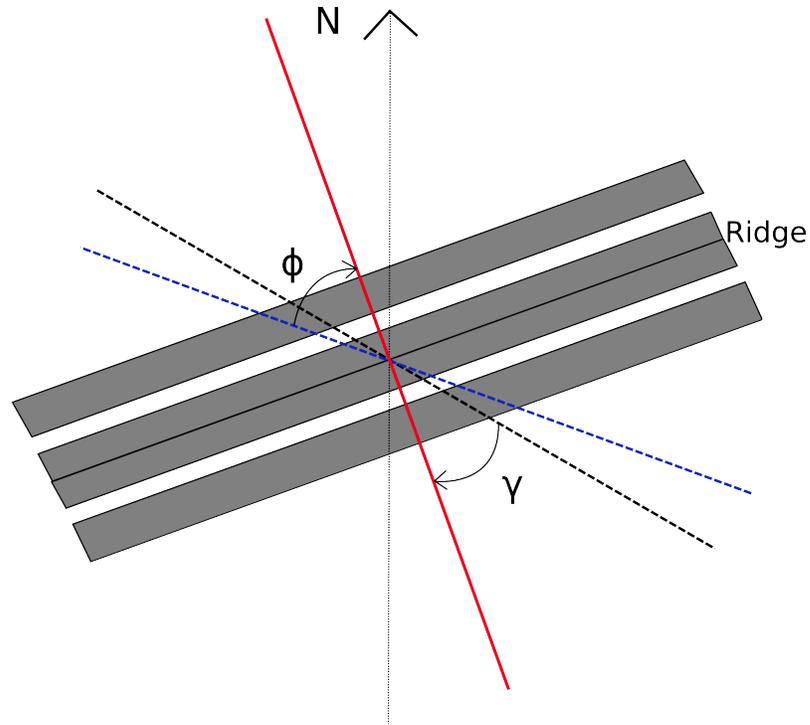


Figure 1.3: A map view of various profiles relative to a ridge. Magnetic lineations are shown (grey normally polarized and white reversly polarized). The red line is a profile oriented perpendicular to the ridge, the black dashed line is an oblique profile along which data are collected and the blue dashed line is the current spreading direction azimuth of the ridge. ϕ and γ are as shown. In the case of conventional perpendicular spreading $\phi=0$ but γ can be any number in the 0-90 degree range. In the case of an oblique spreading ridge where data are collected along the current spreading direction $\gamma=\phi$.

claim it to be much thicker (e.g. *Harrison* [1976]). In Magellan the user can change this parameter, but the default thickness is 500 m. It is worth noting that changing the thickness of the magnetized layer and changing the intensity of magnetization both cause the amplitude of the anomalies to change, and a thicker source layer with low intensity of magnetization can therefore also be represented by a thinner source layer with higher intensity of magnetization. The magnetization vectors of the magnetized polygons are assumed to be aligned parallel to the local field of the geocentric dipole (no declination). This is true except when modeling extinct spreading centers, the user has to specify the paleostrike of the ridge and the paleoinclination at the time of formation. The total magnetic anomaly arising from the polygons is obtained

by projecting the magnetization vector parallel to the direction of the local magnetic field, adding to or subtracting from the total local magnetic field intensity.

Magellan, like Magbath, is able to handle as many parameters as the user wants. With Magellan, the user can simulate symmetric and asymmetric seafloor spreading where, for example, 5% asymmetry implies spreading along the positive and negative x-axis of 1.05 and 0.95 times the half spreading rate, respectively. Using Magellan, the user can choose different magnetization intensities of the source layer for different time intervals. The default intensity of magnetization is 10 A/m. There are no constraints on number of jumps, asymmetry periods, spreading rates or intensity of magnetization parameters. Another improvement Magellan provides over existing programs is the ability to plot other kinds of data when doing magnetic modeling (e.g. free air gravity) (Figure 1.4).

Magellan reads in the previously described parameters along with the following parameters:

- *Source layer thickness.* Default is 0.5 km.
- *Profile Azimuth.* The azimuth of the profile along which the data are collected, measured clockwise from north in degrees (range: 0°-360°). Default is 90°.
- *Inclination.* The inclination of the Earth's magnetic field at the time and position the measurement was taken. This can be found at <http://www.ngdc.noaa.gov/>. Default is 45°.
- *Declination.* The declination of the Earth's magnetic field at the time and position the measurement was taken. This can be found at <http://www.ngdc.noaa.gov/>. Default is 0°.
- *Ridge Azimuth.* The azimuth of the ridge which defines the y-axis of the coordinate system shown in Figure 1.2, measured clockwise from north in degrees (range: 0 – 360°). Default is 180°.

- *Spreading Direction.* The azimuth of the current spreading direction, measured clockwise from north in degrees (range: 0° - 180°). Default is 90° .
- *Model spacing.* The distance between points where the model is calculated. Default is 1 km.
- *Level.* The height in kilometers above sea level where the model is calculated (positive is above sea level and negative is below sea level). Note that the level can never be lower than the depth of any data points (that is the bathymetry). Default is 0 km (sea level).
- *xmin/xmax.* The minimum and maximum extent of the model. Default is the distance obtained from the end of the timescale.
- *Timescale.* The timescale used. Default is *Lourens et al.* [2004].
- *Oldest reversal time.* The time that defines the end of the magnetized body (no magnetized block is older than this parameter). This parameter needs to be smaller or equal to the end of the timescale used. Default is the end of the timescale used.
- *Age at the ridge crest.* When modeling extinct spreading centers the ridge crest is not zero age. The tmin option defines the age at the ridge crest. Default is 0 Ma.
- *Paleoinclination.* the inclination of the magnetization vector at the time of formation; this parameter is used if the magnetized body has moved over many degrees of latitude and therefore does not have the same inclination as the magnetic field around it. Range: -90° to 90° . Not used unless specified.
- *Paleo strike of ridge.* The old strike of the ridge; if the strike of the ridge changes then the magnetization vector undergoes a rotation in the horizontal plane and

this needs to be taken into account when calculating the model. Range: 0° to 360°. Not used unless specified.

Matplotlib, a Python plotting library, is used to plot the results. It includes a zoom in/out feature and an option to save the figure. The user can control the representation of the data by changing the Matplotlib settings in the plot.py module. Magellan writes out ASCII files with the model, data, the organization of the magnetized blocks, and the location (distance away from the ridge and latitude and longitude, if they are provided in the input file) of the pseudofaults and failed rifts, which can be used with other tools such as the Generic Mapping Tools (*Wessel and Smith* [1995]). A simple shell script is included with Magellan which demonstrates how the output ASCII files can be used with the Generic Mapping Tools.

If the input file has longitude and latitude information then the user can pick anomalies with Magellan. This is done by moving the mouse cursor over the desired anomaly and hitting the 'a' button. Furthermore, by editing the plot.py module the user can add more buttons to represent specific features in the modeling.

Table 1.2: Parameters for the modeling of the Reykjanes Ridge data shown in Figure 1.4.

Spreading rates		Jumps		Magnetization	
Period (Ma)	Value (km/Myr)	Time of Jump (Ma)	Distance (km)	Period (Ma)	Value (A/m)
0–6.5	18.8	1.3	-1	0–0.78	20
6.5–23	22.6	4.2	1	0.78–15	8
		6.7	2	15–23	6
		9.1	3		

Ridge Jumps

A ridge jump occurs when the locus of spreading moves laterally from one spreading center to another. Ridge jumps are usually caused by rift propagation. Rift propa-

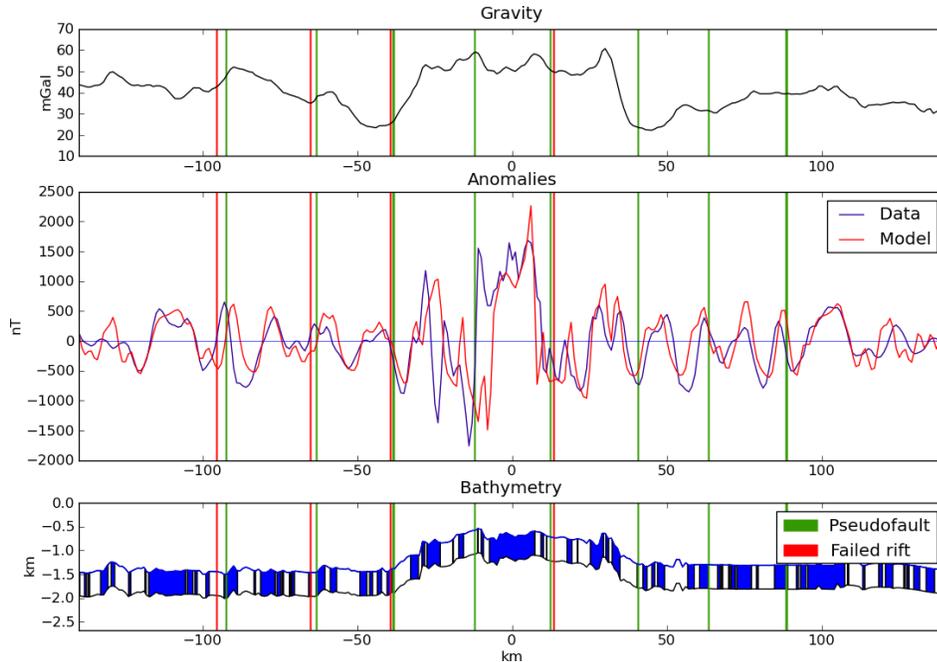


Figure 1.4: The output from Magellan for data from a 2007 Reykjanes Ridge expedition. Top panel; free air gravity data. Middle panel; red is the model and blue line is the data. Bottom panel; blue and white blocks represent normal and reversely magnetized blocks, respectively. The green vertical lines are pseudofaults and red vertical lines failed rifts. Note that with every failed rift there are two pseudofaults, with the failed rift always farther away from the axis than the inner pseudofault.

gation (*Hey* [1977]), where a propagating rift transfers lithosphere from one plate to another, is a three dimensional process which Magellan simulates in two dimensions as an instantaneous axial relocation termed ridge jump. Figure 1.5 shows a propagating rift, in map view, along with the pseudofaults (which offset the magnetic lineations so it looks as if the lineations have been faulted) and failed rift (the extinct trace of the dying ridge) in relation to the active spreading center. A ridge jump appears to be an instantaneous event when the rift propagation is viewed in cross-section, along the spreading flowlines of the ridge. There are two pseudofaults associated with a propagating rift. Magellan, like Magbath, represents this process correctly. Figure 1.1B shows the output from Magellan using the same Galapagos data as shown modeled with MODMAG in Figure 1.1A. Note, however, that rotation of the transferred lithosphere and its magnetization vector (e.g., *Perram et al.* [1993]) is not treated in

our magnetic model. A positive jump distance represents a relocation of the ridge axis in the direction of the positive x-axis so that the failed rift sits to the left of the newly formed ridge axis.

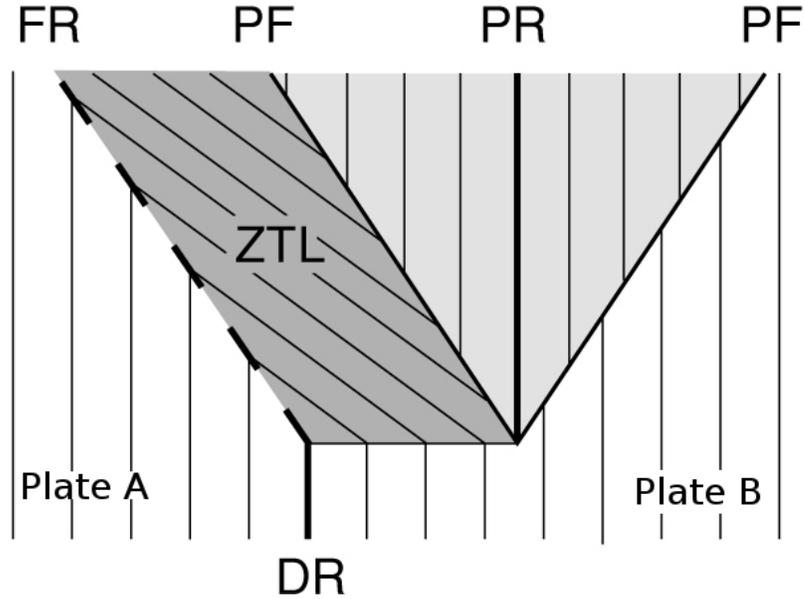


Figure 1.5: A map view of a right-stepping propagating rift (PR) after *Hey et al.* [1980]. The PR propagates down and replaces the dying rift (DR), producing two pseudofaults (PF), a failed rift (FR) and zone of transferred lithosphere (ZTL) where lithosphere was moved from plate B to plate A, producing asymmetric accretion. Thin vertical lines are isochrons, rotated in the ZTL, and offset by the PF. The FR is the extinct trace of DR. A cross-sectional profile perpendicular to the strike of the ridge would show the failed rift outside the pseudofault on plate A.

Smoothing the Model When Seafloor Spreading is Slow

Magnetic anomalies over slow spreading ridges generally lack fine-scale character and resolution because the reversal polarity structure is contaminated by complex seafloor spreading processes (e.g. *Tisseau and Patriat* [1981]; *Macdonald et al.* [1983]; *Mendel et al.* [2005]). This makes it hard to compare the calculated model to real data because the model contains higher frequencies than the data. The method of *Tisseau and Patriat* [1981] was implemented in MODMAG and has been implemented in Magellan to suppress the higher frequencies in the model. This method involves making the

magnetized blocks narrower prior to the calculation of the magnetic model by using a "contamination coefficient" (*Mendel et al.* [2005]) and thus simulating a fictitious spreading rate (*Tisseau and Patriat* [1981]).

The contamination coefficient shrinks the horizontal scale of the magnetized blocks before the anomalies are calculated. This changes their aspect ratio so that they become narrower relative to their depth. The field calculated from these compressed blocks has the shorter wavelengths attenuated more than the longer wavelengths. This is similar to the field formed by a filtered magnetization model, e.g. the Gaussian filter used by *Wilson and Hey* [1995] in their Galapagos Magbath modeling.

Problem With Oblique Spreading and Oblique Profiles

When modeling marine magnetic anomalies using a 2-D forward method, the magnetic source bodies are assumed to extend infinitely along the ridge and the data profiles are assumed to be perpendicular to the ridge. If seafloor spreading is oblique or if magnetic data are collected along oblique profiles (even in the case of conventional orthogonal spreading) the latter assumption does not hold and thus the 2-D forward method can not be used directly. Let ϕ be the angle between the normal to the ridge and the spreading direction and γ be the angle between the normal of the ridge and the profile azimuth, as shown in Fig. 1.3. In Magellan, the following steps are executed in order for the 2-D assumptions to hold:

1. Arrange the synthetic magnetized bodies along the flowlines using spreading rate and jump parameters.
2. Project the synthetic magnetized bodies perpendicular to the trend of the ridge, using ϕ (once this has been done the assumptions for the 2-D method hold).
3. The contamination coefficient is applied to the synthetic magnetized bodies and to the distance scale on which the data are to be evaluated in perpendicular

space.

4. The magnetic model is calculated.
5. The model is scaled back to the original horizontal distance (the compaction of the distance scale due to the contamination coefficient is undone).
6. The data are projected into spreading direction space where they are compared with the model.

If data are collected along the current spreading direction of an oblique spreading ridge, $\phi=\gamma$. If data are not collected along the spreading direction of an oblique spreading ridge then ϕ and γ are not the same. If data are collected perpendicular to a conventional orthogonally spreading ridge, $\phi=\alpha=0$. The timing of a ridge jump is most accurate if data are collected parallel to the spreading flowlines of the ridge and in order to properly model rift propagation development one should use data collected in such a way.

1.5 Case Studies

1.5.1 No data: Teaching purposes

One can use Magellan without providing it with a data file in order to get a better understanding of how the magnetic field changes with different inclination and/or declination, observation elevation and source layer thickness. Figure 1.6 shows the output from Magellan when it is run with its default parameters and $tend=20 Ma$.

1.5.2 Case example: The Reykjanes Ridge

The Reykjanes Ridge is the part of the Mid-Atlantic ridge located southwest off the coast of Iceland, and extends from the Reykjanes Peninsula to the Bight Transform

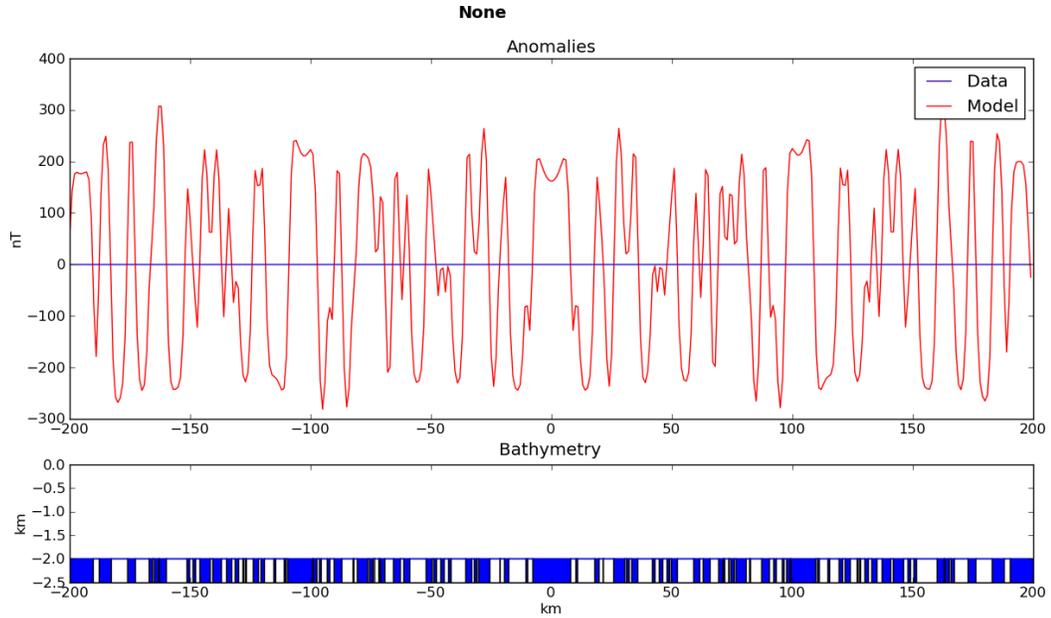


Figure 1.6: The output from Magellan when run with all its default values and $tend=20$ Ma. Red is the model and there is no data. The top of the flat bathymetry is -2 km.

Fault. It is anomalous in many ways, including its 30° oblique spreading from perpendicular to the ridge, a topographic axial high instead of a conventional axial valley for slow spreading ridges, and the diachronous V-shaped patterns (*Vogt* [1971]) observed from topography and gravity data. The spreading direction of the ridge is 100° (*DeMets et al.* [1990, 1994, 2010]), measured clockwise from north, yet the ridge strikes $\sim 40^\circ$ relative to north. Figure 1.4 shows data from a 2007 Reykjanes Ridge expedition (*Hey et al.* [2010]) modeled with Magellan. The profile runs along the Eurasia-North America flowline, not orthogonal to the ridge, and crosses the ridge axis at 62.55°N . The parameters used in the modeling are shown in Table 1.2. A contamination coefficient of 0.7 was applied and the timescale of *Lourens et al.* [2004] used. In the middle panel the data are shown in blue, the model in red, and red and green vertical lines represent failed rifts and pseudofaults, respectively. The top panel shows an example of how additional data can be represented with the magnetic data, in this case it shows free air gravity along the flowlines. The jumps that were used

in the modeling represent propagators coming down from the Iceland plume that explain observed asymmetries in the V-shaped ridges and magnetic patterns (*Hey et al.* [2010]).

1.6 Conclusions

Magellan is a new and improved tool for the forward modeling of marine magnetic anomalies. It is written in the Python programming language which makes it easy for scientists who are not experts in programming to read the code and modify it for their own research and needs. Magellan can be used to model marine magnetic data sets for research or in a class to teach students the foundation of marine magnetic modeling. Software can always be improved and it is the suggestions and comments from the users that contribute to the development of a constantly improving Magellan. Magellan and its manual can be downloaded at www.magellan-project.net.

Chapter 2

Detailed Tectonic Evolution of the Reykjanes Ridge During the Past 15 Ma

2.1 Abstract

We present a new detailed tectonic model of the Reykjanes Ridge which examines the rift propagation hypothesis for the V-shaped ridges and its asymmetric lithospheric accretion. Four major southward rift propagations extend through our entire survey area and several additional small scale rift propagations are observed, including northward propagators. If plume pulses only drive southward propagators, then two mechanically different kinds of propagators must exist. We find that there is a major difference in the crustal accretion asymmetry between the area immediately off the Iceland shelf and that farther south, both in rift propagation pattern and free air gravity lineations. Furthermore, we identify two small offset features coined *ponsu-transforms*, from which rift propagation is both initiated and stopped. The pattern of the V-shaped ridges on the Reykjanes Ridge is not symmetric about the Reykjanes Ridge and the V-shaped ridges are not linear continuous features. Our rift propagation model produces excellent fits to magnetic data and provides a self-consistent model for the evolution of the Reykjanes Ridge during the past 15 Ma.

2.2 Introduction

The Reykjanes Ridge (RR) is part of the Mid-Atlantic Ridge located in the North Atlantic between Iceland and the Bight Transform Fault near 57°N (Figure 2.1). It is a slow spreading ridge with a full spreading rate of ~ 20 km/Myr along an azimuth

of $\sim 100^\circ$ (*Merkouriev and DeMets* [2008]; *DeMets et al.* [2010]). It is anomalous in many ways, including its oblique spreading of 30° from perpendicular to the ridge, and exhibiting a topographic axial high (*Talwani et al.* [1971]) instead of the usual axial valley found at most slow spreading ridges. The axial high morphology has been attributed to excess melting due to a hotspot or a mantle plume beneath Iceland (*Wilson* [1963]; *Morgan* [1971]).

The Reykjanes Ridge is also anomalous because of the diachronous topography and gravity V-shaped ridges (VSRs) flanking it (Figure 2.1). *Vogt* [1971] discovered this phenomenon and hypothesised that a plume underlying Iceland was pulsing, causing the ridge to have zones of thicker crust than normal during pulses and troughs in between pulses. His hypothesis has since been taken as a fact and many models have been proposed to explain the pulses as asthenosphere or temperature pulses (e.g. *Vogt and Johnson* [1972]; *White et al.* [1995]; *White* [1997]; *White and Lovell* [1997]; *Smallwood and White* [1998, 2002]; *Albers and Christensen* [2001]; *Ito* [2001]; *Jones et al.* [2002]; *Jones* [2003]; *Poore et al.* [2006, 2009]).

One other model had previously been suggested for the origin of the V-shaped ridges. *Hardarson et al.* [1997] suggested that ridge relocations on Iceland (*Sæmundsson* [1974]) disrupt the flow of hot plume material to the RR (*Hardarson et al.* [1997, 2008]), forming the troughs. In this model the Iceland plume is a steady-state plume and the V-shaped troughs anomalous rather than the ridges.

Vogt [1971] concluded his discovery paper with the words 'While the interpretation of V-shaped ridges as indicators of mantle flow seems promising, we do not claim that it is fact. Other propagating effects such as fractures and fluid instabilities should be explored'. Also, *Johansen et al.* [1984] and later *Jones et al.* [2002] observed an asymmetry in the VSRs about the RR and suggested that a more complicated explanation was necessary. Based on a 2007 survey of the RR and its flanks, *Hey et al.* [2010] proposed a model which is compatible with but does not require a mantle

plume in which the VSRs are caused by a series of propagating rifts migrating away from Iceland and producing asymmetry by transferring lithosphere from one plate to the other. Here we extend the initial results of *Hey et al.* [2010] and present a more detailed study of the propagating rift model of the Reykjanes Ridge.

2.3 Data

In order to examine in greater detail the propagating rift model for the Reykjanes ridge we carry out detailed modeling of the magnetic data collected on R/V Knorr in June-July 2007. The shiptracks run nearly parallel to the spreading flowlines of the ridge predicted by the Eurasia-North America rotation parameters of DeMets (pers. comm., 2010) (Figure 2.2). The flatness of the Iceland shelf and its topographic step have been attributed to crustal flow caused by differences in zero-age crustal thickness (*Jones and MacLennan* [2005]). Also, the magnetic anomalies become smooth and low in amplitude on the Reykjanes Ridge where it intersects the Iceland shelf (*Talwani et al.* [1971]; *Vogt et al.* [1980]). Because of the structural complexities on the Iceland shelf we only model the off-shelf profiles 17-25 (survey box in Figure 2.1).

2.4 Methods

We use a newly developed forward marine magnetic modeling program, Magellan (*Benediktsdottir et al.*, in prep.), to model our magnetic data.

2.4.1 Modeling oblique spreading centers

Two dimensional modeling of magnetic anomalies over an obliquely spreading ridge requires special treatment. If a ridge is spreading in the conventional orthogonal manner, one can take data collected on any azimuth across the ridge and simply

project them to a perpendicular azimuth and from there do the two dimensional forward modeling. Since the spreading flowlines of the Reykjanes Ridge are not perpendicular to the strike of the ridge one cannot do this simple projection because lithosphere on either side of the ridge on a ridge-perpendicular profile would not have formed at the same point on axis, i.e., equal age points would not be conjugate. To deal with this problem we carry out the following steps.

1. First the magnetized bodies (normal and reversed polarized blocks) need to be arranged according to spreading, jump, and asymmetry parameters. This is done in flowline space to ensure that the modeled anomalies are conjugate, i.e., that they formed at the same point on the axis.
2. Second, the magnetized bodies are projected into a ridge-perpendicular space so that the two-dimensional assumption holds. The ridge is assumed to extend infinitely along its strike and it is therefore possible to use a 2-D method to calculate a magnetic model which arises from the magnetized ridge. Calculating the model from the magnetic blocks obtained from (1) would be equivalent to having the same ridge orientation but faster spreading (wider blocks). The width of the blocks in ridge-perpendicular space is what controls the width of the anomalies. The greater the obliquity the smoother the model will be because of the narrower width of the projected blocks measured in the ridge-perpendicular direction relative to the flowline direction. In general: the slower the spreading rate the smoother the model for purely geometrical reasons.
3. The model is calculated in ridge-perpendicular space.
4. The magnetic anomaly model is projected back into flowline space where it is compared to the data. This involves only a horizontal stretching of the calculated magnetic anomaly back to the original flowline geometry.

2.4.2 Outward displacement

Outward displacement comprises the effects which cause the youngest magnetic polarity zone to be wider than it would be if calculated from the actual spreading rate, and other polarity periods to be shifted away from the spreading center while maintaining their original width (*Atwater and Mudie* [1973]; *Hey et al.* [1980]; *DeMets and Wilson* [2008]). Intrusions of dikes into older crust of opposite polarity, older lava flows flowing over lava of opposite polarity and accumulation of gabbros under crust of opposite polarity are all processes that would cause the central Brunhes anomaly to be wider than true spreading rates predict. Outward displacement is therefore a source of error in global plate motion models and needs to be corrected for. It has been reported to be as high as 5-6 km on the Reykjanes Ridge (*DeMets and Wilson* [2008]).

