

RECURSOS COMPUTACIONAIS PARA REALCE DE ANOMALIAS MAGNETOMÉTRICAS E GRAVIMÉTRICAS EM SALA DE AULA

LUIZEMARA SOARES ALVES SZAMEITAT ¹ & ALESSANDRA DE BARROS E SILVA BONGIOLO ²

1 Universidade do Estado do Rio de Janeiro - UERJ, Instituto GeoAtlântico. R. São Francisco Xavier, 524 - Sala 4006-A - Maracanã, Rio de Janeiro - RJ, 20550-013. E-mail: luizemara@gmail.com

2 Universidade Federal do Paraná, Pós-Graduação em Geologia. Av. Cel. Francisco H. dos Santos, s/n - Jardim das Américas, Curitiba - PR, 81270-460. E-mail: alebongiolo@ufpr.br

Resumo - A equipe do Laboratório de Pesquisas em Geofísica Aplicada (LPGA-UFPR) vem se empenhando no desenvolvimento de trabalhos de pesquisa no ensino e na divulgação da geofísica, promovendo a conscientização e a compreensão dos métodos geofísicos dentro da universidade e para o público externo. Seguindo esta linha de trabalho, as autoras prepararam uma coleção de códigos na linguagem Python para o tratamento de dados gravimétricos e magnetométricos utilizando técnicas básicas: derivadas horizontais e vertical, gradiente horizontal, redução ao polo, amplitude do sinal analítico, e estimativa de anomalia Bouguer. Visando a praticidade no uso em sala de aula, os algoritmos foram preparados no formato Jupyter Notebook, e pensados para serem usados no ambiente de programação do Google Colaboratory. As implementações das técnicas selecionadas foram baseadas em funções clássicas da literatura, e também com o uso de bibliotecas já existentes na mesma linguagem. Os algoritmos das funções estão incluídos neste material, e os programas completos em formato notebook Python estão disponibilizados e sendo atualizados em repositório público GitHub. Como resultado, este trabalho compõe uma ferramenta acessível para estudantes das geociências na análise de dados de campos potenciais.

Palavras-Chave: Ensino; Geofísica Aplicada; Magnetometria; Gravimetria; Python.

Abstract - The Laboratory of Applied Geophysics Research (LPGA-UFPR) has been engaged in the development of research in teaching and science communication of geophysics, promoting the awareness and understanding of geophysical methods in the university and external community. In this sense, we prepared a collection of Python scripts for the enhancement of gravimetric and magnetometric anomalies with basic techniques: horizontal and vertical derivatives, horizontal gradient, reduction to the pole, analytic signal amplitude, and Bouguer Anomaly. Aiming the development of a hands-on training in the classroom, the algorithms were prepared as Jupyter Notebook files, designed to be used in the Google Colaboratory programming environment. The implementation of the selected techniques were based on classic functions in literature, and using existing libraries developed in the same language. The functions are included herein, and complete programs in notebook Python format are available in the public GitHub repository, where the user can check for updates. As a result, this material comprises an accessible tool for geoscience students in potential field data analysis.

Keywords: Teaching; Applied Geophysics; Magnetometry; Gravimetry; Python.

1. INTRODUÇÃO

As melhores práticas da ciência moderna incluem o compartilhamento de dados e de algoritmos de uso livre. Em busca de contribuir com esta linha de atuação, este trabalho traz materiais que poderão ser usados por toda a sociedade, mas que foram pensados para os alunos de graduação e pós-graduação que necessitem aplicar tratamentos de realce em dados geofísicos para suas pesquisas e trabalhos acadêmicos. O LPGA apoia iniciativas que promovem o ensino das Ciências da Terra de forma acessível, e trabalhos que ampliam o acesso da comunidade à instrumentação e dados de geofísica (e.g., Bongiolo et al., 2022).

