



UNIVERSIDADE FEDERAL FLUMINENSE

Instituto De Geociências

Departamento De Geologia e Geofísica

Projeto Final em Geofísica II

Gean Lucas Mello Farias

***Modelagem Gravimétrica Crustal 2-D na Bacia de Pelotas Utilizando
Informações a Priori de Seções Sísmicas e da Geologia Regional***

Niterói

Julho, 2017

Gean Lucas Mello Farias

*Modelagem Gravimétrica Crustal 2-D na Bacia de Pelotas Utilizando
Informações a Priori de Seções Sísmicas e da Geologia Regional*

*Projeto Final em Geofísica II submetido à Universidade
Federal Fluminense como parte dos requisitos para a
obtenção do grau de Bacharel em Geofísica*

Orientador: Dr. Alexandre Motta Borges

Universidade Federal Fluminense

Instituto de Geociências

Departamento de Geologia e Geofísica

Niterói

Julho, 2017

*Modelagem Gravimétrica Crustal 2-D na Bacia de Pelotas Utilizando
Informações a Priori de Seções Sísmicas e da Geologia Regional*

*Projeto Final em Geofísica II submetido à Universidade
Federal Fluminense como parte dos requisitos para a
obtenção do grau de Bacharel em Geofísica*

Orientador: Dr. Alexandre Motta Borges

Aprovado em 20 de julho de 2017.

Comissão examinadora:

Dr. Alexandre Motta Borges

Dr. Cleverson Guizan Silva

Dr. Wagner Moreira Lupinacci

RESUMO

O presente trabalho visou à modelagem de um modelo geológico transversal à linha de costa da Bacia de Pelotas que seja coerente com dados gravimétricos da região. Para tanto se utilizou resultados de interpretações de seções sísmicas localizadas na mesma linha do perfil gravimétrico usado na modelagem, além de informações retiradas da bibliografia da geologia regional, com a finalidade de gerar parâmetros necessários, como profundidade das camadas e densidades, para o modelo inicial. O modelo gerado mostrou-se coerente com os dados gravimétricos e com a geologia, demonstrando como a modelagem direta gravimétrica pode desempenhar um papel importante na investigação geológica em alta profundidade, até mesmo subcrustal, ou em regiões inóspitas em relação ao sinal sísmico.

ABSTRACT

This study aimed to implement a cross-geological model to the shoreline, including the continent-ocean boundary, of the Pelotas Basin that is consistent with gravimetric data of the region. For this purpose we used the results of interpretation of seismic sections located on the same line gravimetric profile used in modelling, and information taken from the regional geology literature, in order to generate the required parameters, such as depth layers and densities for the initial model. The generated model was consistent with the gravimetric data and geology, showing how the Gravimetric direct modelling can play an important role in geological investigation for high depth, even subcrustal, or inhospitable regions in relation to the seismic signal.

ÍNDICE DE FIGURAS

Figura 1 - Delimitação da Bacia de Pelotas sobre a cobertura de dados de anomalia bouguer da GETECH.....	7
Figura 2 - Carta estratigráfica da Bacia de Pelotas por Bueno et al., 2007	11
Figura 3 - Seção sísmica da sub-bacia do norte da Bacia de Pelotas interpretada por Stica et al., 2014	13
Figura 4 - Seção sísmica interpretada da sub-bacia sul da Bacia de Pelotas.....	14
Figura 5 - Esquema da formação dos SDRs (Hinz et al., 1981, apud Mutter, 1985). (A) Intrusão de diques alimentadores de lavas. (B) Empilhamento dos depósitos vulcânicos e (C) posterior subsidência. (D) Com a ruptura continental e acúmulo de subsidência na ruptura, os depósitos vulcânicos ficam inclinados em direção à bacia oceânica.....	15
Figura 6 - Modelo de margem passiva por Péron-Pinvidic et al., (2013)	16
Figura 7 - Divisão da margem sul brasileira em domínios. Baseado no modelo de Péron-Pinvidic et al., (2013).....	17
Figura 8 - Ação gravitacional de várias partículas (bolinhas azuis) sobre um ponto de observação O .	21
Figura 9 - Ação gravitacional de um corpo gravitacional de três dimensões (V) ocupando um conjunto de coordenadas (x',y',z') sobre um ponto de observação O em (x,y,z) . Calcula-se a contribuição de um elemento do corpo, dV , depois integra-se a contribuição para todo o corpo.	22
Figura 10 - Ângulo α da velocidade de aquisição em relação ao norte geográfico.	24
Figura 11 - As diferentes categorias de interpretação dos dados observados e seus respectivos algoritmos simplificados mostrando o que é feito em cada método. Adaptado de Blakely (1995) em tradução livre.....	27
Figura 12 - Esquema de um corpo tridimensional contínuo localizado por coordenadas (x',y',z') no espaço cartesiano. Temos que $P(x,y,z)$ é o ponto de observação.	29
Figura 13 - Região coberta pelo grid da GETECH encontra-se limitada ao quadrado rosa (AOI). Toda a área offshore não coberta pelos levantamentos nas trajetórias dos navios e em estações gravimétricas no fundo marinho foi completada por dados de altimetria por satélite.	31
Figura 14 - Esquema simplificado de modelo 2-D visto como prismas que se estendem ao infinito na direção do strike (direção Y)......	32
Figura 15 - Perfil sísmico da Petrobrás interpretado na região proximal da bacia, ainda em domínio continental.....	33
Figura 16 - Perfil sísmico do LEPLAC interpretado em domínio oceânico, mais distal da bacia.	33
Figura 17 - Visualização em mapa da seção de estudo. A linha tracejada em pontos verdes corresponde exatamente a localização do perfil gravimétrico e sísmico usados no trabalho.....	35
Figura 18 - Anomalia bouguer em perfil da região de estudo na Bacia de Pelotas. O eixo horizontal representa a distância em metros no perfil e o eixo vertical é o valor da anomalia bouguer em miliGals.	36
Figura 19 - Principais horizontes sísmicos plotados na interface do <i>matplotlib</i> . Em vermelho, topo do manto; em azul, topo do embasamento terrestre; em verde, topo do rift, na região proximal, e embasamento marinho, na região distal.	37
Figura 20 - Console de modelagem gravimétrica 2D interativo. O perfil de cima ilustra a anomalia observada em preto.	38
Figura 21 - Horizontes sísmicos plotados como curvas de diversas cores.....	39

Figura 22 - Modelo sobre as interfaces plotadas. A curva em vermelho é a anomalia calculada. O polígono azul equivale ao corpo de água; o azul claro, aos sedimentos; os polígonos em vermelho claro representam a fase rifte/pré-rifte(delimitada pelas interfaces azul e verde), crosta oceânica(delimitada pelas interfaces vermelha e verde), e crosta intermediária(delimitada pela interface azul e logo abaixo pelo manto, representando o polígono vermelho).	41
Figura 23 - Modelo final sem o desenho dos horizontes, apenas com os polígonos desenhados.	41
Figura 24 - Modelo final sem o desenho dos horizontes, apenas com os polígonos desenhados. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização. Os valores associados aos polígonos representam as densidades, ou seja, $\Delta\rho$, atribuídas a cada corpo, e estão todas em Kgm3 , apesar de, nesta imagem estar Kg/m3 por limitação do editor de textos da ferramenta de edição de imagens.	43
Figura 25 - Modelo final sem o desenho dos horizontes, apenas com os polígonos desenhados. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização. Os valores associados aos polígonos representam as densidades, ou seja, $\Delta\rho$, atribuídas a cada corpo, e estão todas em Kgm3 , apesar de, nesta imagem, estarem representadas como 'Kg/m3' devido a limitação do editor de textos da ferramenta de edição de imagens.....	43
Figura 26 – Borda esquerda do modelo gravimétrico.	44
Figura 27 - Borda esquerda do modelo gravimétrico. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização.	45
Figura 28 – Borda direita do modelo gravimétrico.	46
Figura 29 - Borda direita do modelo gravimétrico. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização.	46
Figura 30 – Erro representando a discrepância entre os dados de gravimetria observados e os calculados.....	47

SUMÁRIO

RESUMO	3
ABSTRACT	5
ÍNDICE DE FIGURAS	6
1 INTRODUÇÃO	6
2 ÁREA DE ESTUDO	7
2.1 Estratigrafia da Bacia de Pelotas	8
2.2 Arcabouço Estrutural.....	12
3 PRINCÍPIOS DO MÉTODO GRAVIMÉTRICO	19
3.2 Tratamento de dados gravimétricos	23
3.2.1 Correções preliminares	23
3.2.1 Anomalias gravimétricas	25
4 MODELAGEM GEOFÍSICA	27
4.2 Modelagem inversa.....	28
4.1 Modelagem direta.....	28
5 MÉTODO.....	31
5.1 Preparando a base de dados.....	31
5.2 A modelagem	37
CONCLUSÕES.....	48
REFERÊNCIAS.....	49
ANEXOS	51

1 INTRODUÇÃO

As investigações geofísicas desempenham um papel importante na compreensão da Terra. A sísmica, historicamente, sempre recebeu o maior montante de investimentos entre os métodos geofísicos, devido a sua importância na indústria petrolífera, mas a gravimetria também desempenha um importante papel no estudo da geologia de subsuperfície.

Em alguns casos, a gravimetria pode ser a única opção onde o sinal sísmico não alcança efetivamente a estrutura geológica a ser analisada. Um exemplo, tratado no presente trabalho, é a Bacia de Pelotas, onde a alta refletividade do topo das rochas vulcânicas depositados no rifte torna quase impeditivo o estudo das formações em maiores profundidades. Nestes casos, a gravimetria mostra-se como o método geofísico relevante no estudo da geologia crustal, contribuindo, desta forma, para o melhor entendimento das bacias sedimentares, até elucidando melhor o papel da geodinâmica na formação dos diversos tipos de bacias.

A gravimetria não só é importante quando é o único dado disponível, como também se mostra eficaz aplicada a outras informações independentes. O problema da ambiguidade inerente ao método geofísico é normalmente minimizado com a incorporação de informações independentes oriundas de outros métodos, que vincularam o problema geofísico (Castro, 2005).

O presente trabalho visou a modelagem de um modelo geológico transversal à linha de costa da Bacia de Pelotas que seja coerente com dados gravimétricos da região. O método de interpretação escolhido foi a modelagem direta, utilizando como dado de entrada o conhecimento prévio da geologia regional e dados provenientes de perfis sísmicos de reflexão.

2 ÁREA DE ESTUDO

A área de estudo está situada na Bacia de Pelotas, que faz parte do segmento de bacias da margem continental brasileira e localiza-se na porção austral brasileira, tendo como limites o Alto de Florianópolis ao norte e o Alto de Polônio ao sul. Esta ainda pode ser dividida em duas sub-bacias: uma localizada ao norte, a partir do Terraço de Rio Grande até o Alto de Florianópolis, e Sul, do Terraço em direção ao Alto de Polônio (Silveira e Machado, 2004 apud Bueno et al., 2007).

Entre a costa e a cota batimétrica de 2000 metros, a bacia abrange 210.000 Km², sendo 44.000 Km² emersos. A Figura 1 mostra a delimitação da bacia de Pelotas, assim como de bacias adjacentes, na costa brasileira, dentro da cobertura dos dados de anomalia bouguer utilizados no presente trabalho.

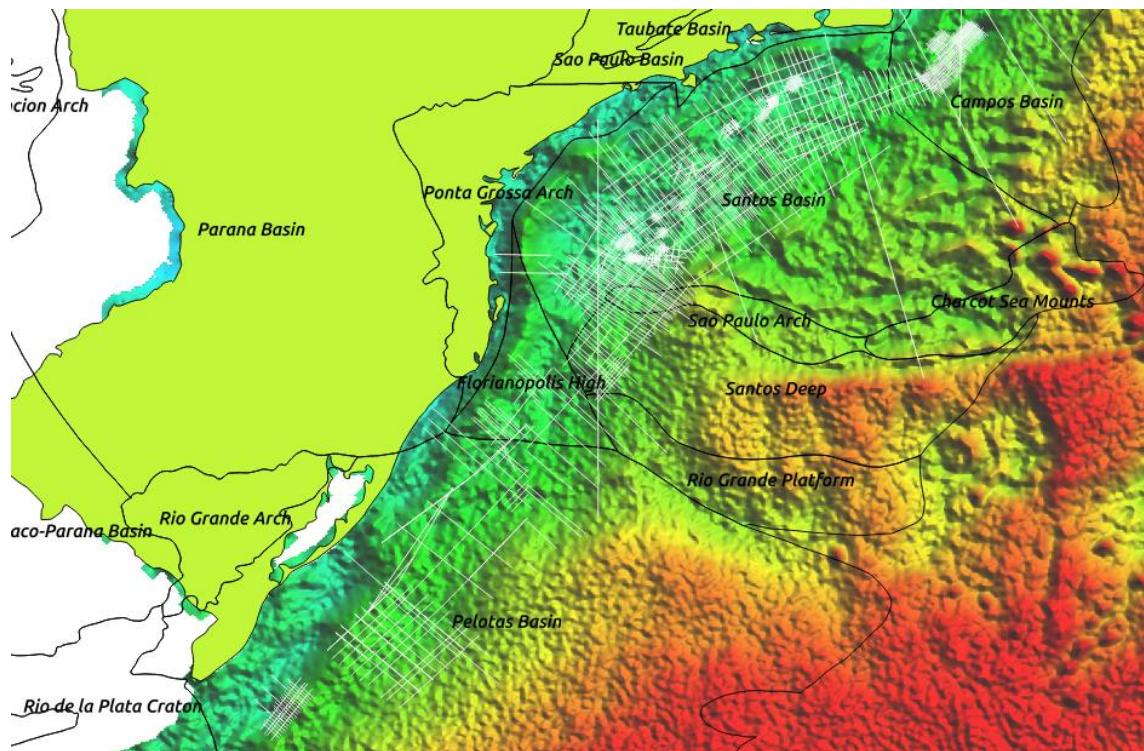


Figura 1 - Delimitação da Bacia de Pelotas sobre a cobertura de dados de anomalia bouguer da GETECH.

A Bacia de Pelotas se apresenta como uma anomalia frente às outras bacias brasileiras por ser um exemplo típico de bacia de margem passiva vulcânica, classificação recente que

leva em conta a quantidade de magmatismo gerado no processo de formação da bacia. Deste modo, a bacia apresenta estruturas relativamente particulares, levando em consideração as bacias da margem continental brasileira.

Mas as particularidades não se limitam apenas ao seu intenso magmatismo. A passagem da fase inicial de rifte para a fase drifte, ou de margem passiva, foi relativamente efêmera na Bacia de Pelotas, resultando em uma sedimentação drifte siliciclástica particularmente espessa, devido à quantidade maior de tempo de exposição a processos sedimentares francamente marinhos em comparação com outras bacias da margem brasileira. Além disso, a bacia possui uma sequência de sal muito modesta em comparação com as bacias da margem leste brasileira, e falhas normais com mergulhos voltados para o continente.

2.1 Estratigrafia da Bacia de Pelotas

É possível considerar o magmatismo do Serra Geral como o início da história da bacia de Pelotas, pois o mesmo marca o início do contexto tectônico que formaria a bacia de pelotas. Sendo assim, as rochas vulcânicas do Serra Geral correspondem à seção pré-rifte da bacia, a mesma exibe estrutura plana paralela, além de manter uma relação de discordância (*rift-onset*) com a sequência rifte.

A sequência rifte, por sua vez, deposita-se em onlap sobre a discordância *rift-onset*, exibindo mergulho contra as falhas na porção distal da bacia. Segundo Bueno et al. (2007), a sequência rifte da bacia de Pelotas constitui-se por duas formações: a Rifte I, correspondente aos basaltos da formação Imbituba, dotada de depósitos vulcânicos, sendo na sísmica reconhecida como cunhas de refletores que avançam para o mar, os Seaward Dipping Reflectors (SDRs), que ainda será alvo de discussão mais detalhada no presente trabalho; e a Rifte II, correspondente às cunhas clásticas da formação Cassino, interpretada como resultado da deposição de leques aluviais na borda flexural do rifte.

A seção pós-rifte, que recobre a formação Cassino em discordância, é formada por basaltos, andesitos e traquiandesitos pertencentes à formação Curumim, que tem ocorrência limitada à porção norte da bacia, além de possuir espessuras superiores a 6 Km. Também são identificados, intercalados na formação Curumim depósitos clásticos continentais gradando

para arenitos marinhos e carbonatos para o topo, o que indica mudança de ambiente continental lacustre para o ambiente marinho (Bueno et al., 2007).

A sequência salífera da bacia de pelotas é praticamente inexistente. A seção mais expressiva localiza-se na parte norte da bacia, da formação Ariri, alcançando no máximo 50 m de espessura, o que não é significativo perto das demais bacias do sudeste brasileiro.

Cunhas de SDRs de diferentes idades são reconhecidas ao longo do limite crosta continental/oceânica na bacia de pelotas por causa da formação diácrona da margem passiva ao longo do litoral brasileiro. As observações indicam as mais velhas na porção sul da bacia, com idades da formação Imbituba; e as mais recentes na porção norte, com idades da formação Curumim. Isso se deve ao fato de que a sub-bacia norte ainda passava pelo processo de rifteamento, a sub-bacia sul já experimentava os efeitos da subsidência termal (Martins - Neto et al., 2006, apud Bueno et al., 2007).

Após a deposição em ambiente de geometria sag da fase pós-rifte, vem a sedimentação drifte, ou seja, sedimentação em ambientes francamente marinhos. A sequência drifte começa com deposição clástica e carbonática em ambiente de plataforma rasa (formação Porto Belo) se interdigitando com os arenitos da formação Tramandaí em regiões proximais. A formação Porto Belo recobre em conformidade os evaporitos da formação Ariri, e, em discordância, as rochas vulcânicas da formação Curumim.

As condições de mar raso começam a dar lugar a condições para sedimentação de tratos transgressivos, favorável para a sedimentação dos pelitos da formação Atlântida, que recobrem em discordância os carbonatos da formação Porto Belo. Os folhelhos, gradativamente, na direção do continente, dão lugar a margas e siltitos, que se interdigitam com os arenitos formação Tramandaí, cuja interpretação é a de arenitos finos de ambiente marinho raso (Bueno et al., 2007). A sequência transgressiva é completada pelos clásticos finos e grossos da formação Cidreira, que se interdigitam com os folhelhos cinza-esverdeados da formação Imbé. Nesta última, intercalados aos folhelhos, encontram-se turbiditos, indicativo de ambiente marinho profundo com margem totalmente desenvolvida (plataforma externa, talude e bacia). Ambas as formações depositam-se em discordância sobre as formações Tramandaí e Atlântida (Bueno et al., 2007). Ainda no topo, sedimentos depositados em condições de mares profundos ganham predominância, fechando a sequência.

A seção regressiva, por sua vez, é representada pela progradação dos clásticos finos da formação Cidreira sobre os folhelhos da formação Imbé em cunhas regressivas do Neógeno, similar ao que ocorre na maior parte das bacias da margem leste brasileira. O grande aumento do aporte sedimentar propiciou a formação do Cone do Rio Grande. O rápido acúmulo de carga sedimentar levou ao colapso gravitacional de massas pouco consolidadas em direção à bacia, resultando em falhamentos normais junto ao talude, além de dobramentos e cavalgamentos na bacia.