For the purposes of magnetic modeling the outward displacement is observed primarily in the central anomaly, causing the first spreading rate period to appear faster (wider) than it actually is. Later spreading stages would not be affected because the outward displacement effect is cancelled out (a reversal period is larger because of outward displacement while it is on the axis but as a new period starts it is shrunk because of the outward displacement of the new period) and all reversal boundaries would be displaced outward (hence the naming, *outward displacement*) so the older spreading rates would not change. In our magnetic modeling the polarity transition zones are vertical which is a likely source of misfit in our Brunhes anomaly modeling.

The central anomaly is generally wider closer to Iceland which could be explained by increased outward displacement. Table 2.1 shows by how many kilometers the central anomaly is wider than the model central anomaly, when summed up for both flanks.

2.4.3 Picking the axis location

We pick the axis in the middle of the observed central anomaly, that is the axis in the model is the time-averaged center of the Brunhes. Therefore small recent propagating rifts might not be detected in the magnetic data; if there is a recent propagating event the axis location should be systematically located on the Eurasia or North America side of the Brunhes. Our ridge axis picks for profiles 17-25 are given in Table 2.1.

2.5 Reykjanes Ridge spreading rates

A fundamental part of forward magnetic modeling is the spreading rates used. They can be determined profile by profile, or predicted by the rotation parameters for the appropriate plate pair at the appropriate part of the ridge. The rotation parameters available are always found by inversions and they are thus best-fit values for a particular plate pair or global plate system. They do not take into account small scale complexities such as asymmetry caused by propagating rifts. Another critical issue is the spreading direction and its possible changes in time. The spreading direction defines the flowlines which are especially important in evaluating ridge propagation events. Unfortunately, the Reykjanes Ridge has no transform faults, which are the best source of information on spreading directions through time. Spreading directions are therefore largely constrained from more regional North Atlantic opening poles.

A rotation pole for the North America - Eurasia plate pair from *Smallwood and White* [2002] (located at $66.85^{\circ}N/135.46^{\circ}E$) predicts our shiptracks which are compared to the flowlines predicted by a new rotation pole of DeMets (2010, pers.comm.) (Table 2.2) in Figure 2.2. We assume our shiptracks are flowlines in this study as they are a good approximation of the newest predicted flowlines and to those of *Merkouriev and DeMets* [2008]. This will introduce an error to our pseudofault location equal to the distance between the shiptracks and the newest flowlines (0 at the axis and ~ 4

km past 6.733 Ma).

The techniques which are used to determine the rotation poles and angular rates (e.g. *Merkouriev and DeMets* [2008]; *DeMets et al.* [2010]) do not detect changes in spreading rate that are less than 1 km/Myr (C. DeMets, pers. comm., 2010). In order to accurately model the magnetic data we need to refine the spreading rates on the Reykjanes Ridge.

Figure 2.3 shows the crustal accretion for chrons 3ro (6.033 Ma), 5n2no (11.04 Ma), 5Bro (15.974 Ma), and 6no (19.722 Ma) (all the chrons discussed here are from *Lourens et al.* [2004]), obtained from our magnetic data (black circles with error bars, from Table 2.3). We measured the distance from the axis to these chrons on either side of the ridge and by summing the amount accreted on either side of the ridge we obtain the total amount of crust accreted along each profile for each time, independent of the ridge axis pick. By using the location of the pole describing our shiptracks we find angular rates which predict a new set of improved spreading rates giving us predicted crustal accretion that best fits the data, minimizing the sum of squares (yellow circles in Figure 2.3). The new stage poles are given in Table 2.2. The new spreading rates, which we ultimately use in our magnetic models are given in Table 2.4 and shown in Figure 2.4 (yellow circles).

If we were to predict spreading rates profile by profile it would result in inconsistent spreading rates between profiles because of complexities such as variable outward displacement in the accretion process. Thus, the spreading rates would generally not increase exactly as the sine of the angular distance away from the location of the rotation pole, as demanded by rigid plate tectonics theory (*Morgan* [1968]). Using spreading rates predicted by best-fitting poles of rotation for all of our profiles causes some imperfection in the forward magnetic modeling but it imposes spreading rate self-consistency between the profiles.

2.6 Reykjanes Ridge accretion asymmetry

The asymmetric accretion of the Reykjanes Ridge is subtle as *Vine* [1966] did not mention it and *Talwani et al.* [1971]; *Herron and Talwani* [1972] stated that the spreading was symmetric. It has now been established that the lithospheric accretion on the Reykjanes Ridge has not been symmetric for the past 20 Ma (*Hey et al.* [2010]). Below we elaborate on different asymmetry-producing mechanisms and show that the propagating rift hypothesis is the most plausible one.

2.6.1 Asymmetric spreading

Continuous asymmetric spreading, where more lithosphere is consistently added to one ridge flank over the other, has been proposed to occur in areas where asymmetrical accretion has been observed (*Menard and Atwater* [1968]; *Weissel and Hayes* [1971]; *Hayes* [1976]; *Stein et al.* [1977]). *Stein et al.* [1977] suggested, by using a fluid mechanical model, that the trailing ridge flank with respect to the ridge migration direction would have a lower viscous dissipation rate and thus have a higher spreading rate. The ridge migration direction of the Reykjanes Ridge is to the northwest relative to Iceland (*Hardarson et al.* [1997]) and thus according to this model the Eurasia plate should accrete more material than the North America plate. Contrary to that prediction, the North American plate accreted more lithosphere between 6.733 and 19.722 Ma, although the Eurasian plate accreted a little more lithosphere since 6.733 Ma (see Table 2.3).

If asymmetric accretion were uniform over many ridge segments asymmetry should be greater further away from the pole of rotation (the increase should follow the sine of the angular distance from the pole). In case of the Reykjanes Ridge, the asymmetry decreases away from the pole of rotation. Asymmetric accretion on the Reykjanes Ridge has not been uniform or in the same sense. It changes from accreting more

lithosphere on the Eurasia plate during the past six million years to accreting more lithosphere on the North America plate 14 million years before that (see Table 2.3).

Rift propagation has been shown to be the asymmetry producing mechanism in the classic areas where asymmetrical accretion has been observed. A ridge rotation model was proposed to be the source of asymmetry in the Northeast Pacific (*Menard and Atwater* [1968]) but the asymmetry producing mechanism in their "Zed" area was later shown to be rift propagation (*Caress et al.* [1988], *Hey et al.* [1988]). Similarly, regional asymmetric spreading was suggested to be the cause of the asymmetrical accretion in the Australia-Antarctic Discordance (*Weissel and Hayes* [1971]) which is now understood to be caused by rift propagation (*Vogt et al.* [1983]; *Phillips Morgan and Sandwell* [1994]; *Christie et al.* [1998]). Furthermore asymmetrical accretion has been observed and attributed to rift propagation in the Juan de Fuca Area (*Shih and Molnar* [1975]; *Wilson et al.* [1984]), the Easter Microplate (*Naar and Hey* [1991]) and Galapagos (*Hey and Vogt* [1977]; *Hey et al.* [1980]; *Wilson and Hey* [1995]).

Figure 2.5 shows our profile 20 modeled using asymmetric spreading (the spreading rates and asymmetry are given in Tables 2.4 and 2.5, respectively). The fit of this model to the data is generally very good until 15.8 Ma when we use 53% asymmetry to the west (accreting more lithosphere to North America). This might partially be caused by the fact that we use best fitting spreading rates over all the profiles, instead of trying to fit spreading rates to individual profiles. However, the sense and amount of asymmetry are both required to change abruptly over a time scale of ~ 1 Ma (see Table 2.5 and Figure 2.6). If this is indeed asymmetric spreading we can think of no plausible explanation for why the ridge would behave in such a way. Instead, we take it as a strong evidence for propagating rifts, which cause abrupt changes in asymmetric accretion by discrete ridge jumps.

For the reasons discussed above we prefer the well documented asymmetry producing mechanism, rift propagation, over continuous regional asymmetric accretion.

2.7 Propagating rift magnetic modeling

2.7.1 How propagating rifts and ridge jumps relate

Figure 2.7 schematically shows a map view of a continuously propagating rift replacing a dying ridge. As it does so lithosphere is transferred from plate B to plate A, causing the lithospheric accretion to be asymmetric. Two pseudofaults on each plate offset the magnetic lineations and the magnetic fabric is rotated in the zone of transferred lithosphere. Comparing magnetic data which are collected along the spreading flowlines of ridges where propagation has and has not occurred will not show the same pattern of magnetic anomalies. In the case where propagation has occurred the magnetic anomalies will look as if a chunk of one plate has been transferred over to the other, thus shuffling the magnetic anomalies around. This process can be imitated by incorporating a ridge jump parameter, to move lithosphere from one plate to the other, when forward calculating the magnetic field from synthetic magnetized blocks. The ridge jump is treated as an instantaneous event in the magnetic models. A propagating rift and ridge jumps are closely related phenomena, looked at from different view points - the ridge jumps observed on individual profiles are caused by new rifts propagating quasi-orthogonally to the profiles (*Hey et al.* [1980]).

2.7.2 Prior results of *Hey et al.* [2010]

The new propagating rift model of *Hey et al.* [2010] shows that an alternative mechanism exists for the origin of the V-shaped ridges south of Iceland. *Hey et al.* [2010] found a self-consistent pattern of jumps that produces most of the observed asymmetric accretion and provides generally good fits to the magnetic anomalies. Each jump results from a propagating rift, traveling away from Iceland. The jump boundaries coincide with linear VSR boundaries, strongly suggesting the VSRs are propagating rift wakes. The data analysis in *Hey et al.* [2010], however, is imperfect as they al-

ways fit one ridge flank better than the other and they do not use the newest poles of rotation (C. DeMets, pers. comm., 2010) to impose a self-consistent spreading rate pattern. Their jump pattern is therefore rather a rough outline of what we observe, rather than a detailed history.

2.7.3 Assumptions

In our magnetic modeling we assume that the A-scarps are of the same age. The A-scarps are two tectonic-looking scarps, on either side of the ridge (Figure 2.5), facing away from the axis. Although they had been thought to be symmetric about the axis (*Vogt [1971]*), they are not the same distance away from the axis when measured along spreading flowlines (*Hey et al. [2010]*). How could two large scarps with similar characteristics, on either side of the ridge, be created at different times? The question is therefore not "why are they the same age", but "how could scarps be the same age yet different distances from the axis?" Within the bounds of the A-scarp there is more lithosphere on the Eurasia side than on the North America side and the asymmetry increases toward Iceland, as seen by the jump parameters in Table 2.6.

Our magnetic models have simple vertical polarity transition zones giving rise to imperfect fits to the central anomaly, specifically closer to Iceland where the central anomaly widens. Incorporating the geometry of outward displacement to our models could improve the fits to the central anomaly; one would need to define new spreading rates based on a pole of rotation corrected for outward displacement (C. DeMets, pers. comm., 2010).

2.7.4 Magnetic models

Figures 2.8-2.16 show the magnetic models for tracks 17-25 modeled with the spreading rates in Table 2.4 and the jump and magnetization parameters in Table 2.6.

Because the Reykjanes Ridge is a slow spreading ridge the magnetic blocks are

not as wide as the ones found over fast spreading ridges and the magnetic anomalies are therefore not as well defined as they lack the characteristics of the magnetic anomalies found over fast spreading ridges. Their resolution is, however, high enough to be approximated by a 2-D block model. To get more realistic fits to our data we apply the method of the contamination coefficient (*Tisseau and Patriat [1981]*), which makes the magnetized blocks narrower prior to calculating the model, thus suppressing the high frequencies. Our models use a contamination coefficient of 0.5-0.7, as seen in Table 2.6. As discussed above, this is equivalent to shrinking the horizontal scale of the magnetization block model by these factors and then forward calculating the magnetic anomaly.

The fit to the data is very good for profiles 19-25. These profiles are farther from the shelf than profiles 17 and 18 which are located in the transition zone between the shelf and the ridge. The fits to profiles 17 and 18 have however been significantly improved from the ones found in *Hey et al. [2010]* where specifically profile 18 was not fitted well.

The size of the jumps is constrained to ~ 0.2 km whereas the time of the jumps is constrained to ~ 0.5 Ma. As mentioned before we take our shiptracks as the flowlines of the ridge and that will result in an error in our pseudofault location of ~ 4 km. This error is comparable to the ~ 0.5 Ma error (0.5 Ma with a half spreading rate of ~ 10 km/Myr = 5 km). The difference in the flowlines should therefore not change our results. Note that one could split up a single jump to two jumps or more, if desired, but that increases the free parameters used in the modeling making it statistically less significant.

When modeling magnetic anomalies we start at the ridge axis and then move out to the ridge flanks because the younger parameters affect the older magnetic anomaly pattern. In order to figure out what size of jumps to put in the modeling for a specific time range we need to view magnetic anomalies that are a few million years older to

see how the younger jump affects the older portion of the magnetic data. For this reason we have only modeled our data out to ~ 15 Ma even though the data range is out to ~ 20 Ma, which explains some of the misfit to the data between ~ 15 Ma and ~ 20 Ma.

Table 2.7 shows the root mean square measure, for our magnetic models compared to the models of *Hey et al.* [2010]. In magnetic modeling most of the signal is a short wavelength and a small shift can produce a large residual, which explains the big difference in these two studies. The new models we present here reduce the r.m.s misfit compared to that of *Hey et al.* [2010] by the amounts shown in Table 2.7.

2.8 Results

Figure 2.17 shows time of jump versus distance away from Iceland compiled for profiles 17-25. Propagation has occurred both away from and toward Iceland, as indicated by the arrows, where lithosphere is transferred to Eurasia (blue dots) or North America (red dots). The size of jumps is indicated by the area of the dots. The southward propagators tend to extend all the way through the survey area while the northward propagators tend to be shorter and less pronounced. This is plausibly associated with a topographic gradient away from Iceland (*Searle et al.* [1998]) which the southward propagators need not overcome like the northward propagators. Also, the pattern of propagators is likely complicated by the growth and evolution of the axial volcanic ridges which form sub-normal to the spreading direction and can propagate to off-axis lithosphere (*Searle and Laughton* [1981]; *Parson et al.* [1993]; *Searle et al.* [1998]).

Figures 2.18 and 2.19 show snapshots of the evolution of the propagating rift history of the Reykjanes Ridge implied by the magnetic anomaly fits in Figures 2.10-2.16. Figure 2.18 shows a schematic evolution of the Reykjanes Ridge where our southernmost profile 25 is at $y=290$ km and the Reykjanes Peninsula is at $y=0$ km.

The y-axis runs perpendicular to the flowlines of the ridge so the positive azimuth of the x-axis is 100° clockwise from north. Pseudofaults of southward and northward propagating rifts are connected with blue and green lines, respectively. Figure 2.19 shows the evolution in a map view on top of satellite gravity (*Sandwell and Smith [2009]*). Pseudofaults of southward and northward propagating rifts are connected with solid and dashed lines, respectively.

We divide our survey area into two sub-areas; profiles 17 and 18 are located where the Reykjanes Ridge meets the Iceland shelf and the crustal accretion appears to be more complex than in the profiles to the south. Below we review the tectonic evolution of these two sub-regions as well as a detailed discussion of the methods used to produce the reconstruction snapshots.

2.8.1 Reconstruction movies

To reconstruct the plate spreading we cut the gravity grid between the span of profiles 17 and 25 and the timespan between 19 Ma and the time of the snapshot, on both ridge flanks. We found the exact location of the desired age on the gravity grid by putting a fictitious zero-distance jump at the snapshot time into Magellan for each of the profiles. We ran Magellan with the modeling parameters from Table 2.6 and spreading rates from Table 2.4 giving us an exact location (longitude, latitude coordinates) of the time-marker (that way we are not assuming symmetric accretion).

To rotate the gravity grid we use the stage poles we found earlier (Table 2.2). Note that these reconstructions are independent of the present day ridge axis location. The ridge axis location at each snapshot is therefore generally not exactly the same as the ridge axis location in present day.

2.8.2 Profile 19 and south

Clear history of propagating rifts is revealed for profiles 19-25 in Figure 2.17. Four southward propagating rifts extend through our entire survey area and can be traced back to Iceland. The oldest propagator left Iceland at 15 ± 0.2 Ma with an initial propagation rate of $\sim 300 \pm 50$ km/Myr. This one was referred to as the E-propagator in *Hey et al.* [2010] but here we call it Loki, after the mischievous Norse god. Loki transfers lithosphere to North America and is a southward propagator so the offset between the propagating rift and the dying rift would have been a right-stepping transform (or non-transform) offset. Loki stalls between profiles 21 and 22 for ~ 0.9 Ma (Figures 2.20(b),(c)) where we detect a small transform-like discontinuity in the gravity data, hereafter termed *ponsu-transform* (ponsu is an Icelandic prefix meaning itty-bitty) (see the ponsu-transform in the gravity in Figure 2.19(b)), and then it continues on with a propagation rate of $\sim 120 \pm 40$ km/Myr, transferring less lithosphere (~ 4 km vs. $\sim 5 - 8$ km) to North America (Figure 2.20). The right-stepping ponsu-transform was formed as Loki stalled (Figure 2.20(d)). The bend in Loki's pseudofaults caused by the pause in propagation is shown schematically in Figure 2.20(d) and is evident in Figures 2.18(a) and 2.19(a) where it is shown as a schematic bend rather than a tiny offset. Interestingly, Loki's pseudofaults coincide with a major escarpment (the E-scarp in *Vogt* [1971] and *Hey et al.* [2010]) on either side of the ridge.

Two small propagators are initiated at the ponsu-transform after Loki has propagated through the survey area. At $\sim 12.5 \pm 0.2$ Ma a northward propagator is initiated with a propagation rate of $\sim 80 \pm 10$ km/Myr (Figures 2.18(b), 2.19(b)). There is no evident pattern in the free air gravity (Figure 2.19(b)) coinciding with the pseudofaults of this propagator. At $\sim 10.9 \pm 0.2$ Ma a southward propagator is initiated with the propagation rate of $\sim 110 \pm 30$ km/Myr. The pseudofaults generated roughly follow the inward facing slope of a gravity ridge as seen in Figure 2.19(b).

An independent southward propagator is observed beginning on profile 17 at

$\sim 10.7 \pm 0.2$ Ma and stopping at profile 20 at $\sim 9.8 \pm 0.2$ Ma, propagating at a rate of $\sim 90 \pm 20$ km/Myr (Figure 2.18(b)). There is no evident pattern in the free air gravity (Figure 2.19(b)) coinciding with the pseudofaults of this propagator.

The second continuous propagator, Fenrir (after the monstrous wolf in Norse mythology), left Iceland at 10 ± 0.2 Ma with a propagation rate of $\sim 100 \pm 10$ km/Myr if we assume it traveled in a linear fashion. Fenrir corresponds closely to the C propagator in *Hey et al.* [2010] and transfers lithosphere from Eurasia to North America and re-organises the ridge by eliminating the ponsu-transform (Figure 2.19(c)). The offset between the propagating rift and the dying rift would have been right-stepping. The pseudofaults generated by Fenrir coincide with a prominent and well established gravity ridge on the North America plate and the inward facing scarp of a less well defined gravity ridge on the Eurasia plate.

Two northward propagators are initiated after Fenrir, both transferring lithosphere to North America. A new ponsu-transform forms between profiles 22 and 23 at which one propagator is initiated at $\sim 7.1 \pm 0.2$ Ma and the other one is stopped at $\sim 5.8 \pm 0.2$ Ma (Figure 2.18(d), 2.19(d)). The propagation rates of both are $\sim 60 \pm 20$ km/Myr. The pseudofaults of these two propagators follow minor gravity ridges on the North America plate, forming two small V's pointing to Iceland (opposite to the general trend of the V-shaped ridges pointing away from Iceland) (Figure 2.18(d), 2.19(d)), suggesting that the V-shaped ridges are complex features affected by small scale complexities such as propagators.

The third continuous propagator left Iceland at 6.5 ± 0.2 Ma with a propagation rate of $\sim 60 \pm 10$ km/Myr assuming a linear propagation rate. This propagator was referred to as the A-propagator in *Hey et al.* [2010] but here we call it Sleipnir after Odin's horse in Norse mythology. Sleipnir transfers lithosphere from North America to Eurasia, opposite to what *Hey et al.* [2010] found, and the offset between the propagating rift and dying rift would have been a left-stepping one. The pseudofaults

generated by Sleipnir coincide with the edges of the gravity ridge high (termed the A scarps in *Vogt [1971]* and *Hey et al. [2010]*) in the free air gravity (Figure 2.19(e)) on either side of the ridge. Similarly to Fenrir, Sleipnir eliminates the ponsu-transform and its pseudofaults coincide with a gravity ridge.

The fourth continuous southward propagator left Iceland at 4.0 ± 0.2 Ma with a propagation rate of $\sim 90 \pm 10$ km/Myr, if assumed to propagate linearly. This propagator was referred to as the A' propagator in *Hey et al. [2010]* but here we call it Hel, after Loki's daughter in the Norse mythology. Hel transfers lithosphere from Eurasia to North America, opposite to what *Hey et al. [2010]* found, and the offset between the propagating rift and dying rift would have been a right-stepping offset. The reason the lithospheric transfer pattern differs from that in *Hey et al. [2010]* is they hypothesised a propagator within the Brunhes whereas here we use a more time averaged axis.

2.8.3 Transitional profiles 17-18

The jump pattern in this area is more complex than in the southern part of the survey area. There are a few key observations about this area that should be mentioned. Firstly, the A-scarp curves out on the Eurasia plate north of profile 19 but not on the North America plate indicating that the crustal accretion in profiles 17 and 18 is different from the area to the south where the A-scarp is much more linear. Secondly, a circular structure interpreted as a central volcano (*Höskuldsson et al. [2010]*), is apparent from the free air gravity at the ridge axis in profiles 17 and 18 (Figure 2.21) which differs considerably from the southern profiles where the free air gravity shows lineations subparallel to the ridge. A third observation is that the pattern of the free air gravity on the ridge flanks changes drastically on profiles 17 and 18. A long gravity low along profile 18 on North America is present and the gravity ridges on the Eurasia side are not detectable. These observations suggest a different and a more

complex crustal accretion process along profiles 17 and 18 compared to the profiles to the south.

The most prominent differences on profiles 17 and 18 are the relatively enormous jumps (13-15 km) occurring at ~ 6 Ma transferring lithosphere from Eurasia to North America. They coincide with a gravity low on the North America plate (Figure 2.19(f), 2.21) but interestingly jumps of this size are not observed farther south.

Furthermore, there are several jumps we have not been able to attribute to a propagator. These jumps need to be there in order for the magnetic models to produce good fits to the anomaly data. As mentioned before, it is often plausible to take a jump and break it down to several smaller ones. Also, by reducing the size of one jump an adjacent jump can be made bigger (or smaller if it is transferring lithosphere in the opposite direction) and thus the amount of asymmetry is kept constant.

The sense of asymmetry in this area is also distinctly different from the rest of the survey area. The first three jumps on these two profiles transfer ~ 15 km of lithosphere to Eurasia within the first 4 Ma compared to 0.8-3.4 km for the southern profiles, explaining the curving of the A scarp on the Eurasia plate.

We observe Loki, Fenrir and Sleipnir in our magnetic models for profiles 17 and 18 but we do not observe Hel, although as stated above we think that it propagated through profiles 17-25. The amount of lithosphere that is being transferred to North America by Hel is very little (~ 1 km) and the magnetic signal arising from the propagation might therefore be contaminated by complex crustal accretion processes (e.g. increased outward displacement) specifically because the A scarp on the Eurasian plate curves out right at profiles 17 and 18.

2.9 Discussion

We provided generally excellent fits to our off-shelf magnetic profiles that are greatly improved over the ones in *Hey et al.* [2010], particularly profiles 17 and 18, better establishing the rift propagation history on the Reykjanes Ridge. A striking new result is that propagating rifts can propagate north toward Iceland which would be counter intuitive for many because of the topographic gradient away from Iceland. A pulsing plume explanation for the origin of the VSRs (e.g. *Vogt* [1971]; *Vogt et al.* [1980]; *Smallwood and White* [1998]; *Jones et al.* [2002]) has been taken as fact for the past 40 years but *Hey et al.* [2010] suggested that the origin of the VSRs is rift propagation. *Hey et al.* [2010] pondered on whether the plume pulses could drive the propagators. If plume pulses drive propagators they would all be southward propagating. As indicated by our magnetic models, northward propagators exist, and they would certainly not be driven by plume pulses.

The northward propagators tend to be shorter (crossing only a few profiles) and not as pronounced as the majority of the southward propagators (crossing all of our profiles). If Iceland plume pulses drive the southward propagators then two sets of mechanically different propagators exist: well established southward propagators (Loki, Fenrir, Sleipnir and Hel) driven by plume pulses re-organising the Reykjanes Ridge eliminating ponsu-transforms, and shorter rift propagations driven by something else.

The VSRs are not simple southward pointing Vs. Figure 2.21 shows the pseudo-fault and failed rift pattern predicted by our magnetic models in relation to Iceland superimposed on free air gravity (*Sandwell and Smith* [2009]). Pseudofaults of southward and northward propagating rifts are connected with solid and dashed lines, respectively, and failed rifts are plotted as red dots. If the VSRs are plume pulses we would expect to see linear symmetrical gravity ridges subparallel to the Reykjanes Ridge but counter to that prediction the gravity ridges are not symmetric about the ridge axis. Their amplitude is greater on the North America plate where the majority

of the failed rifts are located and there is a gap in the gravity ridges between profile 19 and the shelf edge which we explain by a fundamental difference in propagation history north and south of profile 18. *Jones and MacLennan [2005]* noted asymmetry of the Iceland shelf about the Reykjanes Ridge which can be seen in the free air gravity. The shelf reaches ~ 50 km further south on North America compared to Eurasia supporting the overall asymmetric accretion behaviour of the Reykjanes Ridge. A pulsing plume would not cause asymmetric accretion but rift propagation would so something would have to be added to the pulsing plume hypothesis to produce the observed asymmetry.

The two ponsu-transforms we have identified independently with our magnetic models are observable in the free air gravity. The older ponsu-transform was active 8-14 Ma ago and was located between profiles 21 and 22 (Figure 2.17). A discontinuity in the gravity ridges is seen in the reconstruction snapshot at 8.85 Ma (Figure 2.19(b)) as a linear feature paralleling the flowline of the Reykjanes Ridge. On the North America side it is a low and on the Eurasia side it is a high. Two propagators originate from this ponsu-transform and the Loki propagator stalled here for ~ 0.9 Ma (Figure 2.20) causing the amount of lithosphere Loki transferred to the North American plate to decrease. As Fenrir propagated south the older ponsu-transform was eliminated. The younger, and smaller, ponsu-transform was active 5-7 Ma ago and was located between profiles 22 and 23. After Fenrir propagated down the survey area a well established gravity ridge subparallel to the Reykjanes Ridge formed on North America (Figure 2.19(c)). The younger ponsu-transform formed a little later (Figure 2.19(d)) from which one northward propagator was initiated at ~ 7.1 Ma and at which a different one was eliminated at ~ 5.8 Ma.