Neste trabalho, os algoritmos foram formulados com o objetivo de transformar os dados iniciais de gravimetria e magnetometria, para realçar determinadas feições geológicas do terreno. Como material didático, a compreensão das funções exibidas requer pensamento crítico dos alunos e conhecimento básico em conceitos de matemática básica e cálculo. No entanto, os programas no formato notebook Python permitem que o processamento seja feito mesmo que o aluno esteja apenas iniciando nos estudos de geofísica e no uso da programação para suas atividades acadêmicas.

Considerando o futuro profissional dos alunos, a ação de delinear as bordas de corpos geológicos é uma técnica fundamental para o mapeamento geológico. Em particular quando os corpos têm exposição limitada, ou mesmo se estes são não-aflorantes, é comum o uso de dados geofísicos potenciais (gravimétricos e magnetométricos) para definir sua extensão espacial. Neste sentido, as mudanças laterais nas propriedades físicas de magnetização e densidade das rochas fornecem as informações espaciais sobre o corpo anômalo (e.g. Blakely, 1996; Kearey et al., 2009). Esta delimitação de corpos geológicos no espaço tem grande importância para a correlação de dados diretos (dados de afloramentos e/ou de poços). Além disso, os dados potenciais interpretados podem fornecer

informações sobre estilos tectônicos e tendências de deformação dos terrenos geológicos. Ainda, este tipo de dado geofísico e o realce de variações do campo podem ser aplicados em menor escala, para fins de engenharia, arqueológicos ou forenses, por exemplo (e.g., Bongiolo et al., 2022).

2. REVISÃO TEÓRICA

2.1. O realce de características do campo potencial local

O tratamento de dados para realce de anomalias (“filtros de realce”, ou “filtragem de dados”) é utilizado no intuito de facilitar a interpretação dos lineamentos e contatos de corpos anômalos no dado magnetométrico ou gravimétrico, além de melhorar a precisão destas interpretações. Neste trabalho, reunimos algoritmos adaptados da literatura de alguns dos filtros mais utilizados para este tipo de atividade de interpretação: derivadas horizontais, derivada vertical, gradiente horizontal, redução ao polo e amplitude do sinal analítico. No caso do dado gravimétrico, incluímos também a transformação da anomalia Ar-livre em anomalia Bouguer.

2.1.1. Derivadas horizontais

As derivadas horizontais se referem às derivadas parciais das componentes do campo potencial nas direções x e y . Por isso, são usadas no tratamento de dados gravimétricos e magnetométricos para fornecer a taxa de variação do campo anômalo em relação à estas coordenadas. Com isso, auxiliam na localização de variações na distribuição de massa (campo gravitacional) ou magnetização (campo magnético), associadas, por exemplo, a estruturas geológicas ou objetos em subsuperfície (Bongiolo et al., 2022). As derivadas horizontais podem ser utilizadas individualmente, a depender do contexto e do objetivo do estudo. Porém, são frequentemente aplicadas como parte do cálculo de outros filtros

geofísicos, como derivada vertical e amplitude do sinal analítico.

2.1.2. Gradiente horizontal total

A medida de gradiente das anomalias no plano horizontal pode auxiliar no mapeamento de bordas de corpos e estruturas geológicas. Apesar do método analítico ser comumente aplicado neste caso (Cordell e Grauch, 1985), a magnitude deste gradiente pode também ser obtida por relações de diferenças finitas (Blakely e Simpson, 1986; Blakely, 1996). Para o uso com fins didáticos no curso de Geologia ou para alunos com menor experiência em cálculo numérico, a estimativa de valores no domínio do espaço facilita a compreensão dos alunos. O cálculo do gradiente trazido por Blakely (1996, pag. 348) segue a fórmula da Equação 1.

$$GH(x, y) = \left[\left(\frac{\partial g_z(x, y)}{\partial x} \right)^2 + \left(\frac{\partial g_z(x, y)}{\partial y} \right)^2 \right]^{1/2} \quad (\text{Equação 1})$$