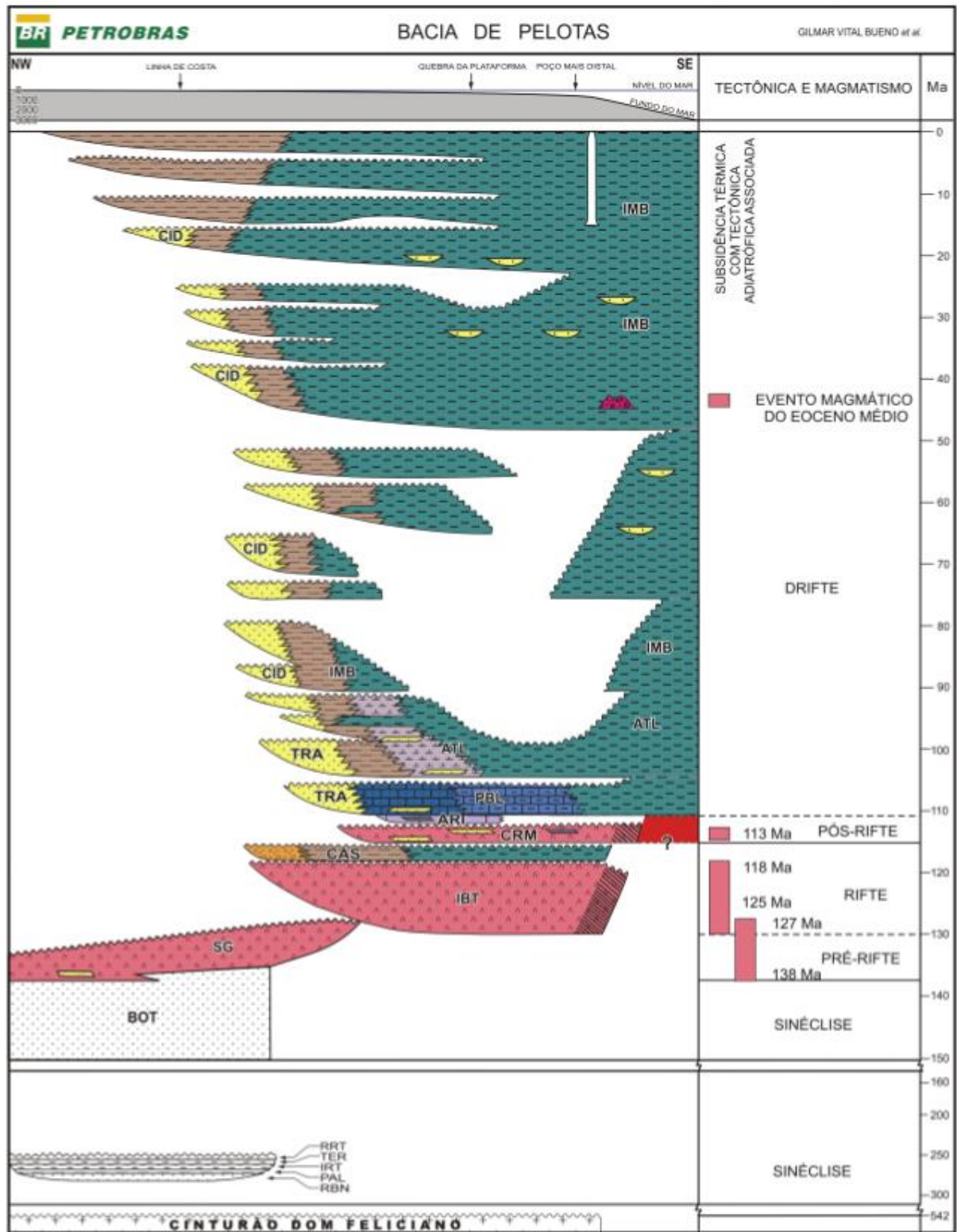


Figura 2 - Carta estratigráfica da Bacia de Pelotas por Bueno et al., 2007

2.2 Arcabouço Estrutural

A história da Bacia de Pelotas remonta ao início do rifteamento ocorrido no Gondwana no Cretáceo, mesmo contexto de formação das principais bacias sedimentares brasileiras. Os efeitos dos processos do rifteamento e da posterior formação da margem passiva são herdados pela bacia, assim como estruturas mais antigas provenientes do Ciclo Orogênico Brasileiro.

Em sua porção norte, a bacia assenta-se sobre os basaltos da formação Serra Geral, depositado sobre a bacia do Paraná em evento magmático do cretáceo inferior precedente à ruptura continental; ao sul, sobre a Faixa móvel Dom Feliciano, pertencente à província da Mantiqueira, que herda em sua estrutura e rochas com alto grau de metamorfismo eventos que datam desde o proterozóico (Ciclo Brasileiro); na parte central, o embasamento constitui-se do batólito de Pelotas, intrusão granítica também datada do proterozóico que corresponderia a uma raiz de arco magmático.

Segundo Geoffroy (2005), a presença de falhas normais com mergulhos voltados para o continente, acomodando as deformações da crosta na formação da margem, é uma das principais características de margens passivas vulcânicas. A seção abaixo ilustra o padrão das falhas da Bacia de Pelotas.

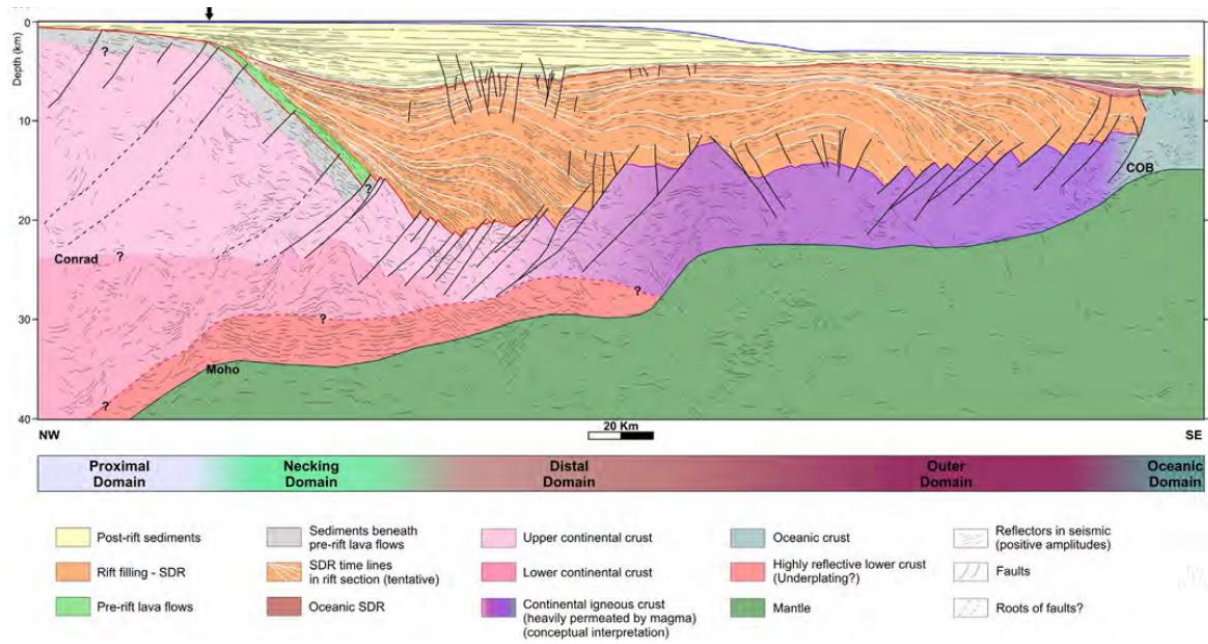


Figura 3 - Seção sísmica da sub-bacia do norte da Bacia de Pelotas interpretada por Stica et al., 2014

Na parte sul da bacia (Figura 4), as falhas normais controlam as deformações da fase rifte desde o início da região de afinamento, onde ocorrem os deslocamentos da seção pré-rifte, com rejeitos de 2-3 Km, formando semi-grábens com depocentros associados às falhas de até 5 Km de profundidade.

Na parte norte (Figura 3), por sua vez, as principais falhas ocorrem em uma região mais distal em relação à região de afinamento do embasamento e da seção pré-rifte. Deste modo, a seção rifte apresenta ausência de falhas, refletores subparalelos com mergulho para a bacia, dando a falsa impressão de uma seção sag pós-rifte. De todo modo, a interpretação desta seção mostra crescimento das camadas em direção às falhas, rotação de blocos e formação de anticlinais de rollover nas capas, típico de ambientes de rifteamento (Stica et al., 2014).

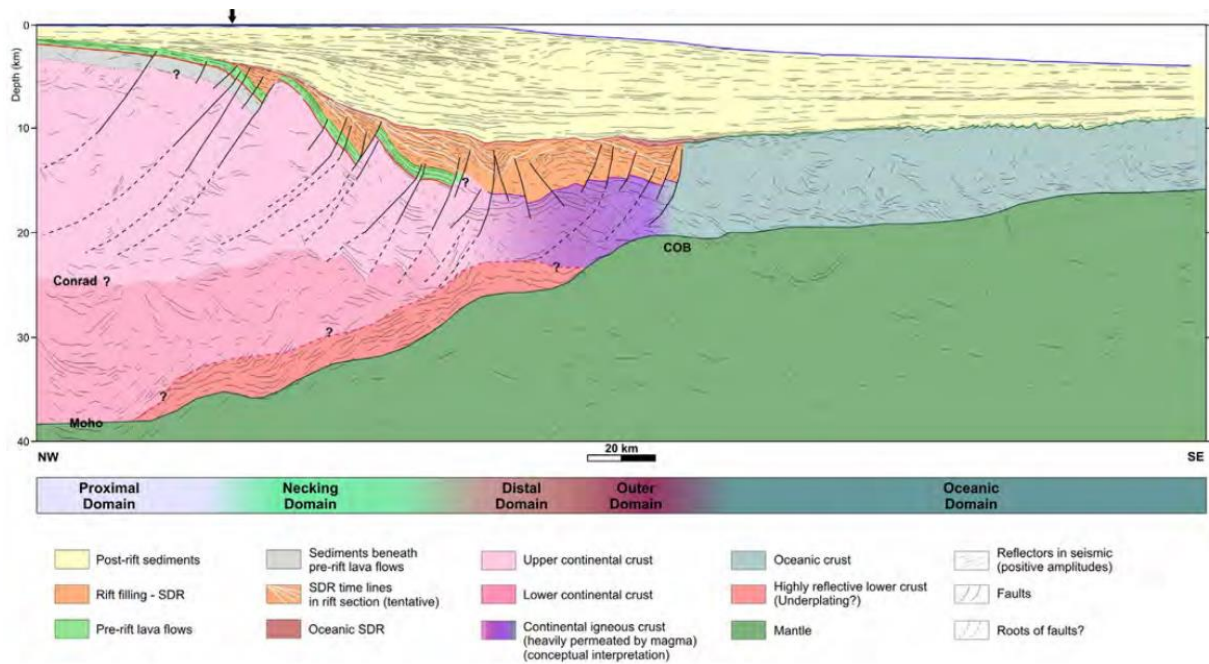


Figura 4 - Seção sísmica interpretada da sub-bacia sul da Bacia de Pelotas

As falhas também controlam a disposição das *Seaward Dipping Reflectors* (SDRs), que é mais uma particularidade da Bacia de Pelotas e característica de margens passivas com alto grau de magmatismo. As SDRs são interpretadas na sísmica como cunhas convexas mergulhantes para o mar, daí a denominação desta feição.

Uma explicação clássica para a formação das SDRs é a de Hinz et al. (1981, apud Mutter, 1985). Em sua tese, a formação das SDRs se dá por magmatismo subaéreo durante a extensão e o afinamento da crosta no processo de rifteamento. No primeiro momento, a deposição do magma ocorre em cunhas voltadas para o continente; com o acúmulo da deposição, subsidência, e posterior ruptura continental, o depósito vulcânico fica mergulhado contra a zona de ruptura, resultando no aspecto mergulhante para o mar (Figura 5). De acordo com a tese, o limite entre a crosta continental e oceânica estaria no limite distal (em direção ao mar) das SDRs, pois os refletores se formariam no estágio final de rifteamento, precedendo, deste modo, a formação de uma zona de espalhamento oceânico.

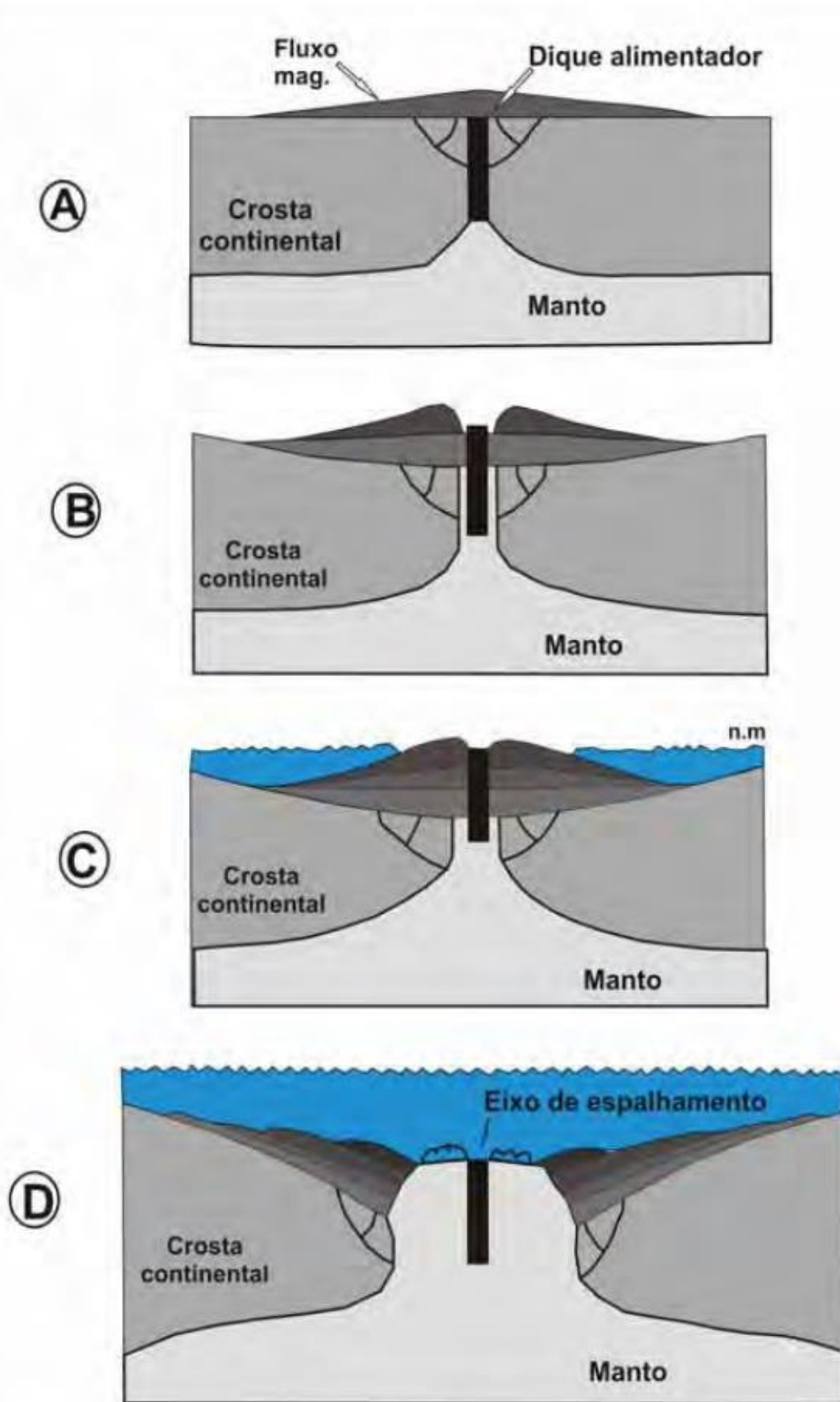


Figura 5 - Esquema da formação dos SDRs (Hinz et al., 1981, apud Mutter, 1985). (A) Intrusão de diques alimentadores de lavas. (B) Empilhamento dos depósitos vulcânicos e (C) posterior subsidência. (D) Com a ruptura continental e acúmulo de subsidência na ruptura, os depósitos vulcânicos ficam inclinados em direção à bacia oceânica.

A distribuição regional de SDRs cresce em largura do sul para o norte, a despeito do que é apresentado do norte da Argentina para o sul do Uruguai, onde cunhas de SDRs decrescem do sul para o norte, com cada segmento limitado por falhas de transferência (Franke et al., 2007 apud Bueno et al., 2014).

Para fins de uma interpretação regional da crosta em ambientes de margens passivas, Péron-Pinvidic et al., (2013) divide a margem em faixas subparalelas às linhas de costa: domínio proximal, domínio de afinamento, domínio distal, domínio externo e domínio oceânico. O seu modelo é uma tentativa de generalizar características específicas do tectonismo, vulcanismo e da deformação resultantes do processo de formação das margens associando-os às porções citadas e ilustradas na Figura 6.

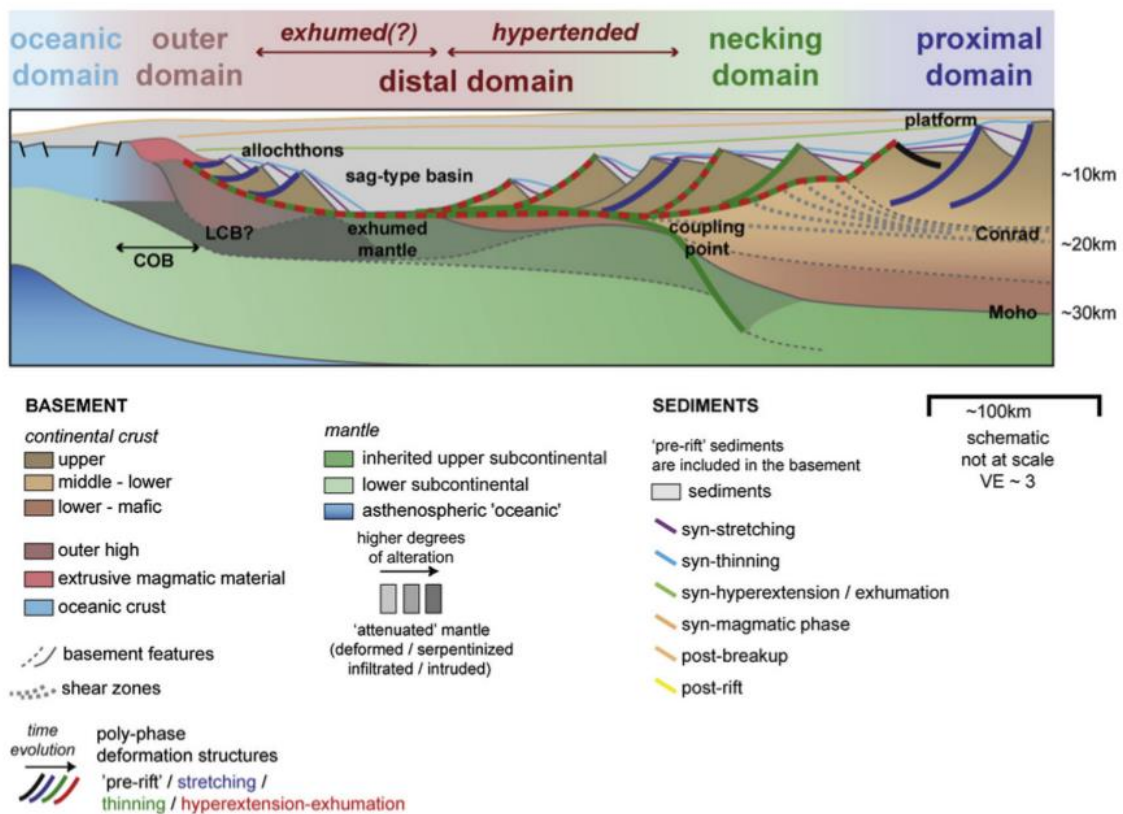


Figura 6 - Modelo de margem passiva por Péron-Pinvidic et al., (2013)

Abaixo temos a interpretação do segmento da margem brasileira associado à Bacia de Pelotas baseada no modelo de Péron-Pinvidic et al., (2013).