Outward displacement has been reported to increase on the Mid-Atlantic Ridge from the Azores to the Reykjanes Ridge (*DeMets and Wilson [2008]*). Outward displacement affects the location of pseudofaults acquired from magnetic modeling

and generally the greater the outward displacement the farther out the pseudofaults should be. The pseudofaults in our magnetic modeling therefore appear to be closer to the ridge by the value of the outward displacement. For the Reykjanes Ridge that value could be as high as $\sim 5\text{-}6$ km (*DeMets and Wilson [2008]*).

The contamination coefficient is an indicator of smoothness of the magnetic data. Table 2.6 shows the contamination coefficient we used for profiles 17-25. The smaller it is the more suppressed the high frequencies are in the data and therefore small reversals are less detectable. There is not a clear gradient or change along the ridge in the contamination coefficient, suggesting that small scale accretion complexities are present independent of distance from Iceland.

2.9.1 Unresolved puzzles

The results of the magnetic modeling suggests many more questions. These mostly come from observations we cannot explain by our present rift propagation model.

- *Bad fits at the end of profiles.* As mentioned above we stop our magnetic modeling at ~ 15 Ma because of the limitation of our data coverage. A large scarp can be seen in the bathymetry of profiles 18-25 at $\sim 210\text{--}220$ km distance from the axis on the North America plate and if we had data coverage ~ 50 km further out we would be able to model whatever is going on. Greater data coverage would further the understanding of the origin of the V-shaped ridges and the asymmetrical accretion of the Reykjanes Ridge.
- *The flowline-parallel gravity low on profile 18.* Why would a gravity low persist for millions of years in one place on the plate boundary and only be noticeable on one ridge flank (the North American flank, see Figure 2.21)? This low is not an artifact in the satellite gravity data because it is observable in the shipboard gravity data as well (Figure 2.9).

- *Large jumps not extending all the way through our survey area.* The two ~ 15 km jumps that transfer lithosphere to the North American plate at ~ 6 Ma in profiles 17 and 18 are not traceable further south. The zone of transferred lithosphere located on the North America plate coincides with a very deep gravity low (Figure 2.21). Something must have caused the plate boundary to shift abruptly toward the east at ~ 6 Ma when the big jumps occurred, and then slowly relocate back west, through three smaller jumps which transferred lithosphere back to the Eurasian plate (Figures 2.17,2.21). This same evolution is not observed to the south.
- *Propagation direction.* Rift propagation has most commonly been observed to occur away from hotspots (*Hey and Vogt [1977]; Delaney et al. [1981]; Vogt et al. [1983]; Schilling et al. [1985]; Naar and Hey [1991]; Wilson and Hey [1995]*), probably because of gravity spreading stresses caused by the topographic gradient away from hotspots (*Phipps Morgan and Parmentier [1985]*), but rift propagation has also been recorded to occur toward hotspots (*Wilson and Hey [1995]; Barckhausen et al. [2008]; Mihut and Müller [1998]*) into a hotter, weaker lithosphere. It is therefore still poorly understood what controls the direction of propagation.
- *Fundamental difference between transitional and southern profiles.* Both the free air gravity anomaly and jump pattern in the transitional profiles (17 and 18) differ highly from the southern profiles. A fundamental difference must exist in the crustal accretion processes between these two areas. The transitional profiles border the Iceland shelf which might be the cause of complex crustal accretion processes.
- *Asymmetric gravity amplitude about the Reykjanes Ridge.* Sediments which derive from Iceland and blanket the Eurasia plate would tend to elevate the

free air gravity signal and reduce the relative amplitudes between the troughs and the ridges. If the crustal thickness and bathymetry are the same on the ridge flanks we would expect to see higher free air gravity amplitudes on the Eurasia plate, which we do not for reasons still unknown. The most notable difference is in the gravity ridge that coincides with the pseudofaults of the Fenrir propagator (Figure 2.21).

2.10 Speculations

The Eastern and Northern Volcanic Zones in Iceland have been propagating away from the Iceland hotspot (*Sæmundsson* [1979]; *Schilling et al.* [1982]; *Hardarson et al.* [1997]; *Einarsson* [2008]) and propagation to the southwest Iceland shelf has been hypothesised to match observed magnetic anomalies on the Iceland shelf (*Kristjánsson and Jónsson* [1998]). Figure 2.21 shows the rift propagation on the Reykjanes Ridge suggested by this study in relation to Iceland. Dash-dotted lines on Iceland indicate locations of paleo-spreading centers (*Sæmundsson* [1974]). We have suggested a relation between the paleo-spreading centers and our results with dotted lines extending from Iceland down to our survey area. The gravity scarp following the North America pseudofault of Loki can be traced on to the shelf to the paleo-spreading center of Vestfirðir as a gravity escarpment. As the Vestfirðir paleo-spreading center became extinct a propagator might have been initiated because a local change in the tectonic geometry limited the supply of magma down the ridge, as proposed by *Hardarson et al.* [1997]. Figure 2.17 shows that by linearly extrapolating the Loki propagator to Iceland it would have left at ~ 15 Ma which coincides roughly with the extinction age of the Vestfirðir paleo-spreading center dated at ~ 15 Ma (*Sæmundsson* [1974]). Similarly, as the Snæfellsnes-Skagi paleo-spreading center became extinct a propagator might have been initiated with a pseudofault coinciding with the gravity

step indicated by the dotted line from the Snæfellsnes peninsula. The North America pseudofault of Sleipnir could be linked to this event. Figure 2.17 shows that by linearly extrapolating the Sleipnir propagator to Iceland it would have left at ~ 6.5 Ma which coincides roughly with the extinction age of the Vestfirðir paleo-spreading center dated at ~ 7 Ma (*Sæmundsson* [1974]).

If these speculations are correct then we can predict the existence of an unknown paleo-spreading center in Iceland between the Vestfirðir and Snæfellsnes-Skagi paleo-spreading centers which would have become extinct at ~ 10 Ma when Fenrir left Iceland (Figure 2.17).

Based on the gravity patterns associated with the northward propagators, which began after Fenrir propagated through the survey area, we predict that other northward propagators will be discovered south of our survey area where similar complicated gravity ridges, wider to the north than to the south, exist (Figure 2.21).

2.11 Conclusions

We have attempted to accurately model the Reykjanes Ridge magnetic anomalies south of Iceland. These models strongly suggest rift propagation both toward and away from Iceland, explaining the observed asymmetric lithospheric accretion. Four major southward rift propagations extend through our entire survey area and all but the second most recent propagator transfer lithosphere to the North American plate. Several small scale rift propagations are observed, including northward propagations suggesting that the evolution of axial volcanic ridges complicates the rift propagation evolution and poses a new problem for the pulsing plume hypothesis. If plume pulses drive southward propagators, two mechanically different propagators must exist. We find that there is a major difference in the crustal accretion asymmetry between the area immediately off the Iceland shelf and further south, both in the rift propagation

pattern and the free air gravity lineations. Furthermore, we identify two small offset features coined ponsu-transforms, from which rift propagation is sometimes initiated and sometimes eliminated. The pattern of the VSRs is not symmetric or identical about the Reykjanes Ridge and the VSRs are not linear continuous features. Also, we have identified northward pointing Vs in the free air gravity and a major flowline-parallel free air gravity low, re-enforcing the conclusion that the VSRs are not simple features. Our rift propagation model provides excellent fits to magnetic data and provides a self-consistent model for the evolution of the Reykjanes Ridge during the past 15 Ma.

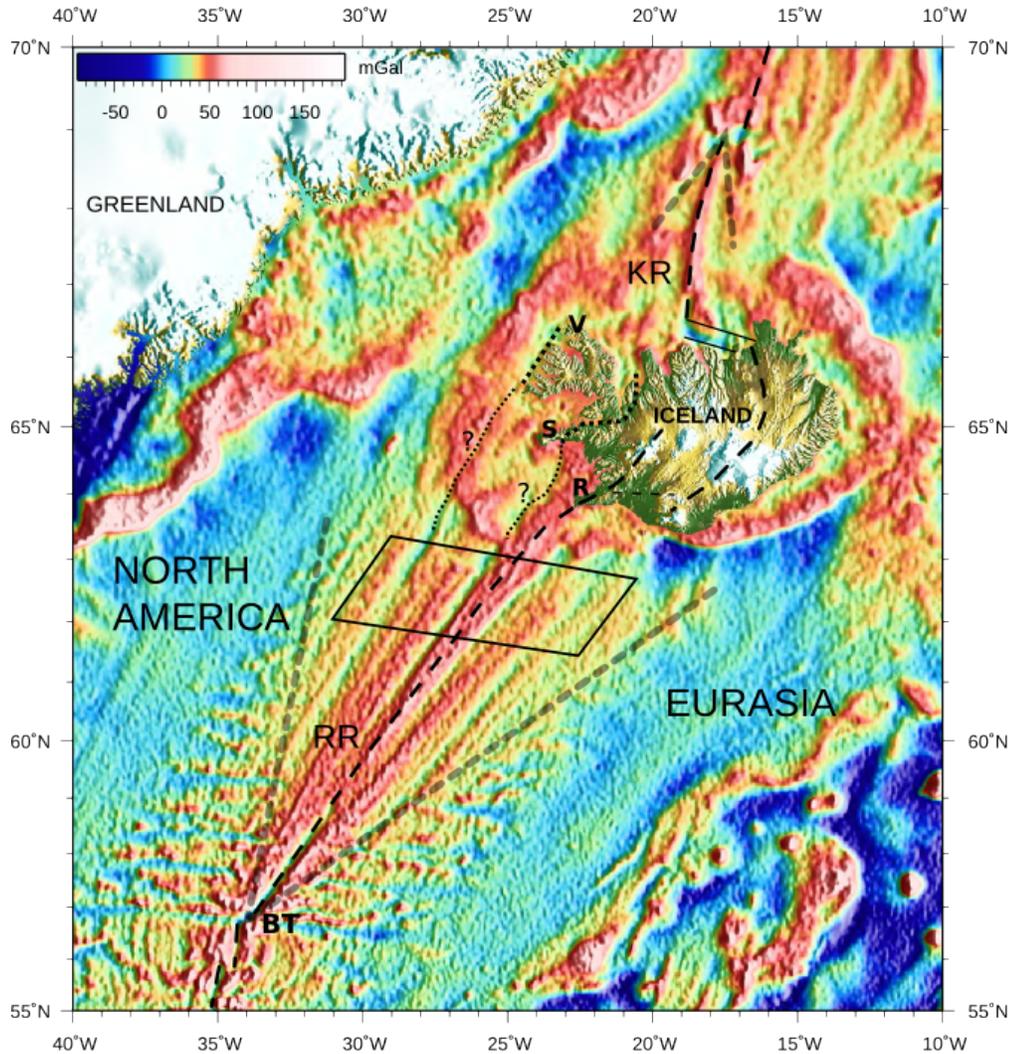


Figure 2.1: Satellite gravity and tectonic boundaries near Iceland (*Sandwell and Smith [2009]*) with gridded land topography superimposed. Heavy black dashes show Reykjanes Ridge (RR), Kolbeinsey Ridge (KR), and their extensions through Iceland. The VSRs we reinterpret here are the ridges and troughs slightly oblique to the Reykjanes Ridge axis enclosed by the southward pointing gray dashed V. Box shows location of profiles 17–25. Heavy dotted lines are paleo-spreading centers on Iceland and less heavy dotted lines show possible extensions down to our survey area. TFZ, Tjörnes Fracture Zone; V, Vestfirðir; S, Snæfellsnes; R, Reykjanes Peninsula; BT, Bight Transform. Modified from *Hey et al. [2010]*.

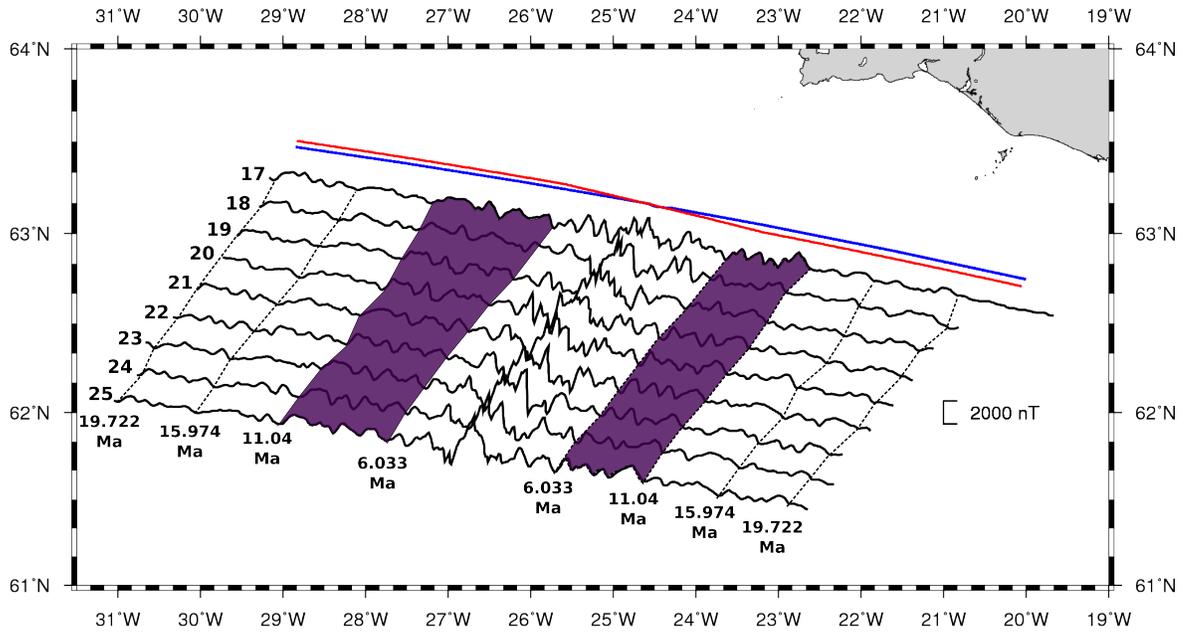


Figure 2.2: Magnetic data from the survey box in Figure 2.1. Dashed lines are magnetic anomalies used to define the new spreading rates for our magnetic modeling. Note the asymmetry of the anomalies, specifically there is more lithosphere between 6.033 Ma (chron 3r0 in *Lourens et al.* [2004]) and 11.04 Ma (chron 5n.2no in *Lourens et al.* [2004]) on North America than on Eurasia (shaded regions) and that asymmetry is independent of the ridge axis picks. Our shiptracks for profile 16 (blue line) and flowlines of the ridge predicted by the Eurasia-North America rotation parameters of DeMets (pers. comm., 2010) (red lines) are shown. Our shiptracks are nearly parallel to the predicted flowlines and the difference between these two is negligible for magnetic modeling purposes. Profile numbers indicate the location of our profiles (17-25).

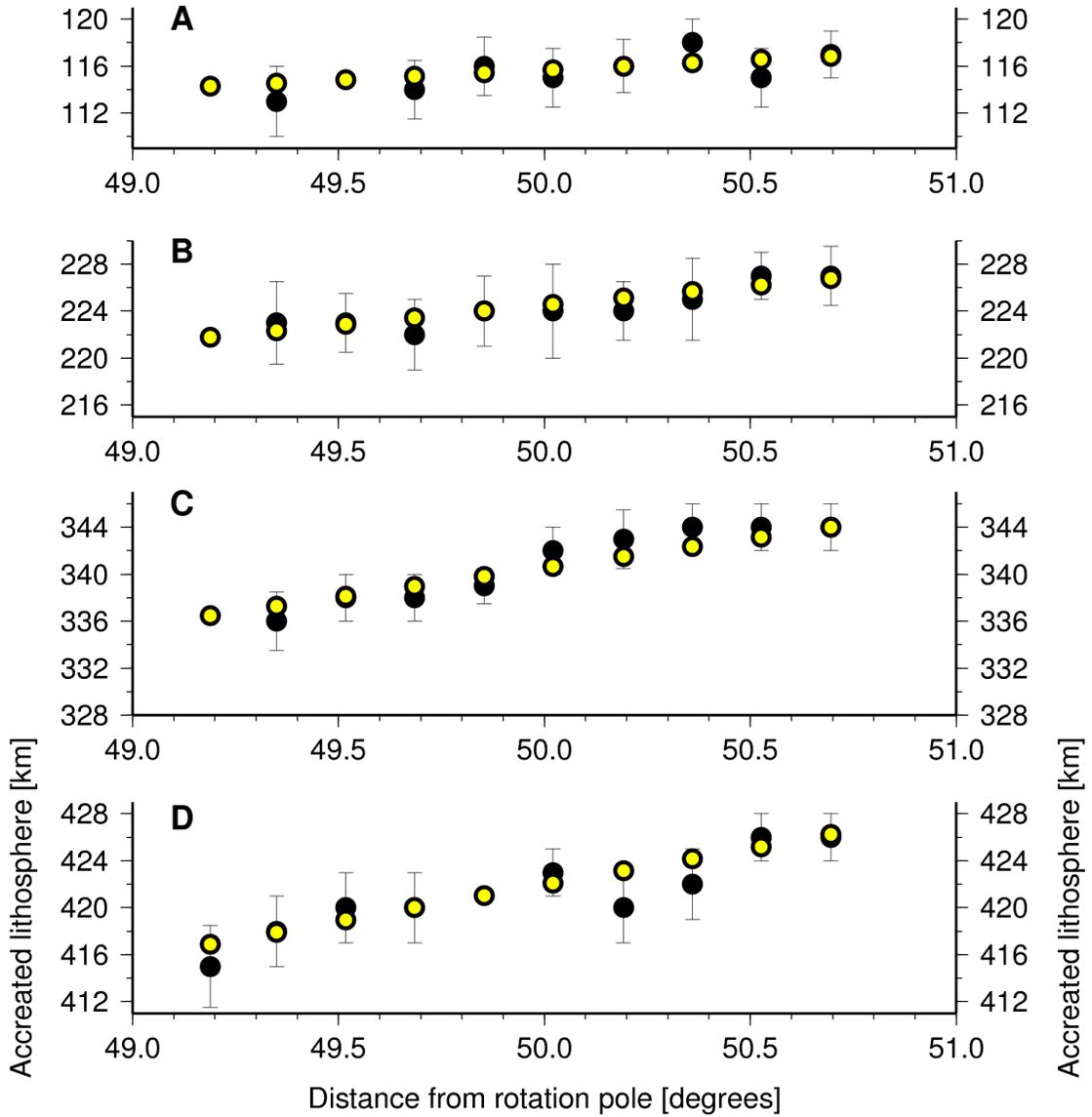


Figure 2.3: Total amount of accreted lithosphere (from Table 2.3) (black dots with error bars) and lithospheric accretion predicted by spreading rates from Table 2.4 (yellow dots) vs. distance from rotation pole (from Table 2.2) at **A.** 6.033 Ma, **B.** 11.04 Ma, **C.** 15.972 Ma, and **D.** 19.722 Ma for tracks 17 (closest to pole) to 25 (farthest from pole).

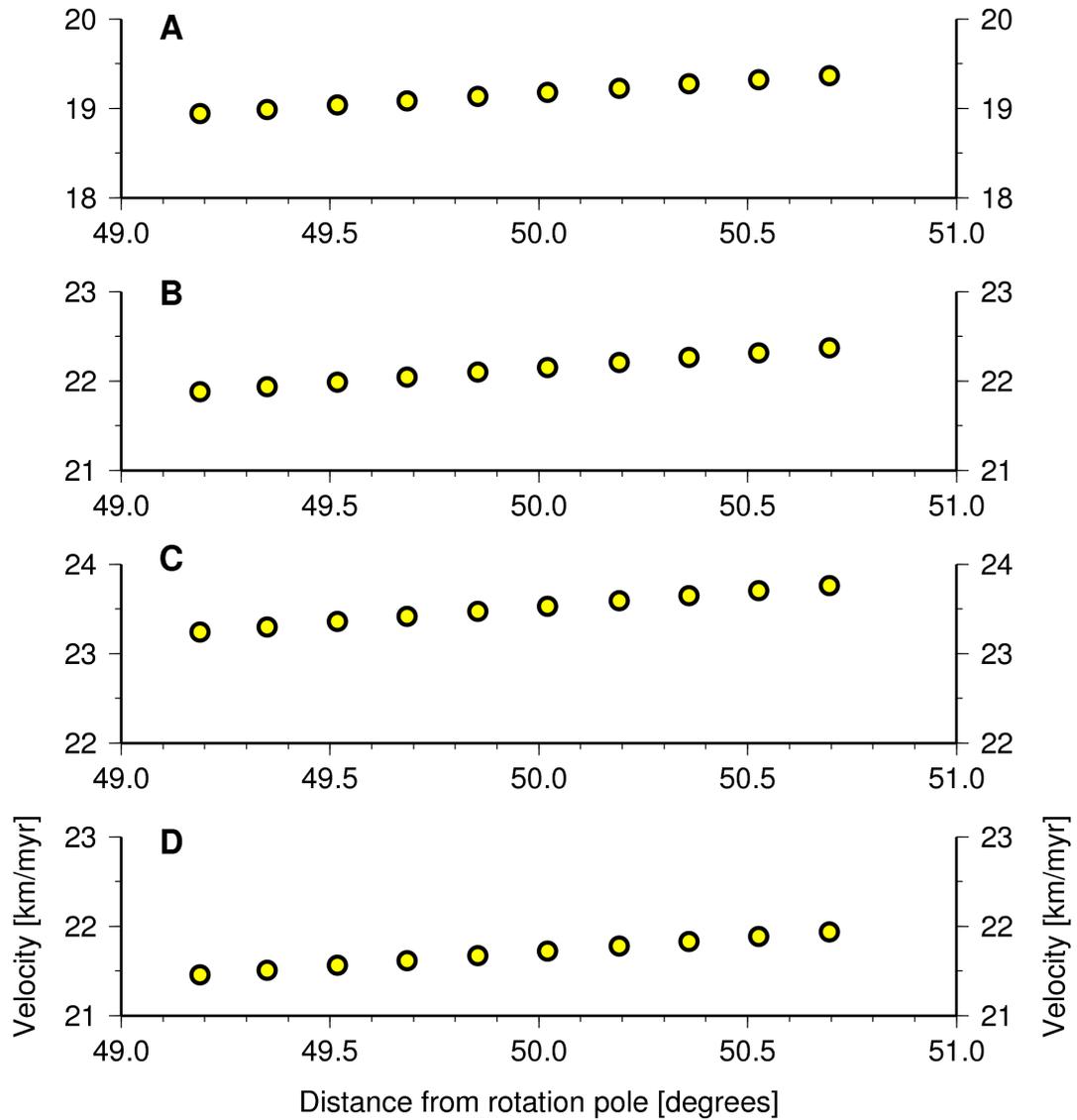


Figure 2.4: Spreading rates (from Table 2.4) vs. angular distance from rotation pole (from Table 2.2) for **A.** 0-6.733 Ma, **B.** 6.733-11.04 Ma, **C.** 11.04-15.974 Ma, and **D.** 15.974-19.722 Ma for tracks 17 (closest to pole) to 25 (farthest from pole). Yellow dots are spreading used in this study

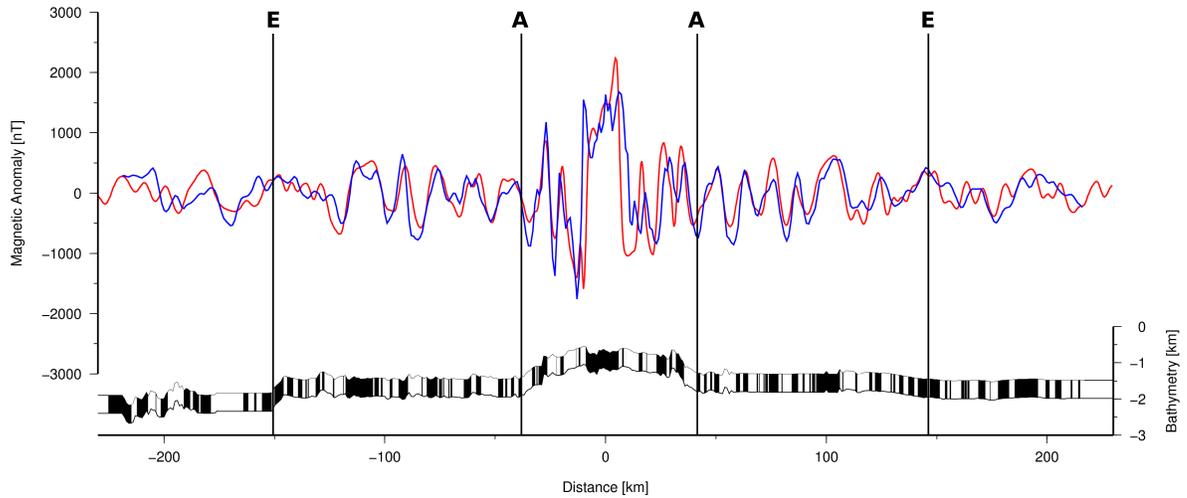


Figure 2.5: Profile 20 modeled with spreading rates shown in Table 2.4 and abrupt shifts in both magnitude and sense of asymmetry shown in Table 2.5. Blue is the data and red is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Note that the bathymetric A scarps and the E scarps are the same age on each side of the ridge (as indicated by the magnetic reversal sequence). Compare this fit with profile 20 modeled with ridge jumps in Figure 2.11, where the fits are similar but mechanism more plausible.

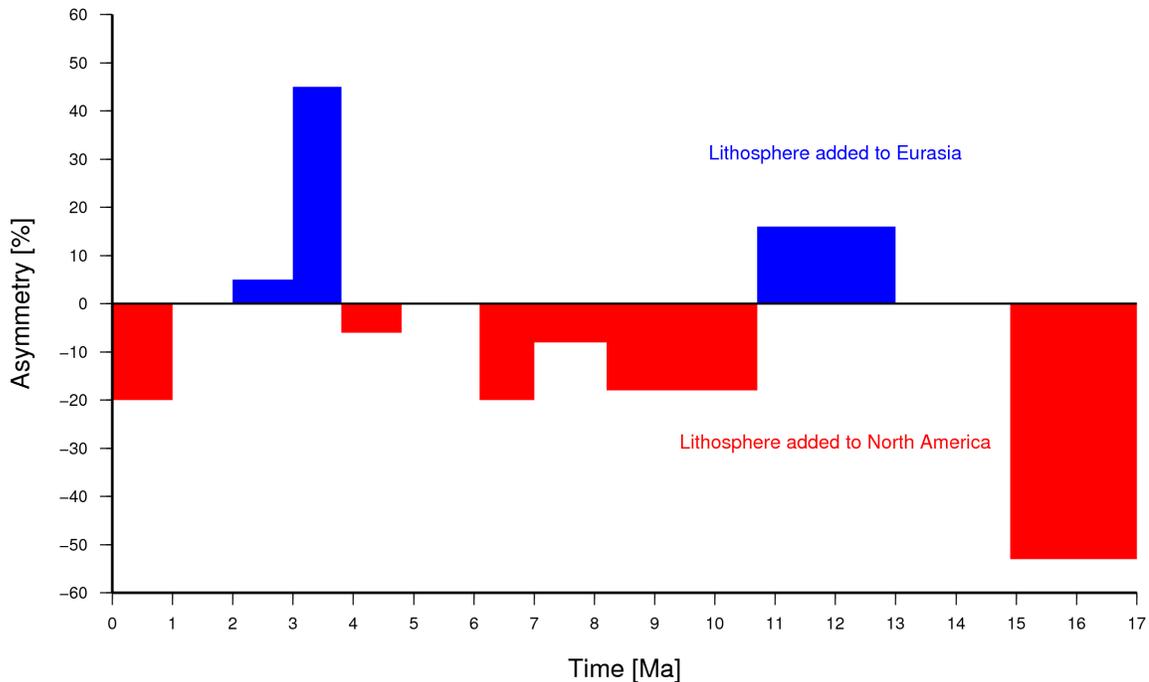


Figure 2.6: A schematic representation of the sense of asymmetric accretion for the model of profile 20 shown in Figure 2.5. The asymmetry values are given in Table 2.5. Positive and negative asymmetries represent more lithosphere added to the Eurasian plate and the North American plate, respectively.