2.1.3. Derivada Vertical

A derivada vertical do campo potencial ressalta alterações de alta frequência na direção z (Evjen, 1936), e é usada principalmente para destacar fontes rasas (Blakely, 1996). Enquanto o gradiente horizontal pode ser calculado com diferenças finitas, a derivada vertical é geralmente obtida em número de onda, e envolve o uso de transformada de Fourier. Isto porque, no contexto deste estudo, os campos potenciais gravitacional e magnético são tratados como funções harmônicas. Este tipo de função satisfaz a equação de Laplace, que é uma equação diferencial parcial de segunda ordem (Equação 2; Blakely, 1996, pag. 117). Pela equação de Laplace, assumimos que a soma das derivadas parciais de segunda ordem de um campo potencial é zero (Equação 2), e que o valor médio do potencial na região entorno de um ponto P é o próprio valor do potencial no ponto P. Já mais distante da fonte, o campo se torna menos intenso. Isso pode ser percebido no início da Equação 2, pois o valor do campo

tem uma relação inversamente proporcional com o raio.

$$\nabla^2 \frac{1}{r} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0 \quad (\text{Equação 2})$$

A partir do conceito anterior, observamos que a variação no eixo z nos campos potenciais é dependente das variações em x e y, e, portanto, a variação do campo em z pode ser estimada a partir do comportamento do campo em x e y (Equação 3; Blakely, 1996, pag. 325). Em síntese, temos que a derivada vertical do campo potencial é consequência do conceito de Laplace. Na Equação 3 é a derivada de segunda ordem, mas a derivação de qualquer ordem é considerada uma transformação válida para ajudar na descrição do campo potencial estudado (Blakely, 1996).

$$-\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial z^2} \quad (\text{Equação 3})$$

Ao estudar os valores do campo no espaço, poderíamos definir a derivada em z com base na variação entre pequenos espaços, considerando o valor de Δz como tendendo a zero (Equação 4; Blakely, 1996, pag. 326). Considerando que se trata de um campo harmônico, a derivada também pode ser efetuada no domínio de Fourier, e sua forma ficaria com a expressão da Equação 5 (Blakely, 1996, pag. 326). Na transformação para o domínio de Fourier, a função recebe a frequência do sinal, ao invés das distâncias em x, y e z. O valor k representa a frequência, que é o inverso do comprimento da onda ($1/\lambda$), e também é chamado de número de onda.

$$\frac{\partial}{\partial z} f(x, y, z) = \lim_{\Delta z \rightarrow 0} \left(\frac{f(x, y, z) - f(x, y, z - \Delta z)}{\Delta z} \right) \quad (\text{Equação 4})$$

$$\mathcal{F} \left[\frac{\partial f}{\partial z} \right] = \lim_{\Delta z \rightarrow 0} \left(\frac{\mathcal{F}[f] - \mathcal{F}[e^{-|k|\Delta z}]}{\Delta z} \right) \quad (\text{Equação 5})$$

Na fórmula, trabalhamos com o valor de $|k|$. No domínio da frequência, o dado possui uma parte

imaginária, onde $|k|$ é o comprimento do vetor k (Equação 6; Li, 2008).

$$|k| = |r| = \sqrt{k_x^2 + k_y^2} \quad (\text{Equação 6})$$

sendo k_x e k_y as projeções do vetor k no plano horizontal (k_x, k_y ; Figura 1). Este detalhamento é importante para a compreensão das transformações descritas a seguir.