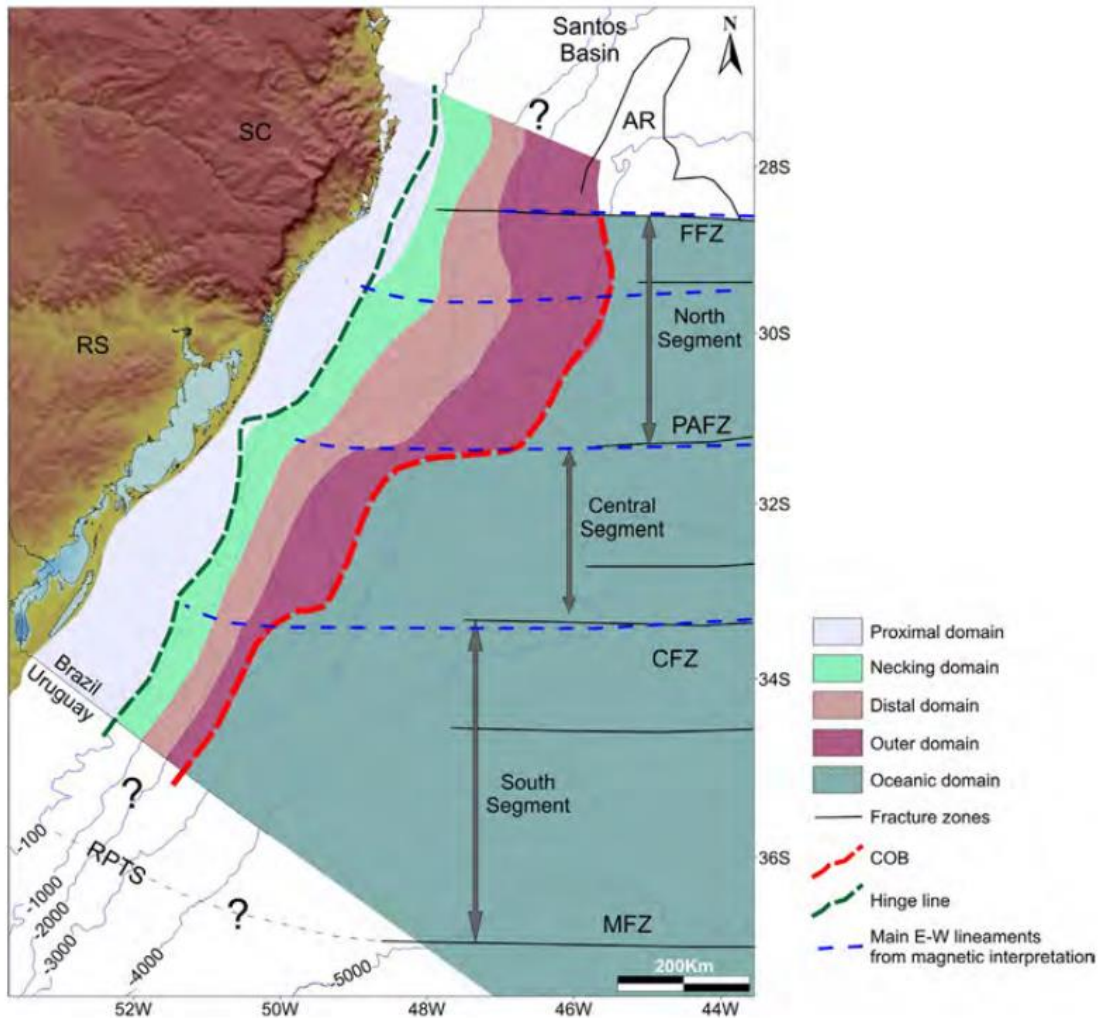


Figura 7 - Divisão da margem sul brasileira em domínios. Baseado no modelo de Péron-Pinvidic et al., (2013)

Interpretações de seções sísmicas de alta profundidade feitas por Stica et. al (2014) mostraram a região proximal com poucas deformações, com uma crosta variando de 30 a 40 km contados a partir do topo do embasamento. As falhas não chegam a alcançar a descontinuidade de Mohorovicic, muitas delas enraizadas no topo da crosta inferior. Um refletor de alta intensidade, localizado no interior da crosta, foi interpretado como um limite reológico entre duas porções crustais com comportamentos mecânicos distintos, ou seja, o limite entre a crosta rúptil, siálica, e a crosta dúctil, variando de milonítica a ultramilonítica, por Zalán et. al (2009, 2011); e, outra hipótese, mais coerente com o contexto de margens passivas vulcânicas, seria a presença do *underplating*, uma soleira depositada por intrusão magmática durante o processo de estiramento crustal (Geoffroy, 2005).

Na linha flexural, o manto começa a diminuir drasticamente de profundidade, em direção ao domínio de afinamento, e a crosta começa a se encontrar mais deformada, devido ao estiramento, com o topo do embasamento atingindo profundidades de 15, 20 e 25 Km abaixo de sua posição no domínio proximal. Nesta porção, se torna claramente notável o padrão de mergulho para o continente das falhas no rifte (Stica et. al, 2014).

No domínio de afinamento, a descontinuidade de Mohorovicic começa a se encontrar em profundidades bem menores, e a crosta se encontra mais deformada, com o afinamento crustal aumentando gradativamente em direção ao oceano até espessuras em torno de 10 Km. As falhas, no domínio flexural, atingem a crosta inferior e mantêm o padrão de mergulho no sentido do continente (Stica et al., 2014).

Nos domínios distal e externo, a crosta apresenta uma variação relativamente menor em sua espessura, mantendo, deste modo, uma espessura aproximadamente constante até a borda que separa a crosta continental e oceânica. Esta região se estende por 50 Km de extensão na parte sul da bacia (Figura 4) e por 200 Km de extensão na parte norte (Figura 3); sendo, nesta última, atípica no contexto de margens passivas vulcânicas, caracterizadas por margens relativamente estreitas (Stica et. al, 2014). Essa diferença na extensão das margens ocorre devido as heranças da tectônica pré-cambriana, ou seja, a tendência NE-SW mostrada nas estruturas da margem leste brasileira.

Finalmente, o domínio oceânico representa a porção crustal essencialmente oceânica com sua fisionomia tabular e relativamente fina, com espessuras variando entre 7 e 10 Km. Zalán et. al (2011) propõem a divisão da crosta oceânica neste domínio em três camadas: uma inferior de composição gabrosa, exibindo forte reflexão sísmica em sua base nas seções sísmicas, onde estaria a descontinuidade de mohorovicic marinha; uma camada intermediária interpretada como enxame de diques, muito devido as fortes reflexões de alto ângulo; e uma camada superior de *pillow lavas*, que não exibem reflexões sísmicas significativas.

3 PRINCÍPIOS DO MÉTODO GRAVIMÉTRICO

A lei de força mais antiga conhecida é a lei da gravitação universal, que exprime as forças de interação gravitacional entre dois corpos. Ela diz que os diferentes corpos materiais se atraem com uma força de magnitude tal que é diretamente proporcional às suas massas e inversamente proporcional ao quadrado de suas distâncias (Nussenzveig, 2002). Sua lei trouxe contribuições a diversas áreas do conhecimento como a física, astrofísica, astronomia, entre outras áreas, trazendo explicação para fenômenos da natureza como a queda de uma maçã até a mecânica que rege todo o cosmos.

A gravimetria, um importante método geofísico, nada mais é do que o estudo do campo de gravitação com a finalidade de fazer inferências a respeito de geologia de subsuperfície. Nos levantamentos de gravimetria, as diferentes densidades, formatos e volumes das rochas no interior da terra causarão variações no campo gravitacional da Terra. Parte dessas variações, ou anomalias, provém de uma ampla variedade de situações geológicas conhecidas, e sua interpretação pode permitir o conhecimento de características do corpo causador da anomalia, como a profundidade, a forma do corpo, etc (Keary, 2009).

Analisando dois corpos considerados pequenos o suficiente em relação a distância que os separa, apresenta-se a seguir a forma analítica da interação gravitacional entre esses corpos:

$$F = \frac{GmM}{r^2}, \quad (1)$$

No qual, F representa a magnitude em módulo da força gravitacional que um corpo aplica no outro, G é a constante de gravitação universal, m e M são as massas dos dois corpos, e r é a distância que os separa.

Pode-se olhar a equação como o módulo da força de interação devido à gravitação da Terra e de uma partícula próximo a sua superfície se o planeta for tratado idealmente como uma partícula com toda a sua massa localizada em seu centro, considerando o centro do planeta como o seu centro de massa.

Analisando apenas o campo gerado pela Terra, de massa M , e aplicando a segunda Lei de Newton, demonstra-se que a Terra gera um campo gravitacional independente da partícula que sofre a influência desse campo:

$$F = m \frac{dv}{dt} = \frac{GMm}{r^2}, \quad (2)$$

No qual, $\frac{dv}{dt}$ é a taxa de variação da velocidade com o tempo, ou seja, a aceleração; G é a constante de gravitação universal; M é a massa da Terra; m é uma massa de prova; e r é a distância entre os centros de massa dos corpos.

$$\frac{dv}{dt} = \frac{GM}{r^2} = g. \quad (3)$$

Logo, o campo de gravitação da Terra é representado pela aceleração gravitacional g . Sendo assim, g tem dimensões de aceleração, representado nas unidades do Sistema Internacional de Medidas (SI) em m/s^2 .

Como as variações das densidades das rochas de subsuperfície causam variações relativamente pequenas, geralmente de $100 \mu m/s^2$, trabalha-se, para fins de praticidade, com a unidade *Gal* na gravimetria, que vale $10^{-2} m/s^2$, sendo mais utilizado a sua subdivisão o *miligal*, que vale $10^{-5} m/s^2$ (Keary, 2009).

Apesar de discutido até aqui apenas a magnitude dos módulos das grandezas envolvidas na teoria da gravitação, o campo gravitacional tem natureza vetorial. Como g é um campo vetorial, é comum a procura de um campo potencial associado a ele por causa da conveniência e facilidade de se trabalhar com campos escalares. No caso do campo gravitacional, é facilmente verificável que

$$U = \frac{-GM}{r}, \quad (4)$$

No qual, U é o campo potencial procurado, pois

$$\frac{dU}{dr} = \frac{GM}{r^2}. \quad (5)$$

O campo gravitacional, entretanto, deve ser tratado de forma adequada à sua natureza vetorial. Dessa forma, o uso de coordenadas esféricas tem sua conveniência neste caso,

devido à ação central do campo gravitacional. O parâmetro r mantém-se por ser a distância radial do centro do planeta ao ponto de estudo. Portanto, a fórmula do potencial continua a mesma, mas o operador diferencial usado é o gradiente, como mostrado abaixo.

$$\vec{\nabla} U = \vec{g} = \frac{GM}{r^2} \hat{r}. \quad (6)$$

Como é claramente notável, \vec{g} depende de r , ou seja, o campo gravitacional depende da distância radial em relação ao centro de massa da Terra.

O campo gravitacional categoriza-se, portanto, como um campo conservativo, pois $\text{rot } \vec{g} = \vec{0}$ e existe uma função potencial associada definida acima.

Em muitos casos, e a gravimetria se encontra neles, o interesse recai na análise do potencial ou do campo gravitacional gerado por uma distribuição contínua de massas, e isso é exatamente o que acontece na realidade. Primeiramente, considera-se o princípio da superposição dos efeitos individuais de diferentes massas, como mostrado abaixo.

$$\vec{g}(O) = \sum_i \vec{g}_i(O) = -G \sum_i m_i \frac{\vec{r}_i}{r_i^3}, \quad (7)$$

$$U(O) = \sum_i U_i(O) = -G \sum_i \frac{m_i}{r_i}. \quad (8)$$

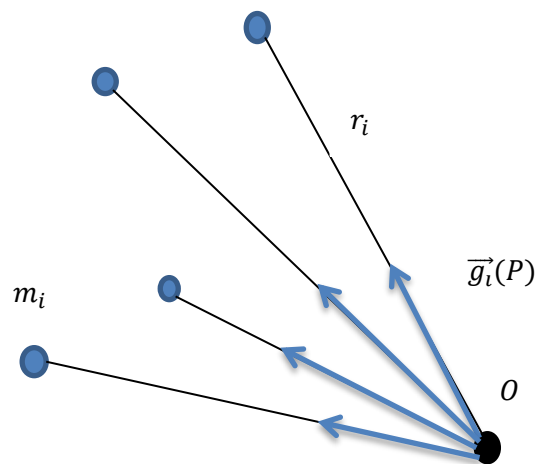


Figura 8 - Ação gravitacional de várias partículas (bolinhas azuis) sobre um ponto de observação O .

No qual, $\vec{g}(O)$ é a resultante da aceleração gravitacional causada pelas partículas de massa m_i e distância r_i , assim como $U(O)$ é a resultante do potencial gravitacional gerado por cada uma dessas partículas, \vec{r}_i é o vetor posição de cada uma dessas partículas, e G é a Constante de Gravitação Universal.

O cálculo de uma distribuição contínua de massas pode ser feito através de integração ou numericamente através da soma de elementos de massa que compõe o corpo.

$$d\vec{g}(O) = -G \cdot dm \cdot \frac{\vec{PO}}{PO^3} = -G \cdot p(P) \cdot \frac{\vec{PO}}{PO^3} dV, \quad (9)$$

$$dU(O) = -G \frac{p(P)dV}{PO}. \quad (10)$$

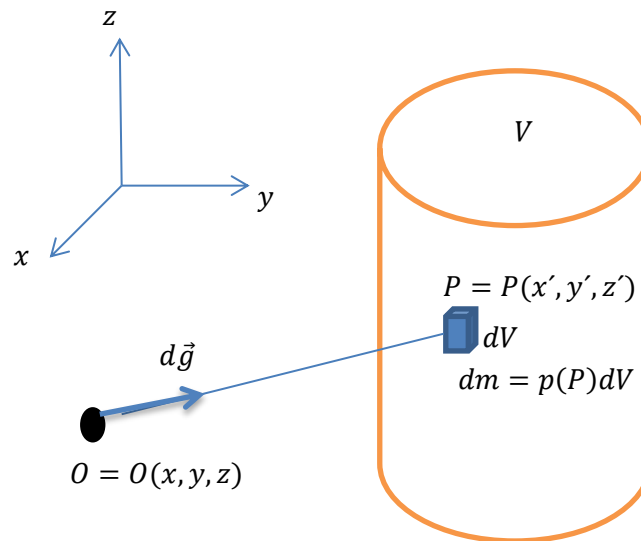


Figura 9 - Ação gravitacional de um corpo gravitacional de três dimensões (V) ocupando um conjunto de coordenadas (x',y',z') sobre um ponto de observação O em (x,y,z). Calcula-se a contribuição de um elemento do corpo, dV, depois integra-se a contribuição para todo o corpo.

Integrando as contribuições de todas as partes do corpo para o campo gravitacional e o potencial, temos:

$$\vec{g}(O) = -G \iiint_{P \in V} p(P) \frac{\vec{PO}}{PO^3} dV, \quad (11)$$

$$U(O) = -G \iiint_{P \in V} \frac{p(P)}{PO} dV. \quad (12)$$

Os gravímetros detectam a componente vertical do campo gravitacional. A derivada na direção vertical, z , fornece deste modo a componente vertical do campo de gravidade gerado pelas massas da Terra (Blakely, 1995).

Em coordenadas cartesianas, $r = \sqrt{(x' - x)^2 + (y' - y)^2 + (z' - z)^2}$, então a derivada de g na direção vertical é

$$\frac{\partial g}{\partial z} = -G \iiint_{(x',y',z') \in V} p(x',y',z') \frac{(z - z') dx' dy' dz'}{r^3}. \quad (13)$$

3.2 Tratamento de dados gravimétricos

Os gravímetros estão sempre sujeitos a efeitos indesejáveis e inconvenientes para a interpretação geológica. Fontes diversas às variações das densidades das rochas de subsuperfície não são nada mais do que ruído para efeitos de investigação da geologia de subsuperfície; logo, precisam ser atenuados nos dados gravimétricos.

Essa limpeza de ruídos na gravimetria, chamada de correção gravimétrica, retira ruídos conhecidos dos dados, como os efeitos da topografia e da rotação da Terra; elevação da aquisição dos dados em relação ao nível do mar; efeitos de maré da lua e do sol; movimento do instrumento de aquisição; efeitos do terreno do local de medição (Blakely, 1995).

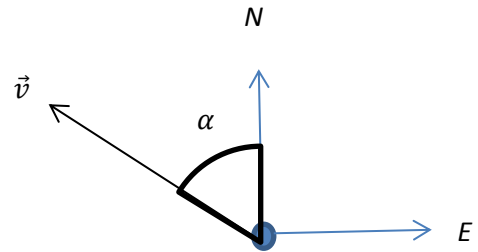
3.2.1 Correções preliminares

Correção de Eötvös

A correção de Eötvös serve para retirar os efeitos do movimento do gravímetro sobre a superfície da Terra. A variação da velocidade relativa em relação à velocidade de rotação da Terra pode gerar efeitos indesejados adicionados aos valores de gravidade, ou pode diminuir a magnitude da gravidade observada. Isso se deve ao efeito da aceleração centrífuga dos corpos na superfície terrestre.

Com certos dados dos parâmetros da aquisição, que serão descritos a seguir, a fórmula abaixo calcula a magnitude da correção à qual os dados devem ser submetidos.

$$g_E = 7,053v \cos \lambda \sin \alpha + 0,004154v^2, \quad (14)$$



Em que λ é a latitude do levantamento; v corresponde à velocidade em nós (1 nó = 0,51444 m/s); α é o ângulo da aquisição em relação ao norte; e g_E é a magnitude da correção que deverá ser aplicada em *miligals* (mGal).

Figura 10 - Ângulo α da velocidade de aquisição em relação ao norte geográfico.

Correção de maré

As marés terrestres também são efeitos observáveis nas medidas de gravidade, e são de magnitude considerável em levantamentos gravimétricos de alta precisão, principalmente em regiões de baixa latitude. O efeito de maré, entretanto, é uma função periódica muito bem definida que tem como variáveis o tempo, longitude e latitude; logo, seu efeito nos dados pode convenientemente ser retirado.

Há diversas fórmulas para o cálculo da correção de maré. A mais usada é a fórmula de Longman (1959, JGR).

Correção de deriva

Muitos gravímetros apresentam o efeito de deriva instrumental, em que o gravímetro adiciona determinado valor em um determinado intervalo de tempo às medições de gravidade, um efeito que se apresenta de forma linear em relação ao tempo, e é visível em medições contínuas no mesmo local.

A correção de deriva instrumental visa retirar esse efeito, e pode ser facilmente feita com medições na estação base, local onde se fazem repetidas medições, a primeira e a última do levantamento. Deste modo, como o efeito de deriva é uma função linear com o tempo, ele é facilmente estimado para todas as medições do levantamento.

3.2.1 Anomalias gravimétricas

Correção de Latitude

A anomalia com respeito à gravidade teórica é feita retirando-se o efeito da gravidade teórica da Terra, cujo referencial é um elipsoide de revolução com o eixo menor na direção dos polos, e com distribuição homogênea de massa. As fórmulas teóricas usadas em gravimetria também consideram o efeito da rotação da Terra, que ocasiona uma diminuição das medidas devido à aceleração centrífuga. A retirada dos efeitos da gravidade teórica devido ao elipsoide de referência também é chamada de correção de latitude, devido à dependência da latitude.

Existem diversos elipsoides de referência para a Terra, desenvolvidos e refinados para maiores precisões ao longo dos anos. O modelo da Terra adotado nos dados do presente trabalho foi o *World Geodetic System 1984* (WGS84) da Associação Internacional de Geodésia, cuja fórmula é apresentada abaixo.

$$g_{ref}(\lambda = latitude) = 9,7803267714 \frac{1 + 0,0019318515138639 \sin^2 \lambda}{\sqrt{1 - 0,00669437999013 \sin^2 \lambda}}. \quad (15)$$

Correção de ar-livre

Subtraído os efeitos da gravidade de referência, apenas um passo separa o processo de retirada dos ruídos para a interpretação geológica de subsuperfície. Antes disso, é necessário retirar os efeitos da altitude do levantamento em relação ao nível do mar, que causa uma influência muito significativa, devido à dependência da gravidade com o inverso do quadrado da distância da fonte.