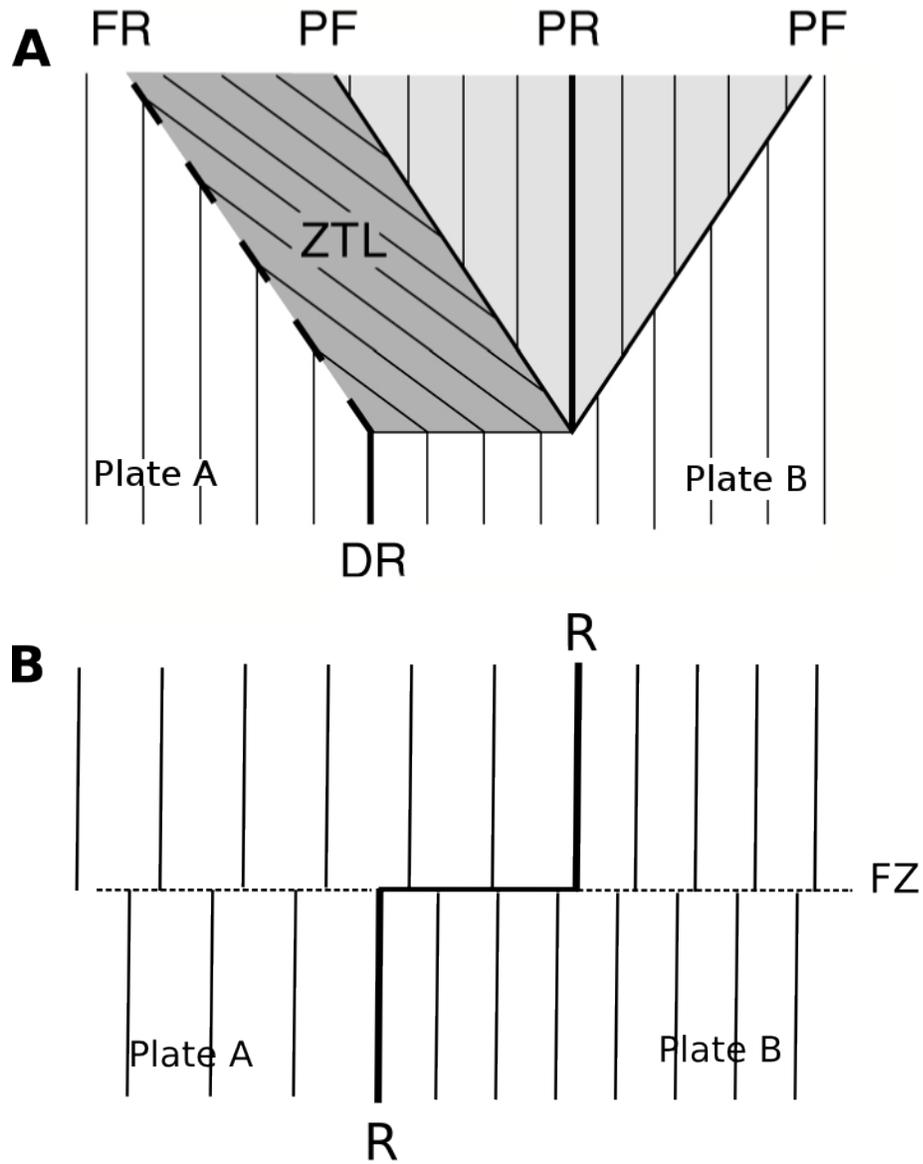


Figure 2.7: Schematic comparison between two asymmetry-producing mechanisms, **A.** propagating rift and **B.** asymmetric spreading. **A.** map view of a propagating rift (PR) after *Hey et al.* [1980]. The PR propagates down and replaces the dying rift (DR), producing two pseudofaults (PF), a failed rift (FR) and zone of transferred lithosphere (ZTL) where lithosphere was moved from plate B to plate A, producing asymmetric accretion. Thin vertical lines are isochrons, rotated in the ZTL. A cross-sectional profile perpendicular to the strike of the ridge would show the failed rift outside the pseudofault on plate A. **B.** Two asymmetric spreading ridge segments (R) are offset by a transform fault. Asymmetric spreading produces no V-shaped pattern. Isochrons are farther apart on plate A than plate B because of asymmetric spreading. Thin vertical lines are isochrons, offset by a fracture zone (FZ).

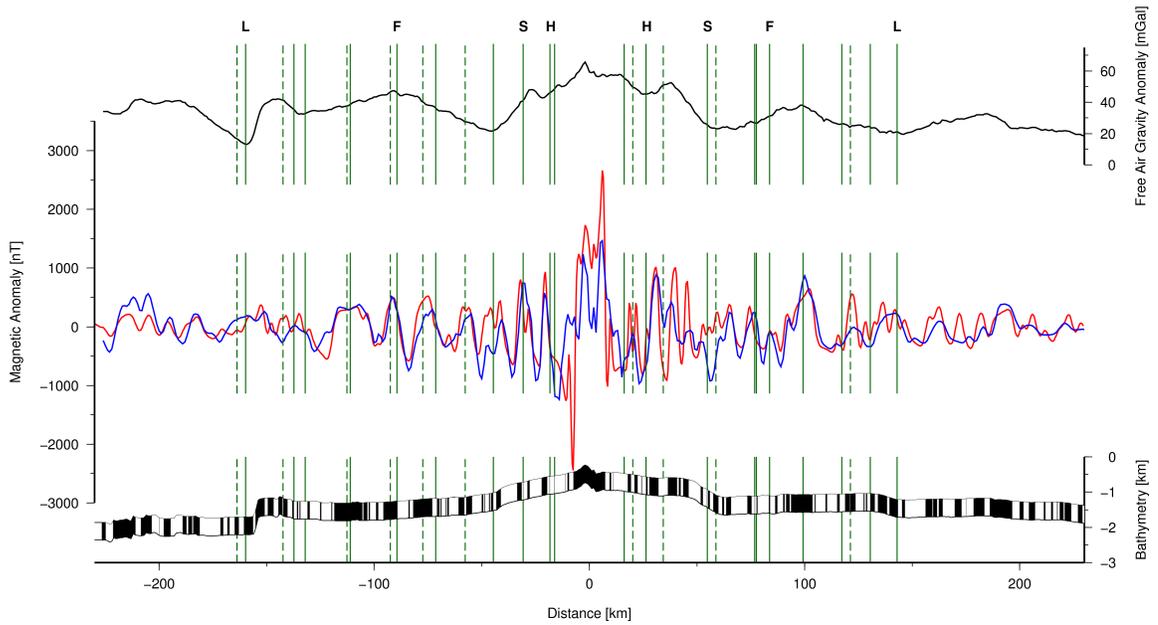


Figure 2.8: Profile 17 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

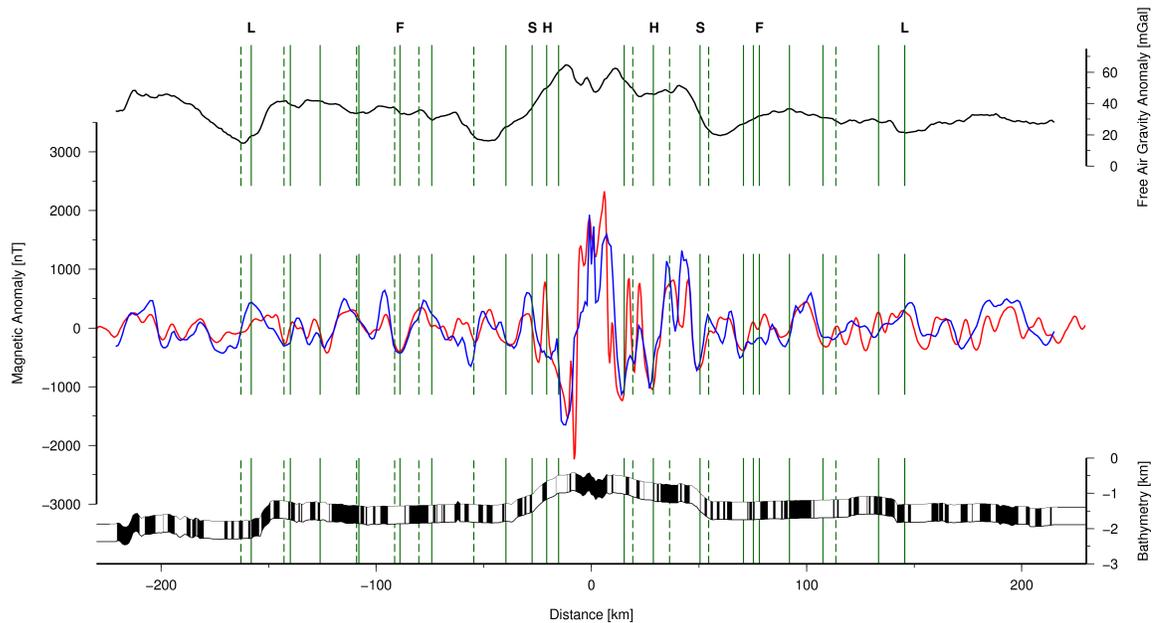


Figure 2.9: Profile 18 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Note that there are not pronounced ridges in the free air gravity on the North American side (negative distance from the ridge) as seen in the profiles to the north and south. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

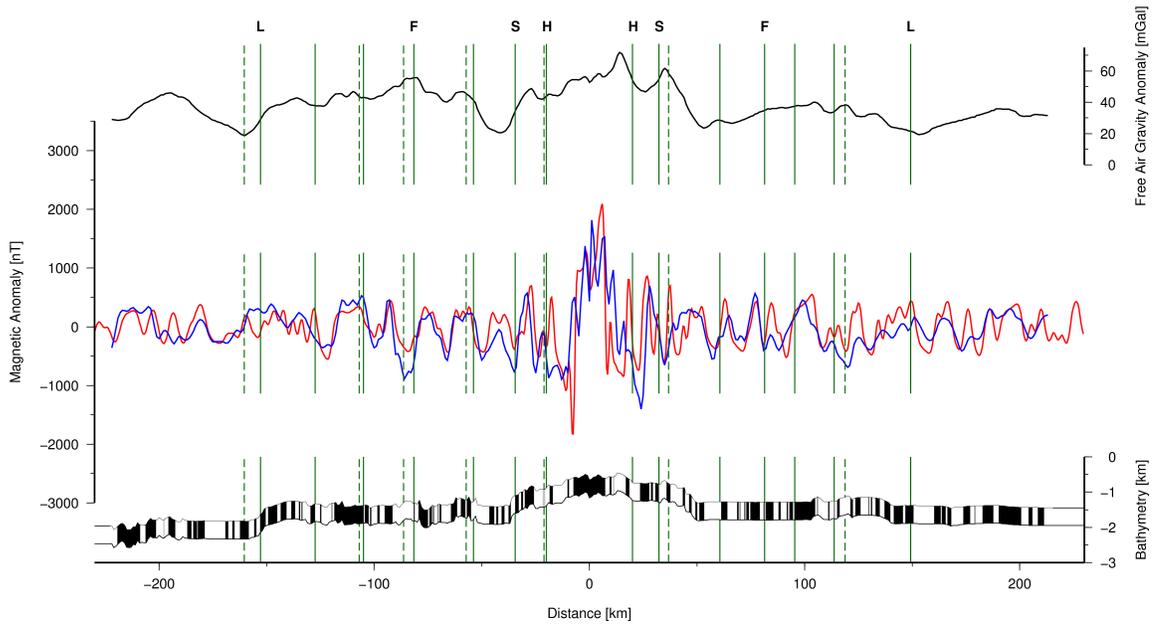


Figure 2.10: Profile 19 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

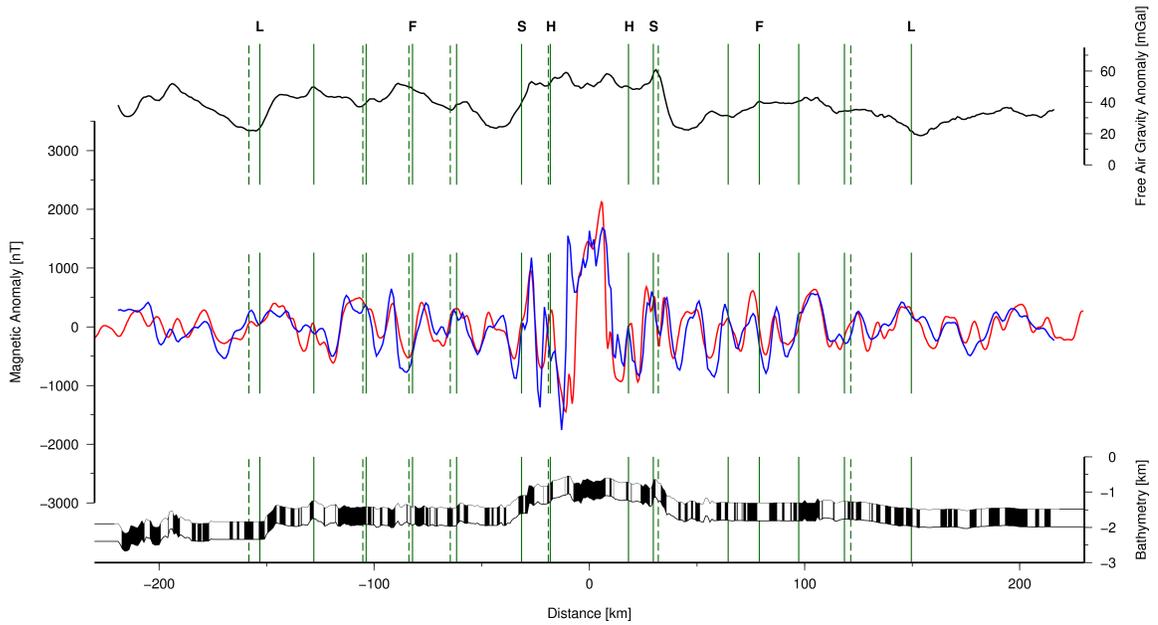


Figure 2.11: Profile 20 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

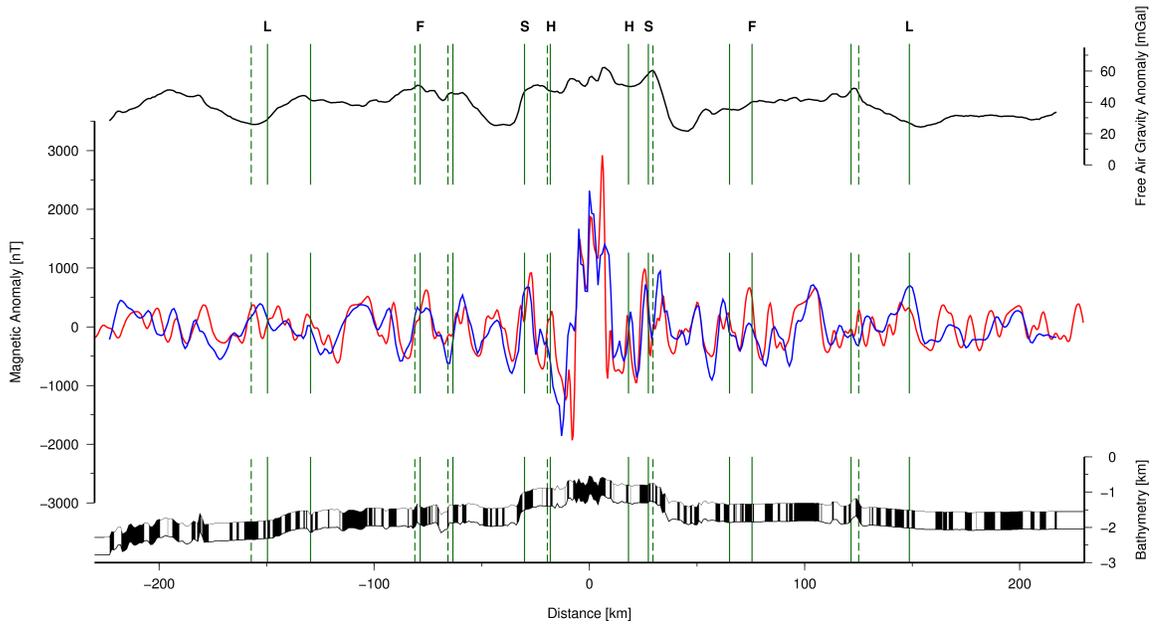


Figure 2.12: Profile 21 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

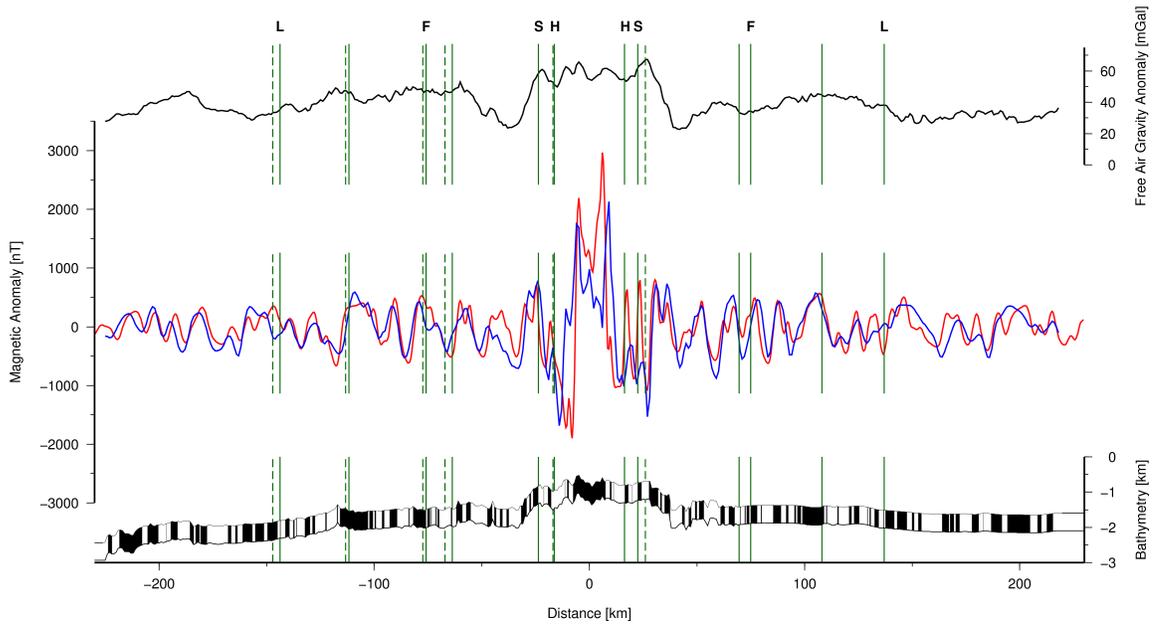


Figure 2.13: Profile 22 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

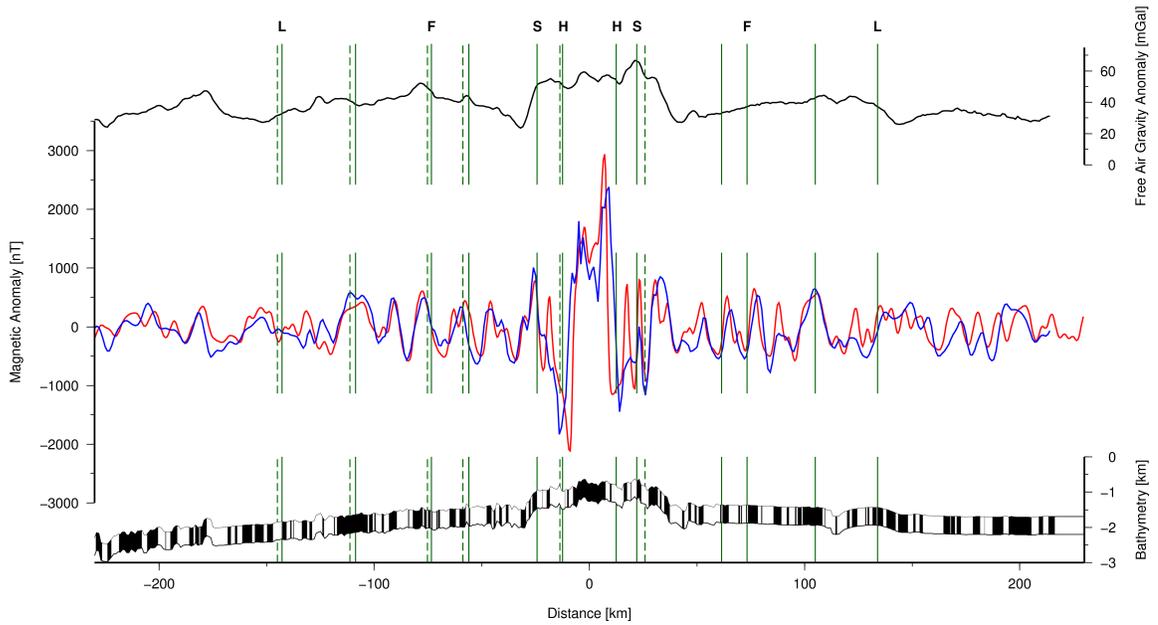


Figure 2.14: Profile 23 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

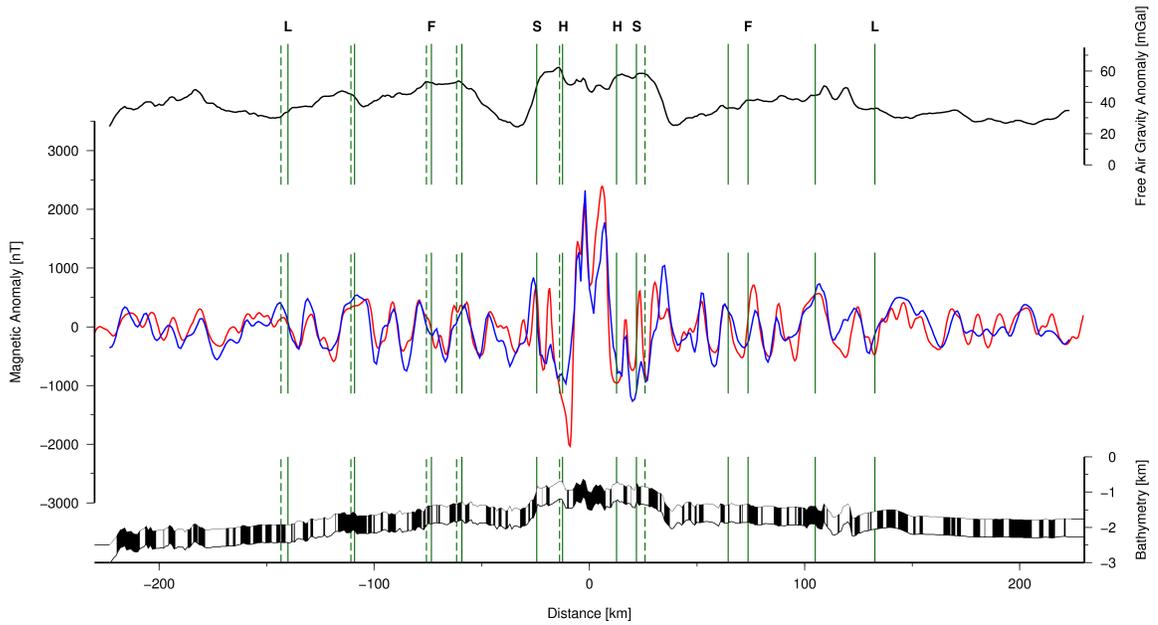


Figure 2.15: Profile 24 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

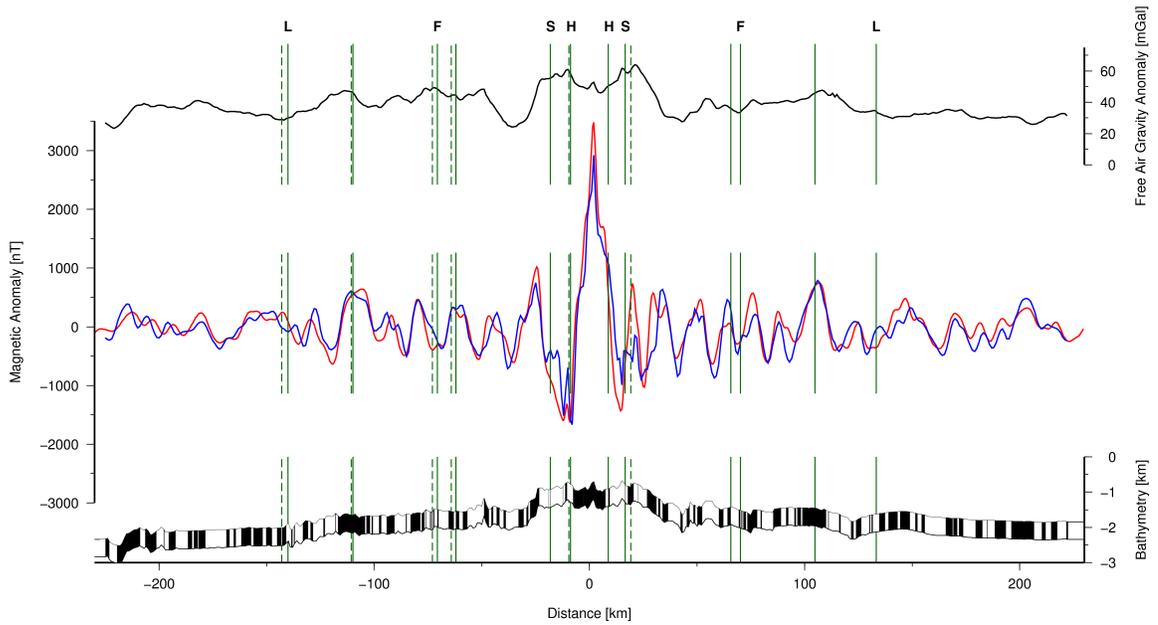


Figure 2.16: Profile 25 modeled with the jump and magnetization parameters given in Table 2.6 and spreading rates from Table 2.4. Black is shipboard free air gravity, red is the magnetic anomaly data and blue is the model. Black and white boxes are normal and reversed magnetized blocks, respectively, following the bathymetry. Green solid and dashed vertical lines are pseudofaults and failed rifts, respectively. H=Hel, S=Sleipnir, F=Fenrir, and L=Loki propagators.