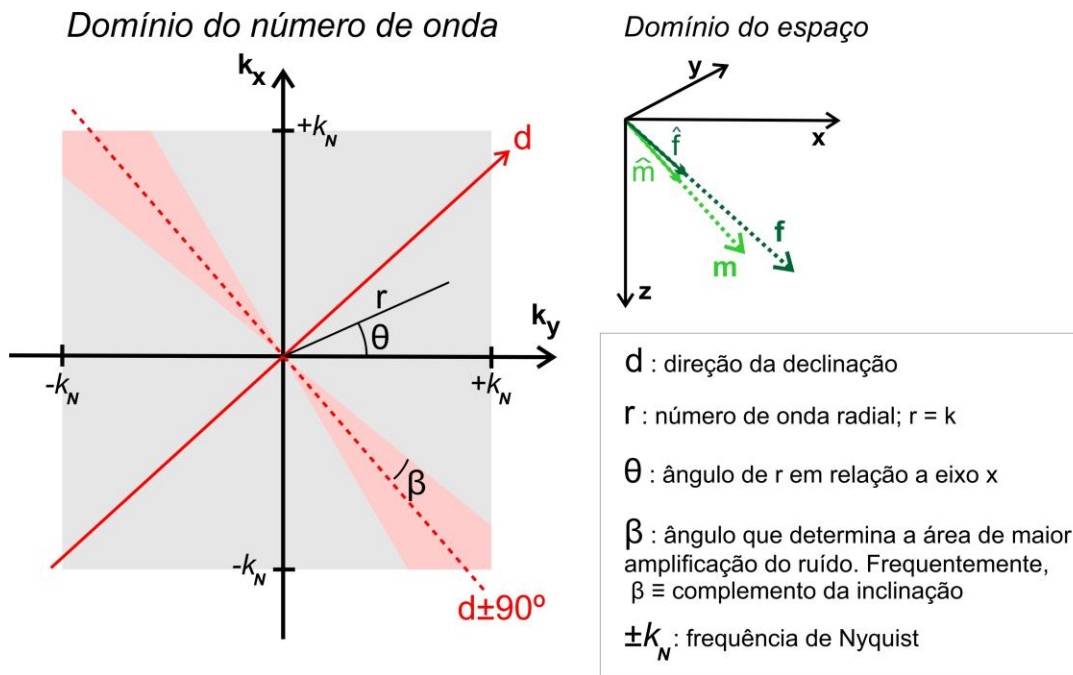


Figura 1 – Esquema do domínio do número de onda (dado transformado pela FFT), mostrando a área em que o filtro RTP se torna mais instável (vermelho claro), e ilustração do vetor de magnetização no domínio do espaço, com a magnetização do corpo m e seu versor \hat{m} , e direção do campo geomagnético f e seu versor \hat{f} . A direção da declinação no domínio de onda é rotacionada a 90 graus em relação ao domínio do espaço. A área de maior instabilidade do filtro RTP é dada pelo ângulo beta (β), que pode ser aproximado para o complemento do ângulo de inclinação. Adaptado de Li (2008).

2.1.4. Redução ao polo (RTP)

Na aplicação do filtro de redução ao polo (RTP), o sinal magnético é transformado para o domínio da frequência usando a transformada de Fourier (Gunn, 1975, Blakely, 1996). O campo potencial magnético anômalo ΔT é representado como $F[\Delta T]$ no domínio de Fourier. Uma vez no domínio da frequência, a resposta fornecida pela fase do dado é ajustada para remover a contribuição da inclinação magnética diferente de 90° . Em seguida, o sinal é transformado novamente para o domínio do espaço usando a transformada inversa.

Consideramos que $\hat{m} = (\hat{m}_x, \hat{m}_y, \hat{m}_z)$ é um vetor unitário na direção da magnetização do corpo, que $\hat{f} = (\hat{f}_x, \hat{f}_y, \hat{f}_z)$ é a direção do campo geomagnético, e que os parâmetros θ_m e θ_f , contribuem para a fase da anomalia e descrevem a direção de magnetização total (Equação 7).

$$\theta_m = \hat{m}_z + i \frac{\hat{m}_x \hat{k}_x + \hat{m}_y \hat{k}_y}{|k|} \quad \text{e}$$

$$\theta_f = \hat{f}_z + i \frac{\hat{f}_x \hat{k}_x + \hat{f}_y \hat{k}_y}{|k|} \quad (\text{Equação 7})$$