A magnitude da correção de ar-livre, como é chamada a correção em relação à altitude das medições, pode ser facilmente verificada através da integração direta sobre o volume idealmente esférico ou através do cálculo do fluxo do campo gravitacional com a lei de Gauss. A fórmula obtida com o cálculo é apresentada abaixo.

$$g_{al}(h) = -0,3086h \text{ mGal} , \quad (16)$$

No qual, h é a altitude em metros.

A anomalia de ar-livre, como é chamada a resultante da correção de ar-livre, apresenta uma boa correlação com a topografia e pode ser usada para uma interpretação geológica. Entretanto, o intérprete pode refinar mais os dados para realçar o máximo possível as anomalias causadas pelos corpos de interesse e, assim, facilitar a interpretação.

Correção Bouguer

Como a correção de ar-livre retira apenas o efeito da altitude em relação ao nível de referência, o efeito da massa topográfica excedente permanece na anomalia de ar-livre. A anomalia bouguer é obtida retirando-se a contribuição da massa topográfica que se encontra acima do nível de referência da altitude (em geral, o nível do mar é a referência).

A correção de bouguer retira o efeito da massa topográfica considerando-as idealmente como uma placa infinita com espessura igual à altitude da massa. A fórmula para a anomalia bouguer, demonstrada abaixo, é facilmente obtida através do cálculo do fluxo do campo gravitacional gerado pela placa, com a aplicação da lei de gauss, na placa infinita, e depende apenas da espessura da placa (considerando-se a densidade das rochas constante).

$$g_b(h) = 2\pi G\rho h. \quad (17)$$

A correção bouguer, entretanto, ainda deixa rastros de efeitos não desejados, pois, como é facilmente notável, esta correção acaba retirando efeitos de uma massa excedente que é inexistente. Para compensar os excessos da correção bouguer, a **correção de terreno** ou a **bouguer completa** utiliza um modelo do terreno para avaliar os excessos da correção bouguer com o modelo de placa infinita, e, assim, refinar a anomalia bouguer.

4 MODELAGEM GEOFÍSICA

Após o processamento dos dados geofísicos, entra a importante e desafiadora tarefa da interpretação dos dados. O problema consiste em apontar as características físicas e geológicas do corpo causador dos padrões que aparecem nos dados.

Blakely (1995) categoriza as várias técnicas de interpretação em três formas diferentes; nas quais, o mesmo objetivo, que é estimar as características das fontes dos dados geofísicos, é alcançado usando diferentes algoritmos, os quais são ilustrados abaixo.

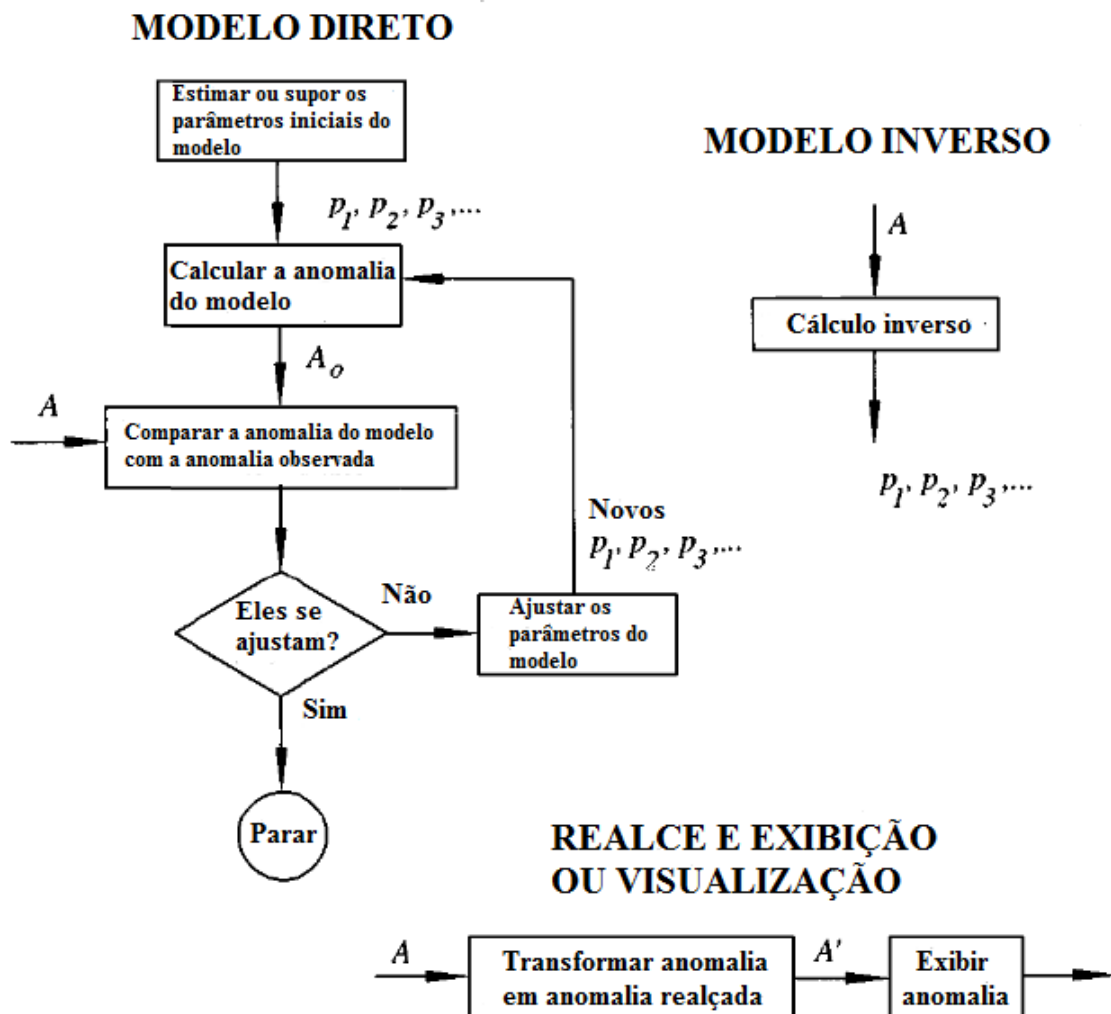


Figura 11 - As diferentes categorias de interpretação dos dados observados e seus respectivos algoritmos simplificados mostrando o que é feito em cada método. Adaptado de Blakely (1995) em tradução livre.

O presente trabalho teve foco na elaboração de modelos diretos.

4.2 Modelagem inversa

Na modelagem inversa, também chamado problema inverso, a gravidade é especificada e as densidades ou a geometria ou ambas são incógnitas, que deverão ser determinadas automaticamente por procedimentos estatísticos (Castro, 2005).

4.1 Modelagem direta

Para a modelagem direta, as densidades e formas geométricas das fontes são assumidas e o valor da gravidade calculado é comparado com o dado observado (Castro, 2005).

A concepção do modelo gravimétrico direto envolve suposições acerca das características do meio, tais como a densidade, caso da gravimetria, da fonte dos dados geofísicos. O objetivo é ajustar o modelo aos dados observados em um processo iterativo de ajustes no modelo, que pode envolver condicionantes baseados em interpretações de dados provenientes de outros métodos, como a sísmica e a geologia conhecida, por exemplo.

O grande problema da modelagem é conseguir representar a geologia através de formas geométricas com o máximo de fidelidade possível e com o mínimo custo computacional (Blakely, 1995).

Blakely (1995) apresenta, em linhas gerais, a base do cálculo feito pelas diferentes técnicas de modelagem direta gravimétrica sobre os modelos para gerar a anomalia calculada e comparar com a observada, analisando a Figura 12.

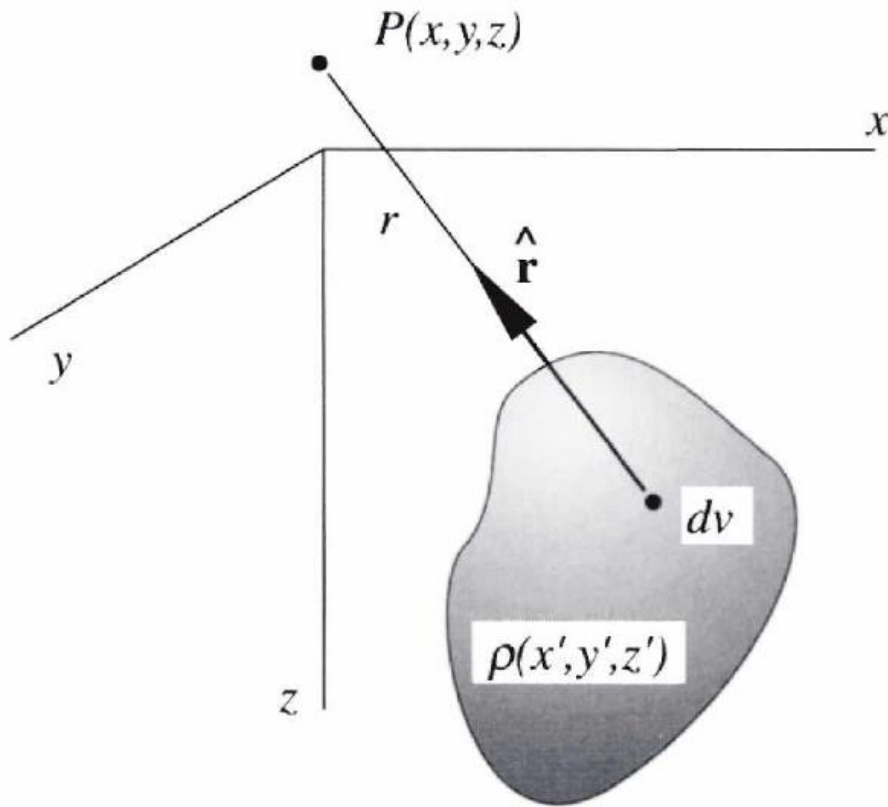


Figura 12 - Esquema de um corpo tridimensional contínuo localizado por coordenadas \$(x', y', z')\$ no espaço cartesiano. Temos que \$P(x, y, z)\$ é o ponto de observação.

A fórmula da componente vertical do campo gravitacional devido ao corpo de densidade \$\rho(x', y', z')\$, apresentado na Figura 12, apresenta-se abaixo como:

$$\frac{\partial U}{\partial z} = -G \iiint_{(x', y', z') \in V} \frac{\rho(x', y', z')(z - z') dV}{r^3}, \quad (18)$$

$$\text{No qual, } r = \sqrt{(x' - x)^2 + (y' - y)^2 + (z' - z)^2}. \quad (19)$$

De forma geral a equação apresentada acima pode ser representada da seguinte forma:

$$g(x, y, z) = \iiint_{(x', y', z') \in V} \rho(x', y', z') \psi(x - x', y - y', z - z') dV , \quad (20)$$

Onde

$$\psi(x, y, z) = -G \frac{z}{(x^2 + y^2 + z^2)^{3/2}} . \quad (21)$$

A modelagem calcula iterativamente a função ψ , que depende apenas da forma geométrica do corpo modelado, apesar do problema da representação geométrica de corpos geológicos complexos. Na prática, dividem-se os corpos modelados em N partes, e adiciona-se a influencia gravitacional de cada um deles, seguindo o principio da superposição. Deste modo, o campo gravitacional é apresentado simplifcadamente como:

$$g_m = \sum_{n=1}^N \rho_n \psi_{mn} , \quad (22)$$

No qual, g_m é a atração vertical no m-ésimo ponto de observação, ρ_n é a densidade da n-ésima parte do corpo, e ψ_{mn} é a atração gravitacional no m-ésimo ponto de observação devido a n-ésima parte do corpo com densidade unitária.

5 MÉTODO

5.1 Preparando a base de dados

Os dados utilizados como parâmetro da modelagem gravimétrica são provenientes dos dados de gravidade da GETECH, um grid com $0,02^\circ$ de resolução, que cobre parte de regiões onshore e offshore do sudeste brasileiro (Figura 13) e integra: levantamentos de estações gravimétricas em terra, medidas em estações gravimétricas em fundo de mar, levantamentos realizados em trajetórias de navios, e dados de altimetria por satélite.

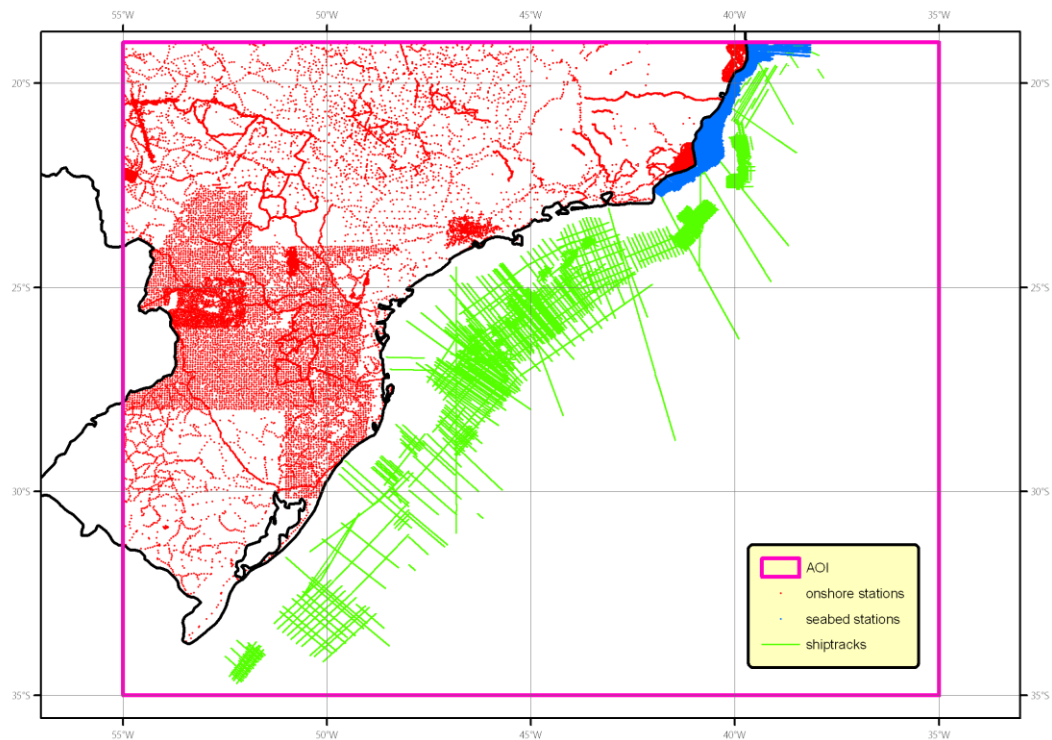


Figura 13 - Região coberta pelo grid da GETECH encontra-se limitada ao quadrado rosa (AOI). Toda a área offshore não coberta pelos levantamentos nas trajetórias dos navios e em estações gravimétricas no fundo marinho foi completada por dados de altimetria por satélite.

Todos os dados foram reprocessados usando fórmula de gravidade WGS84 e ajustados ao datum IGSN71. Correções de terreno e de bouguer foram aplicadas durante o reprocessamento, usando densidades de $2,67 \text{ g/cc}$ (onshore) e $2,2 \text{ g/cc}$ (offshore).

Os primeiros esforços do trabalho foram focados em elaborar um modelo prévio a ser ajustado no *fatiando a terra*, uma biblioteca do *python* de código aberto para modelagem e inversão de dados geofísicos. Uma das sub-bibliotecas do *fatiando a terra* permite criar modelos geológicos hipotéticos para calcular sua resposta gravimétrica. No presente trabalho, optou-se pela criação de um modelo 2-D, assumindo a Terra como sendo bidimensional, variando apenas com a profundidade (direção Z) e a direção do perfil (direção X, perpendicular ao strike). A NGA, 2004 ilustra que o modelo 2-D pode ser visto como construído por uma seção transversal de um prisma que se estende ao infinito na direção do strike (Figura 14).

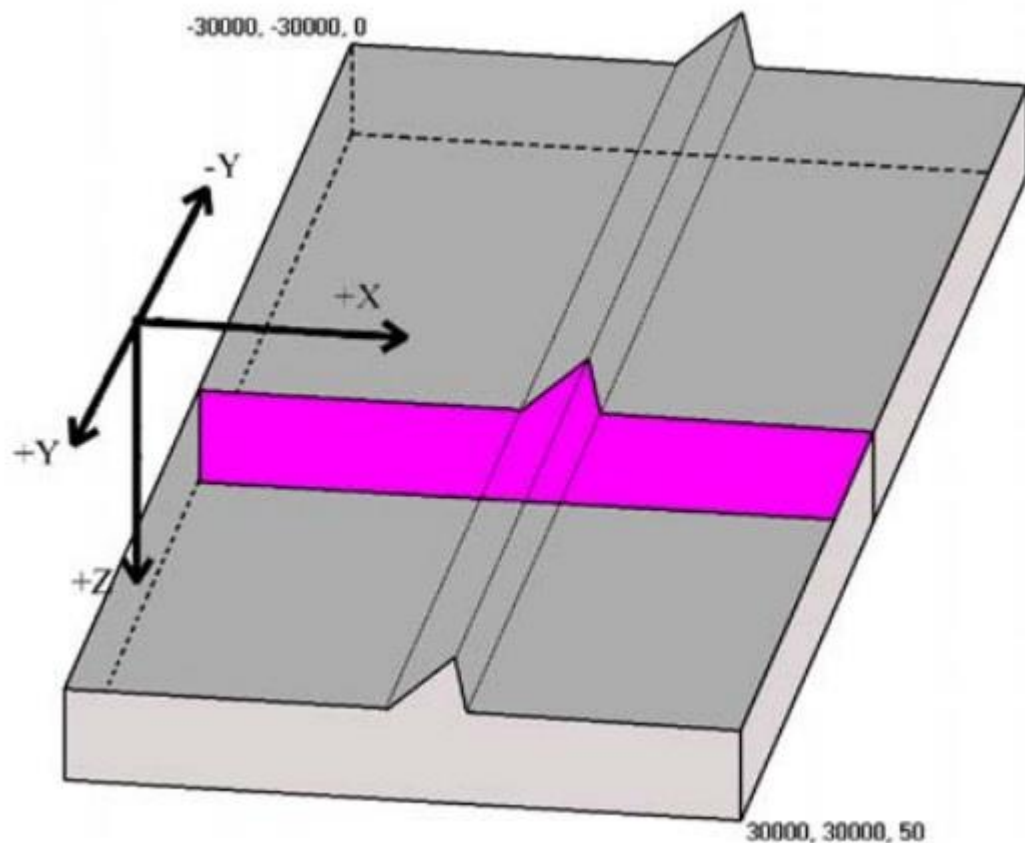


Figura 14 - Esquema simplificado de modelo 2-D visto como prismas que se estendem ao infinito na direção do strike (direção Y).

O cálculo da resposta gravimétrica do modelo no *fatiando a terra* é baseado nos algoritmos de Talwani et al. (1959).

O aplicativo que permite criar os modelos gravimétricos demanda dados sobre os limites da área, em metros, onde o modelo será desenhado usando polígonos.

Os dados em tempo da sísmica tiveram que ser convertidos para profundidade, e as coordenadas UTM foram convertidas para distâncias euclidianas em relação a uma origem.

Para a geração dos horizontes do modelo prévio, duas seções sísmicas (Figura 15 e Figura 16), interligadas em suas extremidades, como se formassem apenas um perfil, foram interpretadas no software de interpretação Petrel 2014.