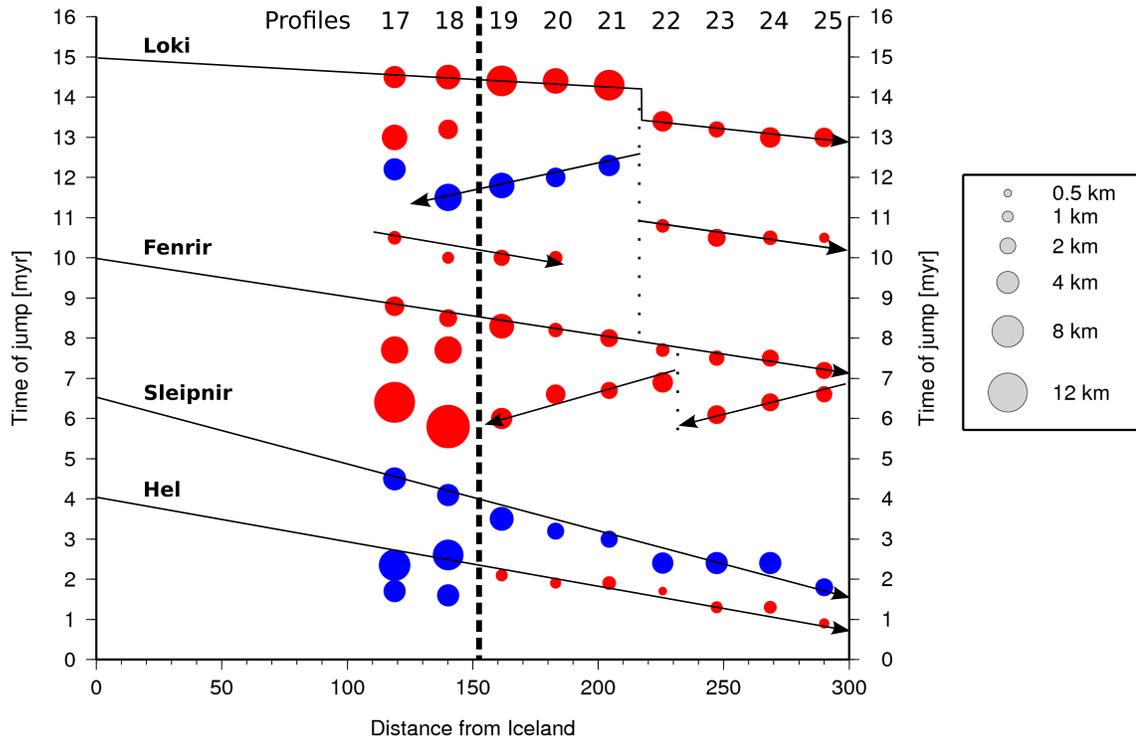


Figure 2.17: Time of jump from the magnetic models vs. distance from the Reykjanes Peninsula (RP)(at $63.67^{\circ}\text{N}/22.75^{\circ}\text{W}$) on Iceland. Red and blue circles correspond to jumps transferring lithosphere to North America and Eurasia, respectively. Arrows show direction of propagation. Solid lines are a linear interpolation of the southward propagating rifts extrapolated to the RP. Heavy dashed line separates the transitional profiles 17 and 18 from profiles 19 and south. Dotted lines are the locations of the two ponsu-transforms. Propagation rates are $\sim 300 \pm 50$ km/Myr, $\sim 120 \pm 40$ km/Myr, $\sim 100 \pm 10$ km/Myr, $\sim 60 \pm 10$ km/Myr and $\sim 90 \pm 10$ km/Myr for Loki before the ponsu-transform, Loki after the ponsu-transform, Fenrir, Sleipnir and Hel, respectively.

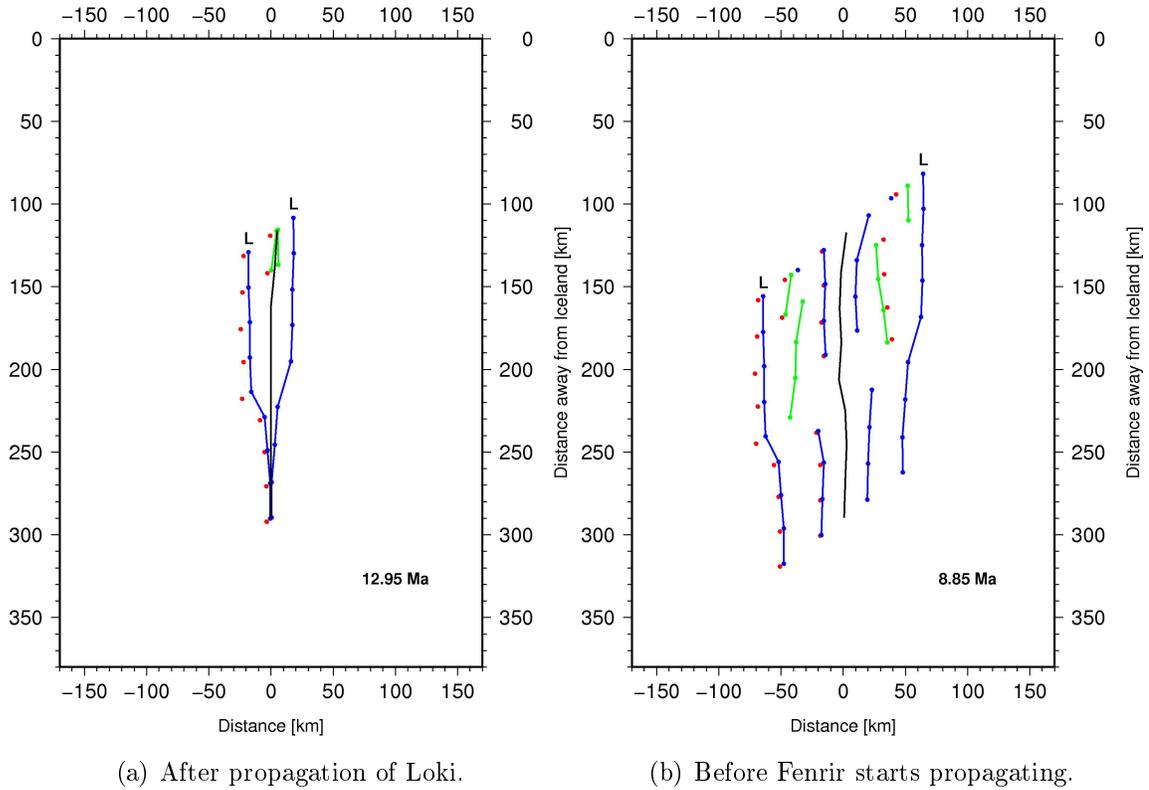
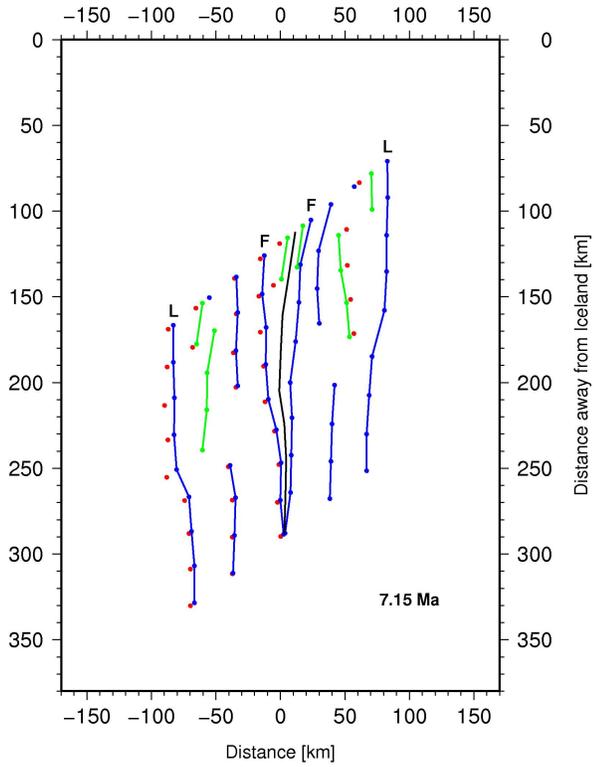
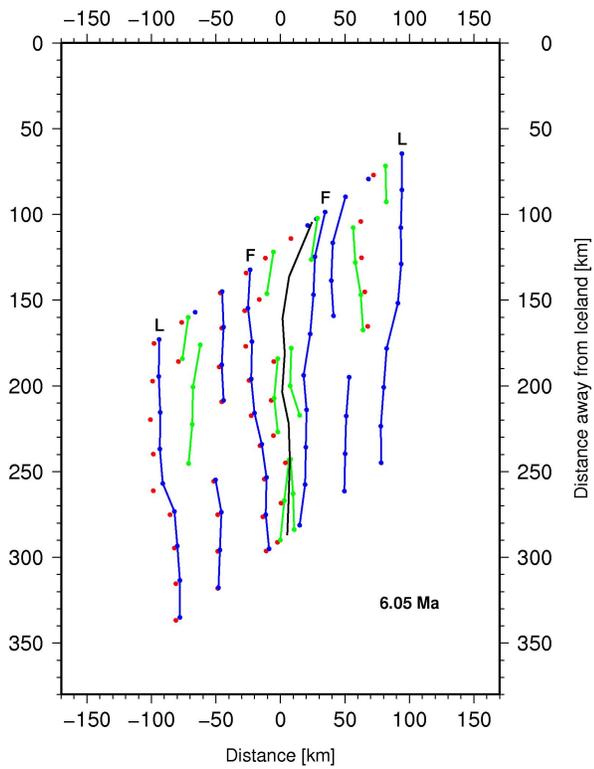


Figure 2.18: Cartoon showing the evolution of the Reykjanes Ridge. Horizontal axis is distance away from ridge and vertical axis is distance from the Reykjanes Peninsula. Green lines connect the pseudofaults of northward propagators, blue lines connect the pseudofaults of southward propagators, and red dots are failed rifts. Time of each snapshot is indicated by the number in the lower right corner. The ridge axis is shown by a black line and its geometry changes as new propagation events occur. L, Loki; F, Fenrir; S, Sleipnir; H, Hel.

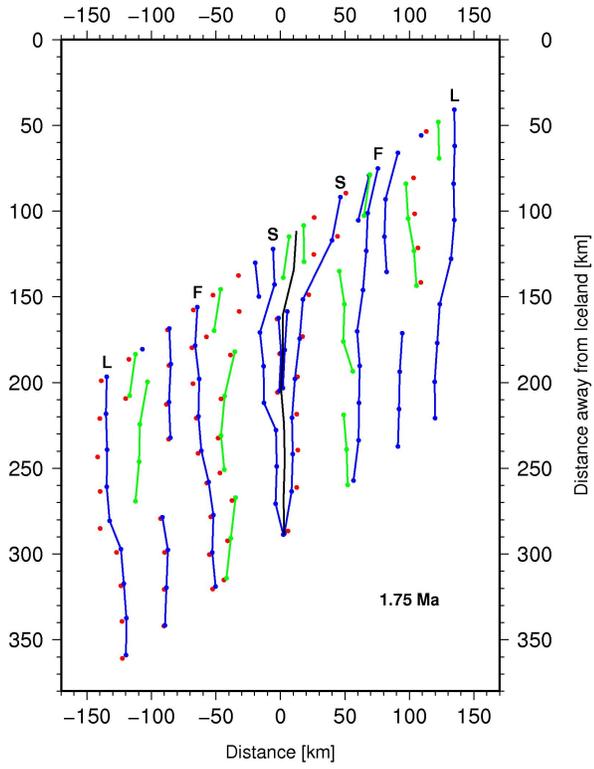


(c) After propagation of Fenrir.

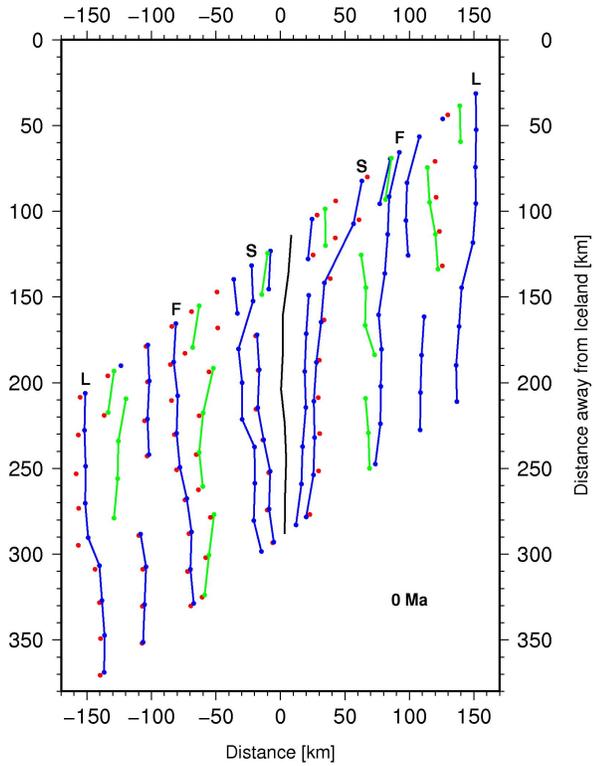


(d) Before Sleipnir starts propagating. The ridge axis at profile 17 is shifted to the right because of a big jump which later shows up in profile 18.

Figure 2.18: (continued)

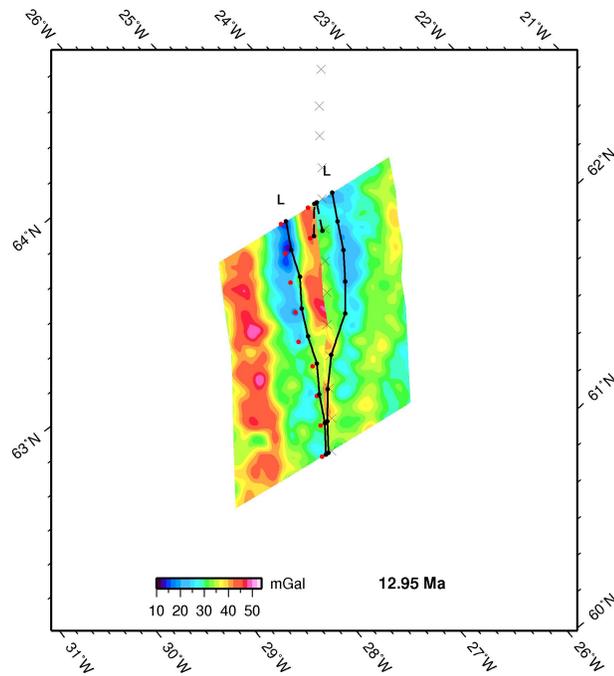


(c) After propagation of Sleipnir and on at the onset of Hel.

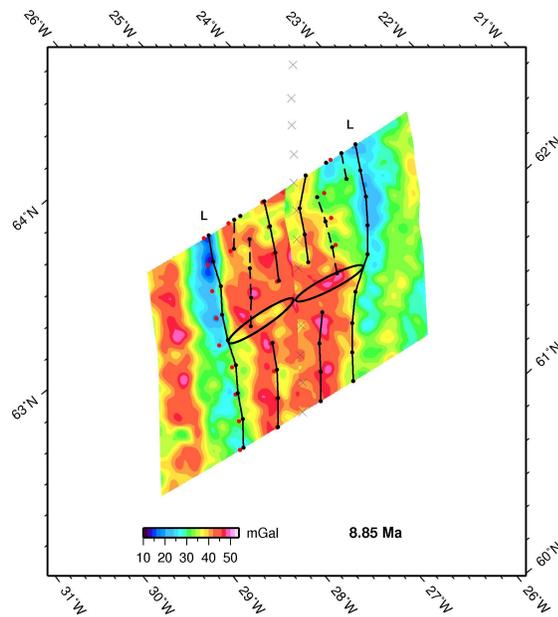


(d) Present day configuration.

Figure 2.18: (continued)

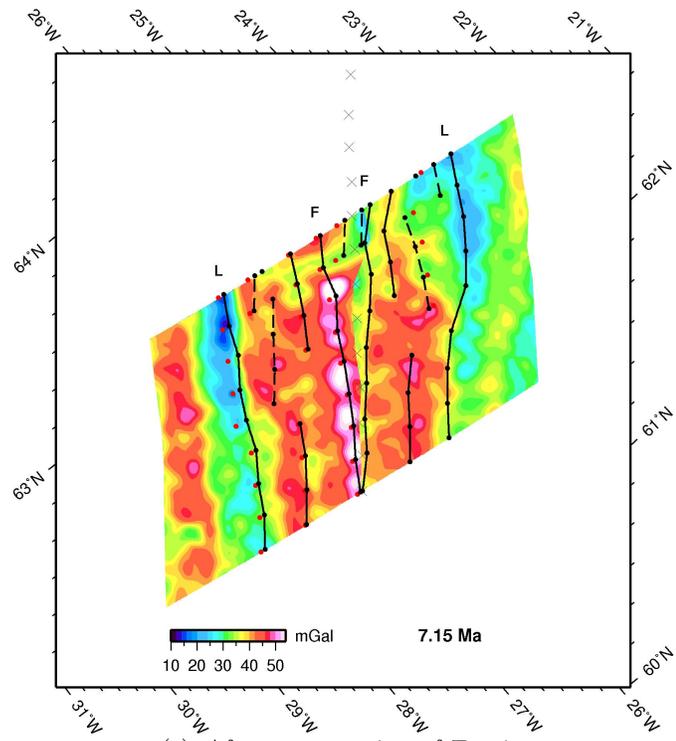


(a) After propagation of Loki.

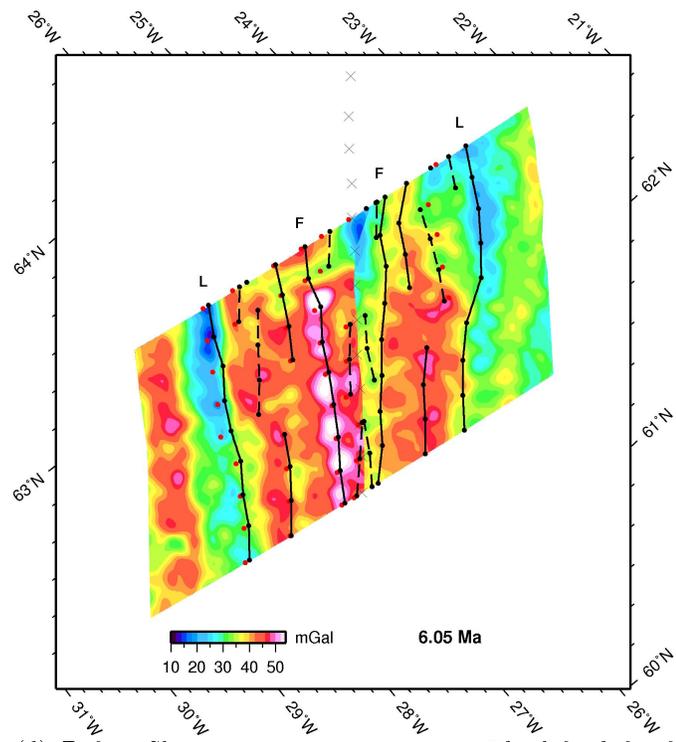


(b) Before Fenrir starts propagating. The black circles indicate the location of the pseudofaults.

Figure 2.19: Snapshots of the evolution of the Reykjanes Ridge. Overlaid on satellite gravity (*Sandwell and Smith [2009]*) are pseudofaults (black circles) connected by solid and dashed lines for southward and northward propagators, respectively. Red circles are failed rifts. The names of the major southward propagators are marked in. For each time the gravity is gridded in the area bounded by 19 Ma, the current time and profiles 17 and 25 on each ridge flank; the two areas are then rotated toward each other to close the space between the ridge and the areas. Time of each snapshot is indicated by the number in the lower right corner. L, Loki; F, Fenrir; S, Sleipnir.

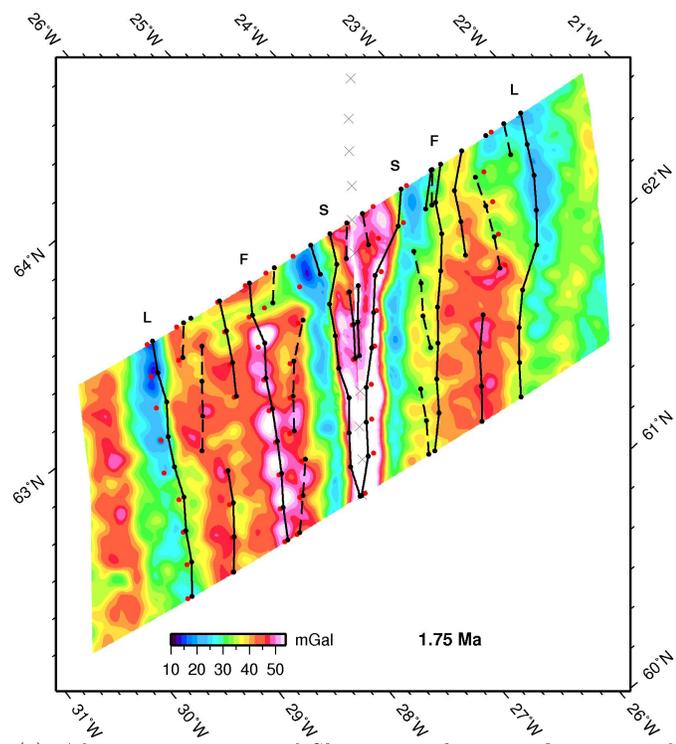


(c) After propagation of Fenrir.

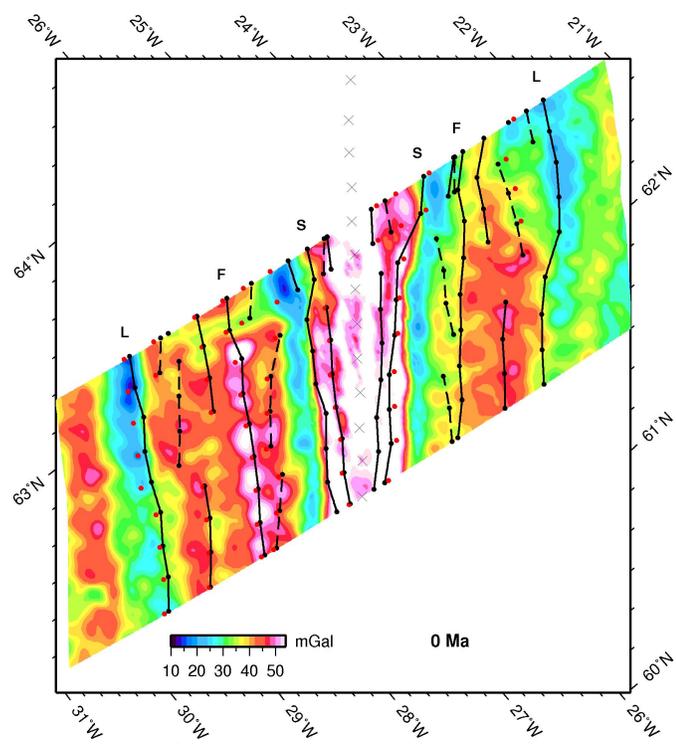


(d) Before Sleipnir starts propagating. The left shift of pseudofaults and failed rifts at profile 17 is caused by a big jump which later shows up in profile 18.

Figure 2.19: (continued)



(e) After propagation of Sleipnir and on at the onset of Hel.



(f) Present day configuration.

Figure 2.19: (continued)

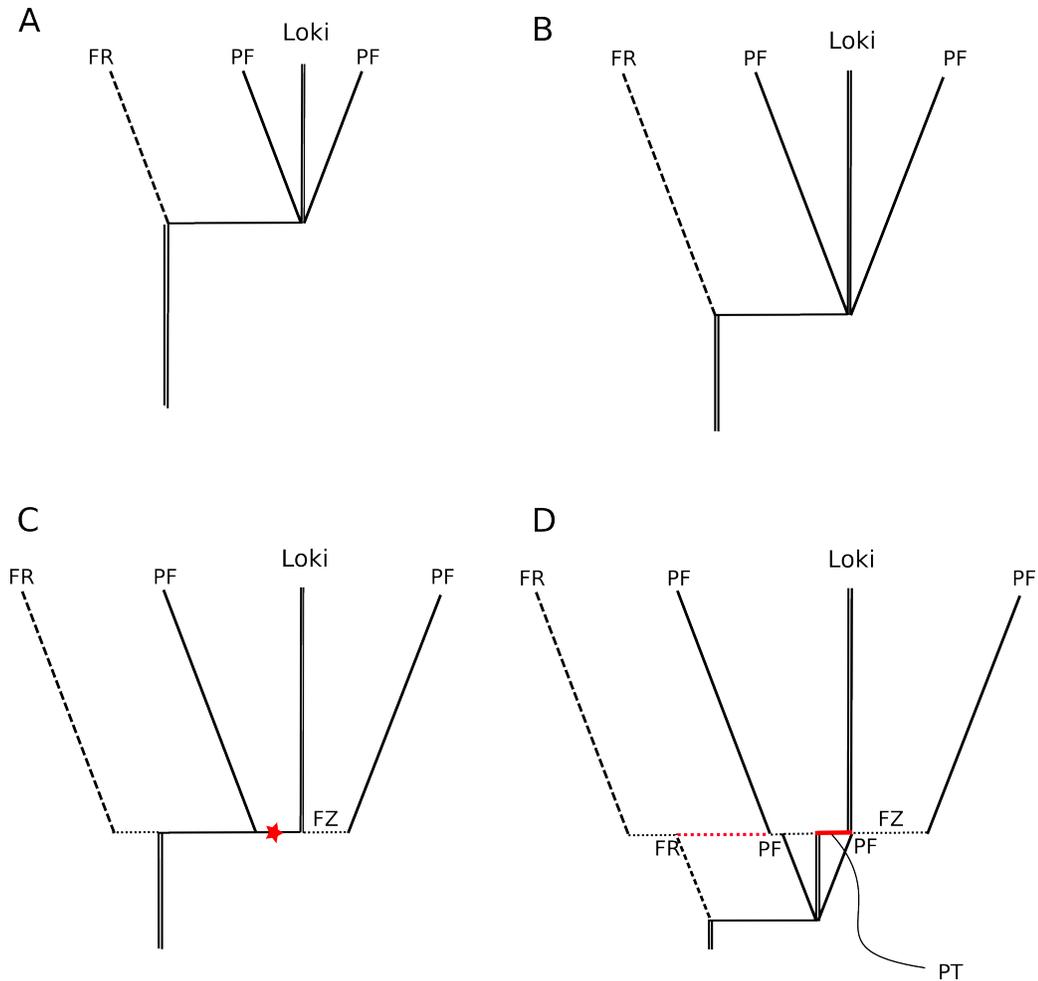


Figure 2.20: Schematic illustration of propagation of Loki. **A.** Tectonic settings as Loki propagates from Iceland, transferring lithosphere to the North American Plate. **B.** Loki stalls. **C.** As Loki is stalled the pseudofaults and failed rift spread away from the ridge. The red star indicates where Loki will continue propagating. **D.** Loki continues to propagate but starts from the middle of the offset. A right-stepping ponsu-transform is created and there is a step in the pseudofaults at its location. The red dashed fracture zone (FZ) is a transform fault that got frozen into the North America plate as Loki continued propagating. FR, failed rift; PF, pseudofault; PT, ponsu-transform; FZ, fracture zone.