Podemos aplicar um filtro ao campo magnético anômalo para transformar o sinal magnético produzido em uma localidade inicial, para que este sinal reproduza a resposta magnética gerada por esta mesma fonte em uma outra localidade no globo. Este filtro é dado pela Equação 8, com o operador $F[\Psi_r]$ dado pela Equação 9. Enquanto os parâmetros θ_m e θ_f refletem a magnetização da fonte e seu campo magnético regional da localidade inicial, θ'_m e θ'_f são a magnetização e o campo regional de saída do filtro de transformação, referentes ao segundo local considerado.

$$F[\Delta T_r] = F[\Psi_r]F[\Delta T] \quad (\text{Equação 8})$$

$$F[\Psi_r] = \frac{\theta'_m \theta'_f}{\theta_m \theta_f} = \frac{|k|^2}{a_1 k_x^2 + a_2 k_y^2 + a_3 k_x k_y + i|k|(b_1 k_x + b_2 k_y)}$$

(Equação 9)

sendo $a_1 = \hat{m}_z \hat{f}_z - \hat{m}_x \hat{f}_x$, $a_2 = \hat{m}_z \hat{f}_z - \hat{m}_y \hat{f}_y$,
 $a_3 = -\hat{m}_z \hat{f}_z - \hat{m}_x \hat{f}_x$, $b_1 = \hat{m}_x \hat{f}_z + \hat{m}_z \hat{f}_x$,
 $b_2 = \hat{m}_y \hat{f}_z + \hat{m}_z \hat{f}_y$ (Blakely, 1996).

Como a transformação que estamos realizando é para o polo, considera-se que, no domínio cartesiano, $\hat{m} = \hat{f} = (0, 0, 1)$. Esta condição nos diz também que assumimos que a magnetização do corpo e o campo geomagnético possuem a mesma direção. A condição apresentada simplifica a equação do operador $F[\Psi_r]$ para a forma da Equação 10, sendo $\theta = \tan^{-1}(k_y/k_x)$.

$$F[\Psi_r] = \frac{1}{\theta_m \theta_f} = \frac{1}{a_1 \cos^2 \theta + a_2 \sin^2 \theta + a_3 \cos \theta \sin \theta + i(b_1 \cos \theta + b_2 \sin \theta)}$$

(Equação 10)

O filtro RTP é considerado estável para latitudes altas, mas em latitudes baixas ($< 20^\circ$ ou $> -20^\circ$; Li, 2008) pode fornecer resultados distorcidos, pois há amplificação de ruídos e geração de anomalias alongadas na direção da declinação (Li, 2008). Por isso, para baixas latitudes, são feitas adaptações na fórmula do RTP (e.g., Hansen e Pawlowsky, 1989; MacLeod et al., 1993; Li, 2008). Como exemplo, usaremos a adaptação de MacLeod et al. (1993), que aplica uma pseudo-inclinação do campo (I'), que deve ser maior do que a inclinação original do campo (I). Este valor pode ser o complemento da inclinação, e também serve como aproximação para o valor do ângulo beta (β) da Figura 1 (Li, 2008). Assim, o filtro RTP em coordenadas polares (Equação 11) é alterado, e o operador adaptado para baixas latitudes é expresso na Equação 12.

$$R(\theta) = \frac{1}{\sin(I) + i \cos(I) \cos(d-\theta)} \quad (\text{Equação 11})$$

$$R(\theta) = \frac{[\sin(I) - i \cos(I) \cos(d-\theta)]^2}{[\sin^2(I') + i \cos^2(I') \cos^2(d-\theta)] * [\sin^2(I) + i \cos^2(I) \cos^2(d-\theta)]}$$

(Equação 12)

2.1.5. Amplitude do Sinal Analítico (ASA)

O filtro de Amplitude do Sinal Analítico (ASA) transforma ΔT de forma a produzir anomalias positivas (em nT/m) sobre a fonte magnética (Nabighian, 1972; Roest et al., 1992). A transformação ASA é um módulo, e é obtida como uma relação entre as derivadas parciais nas três dimensões do espaço (Equação 11).