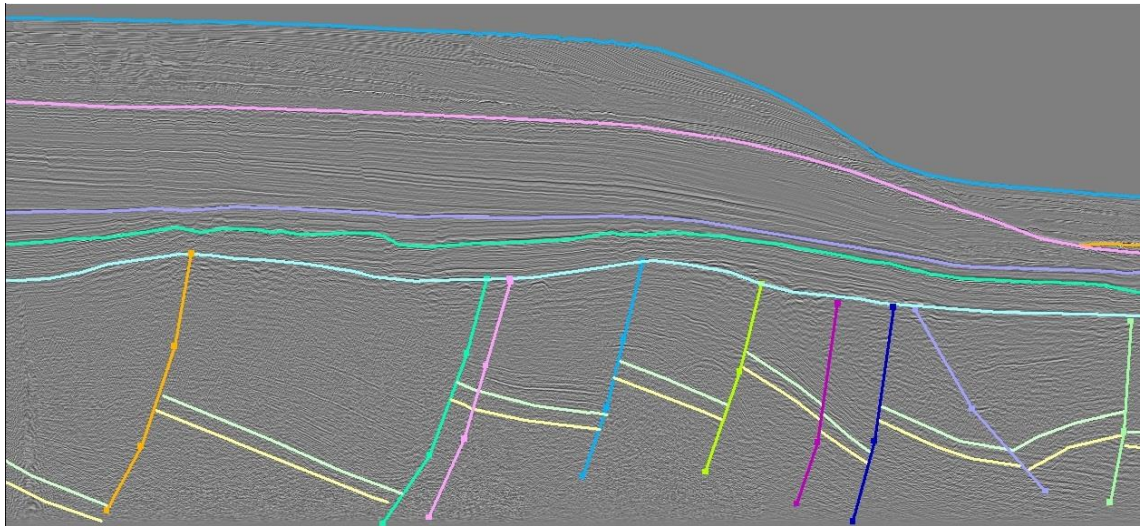


Figura 15 - Perfil sísmico da Petrobrás interpretado na região proximal da bacia, ainda em domínio continental.

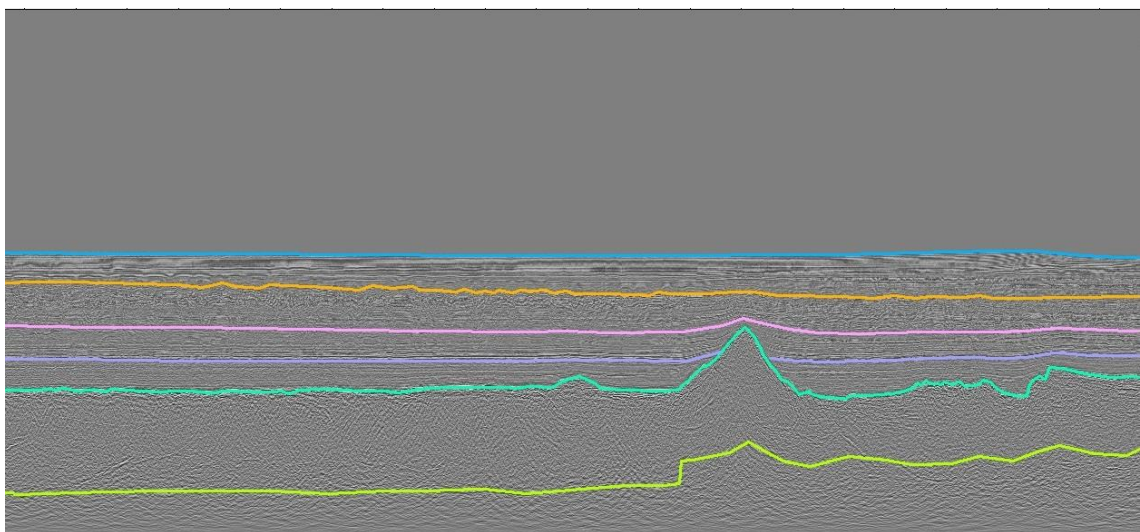


Figura 16 - Perfil sísmico do LEPLAC interpretado em domínio oceânico, mais distal da bacia.

Os horizontes interpretados foram extraídos em formato de dados ASCII e abertos no excell para edições a fim de deixar os horizontes organizados, cada um com as suas colunas de coordenadas e tempos de reflexão. Os tempos de reflexão das interfaces foram transformados em profundidade através da seguinte fórmula base:

$$z_n = \frac{t_n v_n}{2}, \quad (23)$$

Sendo $t_n = t_n - t_{n-1}$ o tempo de reflexão na camada acima da interface n e com velocidade de propagação sísmica v_n . z_n é a distância entre o fundo do mar e a interface n .

A rotina usada como base para o cálculo das espessuras das camadas encontra-se no apêndice. Os cálculos foram realizados no software MATLAB após o carregamento dos dados do excell em formato xls (data). Primeiro calculou-se a espessura das camadas, depois as profundidades de cada uma das interfaces.

A seguir, encontram-se descritas as escolhas das velocidades para a conversão das interfaces interpretadas em tempo para profundidade.

- **Velocidade no manto:** a velocidade escolhida foi a velocidade média das ondas sísmicas logo abaixo da descontinuidade de mohorovicic (moho), 8 Km/s (Decifrando a Terra)
- **Velocidade na crosta** (entre a moho e o topo do embasamento): a velocidade escolhida foi a v_p média entre a crosta superior e a crosta inferior, ou seja, 6,5 Km/s.
- **Velocidade no pré-rifte:** a velocidade adotada foi a velocidade média no basalto (Serra Geral) encontrada em bibliografias, 5,5 Km/s (Dobrin).
- **Velocidade no rifte:** Também foi escolhida a velocidade média do basalto (SDRs), de 5,5 Km/s.
- **Velocidade nos sedimentos** (acima do rifte): Para simplificar, foi escolhida a média entre a velocidade média dos folhelhos e a velocidade média dos arenitos, de 3,8 Km/s (Dobrin).
- **Velocidade na água:** A velocidade encontrada na maioria das bibliografias foi de 1,5 Km/s.

O perfil gravimétrico pôde ser extraído do grid de gravimetria da GETECH com o uso do software *Surfer 2010*. A ferramenta “*slice*” foi usada para retirar a linha gravimétrica exatamente sobre as seções sísmicas, para tal foi usado o grid da gravimetria e as coordenadas da sísmica como dados de entrada. Logo abaixo, encontra-se um mapa com a localização da seção de estudo (Figura 17), gerado no *ArcMap*, aplicativo do *ArcGIS* voltado para o desenho e a investigação de mapas, e o gráfico do perfil da anomalia bouguer (Figura 18) do local de estudo.

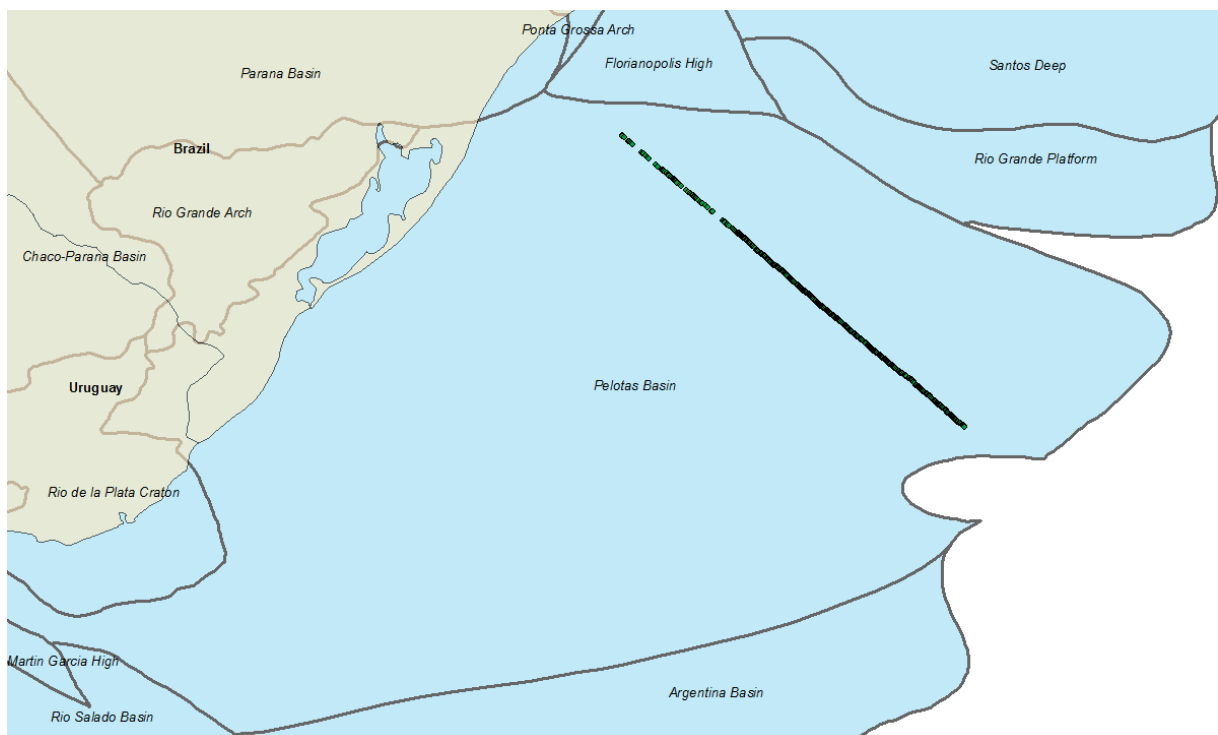


Figura 17 - Visualização em mapa da seção de estudo. A linha tracejada em pontos verdes corresponde exatamente a localização do perfil gravimétrico e sísmico usados no trabalho.

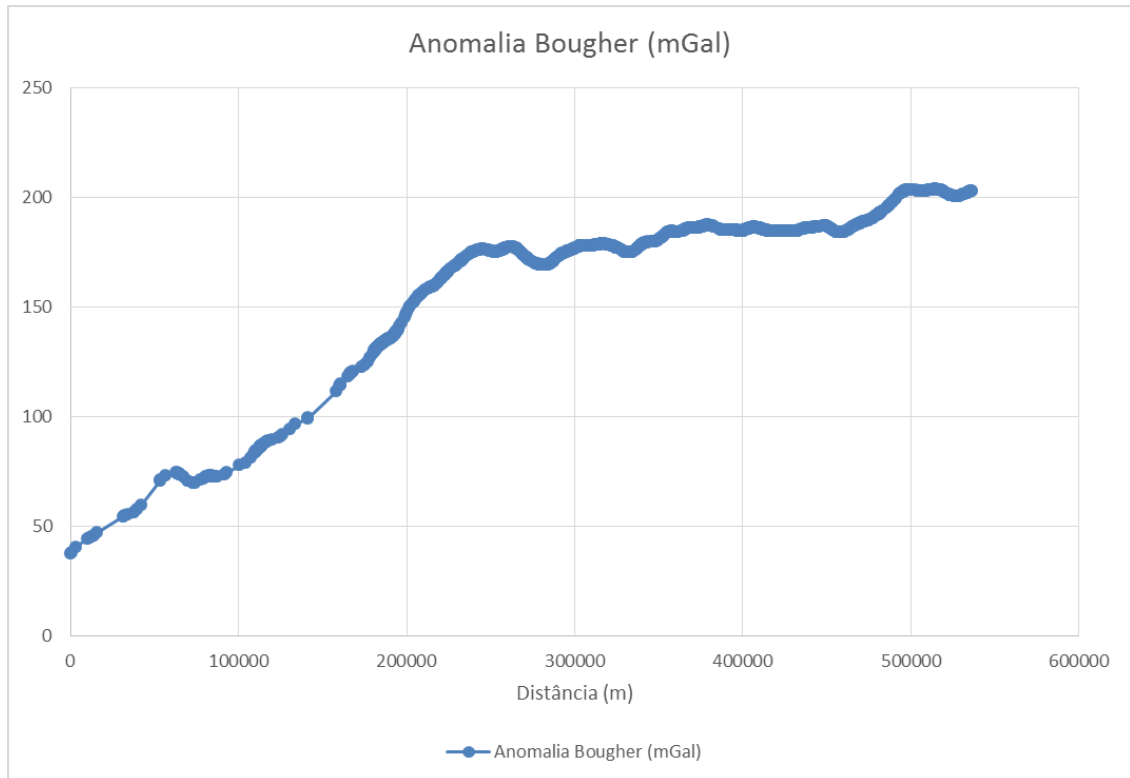


Figura 18 - Anomalia bougher em perfil da região de estudo na Bacia de Pelotas. O eixo horizontal representa a distância em metros no perfil e o eixo vertical é o valor da anomalia bougher em miliGals.

Como é possível visualizar no perfil, a partir da região entre 100 e 200 Km, o perfil gravimétrico fica mais estável, indicando possível mudança para crosta oceânica entre essas coordenadas.

Os pontos no eixo da distância foram gerados a partir das coordenadas UTM do perfil gravimétrico exportado do *surfer*. Para tal, calculou-se a distância (d) entre as coordenadas ao ponto localizado na extremidade leste do perfil. Deste modo, as coordenadas UTM foram transformadas em distâncias em relação à origem do perfil.

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2}. \quad (24)$$

O mesmo processo de conversão de coordenadas foi realizado com as coordenadas dos horizontes exportados da sismica. As distâncias foram calculadas em relação ao mesmo (x_0, y_0) . Os horizontes foram então plotados como gráficos da profundidade pela distância com o *matplotlib*.

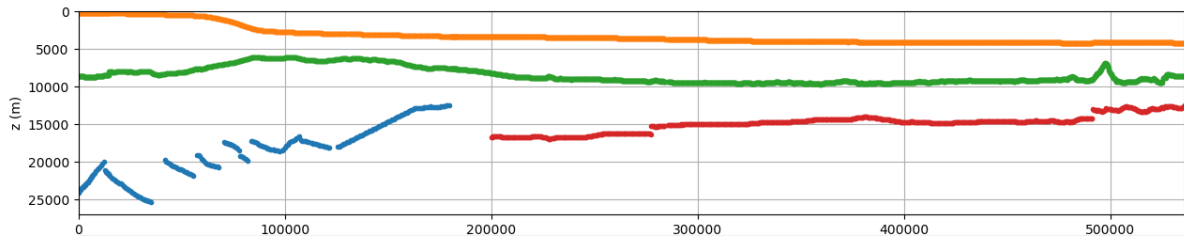


Figura 19 - Principais horizontes sísmicos plotados na interface do *matplotlib*. Em vermelho, topo do manto; em azul, topo do embasamento terrestre; em verde, topo do rift, na região proximal, e embasamento marinho, na região distal.

Com a base de dados pronta para a entrada no software de modelagem, pode-se dar início ao processo de ajustes no modelo inicial com o *fatiando a terra*. As densidades escolhidas puderam ser atribuídas às camadas e os horizontes puderam ser convertidos para polígonos a serem ajustados, iterativamente, de modo a gerar um perfil calculado que seja o mais próximo possível dos dados observados no perfil gravimétrico no console do aplicativo de modelagem.

5.2 A modelagem

Para a modelagem foi usado uma biblioteca do *python* chamado '*fatiando a terra*', provido de sub-bibliotecas e funções específicas para realizar tarefas comuns em geofísica, como modelagem e inversão de dados geofísicos. O pacote pode ser instalado via terminal (ou no prompt de comando, no caso do windows), uma vez que se tenha instalado certos módulos básicos para programação científica disponíveis no [Python Package Index](http://www.python.org/), repositório oficial de pacotes de terceiros desenvolvidos em *python* e de código aberto (para mais informações: <http://www.fatiando.org/>).

O *fatiando a terra* permite realizar a modelagem direta, através da função *Moulder*, com a criação e manipulação de polígonos em um perfil 2D de forma interativa, calculando o

efeito gravitacional. A função *Moulder* é uma aplicação que demanda atributos como a extensão da área, em metros, do perfil em que se deseja criar o modelo, e abre uma interface gráfica, assim que executada, do [matplotlib](#), biblioteca do *python* para se fazer gráficos em 2D, onde o usuário irá criar o modelo e observar a curva da resposta gravimétrica calculada.

Uma vez instalados os programas e pacotes necessários, o dado de gravimetria foi importado em formato csv com a finalidade de exibir, no console de modelagem, a anomalia bougher observada no local de estudo, que seria comparada ao perfil gravimétrico calculado do modelo. Para isso, um script, cuja descrição se encontra no anexo deste trabalho.

Após a execução do script, o console de modelagem foi aberto já com a curva da anomalia observada plotada.

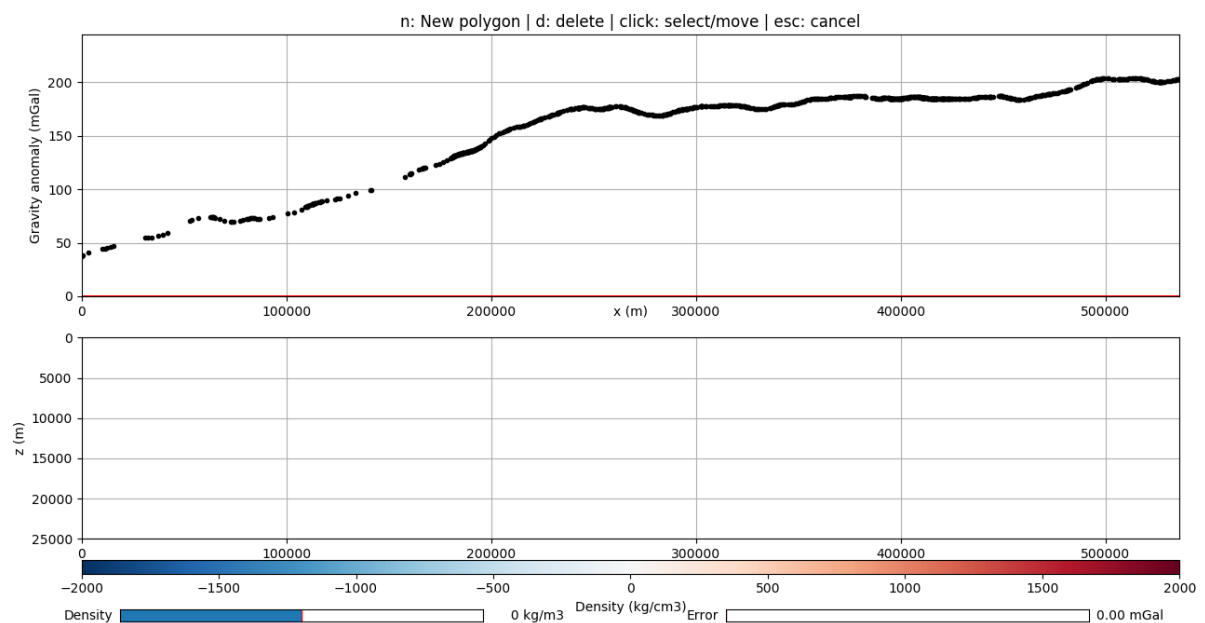


Figura 20 - Console de modelagem gravimétrica 2D interativo. O perfil de cima ilustra a anomalia observada em preto.

No retângulo desenhado sobre o eixo $z(m)$ e $x(m)$, elaborou-se os polígonos do modelo com base nas interfaces extraídas da sísmica sobre eixos com dimensões adequadas. No início, z foi escolhida para variar entre 0 e 40.000 metros, pois era a profundidade até onde se esperava encontrar estruturas interessantes para o trabalho, em analogia ao perfil da Figura 3 e da Figura 4; e x foi escolhida entre 0 e 536.205,9875 metros pois abrange a extensão total do perfil gravimétrico e da sísmica.

Antes de iniciar a modelagem, no entanto, os horizontes sísmicos de profundidade que serviriam de base para o modelo de início tiveram que ser importados. Para realizar esta tarefa, foi necessário ajustar o código fonte do programa, encontrado no anexo deste trabalho, pois o mesmo não foi projetado para receber dados de sísmica. Deste modo, os dados sísmicos foram plotados como gráficos de profundidade X distância.