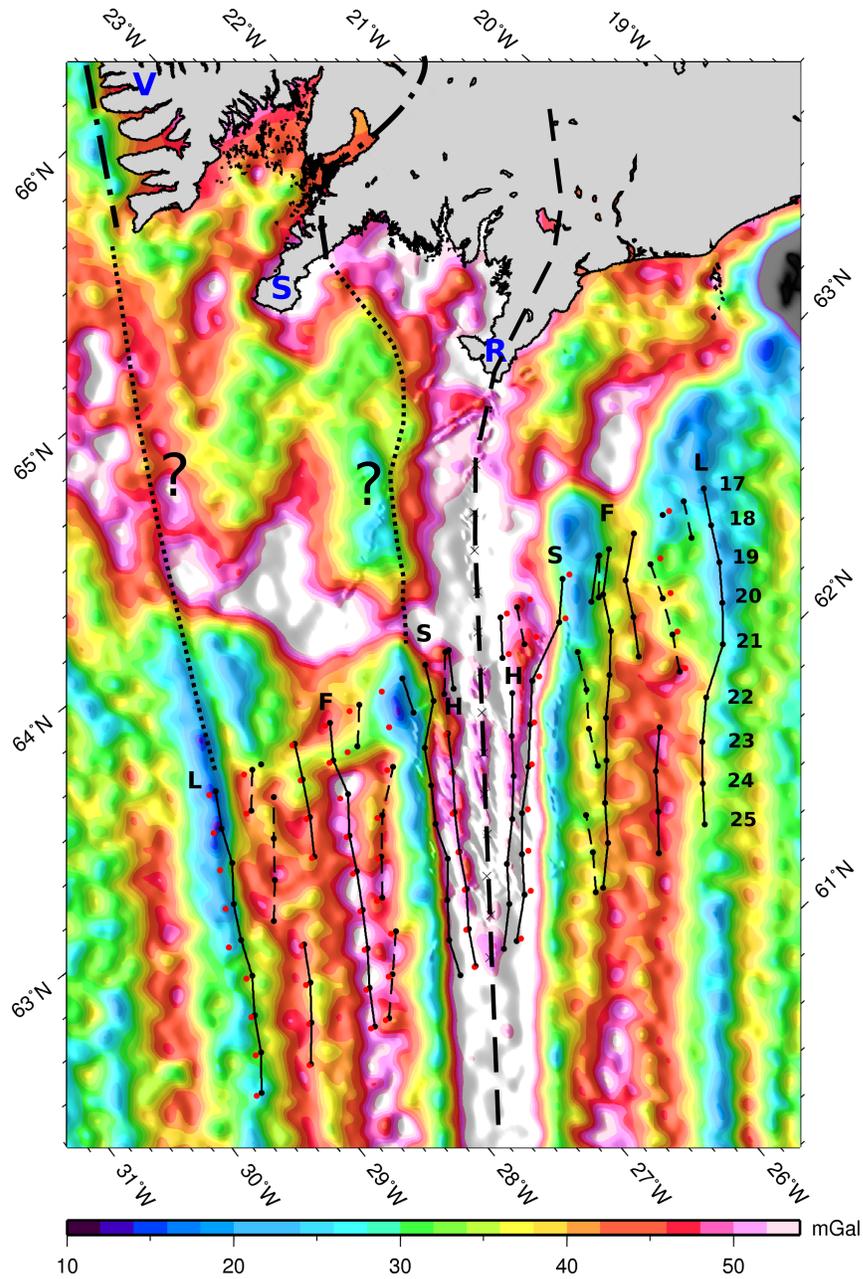


Figure 2.21: Satellite gravity and tectonic boundaries near Iceland (*Sandwell and Smith [2009]*). Oblique Mercator projection. Pseudofaults and failed rifts predicted by our magnetic models are shown; solid lines connect the pseudofaults of southward propagating rifts, dashed lines connect pseudofaults of northward propagators, and red dots are failed rifts. Heavy dashed line is the Reykjanes Ridge and its extension up to Iceland; dash-dotted lines are the locations of the paleo-spreading centers in Iceland and dotted lines are an attempt to trail the paleo-spreading centers down to our survey area. Numbers indicate the location of our profiles (17-25). V, Vestfirðir; S, Snæfellsnes; R, Reykjanes Peninsula; L, Loki; F, Fenrir; S, Sleipnir; H, Hel.

Table 2.1: Location of ridge axis and total difference between the model and data central anomaly (summed up from both ridge flanks) for profiles 17-25. The data central anomaly is always wider and the uncertainty is always 1 km.

Profile Number	Latitude	Longitude	Central Anomaly Difference [km]
17	63.0027	-24.6786	8.5
18	62.8525	-24.9281	4.5
19	62.7050	-25.2071	7.0
20	62.5541	-25.4682	7.0
21	62.4071	-25.7432	5.0
22	62.2522	-25.9888	7.5
23	62.1037	-26.2715	5.0
24	61.9513	-26.5000	2.5
25	61.8012	-26.7803	2.0

Table 2.2: Stage poles of rotation for Eu-Na (Eurasia fixed).

	0-6.733 Ma	6.733-11.04 Ma	11.04-15.974 Ma	15.974-19.722 Ma
DeMets	63.76°N	68.81°N	68.81°N	68.81°N
Latitude	130.82°E	133.96°E	133.96°E	133.96°E
Longitude	0.2133°/Myr	0.2734°/Myr	0.2734°/Myr	0.2734°/Myr
(2010)	66.85°N	66.85°N	66.85°N	66.85°N
This	135.46°E	135.46°E	135.46°E	135.46°E
Latitude	0.2251°/Myr	0.2600°/Myr	0.2762°/Myr	0.2550°/Myr
Longitude				
study				
Rotation Angle				

Table 2.3: Distance of anomalies from the ridge in kilometers.

Profile #	3ro (6.033 Ma)			5n.2no (11.04 Ma)			5Bro (15.974 Ma)			6no (19.722 Ma)		
	Na	Eu	Total	Na	Eu	Total	Na	Eu	Total	Na	Eu	Total
17	55±2	58±4	113±7	122±3	101±4	223±7	178±3	158±2	336±5	222±2	196±4	418±6
18	-	-	-	122±3	101±2	223±5	177±2	161±2	338±4	220±3	200±3	420±6
19	54±2	60±3	114±5	120±2	102±4	222±6	177±2	161±2	338±4	221±2	199±4	420±6
20	57±3	59±2	116±5	117±3	107±3	224±6	176±2	163±1	339±3	-	-	-
21	56±3	59±2	115±5	116±4	108±4	224±8	178±3	164±1	342±4	221±2	202±2	423±4
22	55.5±2.5	60±2	115.5±4.5	115±2	109±3	224±5	177±3	166±2	343±5	220±3	200±3	420±6
23	57±2	61±2	118±4	117±3	108±4	225±7	178±2	166±2	344±4	220±3	202±3	422±6
24	56±3	59±2	115±5	118±2	109±2	227±4	177±2	167±2	344±4	220±2	206±2	426±4
25	58±2	59±1	117±4	118±2	109±3	227±5	177±2	167±2	344±4	221±2	205±2	426±4

Table 2.4: Full spreading rates (km/myr) used in the forward magnetic modeling.

Profile Number	0-6.733 Ma	6.733-11.04 Ma	11.04-15.974 Ma	15.974-19.722 Ma
17	18.99	21.93	23.30	21.51
18	19.04	21.99	23.36	21.56
19	19.08	22.04	23.42	21.62
20	19.13	22.10	23.47	21.67
21	19.18	22.15	23.53	21.72
22	19.23	22.21	23.59	21.78
23	19.27	22.26	23.65	21.83
24	19.32	22.32	23.71	21.89
25	19.37	22.37	23.76	21.94

Table 2.5: Percentage asymmetry for modeling of profile 20 in Figure 2.5. Positive asymmetry indicates more accretion on Eurasia plate.

Time Period	Asymmetry
Ma	%
0-1	-20
1-2	0
2-3	5
3-3.8	45
3.8-4.8	-6
4.8-6.1	0
6.1-7	-20
7-8.2	-8
8.2-10.7	-18
10.7-13	16
13-14.9	0
14.9-17	-53
17-118	0

Table 2.6: Parameters used for the forward magnetic modeling. c , contamination coefficient.

Profile	Jumps		Magnetization	
	Time of Jump (Ma)	Distance (km)	Interval (Ma)	Magnetization (A/m)
17 $c=0.6$	1.7	-4.0	0-0.78	20
	2.35	-8.0	0.78-15	6
	4.5	-4.1	15-20	4
	6.4	13.0		
	7.7	6.0		
	8.8	3.0		
	10.5	1.5		
	12.2	-4.0		
	13.0	5.0		
	14.5	4.0		
18 $c=0.5$	1.6	-4.0	0-0.78	20
	2.6	-7.5	0.78-15	8
	4.1	-4.0	15-20	5
	5.8	15.0		
	7.7	6.0		
	8.5	2.5		
	10.0	1.2		
	11.5	-6.0		
	13.2	3.0		
	14.5	4.8		
19 $c=0.7$	2.1	1.1	0-0.78	20
	3.5	-4.5	0.78-15	6
	6.0	4.0	15-20	6
	8.3	4.7		
	10.0	2.0		
	11.8	-5.0		

Table 2.6: (continued)

	14.4	7.5		
20	1.9	0.9	0-0.78	20
$c=0.5$	3.2	-2.3	0.78-15	8
	6.6	3.0	15-20	6
	8.2	1.7		
	10.0	1.5		
	12.0	-3.8		
	14.4	5.0		
21	1.9	1.4	0-0.78	25
$c=0.7$	3.0	-2.2	0.78-15	8
	6.7	3.0	15-20	6
	8.0	2.5		
	12.3	-3.8		
	14.3	7.5		
22	1.7	0.6	0-0.78	30
$c=0.7$	2.4	-3.5	0.78-15	8
	6.9	3.4	15-20	6
	7.7	1.4		
	10.8	1.5		
	13.4	3.4		
23	1.3	1.2	0-0.78	30
$c=0.7$	2.4	-3.9	0.78-15	8
	6.1	2.8	15-20	6
	7.5	1.8		
	10.5	2.6		
	13.2	2.1		

Table 2.6: (continued)

24	1.3	1.3	0-0.78	30
$c=0.7$	2.4	-3.9	0.78-15	8
	6.4	2.5	15-20	6
	7.5	2.2		
	10.5	1.6		
	13.0	3.3		

25	0.9	0.8	0-0.3	40
$c=0.5$	1.8	-2.6	0.3-0.78	25
	6.6	2.1	0.78-15	10
	7.2	2.3	15-20	6
	10.5	0.8		
	13.0	3.0		

Table 2.7: Magnetic modeling r.m.s. misfit values from this study and *Hey et al.* [2010]

Profile Number	This study [nT]	<i>Hey et al.</i> [2010] [nT]	Reduction in r.m.s. [%]
17	348	368	5
18	346	478	28
19	370	423	13
20	332	431	23
21	352	391	11
22	430	475	9
23	393	439	10
24	295	390	24
25	262	379	31

Bibliography

- Albers, M., and U. Christensen (2001), Channeling of plume flow beneath mid-ocean ridges, *Earth Planet. Sci. Lett.*, *187*(1-2), 207–220.
- Atwater, T., and J. Mudie (1973), Detailed near-bottom geophysical study of the Gorda Rise, *J. Geophys. Res.*, *78*(35), 8665–8686.
- Barckhausen, U., C. Ranero, S. Cande, M. Engels, and W. Weinrebe (2008), Birth of an intraoceanic spreading center, *Geology*, *36*(10), 767.
- Cande, S., and D. Kent (1995), Revised calibration of the geomagnetic polarity timescale for the Late Cretaceous and Cenozoic, *J. Geophys. Res.*, *100*(B4), 6093–6095.
- Caress, D., H. Menard, and R. Hey (1988), Eocene reorganization of the Pacific–Farallon spreading center north of the Mendocino fracture zone, *J. Geophys. Res.*, *93*(B4), 2813–2838.
- Christie, D., B. West, D. Pyle, and B. Hanan (1998), Chaotic topography, mantle flow and mantle migration in the Australian–Antarctic discordance, *Nature*, *394* (6694), 637–644.
- Delaney, J., H. Johnson, and J. Karsten (1981), The Juan de Fuca–hotspot–propagating rift system: New tectonic, geochemical, and magnetic data, *J. Geophys. Res.*, *86*(B12), 11,747–11.
- DeMets, C., and D. Wilson (2008), Toward a minimum change model for recent plate motions: calibrating seafloor spreading rates for outward displacement, *Geophysical J. International*, *174* (3), 825–841.
- DeMets, C., R. Gordon, D. Argus, and S. Stein (1990), Current plate motions, *Geophys. J. Int.*, *101*(2), 425–478.
- DeMets, C., D. Gordon, R.G. Argus, and S. Stein (1994), Effect of recent revisions to the geomagnetic reversal time scale on estimate of current plate motions, *Geophys. Res. Lett.*, *21*(20), 2192–2194.
- DeMets, C., R. Gordon, and D. Argus (2010), Geologically current plate motions, *Geophysical J. International*, *181* (1), 1–80.
- Einarsson, P. (2008), Plate boundaries, rifts and transforms in Iceland, *Jökull*, *58*, 35–58.
- Fernandez, L., and R. Hey (1991), Late Tertiary tectonic evolution of the seafloor spreading system off the coast of California between the Mendocino and Murray fracture zones, *J. Geophys. Res.*, *96*(B11), 17,955–17,979.

- Hardarson, B., J. Fitton, R. Ellam, and M. Pringle (1997), Rift relocation—a geochemical and geochronological investigation of a palaeo-rift in northwest Iceland, *Earth Planet. Sci. Lett.*, *153*(3-4), 181–196.
- Hardarson, B., J. Fitton, and A. Hjartarson (2008), Tertiary Volcanism in Iceland, *Jokull*, *58*, 161–178.
- Harrison, C. (1976), Magnetization of the oceanic crust, *Geophys. J. Int.*, *47*(2), 257–283.
- Hayes, D. (1976), Nature and implications of asymmetric sea-floor spreading—" Different rates for different plates", *Bulletin of the Geological Society of America*, *87*(7), 994.
- Herron, E., and M. Talwani (1972), Magnetic anomalies on the reykjanes ridge, *Nature*, *238*, 390–392.
- Hey, R. (1977), A new class of pseudofaults and their bearing on plate tectonics: a propagating rift model, *Earth Planet. Sci. Lett.*, *37*, 321–325.
- Hey, R., and P. Vogt (1977), Spreading center jumps and sub-axial asthenosphere flow near the Galapagos hotspot, *Tectonophysics*, *37*(1-3), 41–52.
- Hey, R., F. Duenebier, and W. Morgan (1980), Propagating rifts on mid-ocean ridges, *J. Geophys. Res.*, *85*(137), 3647–3658.
- Hey, R., H. Menard, T. Atwater, and D. Caress (1988), Changes in direction of seafloor spreading revisited, *J. Geophys. Res.*, *93*(B4), 2803–2811.
- Hey, R., F. Martinez, A. Höskuldsson, and A. Benediktsdóttir (2010), Propagating rift model for the V-shaped ridges south of Iceland, *Geochem. Geophys. Geosyst.*, *11*(3), Q03,011.
- Höskuldsson, A., R. Hey, M. Fernando, and Á. Benediktsdóttir (2010), Njörður Central volcano, first direct evidence of shallow magma chambers on the Reykjanes Ridge, 29th Nordic Geological Winter Meeting, January 11-13, Oslo.
- Ito, G. (2001), Origin of V-shaped Reykjanes ridges from a pulsing and dehydrating mantle plume, *Nature*, *411*, 681–684.
- Johansen, B., P. Vogt, and O. Eldholm (1984), Reykjanes Ridge: further analysis of crustal subsidence and time-transgressive basement topography, *Earth Planet. Sci. Lett.*, *68*(2), 249–258.
- Jones, S. (2003), Test of a ridge-plume interaction model using oceanic crustal structure around Iceland, *Earth Planet. Sci. Lett.*, *208*(3-4), 205–218.
- Jones, S., and J. Maclennan (2005), Crustal flow beneath Iceland, *J. Geophys. Res.*, *110*, 9410–9429.

- Jones, S., N. White, and J. MacLennan (2002), V-shaped ridges around Iceland: Implications for spatial and temporal patterns of mantle convection, *Geochem. Geophys. Geosyst.*, *3*(10), 1059.
- Keen, C., and C. Tramontini (1970), A seismic refraction survey on the Mid-Atlantic Ridge, *Geophys. J. Int.*, *20*(5), 473–491.
- Kristjánsson, L., and G. Jónsson (1998), Aeromagnetic results and the presence of an extinct rift zone in western Iceland, *J. Geodyn.*, *25*(1-2), 99–108.
- Lourens, L., F. Hilgen, J. Laskar, N. Shackleton, and D. Wilson (2004), The Neogene Period, in *A Geologic Time Scale 2004*, edited by F. Gradstein, J. Ogg, and A. Smith, pp. 409–440, Cambridge University Press Cambridge, UK.
- Macdonald, K., S. Miller, B. Luyendyk, T. Atwater, and L. Shure (1983), Investigation of a Vine-Matthews magnetic lineation from a submersible: The source and character of marine magnetic anomalies, *J. Geophys. Res.*, *88*, 3403–3418.
- Menard, H., and T. Atwater (1968), Changes in the direction of seafloor spreading, *Nature*, *219*, 463–467.
- Mendel, V., M. Munsch, and D. Sauter (2005), MODMAG, a MATLAB program to model marine magnetic anomalies, *Computers and Geosciences*, *31*(5), 589–597.
- Merkouriev, S., and C. DeMets (2008), A high-resolution model for Eurasia–North America plate kinematics since 20 Ma, *Geophysical Journal International*, *173*(3), 1064–1083.
- Mihut, D., and R. Müller (1998), Volcanic margin formation and Mesozoic rift propagators in the Cuvier Abyssal Plain off Western Australia, *J. Geophys. Res.*, *103*(B11), 27,135–27.
- Morgan, W. (1968), Rises, trenches, great faults, and crustal blocks, *J. Geophys. Res.*, *73*(6), 1959–1982.
- Morgan, W. (1971), Convection plumes in the lower mantle, *Nature*, *230*, 42–43.
- Naar, D., and R. Hey (1989), Recent Pacific-Easter-Nazca plate motions, *Evolution of Mid-Oceanic Ridges*, pp. 9–30.
- Naar, D., and R. Hey (1991), Tectonic evolution of the Easter microplate, *J. Geophys. Res.*, *96*(B5), 7961–7993.
- Parson, L., et al. (1993), En echelon axial volcanic ridges at the Reykjanes Ridge: a life cycle of volcanism and tectonics, *Earth Planet. Sci. Lett.*, *117*(1-2), 73–87.
- Perram, L., M. Cormier, and K. Macdonald (1993), Magnetic and tectonic studies of the dueling propagating spreading centers at 20°40′ S on the East Pacific Rise: evidence for crustal rotations, *J. Geophys. Res.*, *98*(B8), 13,835–13,850.

- Phipps Morgan, J., and E. Parmentier (1985), Causes and rate-limiting mechanisms of ridge propagation: a fracture mechanics model, *J. Geophys. Res.*, *90*(B10), 8603–8612.
- Phipps Morgan, J., and D. Sandwell (1994), Systematics of ridge propagation south of 30 S, *Earth Planet. Sci. Lett.*, *121*(1-2), 245–258.
- Poore, H., R. Samworth, N. White, S. Jones, and I. McCave (2006), Neogene overflow of northern component water at the Greenland-Scotland ridge, *Geochem. Geophys. Geosyst.*, *7*, Q06,010.
- Poore, H., N. White, and S. Jones (2009), A Neogene chronology of Iceland plume activity from V-shaped ridges, *Earth Planet. Sci. Lett.*, *283*(1-4), 1–13.
- Sæmundsson, K. (1974), Evolution of the axial rift zone in northern Iceland and the Tjornes fracture zone, *Bulletin of the Geological Society of America*, *85*(4), 495.
- Sæmundsson, K. (1979), Outline of the geology of Iceland, *Jökull*, *29*, 7–28.
- Sandwell, D., and W. Smith (2009), Global marine gravity from retracked Geosat and ERS-1 altimetry: Ridge segmentation versus spreading rate, *J. Geophys. Res.*, *114*.
- Sanner, M. (1999), Python: a programming language for software integration and development, *J. Mol. Graph. Model.*, *17*, 57–61.
- Schilling, J., P. Meyer, and R. Kingsley (1982), Evolution of the Iceland hotspot, *Nature*, *296*, 313–320.
- Schilling, J., H. Sigurdsson, A. Davis, and R. Hey (1985), Easter microplate evolution, *Nature*, *317*(6035), 325–331.
- Schouten, H., M. Tivey, D. Fornari, and J. Cochran (1999), Central anomaly magnetization high: constraints on the volcanic construction and architecture of seismic layer 2A at a fast-spreading mid-ocean ridge, the EPR at 9° 30′–50′ N, *Earth and Planet. Sci. Lett.*, *169*(1-2), 37–50.
- Searle, R., and A. Laughton (1981), Fine-scale sonar study of tectonics and volcanism on the Reykjanes Ridge, *Oceanol. Acta*, *4*, 5–13.
- Searle, R., J. Keeton, R. Owens, R. White, R. Mecklenburgh, B. Parsons, and S. Lee (1998), The Reykjanes Ridge: structure and tectonics of a hot-spot-influenced, slow-spreading ridge, from multibeam bathymetry, gravity and magnetic investigations, *Earth and planetary science letters*, *160*(3-4), 463–478.
- Shih, J., and P. Molnar (1975), Analysis and implications of the sequence of ridge jumps that eliminated the Surveyor transform fault, *J. Geophys. Res.*, *80*(35), 4815–4822.
- Smallwood, J., and R. White (1998), Crustal accretion at the Reykjanes Ridge, 61°–62° N, *J. Geophys. Res.*, *103*(B3), 5185–5201.

- Smallwood, J., and R. White (2002), Ridge-plume interaction in the North Atlantic and its influence on continental breakup and seafloor spreading, in *The North Atlantic Igneous Province: Stratigraphy, Tectonic, Volcanic and Magmatic Processes*, vol. 197, edited by B. Bell and D. Jolley, pp. 15–37, Geol. Soc. Spec. Publ.
- Stein, S., H. Melosh, and J. Minster (1977), Ridge migration and asymmetric sea-floor spreading, *Earth Planet. Sci. Lett.*, *36*(1), 51–62.
- Talwani, M., and J. Heirtzler (1964), Computation of magnetic anomalies caused by 2-D structures of arbitrary shapes, *Computers in the Mineral Industry, Part I: Stanford Univ. Publ. Geol. Sci.*, *9*, 464–480.
- Talwani, M., C. Windisch, and M. Langseth (1971), Reykjanes Ridge crest: a detailed geophysical study, *J. Geophys. Res.*, *76*(2), 473–517.
- Tisseau, J., and P. Patriat (1981), Identification des anomalies magnétiques sur les dorsales à faible taux d'expansion, *Earth Planet. Sci. Lett.*, *52*, 381–396.
- Vine, F. (1966), Spreading of the ocean floor: new evidence, *Science*, *154*(3755), 1405.
- Vine, F., and D. Matthews (1963), Magnetic anomalies over oceanic ridges, *Nature*, *199*(4897), 947–949.
- Vogt, P. (1971), Asthenosphere motion recorded by the ocean floor south of Iceland, *Planet. Sci. Lett.*, *13*(1), 153–160.
- Vogt, P., and G. Johnson (1972), Seismic reflection survey of an oblique aseismic basement trend on the Reykjanes Ridge, *Earth Planet. Sci. Lett.*, *15*(3), 248–254.
- Vogt, P., G. Johnson, and L. Kristjansson (1980), Morphology and magnetic anomalies north of Iceland, *Journal of geophysics: Zeitschrift für Geophysik*, p. 67.
- Vogt, P., N. Cherkis, and G. Morgan (1983), Project Investigator-I: Evolution of the Australia-Antarctic discordance deduced from a detailed aeromagnetic study, in *Antarctic Earth Science*, edited by R. Oliver, P. James, and J. Jago, pp. 608–613, Cambridge University Press Cambridge, UK.
- Weissel, J., and D. Hayes (1971), Asymmetric seafloor spreading south of Australia, *Nature*, *231*, 518–522.
- Wessel, P., and W. Smith (1995), New version of the Generic Mapping Tools released, *Eos Trans. AGU*, *76*(33), 329.
- White, N., and B. Lovell (1997), Measuring the pulse of a plume with the sedimentary record, *Nature*, *387*(6636), 888–891.
- White, R. (1997), Rift-plume interaction in the North Atlantic, *Philos. Trans. R. Soc. London, Ser. A*, *355*(1723), 319–339.

- White, R., J. Bown, and J. Smallwood (1995), The temperature of the Iceland plume and origin of outward-propagating V-shaped ridges, *J. Geol. Soc.*, *152*(6), 1039–1045.
- Wilson, D., and R. Hey (1995), History of rift propagation and magnetization intensity for the Cocos-Nazca spreading center, *J. Geophys. Res.*, *100*(B6).
- Wilson, D., R. Hey, and C. Nishimura (1984), Propagation as a mechanism of reorientation of the Juan de Fuca Ridge, *J. Geophys. Res.*, *89*(B11), 9215–9225.
- Wilson, J. (1963), A possible origin of the Hawaiian Islands, *Can. J. Phys.*, *41*(6), 863–870.

Appendix A

Magellan

Four python modules make up Magellan and they are listed below.