$$|A(x, y, z)| = \sqrt{\left(\frac{\partial \Delta T}{\partial x}\right)^2 + \left(\frac{\partial \Delta T}{\partial y}\right)^2 + \left(\frac{\partial \Delta T}{\partial z}\right)^2}$$

(Equação 11)

Este filtro é frequentemente utilizado como uma alternativa ao uso da transformação RTP em interpretações geológicas, uma vez que o RTP apresenta resultados satisfatórios se a magnetização remanente for conhecida ou negligenciável, o que pode não ser a realidade da área estudada. O filtro ASA muitas vezes apresenta melhor resultado que o filtro RTP quando a fonte magnética tem magnetização remanente importante. Isto não significa que o resultado é livre da influência da magnetização remanente (Li, 2006), mas apenas que é mais eficiente em representar a fonte em mapas. Li (2006) aponta razões pelas quais este filtro pode não ter um resultado satisfatório, mas não entraremos nesta discussão pois não é o escopo deste trabalho. Ainda que haja ressalvas na interpretação do resultado, o ASA é amplamente conhecido e usado pela comunidade geocientífica e costuma apresentar bons resultados. Ressaltamos que, na execução desta ou de qualquer outra transformação de dados potenciais, é preciso analisar o resultado obtido junto ao contexto geológico e a dados complementares, para avaliar a eficiência da representação das anomalias.

2.1.6. Anomalia Gravimétrica Bouguer

A partir do trabalho pioneiro de Pierre Bouguer no século 18, sabemos que há uma distribuição heterogênea de massas no interior da Terra, e que a anomalia gravimétrica é o resultado da interação dessas massas com o campo gravimétrico terrestre. Os dados gravimétricos são adquiridos por medições terrestres, aéreas ou orbitais, assim como o dado magnético. Estas medidas são processadas e corrigidas para eliminar efeitos indesejados, como a influência de marés, variação da altitude, e movimentos relativos da Terra. Após as correções do dado bruto, o valor de gravidade observado (g_{obs}) é reduzido da anomalia calculada a partir do elipsoide de referência escolhido (gravidade

normal, g_N), o que fornece a anomalia ar-livre (Δg_{AL}), conhecida também como anomalia *free-air*. Este é o tipo de anomalia gravimétrica mais comum disponibilizada em repositórios para uso nas interpretações geológicas (e.g., Sandwell et al., 2014). A correção ar-livre é expressa na Equação 12, onde 0,3806 é o valor padrão assumido em estudos gravimétricos para o gradiente vertical da gravidade, e h é a altitude em relação a um *datum*. Como *datum*, pode ser considerada a altitude em relação ao nível do mar (e.g., Blakey, 1996, p.140), ou a altitude ortométrica (altitude em relação ao geoide), ou ainda a menor altitude da área de estudo. A anomalia de ar-livre (Equação 13) pode ser descrita como o resultado da retirada do valor da correção ar-livre no dado observado, e do valor do efeito gravimétrico esperado do elipsoide (gravidade normal, g_N).

$$g_{AL} = 0,3806 h \quad (\text{Equação 12})$$

$$\Delta g_{AL} = g_{obs} - g_{AL} - g_N \quad (\text{Equação 13})$$

No entanto, a anomalia ar-livre não contempla efeitos gravimétricos gerados pelas massas na topografia acima do ponto de referência (*datum*). Como resultado, a anomalia ar-livre é mais sensível a variações geomorfológicas. Em regiões onde não há grandes variações da topografia, a anomalia ar-livre pode ser usada para interpretação de massas anômalas no subsolo sem maiores problemas. Mas, em geral, as feições topográficas ou batimétricas acabam prejudicando a interpretação geológica de corpos em subsuperfície. Por isso, é comum o cálculo da anomalia gravimétrica Bouguer, como forma de mitigar os efeitos do relevo.