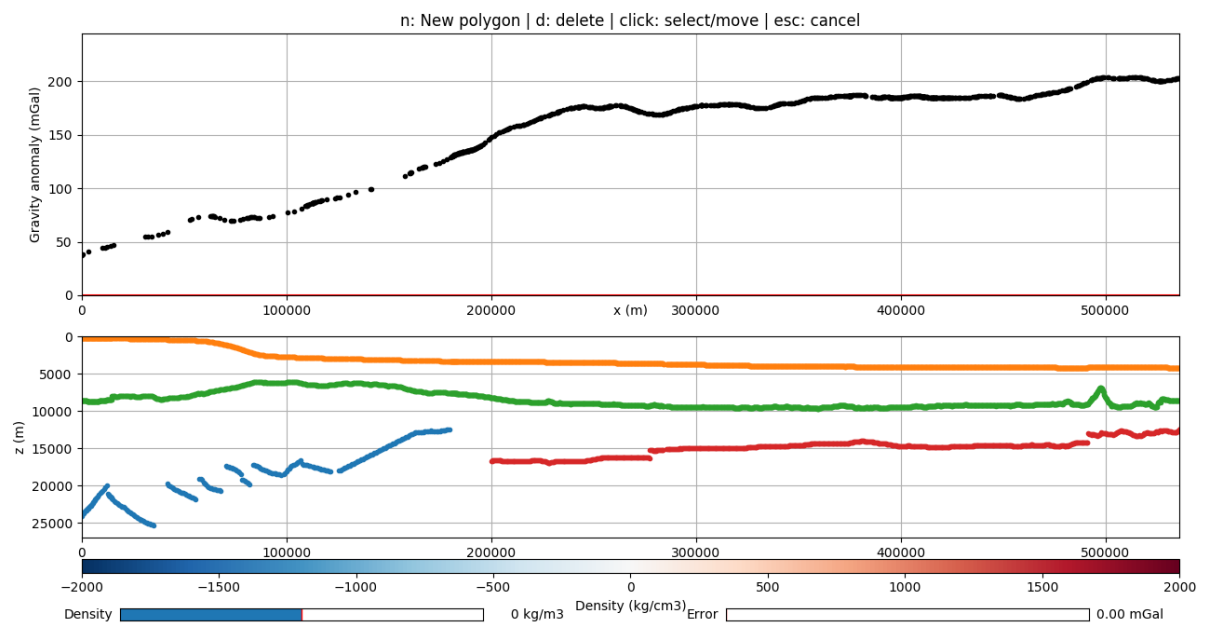


Figura 21 - Horizontes sísmicos plotados como curvas de diversas cores.

As densidades escolhidas para o modelo inicial estão de acordo com as densidades típicas das rochas que compõem cada formação representada pelos polígonos, todas elas derivadas dos trabalhos de Pawlowski (2008) e Blaich et al. (2011).

As densidades escolhidas foram aplicadas à densidade média de referência da crosta, equivalente a 2670 kg/m^3 , para o cálculo do $\Delta\rho$ de cada polígono. Abaixo, encontram-se descritas as escolhas para as densidades.

- Densidade média das rochas do manto = 3300 Kg/m^3 ; logo, $\Delta\rho_{\text{manto}} = 3300 - 2670 = 630 \text{ Kg/m}^3$

- Densidade média da crosta oceânica = 2850 Kg/m^3 ; logo, $\Delta\rho_{\text{crosta_oceânica}} = 2850 - 2670 = 180 \text{ Kg/m}^3$
- Densidade média dos sedimentos = 2050 Kg/m^3 ; logo, $\Delta\rho_{\text{sedimentos}} = 2050 - 2670 = -620 \text{ Kg/m}^3$
- Densidade média dos sedimentos basais = 2400 Kg/m^3 ; logo, $\Delta\rho_{\text{sedimentos_basais}} = 2400 - 2670 = -270 \text{ Kg/m}^3$
- Densidade média da crosta intermediária = 2750 Kg/m^3 ; logo, $\Delta\rho_{\text{crosta_intermediária}} = 2750 - 2670 = 80 \text{ Kg/m}^3$
- Densidade média do *underplating* = 3000 Kg/m^3 ; logo, $\Delta\rho_{\text{underplating}} = 3000 - 2670 = 330 \text{ Kg/m}^3$

Com as densidades em mãos, os polígonos foram desenhados na interface do programa, tendo como referência os horizontes importados da sísmica. Os únicos polígonos que tiveram seus vértices ajustados depois foram os que não eram completamente limitados pelas interfaces importadas da sísmica.

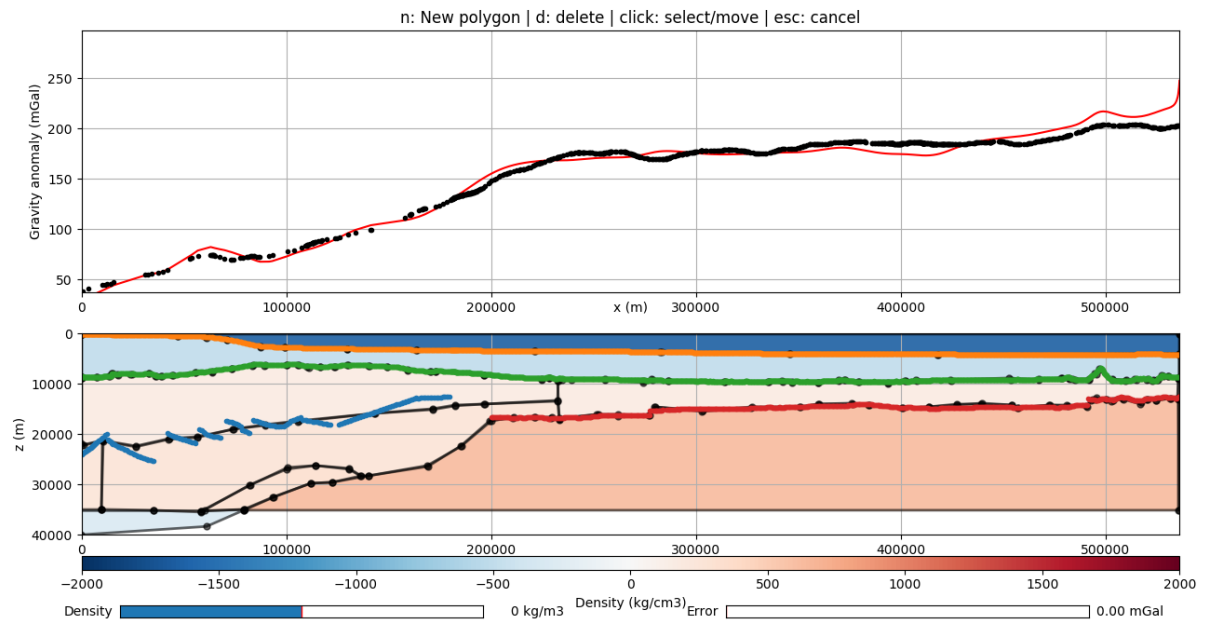


Figura 22 - Modelo sobre as interfaces plotadas. A curva em vermelho é a anomalia calculada. O polígono azul equivale ao corpo de água; o azul claro, aos sedimentos; os polígonos em vermelho claro representam a fase rifte/pré-rifte (delimitada pelas interfaces azul e verde), crosta oceânica (delimitada pelas interfaces vermelha e verde), e crosta intermediária (delimitada pela interface azul e logo abaixo pelo manto, representando o polígono vermelho).

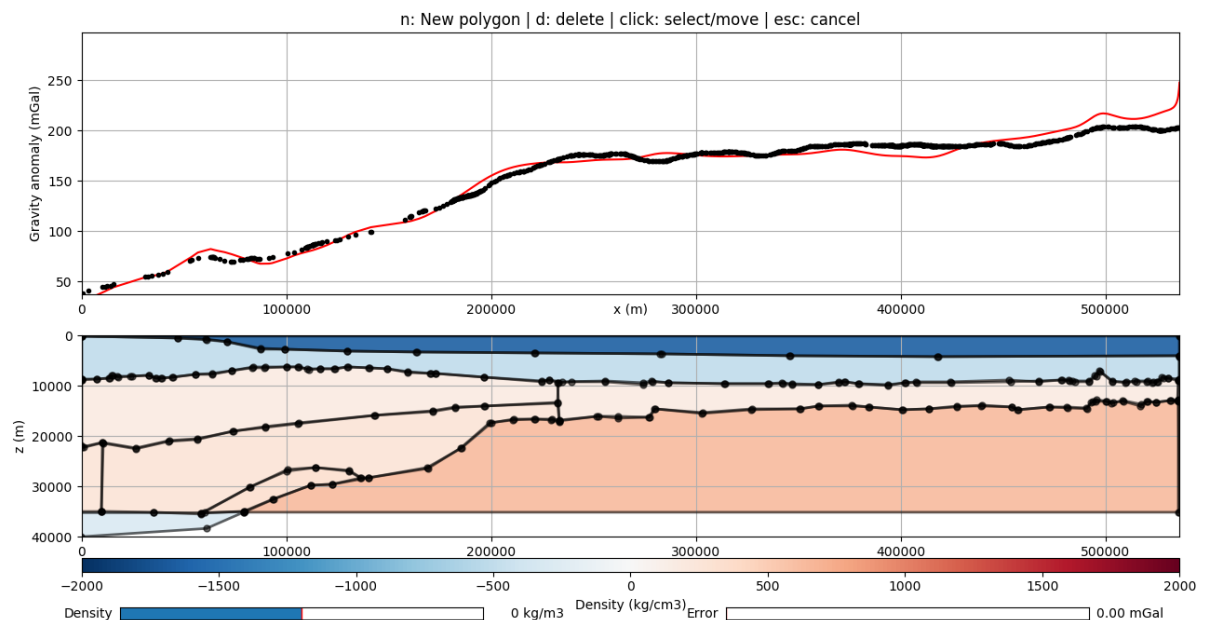


Figura 23 - Modelo final sem o desenho dos horizontes, apenas com os polígonos desenhados.

O programa, no entanto, impôs limitação à atribuição das densidades aos polígonos, pois não foi possível entrar com as densidades exatas na interface do programa. Por causa disso, algumas densidades tiveram que ser escolhidas como as listadas abaixo.

- $\Delta\rho_{\text{água}} = -1642 \text{ Kg/m}^3$
- $\Delta\rho_{\text{sedimentos}} = -505 \text{ Kg/m}^3$
- $\Delta\rho_{\text{rifte}} = 182 \text{ Kg/m}^3$
- $\Delta\rho_{\text{crosta_oceânica}} = 182 \text{ Kg/m}^3$
- $\Delta\rho_{\text{crosta_intermediária}} = 274 \text{ Kg/m}^3$
- $\Delta\rho_{\text{manto}} = 633 \text{ Kg/m}^3$
- $\Delta\rho_{\text{underplating}} = 336 \text{ Kg/m}^3$ e -300 Kg/m^3 (negativo, pois é a porção do *underplating* que se encontra abaixo da profundidade normal da crosta de 35 Km, como é possível notar no polígono violeta da Figura 25, portanto teve que ser compensada pela densidade do manto de referência: $\Delta\rho = 3000 \text{ Kg/m}^3 - 3300 \text{ Kg/m}^3$)

As figuras a seguir foram editadas no *paint* com a finalidade de auxiliar na visualização do modelo.

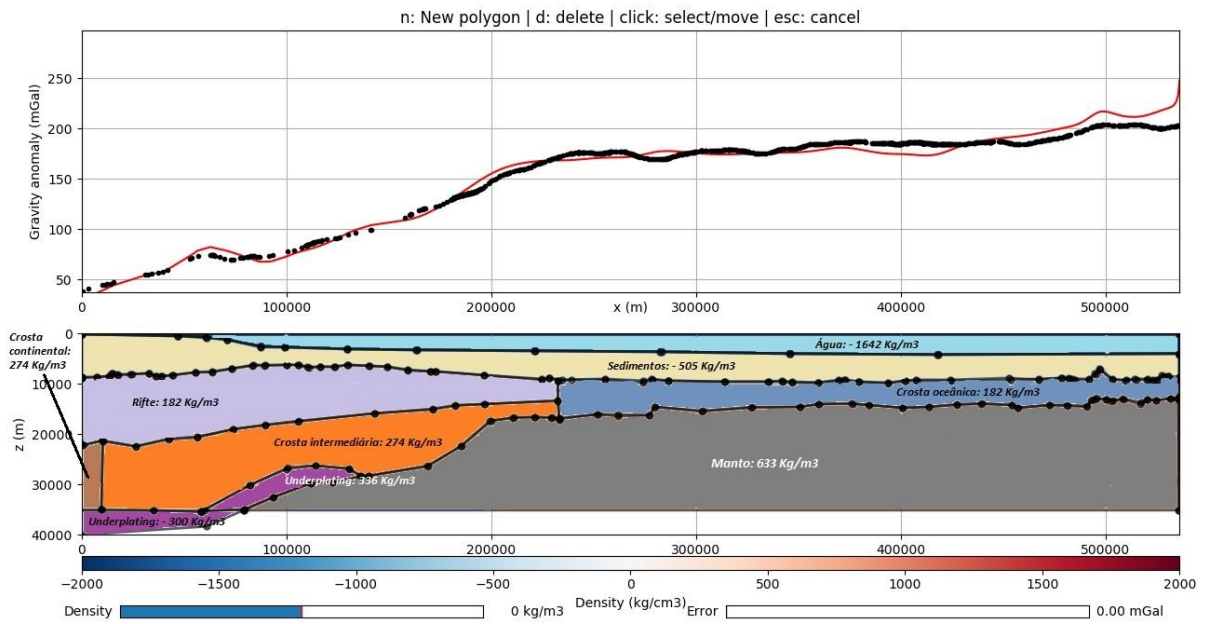


Figura 24 - Modelo final sem o desenho dos horizontes, apenas com os polígonos desenhados. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização. Os valores associados aos polígonos representam as densidades, ou seja, $\Delta\rho$, atribuídas a cada corpo, e estão todas em Kg/m^3 , apesar de, nesta imagem estar Kg/m^3 por limitação do editor de textos da ferramenta de edição de imagens.

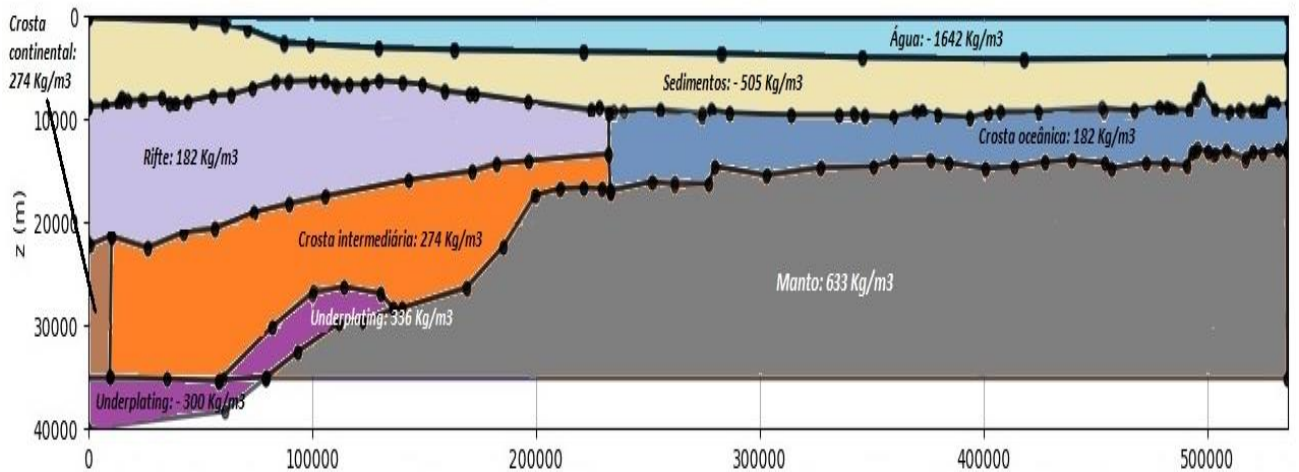


Figura 25 - Modelo final sem o desenho dos horizontes, apenas com os polígonos desenhados. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização. Os valores associados aos polígonos representam as densidades, ou seja, $\Delta\rho$, atribuídas a cada corpo, e estão todas em Kg/m^3 , apesar de, nesta imagem, estarem representadas como 'Kg/m³' devido a limitação do editor de textos da ferramenta de edição de imagens.

O rifte e o pré-rifte foram inseridos no mesmo polígono, nomeado rifte, e a densidade escolhida para esse polígono foi a mesma da crosta oceânica devido à composição similar dessas camadas.

A crosta intermediária, representada pelo polígono de cor laranja na Figura 25, teve uma variação de densidade mais significativa em relação às densidades extraídas da bibliografia, pois foi escolhida conforme a que melhor se ajustava ao modelo, e o seu topo foi contornado de modo a suavizar o horizonte extraído da sísmica, que era mais acidentado (Figura 22). A densidade dos sedimentos, representado pelo polígono amarelo claro, também foi escolhida uma que ajustava bem e estivesse dentro do intervalo $\Delta\rho_{sedimentos}$ - $\Delta\rho_{sedimentos_basais}$ retiradas da bibliografia e listadas anteriormente. As alterações nas demais densidades aconteceram devido a apenas a limitação na entrada dos dados de densidade.

O modelo também contou com outros polígonos fora da cobertura do perfil gravimétrico para eliminar o efeito de borda, como pode ser visto na Figura 26 e na Figura 28, assim como nas suas versões alternativas, que são a Figura 27 e a Figura 29, respectivamente. Na margem direita, foi colocado polígonos que continuaram as mesmas estruturas tendendo ao “infinito”. Na margem esquerda, foi desenhado polígonos para o restante da camada sedimentar e da fase rifte/pré-rifte, além de um polígono para o restante da crosta continental, em marrom na Figura 25 e na Figura 27. Além disso, também foi colocado uma fina camada representando o *underplating*.

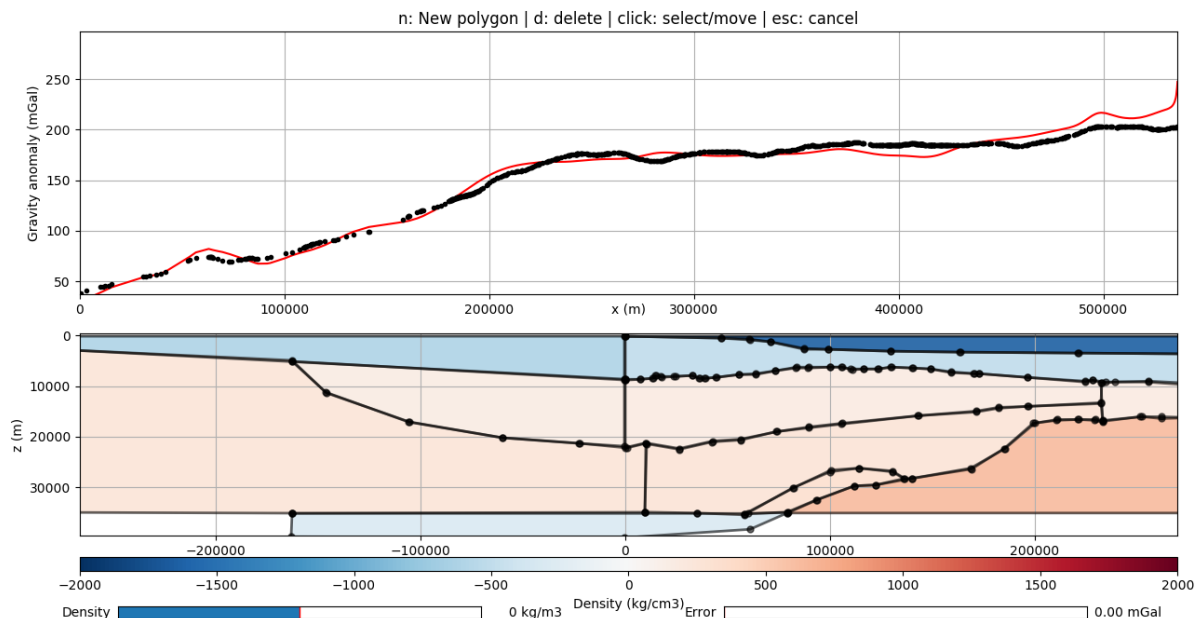


Figura 26 – Borda esquerda do modelo gravimétrico.