A.1 magellan

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

"""
magellan - marine magnetic anomaly modeller
Copyright (C) 2008 Tryggvi Björgvinsson <tryggvib@hi.is>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
"""

import sys, getopt
from lib.Mag.calc import *
from lib.Mag.data import *
from lib.Mag.plot import *
from numpy import *
from timeit import *

_default_pointspacing = '1'
_default_tmin = '0'
# By defining tend as infinity the last stage in the timescale file is used
_default_tend = 'float("infinity")'
def parse_opts():

    options = {'asymmetry':None,
               'config':None,
               'graphs':None,
               'jump':None,
               'magnetization':None,
               'spreadingrate':None,
               'timescale':None,
               'pointspacing':None,
               'inclination': None,
               'declination': None,
               'thickness': None,
               'contam': None,
               'profile_azimuth': None,
               'ridge_azimuth': None,
               'spreading_direction': None,
               'data': None,
               'level': None,
               'tmin': None,
               'tend': None,
               'xmin': None,
               'xmax': None,
```

```

        'paleoincl': None,
        'paleo_ridge_str': None,
        'title': None,}

try:
    opts, args = getopt.getopt(sys.argv[1:],
                                "a:b:c:d:g:i:j:k:l:m:o:r:s:t:z:p:v:x:n:m:w:q: :h",
                                ["asymmetry=",
                                 "profile_azimuth=",
                                 "config=",
                                 "declination=",
                                 "graph=", #Not implemented
                                 "inclination=",
                                 "jump=",
                                 "contam=",
                                 "level=",
                                 "magnetization=",
                                 "spreading_direction=",
                                 "ridge_azimuth=",
                                 "spreadingrate=",
                                 "timescale=",
                                 "thickness=",
                                 "pointspacing=",
                                 "tend=",
                                 "tmin=",
                                 "xmin=",
                                 "xmax=",
                                 "paleoincl=",
                                 "paleo_ridge_str=",
                                 "title=",
                                 "help",])
except getopt.GetoptError:
    #print help information and exit:
    usage()
    sys.exit(2)
for o, a in opts:
    if o in ("-a", "--asymmetry"):
        options['asymmetry'] = a
    if o in ("-b", "--profile_azimuth"):
        options['profile_azimuth'] = a
    if o in ("-c", "--config"):
        options['config'] = a
    if o in ("-d", "--declination"):
        options['declination'] = a
    if o in ("-g", "--graph"):
        options['graphs'] = a
    if o in ("-i", "--inclination"):
        options['inclination'] = a
    if o in ("-j", "--jump"):
        options['jump'] = a
    if o in ("-k", "--contam"):
        options['contam'] = a
    if o in ("-l", "--level"):
        options['level'] = a
    if o in ("-m", "--magnetization"):
        options['magnetization'] = a
    if o in ("-o", "--spreading_direction"):
        options['spreading_direction'] = a
    if o in ("-r", "--ridge_azimuth"):
        options['ridge_azimuth'] = a
    if o in ("-s", "--spreadingrate"):
        options['spreadingrate'] = a
    if o in ("-t", "--timescale"):
        options['timescale'] = a
    if o in ("-z", "--thickness"):
        options['thickness'] = a
    if o in ("-p", "--pointspacing"):
        options['pointspacing'] = a

```

```

    if o in ("-v", "--tmin"):
        options['tmin'] = a
    if o in ("-x", "--tend"):
        options['tend'] = a
    if o in ("-n", "--xmin"):
        options['xmin'] = a
    if o in ("-m", "--xmax"):
        options['xmax'] = a
    if o in ("-w", "--paleoincl"):
        options['paleoincl'] = a
    if o in ("-q", "--paleo_ridge_str"):
        options['paleo_ridge_str'] = a
    if o in (" ", "--title"):
        options['title'] = a
    if o in ("-h", "--help"):
        usage()
        sys.exit()

return (options, args)

def usage():
    print "Usage: magellan [OPTION]... [FILE]"
    print "Options:"
    print "  -a [FILE] \t asymmetry file"
    print "  -j [FILE] \t jump file"
    print "  -s [FILE] \t spreading rate file"
    print "  -t [FILE] \t time scale file"
    print "  -m [FILE] \t magnetization file"
    print "  -c [FILE] \t configuration file"
    print "  -b value \t azimuth of profile"
    print "  -d value \t amount of declination"
    print "  -i value \t amount of inclination"
    print "  -k value \t contamination coefficient"
    print "  -l value \t level of measurement (km)"
    print "  -z value \t thickness of magnetized layer"
    print "  -o value \t spreading direction"
    print "  -r value \t ridge azimuth"
    print "  -p value \t spacing between points in calculations"
    print "  --tmin value \t age at the ridge axis"
    print "  --tend value \t age defining the end of the magnetized body"
    print "  --xmin value \t minimum distance from the ridge where the model is calculated"
    print "  --xmax value \t maximum distance from the ridge where the model is calculated"
    print "  --paleoincl value \t inclination of the magnetized body, if different from present location inclination"
    print "  --paleo_ridge_str value \t strike of the paleoridge (if the magnetic body has been rotated from its original strike)"
    print "  --title \t title of the model"
    print "  -h \t print this help"

if __name__ == '__main__':

    # files contains the information given in the command line prompt
    (files, arguments) = parse_opts()

    # configs and parameters contain information given in the configuration file
    configs = get_configurations(files['config'])
    parameters = get_configurations(files['config'])

    if len(arguments) == 0:
        datafile = configs.pop('data', None)
        #if datafile == None:
            #print "No track data file given\n"
            #sys.exit()
    else:
        datafile = arguments[0]

```

```

# Taking information from configs and putting it into files. The information
# given in the command line are used over the ones given in the configuration file
for key in configs.keys():
    if files[key] == None:
        files[key] = configs.pop(key, None)
    else:
        configs.pop(key, None)

if files['tend'] == None:
    files.pop('tend', None)

tend = eval(files.pop('tend', _default_tend))
# The title of the graph
title = files.pop('title')

(timescale, tend) = get_timescale(tend, files['timescale'])

asym = get_asymmetry(tend, files['asymmetry'])
spread = get_spreadingrate(tend, files['spreadingrate'])
(jump, markers) = get_jumps(tend, files['jump'])

magnet = get_magnetization(tend, files['magnetization'])

# Making sure that, if nothing is specified, that inclination,
# declination, etc. have no value so that magellan uses
# the default values
for key in files.keys():
    if files[key] == None:
        files.pop(key, None)

# Distance between points in model
offset = eval(files.pop('pointspacing', _default_pointspacing))
tmin = eval(files.pop('tmin', _default_tmin))

# If no spreading rate is input the spreading rate is 20 km/myr over all the timescale
if spread=={}:
    a = timescale.keys()
    spread[min(a)] = {'spreadingrate':(10)}

# Making a timeline; what happens when
timeline = create_change_timeline(asym, spread, jump, magnet, timescale, markers)
timeline_copy = timeline[:]

# delta_l and delta_r include information on each little segment (n or r; magnetization)
(delta_l, delta_r, timeline_rest) = create_deltax(timeline, tmin)
delta_lcopy = delta_l[:]
delta_rcopy = delta_r[:]

# Creating magnetized layer in flowline space from given parameters
mag_layer = create_magnetized_layer(delta_l, delta_r)
# Obtain the max on each side of the ridge, this is controlled by tend
# If there is no asymmetry (jumps or spreading) xmax_l=xmax_r
xmax_r = 0
xmax_l = 0
for ((start, end), pol, magnet) in mag_layer:
    if max(start, end) > xmax_r:
        xmax_r = max(start, end)
        if min(start, end) < xmax_l:
            xmax_l = min(start, end)

# dist and deep correspond to the magnetized layer and dist_anom and anom
# correspond to each other (len(anom)=len(dist_anom))
(dist, deep, dist_anom, anom, other_data, lat, lon) =
    get_trackdata(datafile, xmax_l, xmax_r, offset)

```

```

# Calculating how much lithosphere has to be subtracted from the bathymetry
# on each side of the ridge, if any
# If there is no asymmetry (jumps or spreading) xmin_l=xmin_r
if timeline_rest:
    (xmin_l,xmin_r) = get_xminmax(timeline_rest)

# We want to be able to calculate the model with a finer spacing than our
# data spacing is. This is the model_distance vector
# arange(10) = [0 1 2 3 4 5 6 7 8 9]

x_last = dist[len(dist)-1] +1
x_first = dist[0]

if files.has_key('xmin'):
    xmin =float(files['xmin'])
else:
    xmin = x_first

if files.has_key('xmax'):
    xmax =float(files['xmax'])
else:
    xmax = x_last

# The model distance; the spacing is not the same as in the data
dist_model = arange(xmin , xmax, offset)

# phi: the angle between the profile perpendicular to the ridge and the spreading direction
# gamma: the angle between the profile perpendicular to the ridge and the profile azimuth
# perp_azimuth: the angle of the profile perpendicular to the ridge
(phi, gamma, theta, perp_azimuth, contam) = get_angles(files)

# Projecting magnetized layer into perpendicular space (perpendicular to the strike of the
# ridge where the assumptions for 2-D calculations hold)
# The two parameters, phi and contam, control the projection. The phi is there to account for
# obliqueness and the contamination coefficient is there to smooth the model even more in the
# case of slow spreading ridges
(projected_mag_layer) = create_projected_magnetized_layer(mag_layer,files, phi, contam)

faults_and_rifts = create_faults_and_rifts(delta_lcopy, delta_rcopy)

# Projecting the distance from the oblique profile azimuth into perpendicular space including
# the contamination coefficient
perp_projected_dist = []
perp_projected_dist_model = []
if (gamma != 0 or contam != 1):
    # The magnetized block distance
    for distance in dist:
        perp_projected_dist.append(distance*cos(gamma)*contam)

    # The model distance
    for distance in dist_model:
        perp_projected_dist_model.append(distance*cos(gamma)*contam)
else:
    perp_projected_dist_model = dist_model
    perp_projected_dist = dist

# Projecting the distance from the oblique profile azimuth into flowline
# space so the data can be compared to the model
flowline_dist_anom = []

if (theta !=0):
    for distance in dist_model:
        flowline_dist_anom.append(distance*cos(theta))

```

```

else:
    flowline_dist_anom = dist_anom

# Here everything is in perpendicular space; we have accounted for oblique
# profile and oblique spreading. Contamination coefficient has been applied to
# the perp_projected_dist and perp_projected_dist_model but not projected_mag_layer,
# that is done in here
anom_model = create_anomaly_model(perp_projected_dist,perp_projected_dist_model,
deep,files,projected_mag_layer,perp_azimuth,contam)

##### Below are several useful lines that are helpful to write information to files #####
# Prints the location of the pseudofaults to the file 'pf'
f=open('pf', 'w')
for (dista,fault,rift, marker) in faults_and_rifts:
    if fault and not rift:
        if lat:
            lat_write = get_depth(dist_anom, lat, dista)
            lon_write = get_depth(dist_anom, lon, dista)
            f.write(str(dista) + " " + str(lat_write) + " " + str(lon_write) + " " + str(marker) + "\n")
        else:
            f.write(str(dista) + "\n")
f.close()

# Prints the location of the failed rifts to the file 'fr'
f=open('fr', 'w')
for (dista,fault,rift, marker) in faults_and_rifts:
    if rift and not fault:
        if lat:
            lat_write = get_depth(dist_anom, lat, dista)
            lon_write = get_depth(dist_anom, lon, dista)
            f.write(str(dista) + " " + str(lat_write) + " " + str(lon_write) + "\n")
        else:
            f.write(str(dista) + "\n")
f.close()

# Printing the ascarp
f=open('scarp', 'w')
for (dista,fault,rift, marker) in faults_and_rifts:
    if fault and rift:
        if lat:
            lat_write = get_depth(dist_anom, lat, dista)
            lon_write = get_depth(dist_anom, lon, dista)
            f.write(str(dista) + " " + str(lat_write) + " " + str(lon_write) + "\n")
        else:
            f.write(str(dista) + "\n")
f.close()

# Prints the data distance (in flowline space) and to the file 'tryggvi'
f=open('tryggvi','w')
for i in range(0,len(dist_anom)):
    f.write(str(dist_anom[i]) + " " + str(anom[i]) + "\n")
f.close()

# Prints the distance which goes with the depth and the depth
f=open('depth','w')
for i in range(0,len(dist)):
    f.write(str(dist[i]) + " " + str(deep[i]) + "\n")
f.close()

# Prints the model distance and model to the file 'asdis'
# The distance is different in asdis than in tryggvi because the model might
# be calculated at a different interval (because of the 'pointspacing' variable)
# and thus have a different distance vector.
f=open('asdis','w')

for i in range(0,len(dist_model)):
    f.write(str(dist_model[i]) + " " + str(anom_model[i]) + "\n")
f.close()

```

```

# The final model is viewed in flowline space
# If lat and lon are in the input file one can pick features on the plot and
# get their lat and lon
create_plot(dist, dist_model, flowline_dist_anom, deep, anom, mag_layer,
            faults_and_rifts, anom_model, configs, other_data, lat, lon, xmin,
            xmax, title, timeline_copy)

```

A.2 calc.py

```

# -*- coding: utf-8 -*-

"""
calc.py - core computations for magellan.

Copyright (C) 2008 Tryggvi Björnsson <tryggvib@hi.is>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
"""

import sys
import lib.Mag
from math import cos, sin, atan2, radians, degrees, sqrt, log, pi
from numpy import *
# Thickness has to be a decimal number
_default_thickness = '0.5'
_default_profile_azimuth = '90'
_default_inclination = '45'
_default_declination = '45'
_default_contam = '1.0'
_default_ridge_azimuth = '180'
_default_spreading_direction = '90'
_default_level = '0'
_default_paleoincl = 'float("infinity")'
_default_paleo_ridge_str = 'float("infinity")'
# The contam variable has to be global, it is used in two defs
contam = 1.0

def get_angles(parameters):
    """
    Retrieves various useful angles.
    phi: the angle between the profile perpendicular to the ridge
    and the flowline
    gamma: the angle between the profile perpendicular to the
    ridge and the profile azimuth
    perp_azimuth: the angle of the profile perpendicular to the ridge

    """
    # If paleo_ridge_strike of a ridge is defined, it is assigned
    # to ridge_azimuth
    paleo_ridge_str = radians(eval(parameters.pop('paleo_ridge_str',
    _default_paleo_ridge_str)))
    profile_azimuth = radians(eval(parameters.pop('profile_azimuth',
    _default_profile_azimuth)))
    ridge_azimuth = radians(eval(parameters.pop('ridge_azimuth',
    _default_ridge_azimuth)))

```

```

spreading_direction = radians(eval(parameters.pop('spreading_direction',
_default_spreading_direction)))
perp_azimuth = ridge_azimuth - pi/2
theta = abs(profile_azimuth-spreading_direction)

gamma = abs(perp_azimuth - profile_azimuth)
phi = abs(perp_azimuth - spreading_direction)

# We want the more acute angle
if gamma > pi/2 and gamma < 3*pi/2:
    gamma = pi-gamma
elif phi > 3*pi/2:
    gamma = 2*pi -gamma

if phi > pi/2 and phi < 3*pi/2:
    phi = pi-phi
elif phi > 3*pi/2:
    phi = 2*pi -phi

if theta > pi/2 and theta < 3*pi/2:
    theta = pi-theta
elif theta > 3*pi/2:
    theta = 2*pi -theta

print "phi:", phi*180/pi, "gamma:", gamma*180/pi, "perp",
perp_azimuth*180/pi, "theta", degrees(theta)

contam = eval(parameters.pop('contam', _default_contam))
return (phi, gamma, theta, perp_azimuth, contam)

def create_change_timeline(asym,spread,jump,magnet,time, markers):
    """
    creates a timeline of changes from arrays containing
    asymmetry, spreading rates, jumps, and timescale for
    easier lookup when processing bathymetry. Returns a
    list of tuples (each tuple contains start of period
    and dictionary of changing values) sorted by start
    of period:
    [(start_of_period, {change:value})]
    Note: Updates the asymmetry dictionary
    """

    asym_spread = set(asym).intersection(set(spread))
    for same in asym_spread:
        spread[same].update(asym[same])

    asym.update(spread)

    asym_magnet = set(asym).intersection(set(magnet))
    for same in asym_magnet:
        magnet[same].update(asym[same])

    asym.update(magnet)

    asym_time = set(asym).intersection(set(time))
    for same in asym_time:
        time[same].update(asym[same])

    asym.update(time)

    asym_jump = set(asym).intersection(set(jump))
    for same in asym_jump:
        jump[same].update(asym[same])
    asym.update(jump)

```

```

    asym_markers = set(asym).intersection(set(markers))
    for same in asym_markers:
        markers[same].update(asym[same])

    asym.update(markers)
    asym[0] = {'asymmetry':0, 'spreadingrate':0}
    return [(k,asym[k]) for k in sorted(asym.keys())]

def create_anomaly_model(projected_dist, projected_dist_model,
    deep, parameters, magnet_layer, perp_azimuth, contam):
    """
    creates an anomaly model from distance and depth.
    Other parameters needed are thickness, declination,
    inclination, azimuth, and magnetization
    The model is based on theoretical computations.
    Returns a list of depths sorted by distance in x
    direction: [depth]
    """

    thickness = eval(parameters.pop('thickness', _default_thickness))
    declination = radians(eval(parameters.pop('declination',
        _default_declination)))
    inclination = radians(eval(parameters.pop('inclination',
        _default_inclination)))
    level = eval(parameters.pop('level', _default_level))
    paleoincl = radians(eval(parameters.pop('paleoincl', _default_paleoincl)))
    paleo_ridge_str = radians(eval(parameters.pop('paleo_ridge_str',
        _default_paleo_ridge_str)))
    # Shifting the observation point to a level
    deep = deep + ones(len(deep))*level

    # There are two distance lists, one for the model [projected_dist_model]
    #(the model can have a different spacing than the data spacing) and the
    #other for the data [projected_dist] to find the corresponding depths at
    #those data points.
    mag_dict = {}

    min_mag_dict = 0
    max_mag_dict = 0

    for ((first_pos, last_pos), pol, magnet) in magnet_layer:
        min_value = projected_dist[next_index(projected_dist, first_pos)]
        max_value = projected_dist[next_index(projected_dist, last_pos)]
        if (max_value > max_mag_dict):
            max_mag_dict = max_value
        elif (min_value < min_mag_dict):
            min_mag_dict = min_value
            mag_dict[min_value] = (pol, magnet)

    sinI = sin(inclination)
    cosI = cos(inclination)
    cosCminD = cos(perp_azimuth-declination)
    cosC = cos(perp_azimuth)
    # if paleoinclination is defined the magnetized bodies have that
    # inclination, else they have the current inclination
    if paleoincl > 400:
        sinI_body = sinI
        cosI_body = cosI
    else:
        sinI_body = sin(paleoincl)
        cosI_body = cos(paleoincl)
    pol_direction = 1

    if paleo_ridge_str < 400:
        cosC_body = cos(paleo_ridge_str -2*pi*90/360)

```

```

else:
    cosC_body = cosC
model = array(zeros(len(projected_dist_model)))
# We can only calculate a model for the timespan of the timescale.
# We have to make sure that the x1's are defined in mag_dict
# (which is where information about reversals given input parameters is kept).
# We have two loops, the first one loops over the points in the bathymetry
# and the second one in the observation points (the points where the model
# is evaluated)

# We first loop over all the blocks that are within the observation area.
# The first point is the first point in the magnet_layer
start_index = next_index(projected_dist,min_mag_dict)
index1 = arange(start_index,len(projected_dist)-1)
mag = array([float(i) for i in zeros(len(index1))])

x = projected_dist[start_index]
i = 0
for position in range(start_index+1,len(projected_dist)):
    if x in mag_dict:
        if (mag_dict[x][0] == 'n'):
            pol_direction = 1
        else:
            pol_direction = -1
        mag_field = pol_direction*mag_dict[x][1]*pow(10,-7)
    x = projected_dist[position]
    mag[i] = mag_field
    i += 1

index1 = arange(start_index,len(projected_dist)-1)
dummy1 = zeros([len(projected_dist_model),len(index1)], 'f')
dummy2 = zeros([len(index1), len(projected_dist_model)], 'f')
x1 = array(dummy1)
x2 = array(dummy1)
z1 = array(dummy1)
z2 = array(dummy1)
z21 = array(dummy1)
mag_field = array(dummy1)
distance = array(dummy2)

for i in range(len(projected_dist_model)):
    x1[:,i] = projected_dist[start_index:len(projected_dist)-1]
    x2[:,i] = projected_dist[start_index+1:len(projected_dist)]
    z1[:,i] = deep[start_index:len(projected_dist)-1]
    z2[:,i] = deep[start_index+1:len(projected_dist)]
    mag_field[:,i] = mag
for i in range(len(index1)-1):
    distance[i][:] = projected_dist_model

distance = transpose(distance)
z3 = z1 + thickness
z4 = z2 + thickness
z1_pow2 = z1**2
z2_pow2 = z2**2
z3_pow2 = z3**2
z4_pow2 = z4**2

# Splitting the magnetization vector of the body into its x and z
# components. Geocentric dipole, declination is zero and inclination
# is the same as that of the Earth's field. In the case of an body that
# formed in a different place than where it is now, its inclination is
# different than the inclination today.
Jx = mag_field*cosI_body*cosC_body
Jz = mag_field*sinI_body
# Matrix method:
x1_calc = x1 - distance
x2_calc = x2 - distance

```

```

theta1 = arctan2(z2,x2_calc)
theta2 = arctan2(z4,x2_calc)
theta3 = arctan2(z3,x1_calc)
theta4 = arctan2(z1,x1_calc)

x2_calc_pow2 = x2_calc**2
x1_calc_pow2 = x1_calc**2

# Right surface; from (x2,z2) to (x2,z4) z21 = z4-z2
x12 = 0
r1 = sqrt(x2_calc_pow2 + z2_pow2)
r2 = sqrt(x2_calc_pow2 + z4_pow2)
P_r = (theta1-theta2)
Q_r = -1*log(r2/r1)
V_r = 2*(Jx*Q_r - Jz*P_r)
H_r = 2*(Jx*P_r + Jz*Q_r)
T_r = V_r*sinI + H_r*cosI*cosCminD

# Left surface; from (x1,z3) to (x1,z1)
z21 = z1-z3
x12 = 0
r1 = sqrt(x1_calc_pow2 + z3_pow2)
r2 = sqrt(x1_calc_pow2 + z1_pow2)
P_l = (theta3-theta4)
Q_l = -1*log(r2/r1)
V_l = 2*(Jx*Q_l - Jz*P_l)
H_l = 2*(Jx*P_l + Jz*Q_l)
T_l = V_l*sinI + H_l*cosI*cosCminD

# Top surface; from (x1,z1) to (x2,z2)
z21 = z2-z1
x12 = (x1 - x2)

r1 = sqrt(x1_calc_pow2 + z1_pow2)
r2 = sqrt(x2_calc_pow2 + z2_pow2)
const1 = z21**2/(z21**2 + x12**2)
const2 = z21*x12/(z21**2 + x12**2)

P_t = const1*(theta4 - theta1) + const2*log(r2/r1)
Q_t = const2*(theta4-theta1) - const1*log(r2/r1)
V_t = 2*(Jx*Q_t - Jz*P_t)
H_t = 2*(Jx*P_t + Jz*Q_t)
T_t = V_t*sinI + H_t*cosI*cosCminD
# Bottom surface; from (x2,z4) to (x1,z3)
z21 = z3-z4
x12 = (x2-x1)
r1 = sqrt(x2_calc_pow2 + z4_pow2)
r2 = sqrt(x1_calc_pow2 + z3_pow2)
const1 = z21**2/(z21**2 + x12**2)
const2 = z21*x12/(z21**2 + x12**2)
P_b = const1*(theta2-theta3) + const2*log(r2/r1)
Q_b = const2*(theta2-theta3) - const1*log(r2/r1)
V_b = 2*(Jx*Q_b - Jz*P_b)
H_b = 2*(Jx*P_b + Jz*Q_b)
T_b = V_b*sinI + H_b*cosI*cosCminD

# If the field is reversed we have sinI changing sign
# (sinI=-sin(-I)) and cosCminD changing sign (cos(C) = -cos(180-C)
# and therefore we can just multiply the total field by -1 for a
# reversed block.
model = (T_b + T_t + T_l + T_r) *pow(10,9)

model = add.reduce(transpose(model))
return model

def inv_project_anomaly_model(dist, angle):

```

```

"""
projects the anomaly_model back to the original track.
"""

projected_dist = []

for value in dist:
    projected_dist.append(value/cos(angle))

return projected_dist

def project(distance,angle,contam):
    """
    project distance on to a profile oriented angle away from the
    original profile
    """
    projected_distance = []

    for value in distance:
        projected_distance.append(value*cos(angle)*contam)

    return projected_distance

def create_deltax(timeline, tmin):
    """
    Create the total difference traveled between
    changes in a timeline given a change timeline
    as input. Returns a tuple of lists containing
    a tuple of distance and a dictionary which
    contains polarity, pseudo-faults and failed
    rifts:
    ((distance, {polarity,pseudofaults,failed rift})),
    ((distance, {polarity,pseudofaults,failed rift}))

    The former tuple is distances in left direction and
    the latter tuple is distances in right direction

    polarity is either the string 'n' (normal) or 'r' (reverse)
    pseudofault and failed rift are booleans True if it is either
    """

    # Delta movement in right direction
    deltax_r = []
    # Delta movement in left direction
    deltax_l = []
    # The part of the time that isn't used for the model
    timeline_rest = []
    action_new = {}
    time_old = []

    # Initialize with time zero
    (prev_time, action) = timeline.pop(0)

    if action.has_key('polarity'):
        polarity = action['polarity']
    if action.has_key('spreadingrate'):
        spreading_rate = action['spreadingrate']
    asymmetry = 0 # Default asymmetry
    if action.has_key('asymmetry'):
        asymmetry = action['asymmetry']

    magnetization = 10 #A/m -- # Default magnetization
    if action.has_key('magnetization'):
        magnetization = action['magnetization']

    if action.has_key('jump'):
        jump = action['jump']

```

```

markers = action['markers']
if (jump == 0):
    failed_rift = True
    if (jump > 0):
        (deltax_r, deltax_l) = _jump_reposition(deltax_r,deltax_l,jump)
    if (jump < 0):
        (deltax_l, deltax_r) = _jump_reposition(deltax_l,deltax_r,jump)
    pseudo_fault = True

pseudo_fault = False # Let know of pseudo faults
failed_rift = False
markers = None
# Spreading rate in right direction
spread_r = spreading_rate * (1 + asymmetry)

# Spreading rate in left direction
spread_l = spreading_rate * (1 - asymmetry)

for (time,action) in timeline:
    if time_old:
        timeline_rest.append((time_old, action))
        time_old = []
    # We start calculating the model where tmin is
    if abs(prev_time) > tmin and abs(time) <= tmin:
        action_new['spreadingrate'] = spreading_rate
        action_new['asymmetry'] = asymmetry
    if action.has_key('jump'):
        action_new['jump'] = action['jump']
        action_new['markers'] = action['markers']
    action['jump'] = 0
    xremove_l = spread_l*(time - tmin)
    xremove_r = spread_r*(time - tmin)

    if time == -1*tmin:
        time_old=[]
    else:
        time_old = time
    time = -1*tmin
    timeline_rest.append((time, action_new))

    # Figure out how much lithosphere, on each site of the ridge,
    # needs to be removed from the bathymetry
    if -1*time < tmin:
        timeline_rest.append((time, action))
        continue

    delta_t = time - prev_time
    # Delta distance in right direction
    distance_r = delta_t * spread_r
    # Delta distance in left direction
    distance_l = -delta_t * spread_l

    deltax_r.append((distance_r, {'polarity':polarity,
        'magnetization':magnetization,
        'pseudo fault':pseudo_fault,
        'failed rift':failed_rift,
        'start':time,
        'end':prev_time,
        'marker':markers,
        'spreadingrate':spread_r}))
    deltax_l.append((distance_l, {'polarity':polarity,
        'magnetization':magnetization,
        'pseudo fault':pseudo_fault,
        'failed rift':failed_rift,
        'start':time,

```