A correção Bouguer simples (g_{SB}) aproxima o terreno acima do *datum* de referência para uma camada infinita. A atração gravitacional desta

camada é dada pela Equação 12, onde G é a constante gravitacional ($6,67430 \times 10^{-11} \text{ m}^3/(\text{kg} \cdot \text{s}^2)$), ρ é a densidade média dos materiais da camada, e h é a altura da camada. A anomalia Bouguer é então definida pela Equação 13.

$$g_{SB} = 2\pi G\rho h \quad (\text{Equação 12})$$

$$\Delta g_{SB} = g_{obs} - g_{AL} - g_{SB} - g_N \quad (\text{Equação 13})$$

É importante salientar que a densidade (ρ) estimada deve considerar o contraste de densidade médio do terreno, o qual dependerá dos tipos de materiais que compõem o terreno (natureza sedimentar ou ígnea e metamórfica), e o meio que está em contato com o terreno (ar ou água). Este contraste pode ser estimado por amostras de rocha, se a área for relativamente pequena, mas pode ser estimado também por valores de densidade indicados na literatura.

3. MÉTODOS

3.1. A escolha pela Linguagem Python

A análise científica é um campo complexo que envolve a manipulação e o processamento de dados para obter resultados relevantes. A programação em Python tem se tornado cada vez mais popular para realizar análises de dados, não apenas para fins científicos, mas também econômicos. A linguagem Python é uma linguagem de alto nível, conhecida por possibilitar um aprendizado relativamente rápido, mesmo para pessoas com nenhuma ou pouca experiência. Sua simplicidade e legibilidade fazem com que seja frequentemente aplicada para o ensino de programação. Além disso, há um grande número de recursos gratuitos disponíveis em Python, além de uma grande comunidade de usuários que colaboram entre si, o que cria mais oportunidades de troca de informações e de

trechos de algoritmos. Para programadores mais experientes, o Python é visto como uma linguagem versátil e de alta aplicabilidade em diversos campos, desde projetos puramente científicos até desenvolvimento web. Por isso, entendemos que o ensino por esta linguagem fornece uma base sólida, que contribuirá para o desenvolvimento dos alunos em carreiras futuras.

3.2. Funções

Nos códigos implementados, exploramos o uso de algumas bibliotecas populares da linguagem Python, comumente usados para análises científicas, incluindo NumPy, SciPy, Math, Matplotlib e Pandas. Essas bibliotecas fornecem ferramentas para processamento, manipulação e visualização de dados. Além disso, foram utilizadas funções do projeto Fatiando a Terra (Uieda et al., 2013), no algoritmo de geração da anomalia Bouguer a partir do dado de Ar-livre.

3.2. Funções Python importadas

Para o processamento dos dados, são usadas as bibliotecas Math, NumPy e SciPy. A Math traz funções básicas como raiz quadrada, seno e cosseno, por exemplo. O NumPy também fornece ferramentas básicas para a análise numérica em Python, mas consideramos que é interessante apresentar aos estudantes a Math, pois é uma biblioteca mais leve, e que pode ser uma boa opção em alguns algoritmos. O NumPy tem aplicação em dados multidimensionais, ferramentas para uso de objetos do tipo *array* e funções matemáticas de alto desempenho. Exemplos de função de alto desempenho usadas neste trabalho, importadas da NumPy, são `reshape()`, para modificação de dimensões de matrizes, e `ones()`, que cria uma matriz com dimensões determinadas com valores "1".

O NumPy e o SciPy são usados para realizar cálculos envolvendo vetores e matrizes, e a aplicação da transformada rápida de Fourier e sua transformação inversa (Jones et al., 2001, Virtanen et al., 2020), os quais são essenciais

para o processamento de dados geofísicos. Nos códigos aqui utilizados, foram aplicadas as funções `fft2()` e `ifft2()` da biblioteca SciPy, para realizar a transformação para o domínio de Fourier e a transformada inversa, respectivamente.