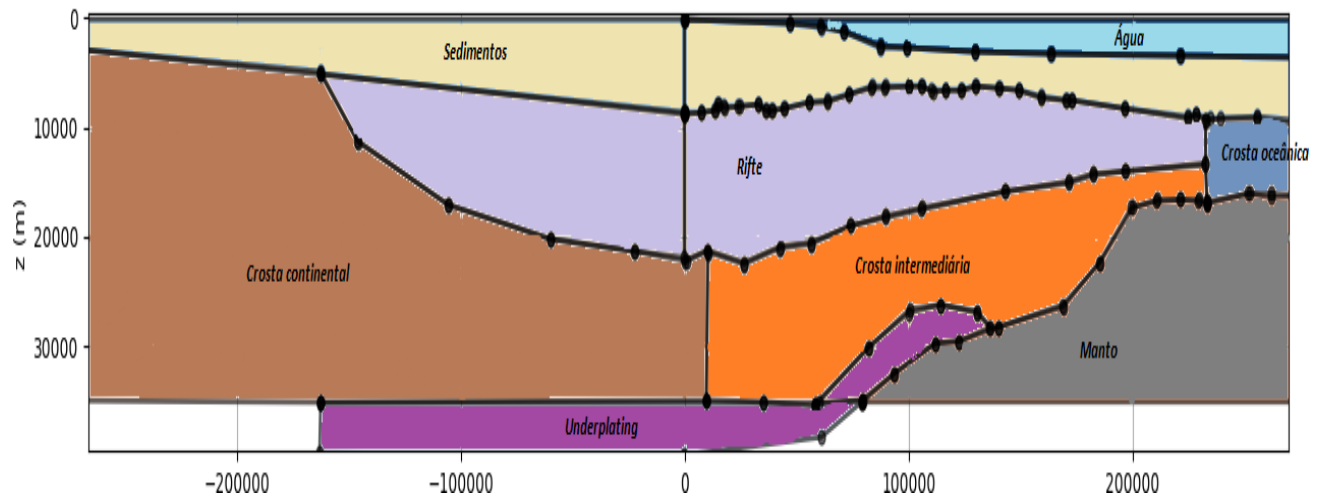


Figura 27 - Borda esquerda do modelo gravimétrico. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização.

Algo que pode chamar a atenção é a longa extensão do rifte, mas o objetivo dos polígonos para a esquerda do perfil (coordenadas negativas na Figura 27) era apenas contornar o efeito de borda e o traçado dos polígonos muito longe da borda não teriam efeito considerável, então os traçados foram os mais simples possíveis. O polígono marrom, nomeada simplesmente crosta continental, abrange tanto a crosta continental superior e crosta continental inferior, e a sua densidade foi escolhida de forma a encontrar o melhor ajuste para o modelo, que foi de $\Delta\rho = 254 \text{ Kg/m}^3$, relativamente mais denso do que o normal, assim como o resultado do trabalho de Blaich (2013), que só conseguiu um bom ajuste com uma crosta de alta densidade.

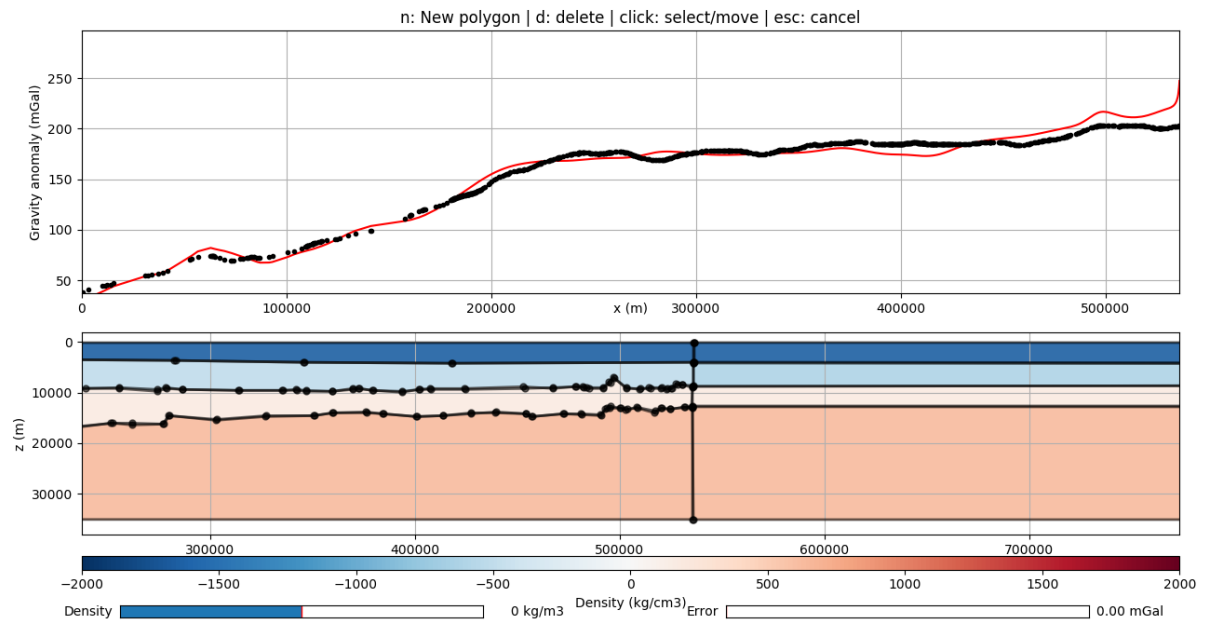


Figura 28 – Borda direita do modelo gravimétrico.

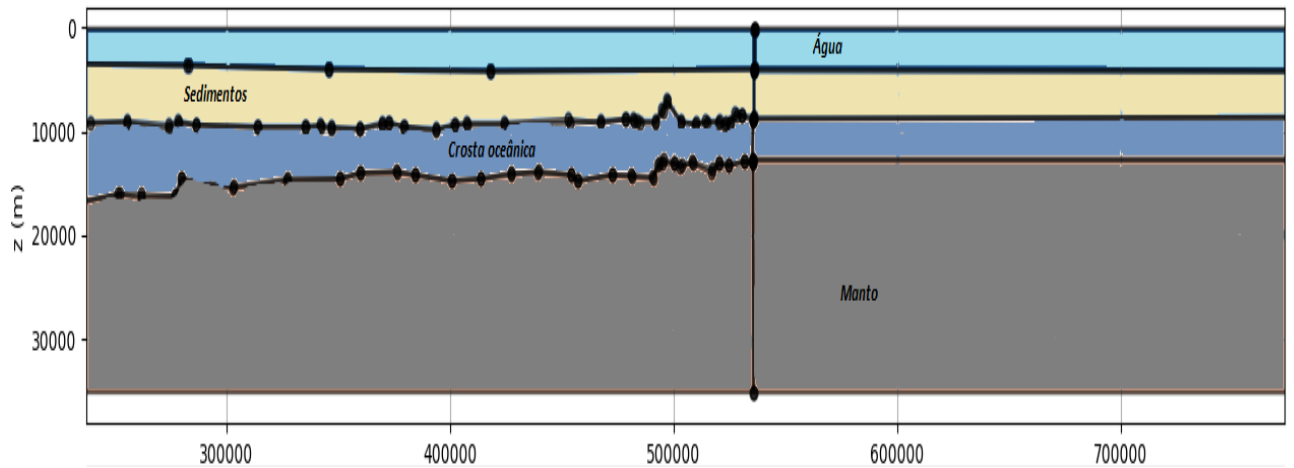


Figura 29 - Borda direita do modelo gravimétrico. Os polígonos foram coloridos e nomeados a fim de facilitar a visualização.

No final, a diferença entre a curva calculada e a observada foi calculada, e foi feito um gráfico, que pode ser visto na Figura 30, para apresentar o erro. O código usado no IDE *spyder* se encontra no anexo.

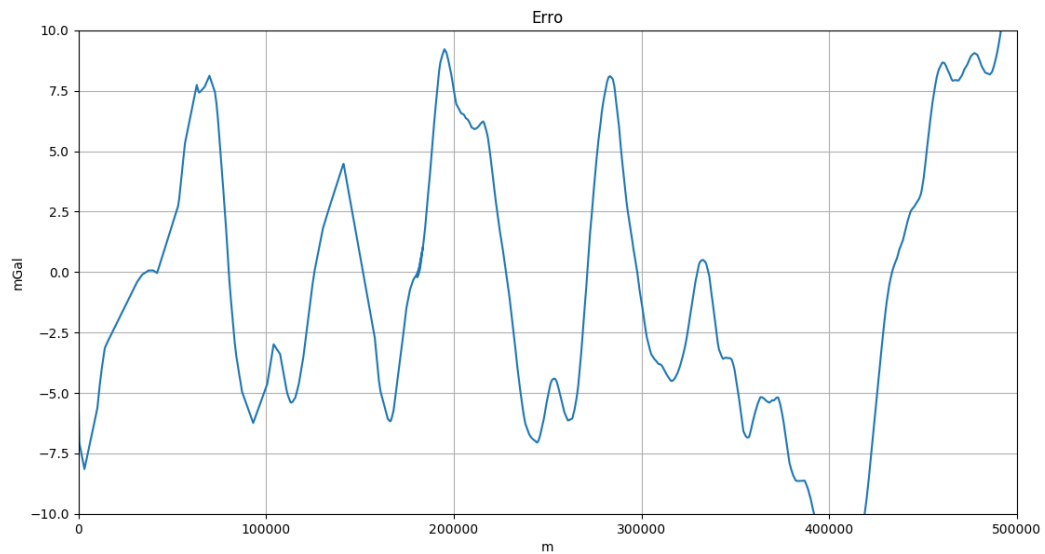


Figura 30 – Erro representando a discrepância entre os dados de gravimetria observados e os calculados.

O erro se encontra entre -10 e 10 na maior parte da seção, e foi o melhor ajuste encontrado. Entre as causas dos erros podem estar imprecisões nos traçados dos polígonos e atribuição das densidades, que já foi comentado, ou equívocos na interpretação sísmica, ou mesmo ambos.

Percebe-se, pela análise do gráfico do erro, que a maior parte do erro se encontra na porção em que os polígonos são limitados apenas pelas sísmicas. Isto ocorreu porque a técnica usada para suprimir o efeito de borda não foi tão bem sucedida na borda direita quanto foi na esquerda, e também porque os lugares onde não tinha dados provenientes da sísmica permitiu mais especulação e mais possibilidades de ajustes, o que ajudou a reduzir os erros nessas partes, que foi o caso do topo do manto e do *underplating* na margem esquerda. Uma alteração nas densidades da margem direita, a fim de melhorar o ajuste nesta porção do perfil, tiraria o melhor ajuste do restante da seção.

Os erros poderiam ter sido reduzidos se o contorno dos polígonos pudesse ser contornado de forma mais precisa. A escala da representação do modelo colocava limites a precisão na elaboração do modelo junto aos horizontes importados da sísmica. Afinal, era 40 Km de profundidade para mais de 500 Km de extensão para modelar. Na interface do programa, um movimento mínimo do mouse, representavam centenas de metros no modelo. Portanto, erros de centenas de metros no desenho dos polígonos foram inevitáveis.

CONCLUSÕES

O trabalho visou a modelagem de um modelo geológico transversal à linha de costa da Bacia de Pelotas que seja coerente com dados gravimétricos da região. O método de interpretação escolhido foi a modelagem direta, utilizando como dado de entrada o conhecimento prévio da geologia regional e dados provenientes de perfis sísmicos de reflexão. O modelo gerado apresentou uma anomalia predita que é, dentro de certas proporções, similar à observada e coerente com o conhecimento que há sobre a região.

Os horizontes provenientes da sísmica formaram uma boa referência para os polígonos, reduzindo o número de soluções possíveis para a interpretação da bacia. As informações retiradas da bibliografia sobre a geologia da Bacia de Pelotas também foram de bom uso, pois permitiram produzir resultados mais coerentes nos locais onde não havia dados da sísmica e que, deste modo, permitiam mais especulações até que se encontrasse o melhor ajuste.

O programa usado na modelagem não fazia algumas coisas que outros softwares do mercado fazem automaticamente, mas este problema foi relativamente fácil de contornar, como o do efeito das bordas, permitindo uma contribuição maior para o aprendizado do usuário. A ferramenta usada no trabalho mostrou-se muito promissora para o estudo e prática de modelagem direta em métodos potenciais.

REFERÊNCIAS

- BLAICH, O.A. et al. **Crustal breakup and continent-ocean transition at South Atlantic conjugate margins**. *Journal of Geophysical Research*, vol. 116, B01402, doi:10.1029/2010JB007686, 2011.
- BLAICH, O.A. et al. **Crustal-scale architecture and segmentation of the South Atlantic volcanic margin**. Geological Society, London, Special Publications, 369, 167–183.
- BLAKELY, R.J. **Potential Theory in Gravity and Magnetic Applications**. Cambridge University Press, 1995.
- BUENO, G.V. et al. **Bacia de Pelotas**. *Boletim de Geociências da Petrobras*, Rio de Janeiro, v.15, n.2, p.551-559, 2007.
- CASTRO, D.L. **Modelagem Gravimétrica 3-D de Corpos Graníticos e Bacias Sedimentares com Embasamento Estrutural de Densidade Variável**. *Revista Brasileira de Geofísica*, v.23, n.3, p.295-308, 2005
- DOBRIN, M.B., SAVIT, C.H. **Introduction to Geophysical Prospecting**, McGraw-Hill International Editions, Singapore, 888p, 1988.
- GEOFFROY, L. **Volcanic passive margins**. *Comptes Rendus Geosci.* 337,1395-1408, 2005
- HUNTER, J. D., **Matplotlib: A 2D Graphics Environment**, *Computing in Science & Engineering*, 9(3), 90-95, doi:10.1109/MCSE.2007.55, 2007.
- KEARY, P., BROOKS, MICHAEL., HILL, I. **Geofísica de Exploração**, Trad. Maria Coelho (coord.) et al. Ed. Oficina do Texto, São Paulo, SP, 2009.
- MUTTER, J.C. **Seaward dipping reflectors and the continental-ocean boundary at passive continental margins**. *Tectonophysics*, 114, p. 117-131, 1985.
- NGA. **GM-SYS - Gravity/Magnetic modelling software user's guide**. Corvallis: Northwest Geophysical Associate, 2004.
- NUSSENZVEIG, H.M. **Curso de Física Básica: Mecânica**. São Paulo – Edgard Blucher, 2002.
- PAWLOWSKI, R. **The use of gravity anomaly data for offshore continental margin demarcation**, *The Leading Edge*, p. 722-727, June 2008.
- PÉRON-PINVIDIC, G., MANATSCHAL, G., OSMUNDSEN, P.T. **Structural comparison of archetypal Atlantic rifted margins: a review of observations and concepts**. *Mar. Pet. Geology*. 43, 21-47, 2013.

TALWANI, M., J. L. WORZEL, M. LANDISMAN. **Rapid gravity computations for two-dimensional bodies with application to the Mendocino Submarine Fracture Zone.** Journal of Geophysical Research, v.64. n.1, p. 49-59, 1959.

TEIXEIRA, W.; TOLEDO, M.C.M. de; FAIRCHILD, T.R.; TAIOLI, F. (Orgs.) **Decifrando a Terra.** São Paulo: Oficina de Textos, 568 p., 2000.

UIEDA, L., V. C. OLIVEIRA JR, and V. C. F. BARBOSA, **Modeling the Earth with Fatiando a Terra,** Proceedings of the 12th Python in Science Conference, pp. 91 – 98, 2013.

ZALÁN, P.V., SEVERINO, M.C.G., RIGOTI, C.A., MAGNAVITA, L.P., OLIVEIRA, J.A., VIANNA, A.R. **An entirely new 3D-View of the crustal and mantle structure of a south Atlantic passive margin -Santos, Campos and Espírito Santo basins, Brazil.** In: AAPG Annual Convention and Exhibition Abstracts Volume CDROM. AAPG, Houston, Texas, USA, 2011.

ANEXOS

Código fonte do *fatiando.gravmag.interactive* com as alterações feitas para importar as interfaces provenientes da sísmica.

```

"""
Interactivity functions and classes using matplotlib and IPython
widgets

**Gravity forward modeling**

* :class:`~fatiando.gravmag.interactive.Moulder`: a matplotlib GUI for
2D
  forward modeling using polygons

----

"""
from __future__ import division, absolute_import
from future.builtins import zip
try:
    import cPickle as pickle
except ImportError:
    import pickle

import numpy
from matplotlib import pyplot, widgets, patches
from matplotlib.lines import Line2D
from IPython.core.pylabtools import print_figure
from IPython.display import Image

import utils
import talwani
from geometry import Polygon

# Importando os dados provenientes da sísmica

file1=numpy.loadtxt('Data-XZ-Embasamento.csv')
file2=numpy.loadtxt('Data-XZ-FundoDoMar.csv')
file3=numpy.loadtxt('Data-XZ-H.csv')
file4=numpy.loadtxt('Data-XZ-Moho.csv')

xfile1=file1[:,0]
yfile1=file1[:,1]
xfile2=file2[:,0]
yfile2=file2[:,1]
xfile3=file3[:,0]
yfile3=file3[:,1]
xfile4=file4[:,0]
yfile4=file4[:,1]

class Moulder(object):

```

```
"""
```

```
Interactive 2D forward modeling using polygons.
```

```
A matplotlib GUI application. Allows drawing and manipulating
polygons and
  computes their predicted data automatically. Also permits
contaminating the
  data with gaussian pseudo-random error for producing synthetic data
sets.
```

```
Uses :mod:`fatiando.gravmag.talwani` for computations.
```

```
*Moulder* objects can be persisted to Python pickle files using the
:meth:`~fatiando.gravmag.interactive.Moulder.save` method and later
restored using :meth:`~fatiando.gravmag.interactive.Moulder.load`.
```

```
.. warning::
```

```
Cannot be used with ``%matplotlib inline`` on IPython notebooks
because
  the app uses the matplotlib plot window. You can still embed
the
  generated model and data figure on notebooks using the
:meth:`~fatiando.gravmag.interactive.Moulder.plot` method.
```

```
Parameters:
```

```
* area : list = (x1, x2, z1, z2)
  The limits of the model drawing area, in meters.
```

```
* x, z : 1d-arrays
  The x- and z-coordinates of the computation points (places
where
  predicted data will be computed). In meters.
```

```
* data : None or 1d-array
  Observed data measured at *x* and *z*. Will plot this with
black dots
  along the predicted data.
```

```
* density_range : list = [min, max]
  The minimum and maximum values allowed for the density.
```

```
Determines the
  limits of the density slider of the application. In kg.m-3.
```

```
Defaults
  to [-2000, 2000].
```

```
* kwargs : dict
  Other keyword arguments used to restore the state of the
application.
```

```
Used by the :meth:`~fatiando.gravmag.interactive.Moulder.load`
method.
```

```
Not intended for general use.
```

```
Examples:
```

```
Make the Moulder object and start the app::
```

```
import numpy as np
area = (0, 10e3, 0, 5e3)
# Calculate on 100 points
x = np.linspace(area[0], area[1], 100)
```

```

z = np.zeros_like(x)
app = Moulder(area, x, z)
app.run()
# This will pop-up a window with the application (like the
screenshot
# below). Start drawing (follow the instruction in the figure
title).
# When satisfied, close the window to resume execution.

.. image:: ../_static/images/Moulder-screenshot.png
:alt: Screenshot of the Moulder GUI

```

After closing the plot window, you can access the model and data from the

```

*Moulder* object::

app.model # The drawn model as fatiando.mesher.Polygon
app.predicted # 1d-array with the data predicted by the model
# You can save the predicted data to use later
app.save_predicted('data.txt')
# You can also save the application and resume it later
app.save('application.pkl')
# Close this session/IPython notebook/etc.
# To resume drawing later:
app = Moulder.load('application.pkl')
app.run()

"""

# The tolerance range for mouse clicks on vertices. In pixels.
epsilon = 5
# App instructions printed in the figure suptitle
instructions = ' | '.join([
    'n: New polygon', 'd: delete', 'click: select/move', 'esc:
cancel'])

def __init__(self, area, x, z, data=None, density_range=[-2000,
2000],
            **kwargs):
    self.area = area
    self.x, self.z = numpy.asarray(x), numpy.asarray(z)
    self.density_range = density_range
    self.data = data
    # Used to set the ylims for the data axes.
    if data is None:
        self.dmin, self.dmax = 0, 0
    else:
        self.dmin, self.dmax = data.min(), data.max()
    self.predicted = kwargs.get('predicted', numpy.zeros_like(x))
    self.error = kwargs.get('error', 0)
    self.cmap = kwargs.get('cmap', pyplot.cm.RdBu_r)
    self.line_args = dict(
        linewidth=2, linestyle='-', color='k', marker='o',
        markerfacecolor='k', markersize=5, animated=False,
alpha=0.6)
    self.polygons = []
    self.lines = []

```



```

self.densities = kwargs.get('densities', [])
vertices = kwargs.get('vertices', [])
for xy, dens in zip(vertices, self.densities):
    poly, line = self._make_polygon(xy, dens)
    self.polygons.append(poly)
    self.lines.append(line)

def save_predicted(self, fname):
    """
    Save the predicted data to a text file.

    Data will be saved in 3 columns separated by spaces: x z data

    Parameters:

    * fname : string or file-like object
      The name of the output file or an open file-like object.

    """
    numpy.savetxt(fname, numpy.transpose([self.x, self.z,
self.predicted]))

def save(self, fname):
    """
    Save the application state into a pickle file.

    Use this to persist the application. You can later reload the
entire
    object, with the drawn model and data, using the
:meth:`~fatiando.gravmag.interactive.Moulder.load` method.

    Parameters:

    * fname : string
      The name of the file to save the application. The extension
doesn't
      matter (use ``.pkl`` if in doubt).

    """
    with open(fname, 'w') as f:
        vertices = [numpy.asarray(p.xy) for p in self.polygons]
        state = dict(area=self.area, x=self.x,
                    z=self.z, data=self.data,
                    density_range=self.density_range,
                    cmap=self.cmap,
                    predicted=self.predicted,
                    vertices=vertices,
                    densities=self.densities,
                    error=self.error)
        pickle.dump(state, f)

@classmethod
def load(cls, fname):
    """
    Restore an application from a pickle file.

    The pickle file should have been generated by the
:meth:`~fatiando.gravmag.interactive.Moulder.save` method.