```

        'end':prev_time,
        'marker':markers,
        'spreadingrate':spread_l}))
pseudo_fault = False
failed_rift = False
if action.has_key('polarity'):
    polarity = action['polarity']

if action.has_key('magnetization'):
    magnetization = action['magnetization']

if action.has_key('spreadingrate'):
    spreading_rate = action['spreadingrate']
    spread_r = spreading_rate * (1 + asymmetry)
    spread_l = spreading_rate * (1 - asymmetry)

if action.has_key('asymmetry'):
    asymmetry = action['asymmetry']
    spread_r = spreading_rate * (1 + asymmetry)
    spread_l = spreading_rate * (1 - asymmetry)
# Jump computations
if action.has_key('jump'):
    jump = action['jump']
    markers = action['markers']
    if jump == 0:
        failed_rift=True
        if (jump > 0):
            (deltax_r, deltax_l) = _jump_reposition(deltax_r,deltax_l,jump)
        if (jump < 0):
            (deltax_l, deltax_r) = _jump_reposition(deltax_l,deltax_r,jump)

    pseudo_fault = True

    prev_time = time

deltax_l.reverse()
deltax_r.reverse()

return (deltax_l, deltax_r, timeline_rest)
def _jump_reposition(jump_in, move_to, jump):
    """
    Repositions the ridge axis when a jump occurs.
    The method takes in the jump_in which represents
    the half which the jump will land in and moves
    every distance up to the jump point into the
    move_to half, setting pseudo faults and failed rifts
    accordingly. Returns the recomputed halves in a
    tuple: (jump_in, move_to)
    """
    pseudo_fault = False
    failed_rift = True

    (dx,prefs) = jump_in.pop()
    while(abs(jump) > abs(dx)):
        fr_tmp = failed_rift
        failed_rift = prefs['failed rift']
        prefs['failed rift'] = fr_tmp

        pf_tmp = pseudo_fault
        pseudo_fault = prefs['pseudo fault']
        prefs['pseudo fault'] = pf_tmp

    start_tmp = prefs['start']
    end_tmp = prefs['end']

```

```

    prefs['start'] = end_tmp
    prefs['end'] = start_tmp

    move_to.append((-dx,prefs))
    jump -= dx

    (dx,prefs) = jump_in.pop()
    new_prefs = {}

    new_prefs['polarity'] = prefs['polarity']
    new_prefs['magnetization'] = prefs['magnetization']
    new_prefs['start'] = prefs['end']
    new_prefs['end'] = prefs['start']
    new_prefs['failed rift'] = failed_rift
    new_prefs['pseudo fault'] = pseudo_fault
    new_prefs['spreadingrate'] = prefs['spreadingrate']
    new_prefs['marker'] = prefs['marker']

    move_to.append((-jump, new_prefs))
    jump_in.append((dx-jump),prefs))

    return(jump_in,move_to)

def create_magnetized_layer(deltax_l, deltax_r):
    """
    create a magnetized layer between min_l (maximum distance
    to the left - or minimum distance since it is a negative
    number) and max_r (maximum distance to the right) from the
    two halves. Returns a list of tuples. Each tuple includes
    another tuple with start and end positions, and the
    polarity and magnetization for that distance:
    [(start,end),polarity]
    """

    magnetized_layer = []
    distance_sum = 0
    start = 0

    sum_right = []
    sum_left = []

    for (deltax, point_prefs) in deltax_r:

# Initializing the polarity and magnetization
    if (distance_sum == 0):
        polarity = point_prefs['polarity']
        magnet = point_prefs['magnetization']

        if ((polarity != point_prefs['polarity'] or magnet !=
            point_prefs['magnetization'])):
            sum_right.append(((start, distance_sum), polarity, magnet))
            start = distance_sum
        polarity = point_prefs['polarity']
        magnet = point_prefs['magnetization']
        distance_sum += deltax
    # Adding the last block
    sum_right.append(((start, distance_sum), polarity, magnet))

    distance_sum = 0
    start = 0

    for (deltax, point_prefs) in deltax_l:

    if (distance_sum == 0):

```

```

# Initializing the polarity and magnetization
polarity = point_prefs['polarity']
magnet = point_prefs['magnetization']

if ((polarity != point_prefs['polarity'] or magnet !=
point_prefs['magnetization'])):

    sum_left.append(((distance_sum, start), polarity, magnet))
        start = distance_sum
    polarity = point_prefs['polarity']
        magnet = point_prefs['magnetization']
        distance_sum += deltax

# Adding the last block
sum_left.append(((distance_sum, start), polarity, magnet))
sum_left.reverse()

((start_l,end_l),polarity,magnet) = sum_left[-1]
((start_r,end_r),polarity,magnet) = sum_right[0]

magnetized_layer.extend(sum_left[:-1])
magnetized_layer.append(((start_l,end_r),polarity,magnet))
magnetized_layer.extend(sum_right[1:])

return magnetized_layer

def create_projected_magnetized_layer(magnetized_layer,parameters, phi, contam):
    """
    projects previously made magnetized layer onto a profile
    perpendicular to the ridge, if the original profile is oblique.
    Angle of projection is 90 - abs(ridge_orientation - track_orientation).
    """

    # Contamination coefficient applied to the magnetized layer.
    projected_magnetized_layer = []
    for ((start,end),polarity, magnet) in magnetized_layer:
        new_start1 = start * cos(phi) * contam
        new_end1 = end * cos(phi) * contam
        projected_magnetized_layer.append(((new_start1,new_end1),polarity,magnet))

    return (projected_magnetized_layer)

def create_faults_and_rifts(deltax_l, deltax_r):
    """
    """

    faults_and_rifts = []

    distance_sum = 0

    for (deltax, point_prefs) in deltax_l:
        distance_sum += deltax
        if (point_prefs['pseudo fault'] or point_prefs['failed rift']):
            faults_and_rifts.append((distance_sum,
                point_prefs['pseudo fault'],
                point_prefs['failed rift'],

                point_prefs['marker']))
            distance_sum = 0

    for (deltax, point_prefs) in deltax_r:
        distance_sum += deltax
        if (point_prefs['pseudo fault'] or point_prefs['failed rift']):
            faults_and_rifts.append((distance_sum,
                point_prefs['pseudo fault'],
                point_prefs['failed rift'],

                point_prefs['marker']))

```

```

    return faults_and_rifts

def next_index(indexed_list, search_value):
    """
    Returns the index of the number in indexed_list that search_value is closest to.
    """

    # If the search_value is smaller than the smallest number in
    # the indexed_list (which is always distance), return the
    # index of the smallest number in indexed_list
    if search_value < min(indexed_list):
        return indexed_list.index(min(indexed_list))

    for index in range(len(indexed_list)):
        if index == len(indexed_list)-1:
            return index

    if (search_value < indexed_list[index+1]) and (search_value >= indexed_list[index]):
        if (indexed_list[index+1] - search_value) > (search_value - indexed_list[index]):
            return index
        else:
            return index+1

def next_upper(indexed_list, search_value):
    """
    Returns the index of the number in indexed_list that search_value
    is rounded UP to.
    """

    # If the search_value is smaller than the smallest number in the
    # indexed_list (which is always distance), return the index of the
    # smallest number in indexed_list
    if search_value < min(indexed_list):
        return indexed_list.index(min(indexed_list))

    # If the search_value is bigger than the biggest number in the
    # indexed_list (which is always distance), return the index of the
    # biggest number in indexed_list
    if search_value > max(indexed_list):
        return indexed_list.index(max(indexed_list))

    for index in range(len(indexed_list)):
        if index == len(indexed_list)-1:
            return index

    if (search_value <= indexed_list[index+1]) and (search_value > indexed_list[index]):
        return index+1

def next_lower(indexed_list, search_value):
    """
    Returns the index of the number in indexed_list that search_value
    is rounded DOWN to.
    """

    # If the search_value is smaller than the smallest number in the
    # indexed_list (which is always distance), return the index of the
    # smallest number in indexed_list
    if search_value < min(indexed_list):
        return indexed_list.index(min(indexed_list))

    # If the search_value is bigger than the biggest number in the
    # indexed_list (which is always distance), return the index of
    # the biggest number in indexed_list
    if search_value > max(indexed_list):
        return indexed_list.index(max(indexed_list))

```



```

# Delta movement in left direction
deltax_l = 0

# Initialize with time zero
(prev_time, action) = timeline.pop(0)

if action.has_key('spreadingrate'):
spreading_rate = action['spreadingrate']
asymmetry = 0 # Default asymmetry
if action.has_key('asymmetry'):
    asymmetry = action['asymmetry']

if action.has_key('jump'):
jump = action['jump']
deltax_r -= jump
deltax_l += jump

pseudo_fault = False # Let know of pseudo faults
# Spreading rate in right direction
spread_r = spreading_rate * (1 + asymmetry)

# Spreading rate in left direction
spread_l = spreading_rate * (1 - asymmetry)

for (time,action) in timeline:
    delta_t = time - prev_time
    # Delta distance in right direction
    deltax_r += delta_t * spread_r
    # Delta distance in left direction
    deltax_l += delta_t * spread_l
    pseudo_fault = False

    if action.has_key('spreadingrate'):
        spreading_rate = action['spreadingrate']
        spread_r = spreading_rate * (1 + asymmetry)
        spread_l = spreading_rate * (1 - asymmetry)

    if action.has_key('asymmetry'):
        asymmetry = action['asymmetry']
        spread_r = spreading_rate * (1 + asymmetry)
        spread_l = spreading_rate * (1 - asymmetry)
    if action.has_key('jump'):
        jump = action['jump']
        deltax_r -= jump
        deltax_l += jump

    prev_time = time

return (deltax_l, deltax_r)

```

A.3 plot.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
plot.py - plots data for magellan
```

```
Copyright (C) 2008 Tryggvi Bj\{o}rgvinsson <tryggvib@hi.is>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<http://www.gnu.org/licenses/>>.
"""

```

from pylab import *
import lib.Mag
from calc import *
from numpy import *
_default_thickness = '0.5'
xdata = []
marker = []

# Defining the key-press-events;
# p is a pseudofault
# f is a failed rift
# a is anomaly
# You can put your own letters in there ...
def key_press(event):
    'whenever a key is pressed'

    if event.inaxes:
        if event.key == 'p':
            marker.append('p')
            xdata.append(event.xdata)
        elif event.key == 'f':
            marker.append('f')
            xdata.append(event.xdata)
        elif event.key == 'a':
            marker.append('a')
            xdata.append(event.xdata)

def create_plot(dist, dist_model, dist_anom, deep, anom, layer,
faultrift, model, parameters, other_data, lat ,lon, xmin, xmax,title,timeline):
    """
    Plot bathymetry profiles from distance, depth,
    anomalies, the magnetic layer and a model.
    Uses matplotlib to plot a nice graph.
    """

    thickness = eval(parameters.get('thickness', _default_thickness))
    fig = figure(figsize=(12,8))
    fig.suptitle(title, fontweight='bold', fontsize=14)
    if other_data:
        anomplot = fig.add_subplot(311)
        anomplot.set_position([0.2, 0.3, 0.7, 0.38])
        bathplot = fig.add_subplot(312,sharex=anomplot)
        bathplot.set_position([0.2, 0.06, 0.7, 0.18])
        otherplot = fig.add_subplot(313, sharex=anomplot)
        otherplot.set_position([0.2, 0.73, 0.7, 0.18])
        # Data given in the 4th column of the data file.
        otherplot.set_title('Gravity')
        otherplot.set_ylabel('mGal')
        otherplot.plot(dist_anom, other_data,'black')
    else:
        anomplot = fig.add_subplot(211)
        anomplot.set_position([0.2, 0.35, 0.7, 0.55])
        bathplot = fig.add_subplot(212,sharex=anomplot)
        bathplot.set_position([0.2, 0.1, 0.7, 0.18])

##### Text parameters #####
props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
timeline.reverse()

```

```

y = 0.9
x = 0.02
incr = 0.03
fig.text(x ,y,'Spreading Rates', style='italic', fontsize=12, bbox={})
prev_time_spr = 0
prev_time_asymm = 0
prev_time_magnet = 0
for (time, action) in timeline:
    if time == 0:
        continue
    if action.has_key('spreadingrate'):
        spr = action['spreadingrate']
        y = y - incr
        time = -1*time
        fig.text(x,y,str(prev_time_spr) + '-' + str(time) + ': '+str(spr))
        prev_time_spr = time
y = y - incr - 0.01
fig.text(0.02,y,'Jumps', style='italic')
for (time, action) in timeline:
    if time == 0:
        continue
    if action.has_key('jump'):
        jump = action['jump']
        y = y - incr
        time = -1*time
        fig.text(x,y,str(time)+' ': +str(jump))
y = y - incr - 0.01
fig.text(0.02,y,'Magnetization', style='italic')
for (time, action) in timeline:
    if time == 0:
        continue
    if action.has_key('magnetization'):
        mag = action['magnetization']
        y = y - incr
        time = -1*time
        fig.text(x,y,str(prev_time_magnet) + '-' + str(time)+ ': ' + str(mag))
        prev_time_magnet = time
y = y - incr -0.01
fig.text(0.02,y,'Asymmetry', style='italic')
for (time, action) in timeline:
    if time == 0:
        continue
    if action.has_key('asymmetry'):
        asymm = action['asymmetry']
        y = y - incr
        time = -1*time
        fig.text(x,y,str(prev_time_asymm) + '-' + str(time)+ ': ' + str(asymm))
        prev_time_asymm = time

##### END of TEXT parameters #####

anomplot.set_title('Anomalies')
anomplot.set_ylabel('nT')
anomplot.plot(dist_anom, anom, '#330099') # Data
anomplot.plot(dist_model, model, '#FF0000') # Model
anom_leg = anomplot.legend(('Data','Model'))

bathplot.set_title('Bathymetry')
bathplot.set_xlabel('km')
bathplot.set_ylabel('km')
stuff=[]
for index in range(len(deep)):
    deep[index] = deep[index]*-1
    stuff.append(0)

deepthick = map(lambda x: x-thickness, deep)
# Opening the file 'blocks' which holds information about the blue
# and white blocks. It has to be opened here, before the for loop and

```

```

# closed after the for loop
f=open('blocks','w')

for ((start,end),polarity,magnet) in layer:
    if polarity == 'n': fillcolor = 'b'
    else: fillcolor = 'w'

# The next index above start
    index_lower = next_upper(dist,start)
# The next index below end
    index_upper = next_lower(dist,end)

# Finding the exact depth at start and end
    y_start = get_depth(dist,deep,start)
    y_end = get_depth(dist,deep, end)
    y_start_lower = y_start-thickness
    y_end_lower = y_end-thickness

    if (index_lower == index_upper +1):
        xs = [start,end]
        lys = [y_start, y_end]
        tys = [y_start_lower, y_end_lower]
    else:
        xs = [start]+dist[index_lower:index_upper+1]+[end]
        lys = [y_start]+deep[index_lower:index_upper+1]+[y_end]
        tys = [y_start_lower]+ deepthick[index_lower:index_upper+1] + [y_end_lower]

    x = concatenate( (xs, xs[::-1]) )
    y = concatenate( (lys, tys[::-1]) )
    m = []
    depth = []

    if fillcolor == 'b':
        for i in range(len(xs)):
            m.append(xs[i])
    depth.append(lys[i])
    if fillcolor == 'w':
        for i in range(len(xs)):
            m.append(xs[i])
            depth.append(-1*lys[i])

# Writing to file the organisation of the blue and white blocks in a GMT format
if (end >= dist_anom[0] and start <= dist_anom[-1]):
    if fillcolor == 'b':
        f.write("> -Gblack\n")
        for i in range(len(x)):
            f.write(str(x[i])+ " " + str(y[i]) + "\n")
    else:
        f.write("> -Gwhite\n")
        for i in range(len(x)):
            f.write(str(x[i])+ " " + str(y[i]) + "\n")

    bathplot.fill(x, y, facecolor=fillcolor)

f.close()

# Fix to get one entry per fault/rift in legend
fault_plotted = False
rift_plotted = False
dx = 0.3
for (position,fault,rift, marker) in faultrift:
    if fault:
        anomplot.axvspan(position-dx,position+dx,
            facecolor='#339900',edgecolor='none')
    if other_data:
        otherplot.axvspan(position-dx,position+dx,
            facecolor='#339900',edgecolor='none')
    if fault_plotted:

```

```

        bathplot.axvspan(position-dx,position+dx,
                        facecolor='#339900',edgecolor='none')
    else:
        bathplot.axvspan(position-dx,position+dx,
                        facecolor='#339900',edgecolor='none',
                        label="Pseudofault")
        fault_plotted = True
    if rift:
        anomplot.axvspan(position-dx,position+dx,
                        facecolor='r',edgecolor='none')
    if other_data:
        otherplot.axvspan(position-dx,position+dx,
                        facecolor='r',edgecolor='none')
    if rift_plotted:
        bathplot.axvspan(position-dx,position+dx,
                        facecolor='r',edgecolor='none')
    else:
        bathplot.axvspan(position-dx,position+dx,
                        facecolor='r',edgecolor='none', linewidth=4,
label="Failed rift")
        rift_plotted = True

    bathplot.legend(loc=1)

    bathplot.plot(dist, deep, linewidth=1.0)
    anomplot.plot(dist, stuff, linewidth=0.5)
    bathplot.set_xlim(xmin, xmax)
    bathplot.set_ylim(min(deepthick),0)
    connect('key_press_event', key_press)

    # If you don't want to see the output, just comment this line out
    show()

    # Writing the lat and lon of features to a file
    # dist_anom, lat, lon
    f=open('picks','w')

    for i in range(len(xdata)):
        lat_use = get_depth(dist_anom, lat, xdata[i])
        lon_use = get_depth(dist_anom, lon, xdata[i])
        f.write(str(marker[i]) + " " + str(xdata[i]) + " " + str(lat_use) +
        " " + str(lon_use) + " " + "\n")

    f.close()

```

A.4 data.py

```

# -*- coding: utf-8 -*-

"""
data.py - gathers data from different files and returns data structures

Copyright (C) 2008 Tryggvi Bj\{o}rgvinsson <tryggvib@hi.is>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
"""

```

```

import os,sys,re
import lib.Mag
from numpy import *

# Filenames for default files
data_path = os.path.split(lib.Mag.__file__)[0]
_default_timescale = os.path.join(data_path, 'data', 'candekent.dat')

def _read_period_file(period_file, key_column, value_column):
    """
    A generator function which reads information from files
    with time periods and a specific value for those periods
    Theoretically it can be used for similar, non-time period
    files since it returns the content of given columns (through
    the parameters key_column and value_column). Returns a tuple
    containing the evaluated value of the key column and the
    value column
    """

    filepath = os.path.expanduser(period_file)
    file_content = open(filepath).read()
    lines = file_content.splitlines()
    for line in lines:
        """
        Format of files:
        period_start period_end value

        where
        period_start is start of period in myrs (million years)
        period_end   is end of period in myrs
        value        is the value for that period

        % at start of line is a comment

        Observe that period_end is only for the user's convenience
        and the only period_start is used to indicate the asymmetry
        of a period until next period starts
        """

        #Ignore comments and blank lines
        if re.match('^(%)|^#|(\s*$)',line):
            continue
        #Split line into columns
        columns = line.split()

        yield (eval(columns[key_column]), eval(columns[value_column]))

def get_asymmetry(tend, asymmetry_file=None, key_name='asymmetry'):
    """
    retrieves asymmetry percentage and corresponding time.
    Returns a dictionary consisting of
    {start_of_period:asymmetry_for_period}
    i.e. beginning of time period maps to asymmetry of
    that particular period.
    If no file is provided an empty directory is returned.
    """

    if asymmetry_file is None: return {}

    asymmetry = {}

    for (key,value) in _read_period_file(asymmetry_file,1,2):
        if key >= tend:
            asymmetry[-tend] = {key_name:value}
            break
        asymmetry[-key] = {key_name:value}

```

```

return asymmetry

def get_magnetization(tend, magnetization_file=None):
    """
    retrieves magnitude of magnetization and corresponding time.
    Returns a dictionary consisting of
    {start_of_period:magnetization_for_period}
    i.e. beginning of time period maps to magnetization of
    that particular period.
    If no file is provided an empty directory is returned
    """

    if magnetization_file is None: return {}

    # It is actually the same process as with asymmetry, so we use
    # the asymmetry function
    return get_asymmetry(tend, magnetization_file,'magnetization')

def get_jumps(tend, jump_file=None):
    """
    retrieves jump distances and corresponding time.
    If no file is provided it is assumed that no jumps
    have taken place. Returns a dictionary consisting of
    {time_of_jump:jump_distance}
    i.e. time of jump maps to distance of that particular
    jump.
    """

    if jump_file is None: return ({},{})

    # I want to mark the jumps with numbered identifiers
    filepath = os.path.expanduser(jump_file)
    file_content = open(filepath).read()
    lines = file_content.splitlines()
    for line in lines:
        #Ignore comments and blank lines
        if re.match('^(%)|^#|(\s*$)',line):
            continue
        #Split line into columns
        columns = line.split()

    jumps = {}
    markers = {}
    if len(columns) == 3:
        for (key,value) in _read_period_file(jump_file,0,2):
            if key >= tend:
                break
            markers[-key] = {'markers':value}

        for (key,value) in _read_period_file(jump_file,0,1):
            if key >= tend:
                break
            jumps[-key] = {'jump':value}
    else:
        for (key,value) in _read_period_file(jump_file,0,1):
            if key >= tend:
                break
            jumps[-key] = {'jump':value}
            markers[-key] = {'markers':None}

    return (jumps,markers)

def get_spreadingrate(tend, spreadingrate_file):
    """
    retrieves spreading rates and corresponding time. A
    filename must be provided. Returns a dictionary
    consisting of {start_of_period:spreading_rate_for_period},
    i.e. beginning of time period maps to spreading rate of

```

```

that particular period.
"""
if spreadingrate_file is None: return {}

spr_rates = {}

for (key,value) in _read_period_file(spreadingrate_file,1,2):
if key >= tend:
    spr_rates[-tend] = {'spreadingrate':(value/2)}
    break
    spr_rates[-key] = {'spreadingrate':(value/2)}

return spr_rates

def get_timescale(tend, timescale=None):
    """
    gets the reversed timescale. If no input file is
    provided, use default file of Cande Kent from '95.
    Assumes normal polarity at time zero.
    Returns reversed timescale as a list of tuples:
    [(name of period, start of period, end of period)...]
    """

    if timescale is None: timescale = _default_timescale

    reversed_timescale = {}
    filepath = os.path.expanduser(timescale)
    file_content = open(filepath).read()
    lines = file_content.splitlines()

    number_of_periods = len(lines)
    polarity='n'

    for (key,value) in _read_period_file(filepath,1,0):
        tend_tmp = max(key, value)
        if key >= tend and value < tend:
            reversed_timescale[-tend] = {'polarity':polarity}
            break
            reversed_timescale[-key] = {'polarity':polarity}
            #Swap polarities
            if polarity == 'n': polarity = 'r'
            else: polarity = 'n'

    tend_new = min(tend_tmp, tend)

    return (reversed_timescale,tend_new)

def get_trackdata(input_file, xmin, xmax,offset):
    """
    gathers data from track file. Input file must be
    provided. Data gathered is distance, anomaly and
    depth. Returns...
    """

    distance = []
    anomaly = []
    depth = []
    other_data = []
    lat = []
    lon = []
    if input_file:
        filepath = os.path.expanduser(input_file)
        file_content = open(filepath).read()
        lines = file_content.splitlines()
        for line in lines:
            """
            Format of file:
            distance longitude latitude depth anomaly

```

```

where
distance    is the distance in kilometers
longtitude  is longditude coordinate (not needed)
latitude    is latitude coordinate (not needed)
depth       is depth in kilometers from ocean top
(must be negated)
anomaly     is anomaly of magnetic measurements in nanoTesla

% at start of line is a comment
"""

# Ignore comments and blank lines
if re.match('^(%)|(\s*$)',line):
    continue

columns = line.split()
distance.append(eval(columns[0])) # Evaluate as number
depth.append(eval(columns[1])) # Evaluate as number and negate
anomaly.append(eval(columns[2])) # Evaluate as number
if len(columns) == 6:
    other_data.append(eval(columns[3])) # Evaluate as number
lat.append(eval(columns[5]))
lon.append(eval(columns[4]))
if len(columns) == 4:
    other_data.append(eval(columns[3]))
if distance[0] > distance[1]:
    distance.reverse()
    depth.reverse()
    anomaly.reverse()

# Making sure the bathymetry extends all the way through tend (data
# might not extend that far so we add flat bathy)
addl = int(abs(distance[0]-floor(xmin)))
addr = int(abs(distance[-1]-ceil(xmax)))
if distance[0] > xmin:
    distance_first = [distance[0] - 1 - x for x in range(addl)]
    distance_first.reverse()
    distance_calc = distance_first + distance + [x + 1 + distance[-1]
for x in range(addr)]
    depth_calc = [depth[0]]*addl + depth + [depth[-1]]*addr
else:
    distance_calc = distance
    depth_calc = depth
else:
    distance = array(arange(xmin,xmax,offset))
    anomaly = zeros(len(distance))
    # If there is no data given this is the default depth of the source layer
    depth = ones(len(distance))*2.0
    distance = distance.tolist()
    depth = depth.tolist()
    distance_calc = distance
    depth_calc = depth
    anomaly = anomaly.tolist()

return (distance_calc, depth_calc, distance, anomaly, other_data, lat, lon)

def get_configurations(config_file=None):
    """
    Go through a configuration file (project file)
    where the user can set different values to settings
    otherwise provided through flags. Makes it easier
    to manage flags and use magellan
    Returns an empty dictionary if no file is provided
    """

    configurations = {}
    if config_file is None: return configurations

```

```

file_content = open(config_file).read()

lines = file_content.splitlines()
for line in lines:
    """
    Format of files:
    item=value

    where
    item is the configurable setting
    value is the value for that item

    % at start of line is a comment

    Configurations which do something in magellan are:

    asymmetry = location of asymmetry file
    data = location of data file (track data)
    jump = location of jumps file
    magnetization = location of magnetization file
    spreading rate = location of spreading rate file

    declination = amount of declination
    inclination = amount of inclination
    azimuth = azimuth of the ridge relative to north
    thickness = thickness of magnetized layer
    contam = value of contamination coefficient
    pointspacing = spacing between points in calculation

    graphs = which graphs to plot (not implemented yet)
    """

    # Ignore comments and blank lines
    if re.match('^(%)|(\s*$)',line):
        continue

    # Split line into columns by '='
    # Remove preceeding and trailing whitespaces
    # Pick out the words on each side of
    pattern = re.compile(r'^\s*(\w.*\w)\s*=\s*(\S.*\S)\s*$')
    match = pattern.match(line)

    configurations[match.group(1).lower()] = match.group(2)

return configurations

```