```

```

Parameters:

* fname : string
    The name of the file.

Returns:

* app : Moulder object
    The restored application. You can continue using it as if
nothing
    had happened.

"""
with open(fname) as f:
    state = pickle.load(f)
app = cls(**state)
return app

@property
def model(self):
    """
    The polygon model drawn as :class:`fatiando.mesher.Polygon`
objects.
    """
    m = [Polygon(p.xy, {'density': d})
         for p, d in zip(self.polygons, self.densities)]
    return m

def run(self):
    """
    Start the application for drawing.

    Will pop-up a window with a place for drawing the model (below)
and a
    place with the predicted (and, optionally, observed) data
(top).

    Follow the instruction on the figure title.

    When done, close the window to resume program execution.
    """
    fig = self._figure_setup()
    # Sliders to control the density and the error in the data
    self.density_slider = widgets.Slider(
        fig.add_axes([0.10, 0.01, 0.30, 0.02]), 'Density',
        self.density_range[0], self.density_range[1], valinit=0.,
        valfmt='%6.0f kg/m3')
    self.error_slider = widgets.Slider(
        fig.add_axes([0.60, 0.01, 0.30, 0.02]), 'Error',
        0, 5, valinit=self.error, valfmt='%1.2f mGal')
    # Put instructions on figure title
    self.dataax.set_title(self.instructions)
    # Markers for mouse click events
    self._invert = None
    self._ipoly = None
    self._lastevent = None
    self._drawing = False

```

```

self._xy = []
self._drawing_plot = None
# Used to blit the model plot and make
# rendering faster
self.background = None
# Connect event callbacks
self._connect()
self._update_data()
self._update_data_plot()
self.canvas.draw()
pyplot.show()

def _connect(self):
    """
    Connect the matplotlib events to their callback methods.
    """
    # Make the proper callback connections
    self.canvas.mpl_connect('button_press_event',
                            self._button_press_callback)
    self.canvas.mpl_connect('key_press_event',
                            self._key_press_callback)
    self.canvas.mpl_connect('button_release_event',
                            self._button_release_callback)
    self.canvas.mpl_connect('motion_notify_event',
                            self._mouse_move_callback)
    self.canvas.mpl_connect('draw_event',
                            self._draw_callback)
    # Call the cleanup and extra code for a draw event when
resizing as
    # well. This is needed so that tight_layout adjusts the figure
when
    # resized. Otherwise, tight_layout snaps only when the user
clicks on
    # the figure to do something.
    self.canvas.mpl_connect('resize_event',
                            self._draw_callback)
    self.density_slider.on_changed(self._set_density_callback)
    self.error_slider.on_changed(self._set_error_callback)

def plot(self, figsize=(10, 8), dpi=70):
    """
    Make a plot of the data and model for embedding in IPython
notebooks

    Doesn't require ``%matplotlib inline`` to embed the plot (as
that would
    not allow the app to run).

    Parameters:

    * figsize : list = (width, height)
      The figure size in inches.
    * dpi : float
      The number of dots-per-inch for the figure resolution.
    """
    fig = self._figure_setup(figsize=figsize, facecolor='white')
    self._update_data_plot()

```

```

pyplot.close(fig)
data = print_figure(fig, dpi=dpi)
return Image(data=data)

def _figure_setup(self, **kwargs):
    """
    Setup the plot figure with labels, titles, ticks, etc.

    Sets the *canvas*, *dataax*, *modelax*, *polygons* and *lines*
    attributes.

    Parameters:

    * kwargs : dict
        Keyword arguments passed to ``pyplot.subplots``.

    Returns:

    * fig : matplotlib figure object
        The created figure

    """
    fig, axes = pyplot.subplots(2, 1, **kwargs)
    ax1, ax2 = axes
    self.predicted_line, = ax1.plot(self.x, self.predicted, '-r')
    if self.data is not None:
        self.data_line, = ax1.plot(self.x, self.data, '.k')
    ax1.set_ylabel('Gravity anomaly (mGal)')
    ax1.set_xlabel('x (m)', labelpad=-10)
    ax1.set_xlim(self.area[:2])
    ax1.set_ylim((-200, 200))
    ax1.grid(True)
    tmp = ax2.pcolor(numpy.array([self.density_range]),
cmap=self.cmap)
    tmp.set_visible(False)
    pyplot.colorbar(tmp, orientation='horizontal',
                    pad=0.08, aspect=80).set_label(r'Density
(kg/cm3)')
    # Remake the polygons and lines to make sure they belong to the
right
    # axis coordinates
    vertices = [p.xy for p in self.polygons]
    newpolygons, newlines = [], []
    for xy, dens in zip(vertices, self.densities):
        poly, line = self._make_polygon(xy, dens)
        newpolygons.append(poly)
        newlines.append(line)
        ax2.add_patch(poly)
        ax2.add_line(line)
    self.polygons = newpolygons
    self.lines = newlines
    ax2.set_xlim(self.area[:2])
    ax2.set_ylim(self.area[2:])
    ax2.grid(True)
    ax2.invert_yaxis()
    ax2.set_ylabel('z (m)')
    fig.subplots_adjust(top=0.95, left=0.1, right=0.95,
bottom=0.06,

```

```

        hspace=0.1)

    'Plotando a simica na segunda figura'

    ax2.plot(xfile1,yfile1)
    ax2.plot(xfile2,yfile2)
    ax2.plot(xfile3,yfile3)
    ax2.plot(xfile4,yfile4)
    self.figure = fig
    self.canvas = fig.canvas
    self.dataax = axes[0]
    self.modelax = axes[1]
    fig.canvas.draw()
    return fig

def _density2color(self, density):
    """
    Map density values to colors using the given *cmap* attribute.

    Parameters:

    * density : 1d-array
        The density values of the model polygons

    Returns

    * colors : 1d-array
        The colors mapped to each density value (returned by a
matplotlib
        colormap object.

    """
    dmin, dmax = self.density_range
    return self.cmap((density - dmin)/(dmax - dmin))

def _make_polygon(self, vertices, density):
    """
    Create a polygon for drawing.

    Polygons are matplotlib.patches.Polygon objects for the fill
and
matplotlib.lines.Line2D for the contour.

    Parameters:

    * vertices : list of [x, z]
        List of the [x, z] coordinate pairs of each vertex of the
polygon
    * density : float
        The density of the polygon (used to set the color)

    Returns:

    * polygon, line
        The matplotlib Polygon and Line2D objects

    """
    poly = patches.Polygon(vertices, animated=False, alpha=0.9,

```

```

        color=self._density2color(density))
    x, y = list(zip(*poly.xy))
    line = Line2D(x, y, **self.line_args)
    return poly, line

def _update_data(self):
    """
    Recalculate the predicted data (optionally with random error)
    """
    self.predicted = talwani.gz(self.x, self.z, self.model)
    if self.error > 0:
        self.predicted = utils.contaminate(self.predicted,
self.error)

def _update_data_plot(self):
    """
    Update the predicted data plot in the *dataax*.

    Adjusts the xlim of the axes to fit the data.
    """
    self.predicted_line.set_ydata(self.predicted)
    vmin = 1.2*min(self.predicted.min(), self.dmin)
    vmax = 1.2*max(self.predicted.max(), self.dmax)
    self.dataax.set_ylim(vmin, vmax)
    self.dataax.grid(True)
    self.canvas.draw()

def _draw_callback(self, value):
    """
    Callback for the canvas.draw() event.

    This is called everytime the figure is redrawn. Used to do some
    clean up and tuning whenever this is called as well, like
calling
    ``tight_layout``.
    """
    self.figure.tight_layout()

def _set_error_callback(self, value):
    """
    Callback when error slider is edited
    """
    self.error = value
    self._update_data()
    self._update_data_plot()

def _set_density_callback(self, value):
    """
    Callback when density slider is edited
    """
    if self._ipoly is not None:
        self.densities[self._ipoly] = value

self.polygons[self._ipoly].set_color(self._density2color(value))
    self._update_data()
    self._update_data_plot()
    self.canvas.draw()

```

```

def _get_polygon_vertice_id(self, event):
    """
    Find out which vertex of which polygon the event happened in.

    If the click was inside a polygon (not on a vertex), identify
that
    polygon.

    Returns:

    * p, v : int, int
      p: the index of the polygon the event happened in or None
if
      outside all polygons.
      v: the index of the polygon vertex that was clicked or None
if the
      click was not on a vertex.

    """
    distances = []
    indices = []
    for poly in self.polygons:
        x, y = poly.get_transform().transform(poly.xy).T
        d = numpy.sqrt((x - event.x)**2 + (y - event.y)**2)
        distances.append(d.min())
        indices.append(numpy.argmin(d))
    p = numpy.argmin(distances)
    if distances[p] >= self.epsilon:
        # Check if the event was inside a polygon
        x, y = event.x, event.y
        p, v = None, None
        for i, poly in enumerate(self.polygons):
            if poly.contains_point([x, y]):
                p = i
                break
    else:
        v = indices[p]
        last = len(self.polygons[p].xy) - 1
        if v == 0 or v == last:
            v = [0, last]
    return p, v

def _button_press_callback(self, event):
    """
    What actions to perform when a mouse button is clicked
    """
    if event.inaxes != self.modelax:
        return
    if event.button == 1 and not self._drawing and self.polygons:
        self._lastevent = event
        for line, poly in zip(self.lines, self.polygons):
            poly.set_animated(False)
            line.set_animated(False)
            line.set_color([0, 0, 0, 0])
        self.canvas.draw()
        # Find out if a click happened on a vertice
        # and which vertice of which polygon

```

```

        self._ipoly, self._invert =
self._get_polygon_vertice_id(event)
        if self._ipoly is not None:

self.density_slider.set_val(self.densities[self._ipoly])
        self.polygons[self._ipoly].set_animated(True)
        self.lines[self._ipoly].set_animated(True)
        self.lines[self._ipoly].set_color([0, 1, 0, 0])
        self.canvas.draw()
        self.background =
self.canvas.copy_from_bbox(self.modelax.bbox)
        self.modelax.draw_artist(self.polygons[self._ipoly])
        self.modelax.draw_artist(self.lines[self._ipoly])
        self.canvas.blit(self.modelax.bbox)
    elif self._drawing:
        if event.button == 1:
            self._xy.append([event.xdata, event.ydata])
            self._drawing_plot.set_data(list(zip(*self._xy)))
            self.canvas.restore_region(self.background)
            self.modelax.draw_artist(self._drawing_plot)
            self.canvas.blit(self.modelax.bbox)
        elif event.button == 3:
            if len(self._xy) >= 3:
                density = self.density_slider.val
                poly, line = self._make_polygon(self._xy, density)
                self.polygons.append(poly)
                self.lines.append(line)
                self.densities.append(density)
                self.modelax.add_patch(poly)
                self.modelax.add_line(line)
                self._drawing_plot.remove()
                self._drawing_plot = None
                self._xy = None
                self._drawing = False
                self._ipoly = len(self.polygons) - 1
                self.lines[self._ipoly].set_color([0, 1, 0, 0])
                self.dataax.set_title(self.instructions)
                self.canvas.draw()
                self._update_data()
                self._update_data_plot()

def _button_release_callback(self, event):
    """
    Reset place markers on mouse button release
    """
    if event.inaxes != self.modelax:
        return
    if event.button != 1:
        return
    if self._invert is None and self._ipoly is None:
        return
    self.background = None
    for line, poly in zip(self.lines, self.polygons):
        poly.set_animated(False)
        line.set_animated(False)
    self.canvas.draw()
    self._invert = None
    # self._ipoly is only released when clicking outside

```



```

# the polygons
self._lastevent = None
self._update_data()
self._update_data_plot()

def _key_press_callback(self, event):
    """
    What to do when a key is pressed on the keyboard.
    """
    if event.inaxes is None:
        return
    if event.key == 'd':
        if self._drawing and self._xy:
            self._xy.pop()
            if self._xy:
                self._drawing_plot.set_data(list(zip(*self._xy)))
            else:
                self._drawing_plot.set_data([], [])
                self.canvas.restore_region(self.background)
                self.modelax.draw_artist(self._drawing_plot)
                self.canvas.blit(self.modelax.bbox)
        elif self._invert is not None:
            poly = self.polygons[self._ipoly]
            line = self.lines[self._ipoly]
            if len(poly.xy) > 4:
                verts = numpy.atleast_1d(self._invert)
                poly.xy = numpy.array([xy for i, xy in
enumerate(poly.xy)
                                     if i not in verts])
                line.set_data(list(zip(*poly.xy)))
                self._update_data()
                self._update_data_plot()
                self.canvas.restore_region(self.background)
                self.modelax.draw_artist(poly)
                self.modelax.draw_artist(line)
                self.canvas.blit(self.modelax.bbox)
                self._invert = None
            elif self._ipoly is not None:
                self.polygons[self._ipoly].remove()
                self.lines[self._ipoly].remove()
                self.polygons.pop(self._ipoly)
                self.lines.pop(self._ipoly)
                self.densities.pop(self._ipoly)
                self._ipoly = None
                self.canvas.draw()
                self._update_data()
                self._update_data_plot()
        elif event.key == 'n':
            self._invert = None
            self._ipoly = None
            for line, poly in zip(self.lines, self.polygons):
                poly.set_animated(False)
                line.set_animated(False)
                line.set_color([0, 0, 0, 0])
            self.canvas.draw()
            self.background =
self.canvas.copy_from_bbox(self.modelax.bbox)
            self._drawing = True

```

```

        self._xy = []
        self._drawing_plot = Line2D([], [], **self.line_args)
        self._drawing_plot.set_animated(True)
        self.modelax.add_line(self._drawing_plot)
        self.dataax.set_title(' | '.join([
            'left click: set vertice', 'right click: finish',
            'esc: cancel']))
        self.canvas.draw()
    elif event.key == 'escape':
        self._drawing = False
        self._xy = []
        if self._drawing_plot is not None:
            self._drawing_plot.remove()
            self._drawing_plot = None
        for line, poly in zip(self.lines, self.polygons):
            poly.set_animated(False)
            line.set_animated(False)
            line.set_color([0, 0, 0, 0])
        self.canvas.draw()

def _mouse_move_callback(self, event):
    """
    Handle things when the mouse move.
    """
    if event.inaxes != self.modelax:
        return
    if event.button != 1:
        return
    if self._ivert is None and self._ipoly is None:
        return
    x, y = event.xdata, event.ydata
    p = self._ipoly
    v = self._ivert
    if self._ivert is not None:
        self.polygons[p].xy[v] = x, y
    else:
        dx = x - self._lastevent.xdata
        dy = y - self._lastevent.ydata
        self.polygons[p].xy[:, 0] += dx
        self.polygons[p].xy[:, 1] += dy
    self.lines[p].set_data(list(zip(*self.polygons[p].xy)))
    self._lastevent = event
    self.canvas.restore_region(self.background)
    self.modelax.draw_artist(self.polygons[p])
    self.modelax.draw_artist(self.lines[p])
    self.canvas.blit(self.modelax.bbox)

```

Código para importar o dado de gravimetria observado na região de estudo e para iniciar a modelagem:

```

import numpy as np
file=np.loadtxt('PerfilBougher.csv')
x=file[:,0]
y=file[:,1]

```

```

n=np.max(x)
area=(0.0,n,0.0,40000.0)
z=np.zeros_like(x)
app=Moulder(area,x,z,y)
app.run()

```

Código para calcular o erro importando o dado de gravimetria observado ('PerfilBougher.csv') e o calculado ('PerfilCerto2.csv') na região de estudo:

```

''' Calcular Erro '''

import numpy as np
import scipy.interpolate
import matplotlib.pyplot as plt

GO = np.loadtxt('PerfilBougher.csv')
GC = np.loadtxt('PerfilCerto2.csv')

Y = scipy.interpolate.interpld(GO[:,0],GO[:,1],kind='quadratic')

Erro = np.zeros_like(GO[:,0])

for i in range(0,len(Erro)):
    Erro[i] = GC[i,1]-Y(GC[i,0])

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(GC[:,0],Erro[:])
ax.set_title('Erro')
ax.set_ylabel('mGal')
ax.set_xlabel('m')
ax.set_ylim(-10,10)
ax.set_xlim(0,500000)
ax.grid()
plt.show()

```

Rotina base, feita no matlab, para o cálculo das espessuras das camadas:

```

%cálculo da espessura da camada de base #horizon e topo #Fundodomar de
%velocidade 3.8 km/s
a=length(data);
z=zeros(a,1);
for ii=1:a;
    for jj=1:a;
        if((data(ii,1)==data(jj,4)) && (data(ii,2)==data(jj,5)))
            z(ii) = ((data(jj,6)-data(ii,3))/2)*3.8;
        end
    end
end
end

```

```
%cálculo da espessura do rifte, velocidade=5.5 Km/s
%camada base #Topodopre-rifte, topo #Topodorifte
a=length(data);
z=zeros(a,1);
for ii=1:a;
    for jj=1:a;
        if((data(ii,19)==data(jj,22))&&(data(ii,20)==data(jj,23)))
            z(ii)=((data(jj,24)-data(ii,21))/2)*5.5;
        end
    end
end
end
```