3.3. Algoritmos adaptados

Grande parte das funções utilizadas para a transformação dos dados potenciais foram feitas pela tradução de algoritmos disponibilizados na linguagem Fortran em Blakely (1996), fórmulas do mesmo autor, e do

método de MacLeod et al. (1993) da pseudo-inclinação, discutido também em Li (2008).

4. RESULTADOS

4.1. Funções Adaptadas

A Tabela 1 traz as funções adaptadas de Blakely (1996), com a referência da função original em Fortran, e o objetivo de cada função. Os algoritmos correspondentes estão apresentados no Apêndice.

Tabela 1 – Funções adaptadas de algoritmos de Blakely (1996).

Função original (Blakely, 1996)	Função adaptada	Objetivo
Kvalue()	Kvalue()	Parte das funções de transformação de dados dependem da função Kvalue(). Esta função é parte da estimativa da constante k, que está relacionada ao número de onda (Equação 6).
VerDer()	Derivada()	Função modificada que executa as derivadas de primeira ordem em x, y e z.
DirCosr()	DirCos()	Função para cosseno direcional, fornece a projeção do vetor de magnetização nos eixos cartesianos (x, y, z).
HGrad()	GradHoriz()	Função que fornece o gradiente horizontal total (Equação 1).
NewVec()	NewVec() e RTPL()	Função que segue a lógica do algoritmo original NewVec(). RTPL() foi adaptada de NewVec() para o caso de latitudes baixas, com a técnica de MacLeod et al. (1993) (Equação 12)

4.2. Algoritmos Comentados – Formato Notebook

Os arquivos gerados para uso como material didático no ensino de métodos geofísicos potenciais estão disponibilizados em repositório público (<https://github.com/lizam/ensino>). Através do uso do GitHub, pretende-se manter os códigos atualizados conforme a necessidade. Como tempo, este repositório deverá receber mais algoritmos com finalidade didática.

Os algoritmos estão apresentados como arquivos do tipo Jupyter Notebook (formato IPYNB), acompanhados de arquivos de exemplo para entrada de dados (formato TXT e CSV), e exemplos de arquivos de saída (formato PNG e CSV). Uma vez que o usuário salve o arquivo IPYNB do código escolhido, este poderá executar e personalizar o arquivo em seu ambiente de trabalho particular. No entanto, os arquivos IPYNB disponibilizados estão adaptados para execução no ambiente Google Colaboratory

(<https://colab.research.google.com/>), da empresa Google. O Google Colab é reconhecido como uma boa plataforma para desenvolver e executar códigos em Python sem a necessidade de instalar aplicativos, além de fornecer recursos avançados de processamento em nuvem caso seja necessário, o que dispensa a necessidade de uma máquina local com grande poder computacional. Ao executar a aplicação neste ambiente, os arquivos de saída do usuário (imagens e dados no formato CSV) permanecem no ambiente temporário, devendo ser acessados e salvos pelo usuário em seu computador ou ambiente pessoal em nuvem. Sobre os resultados obtidos, é possível observar que, em alguns casos (por exemplo, no mapa da derivada vertical), o algoritmo cria “artefatos”, principalmente nas bordas (efeito de borda). Este tipo de efeito é esperado, pois a análise de dados no domínio da frequência trata o dado de campo anômalo como tendo periodicidade no espaço (Blakely, 1996). Neste caso, Blakely (1996) indica o uso de funções de expansão de matrizes no dado de entrada, antes da aplicação dos filtros, para mitigar este tipo de efeito. No

contexto deste trabalho, foi feita a opção por não incluir tal alteração, uma vez que o objetivo é apresentar o funcionamento básico dos filtros. No entanto, é provável que esta e outras ferramentas computacionais mais específicas ou mais complexas, como funções para modelagem de regiões anômalas com geometria regular e irregular, sejam adicionadas ao repositório no futuro.

4.3. Filtros de realce para dados potenciais

Os filtros de realce trazidos, e que podem ser utilizados tanto para dados magnetométricos como para dados gravimétricos, são as derivadas em x, em y e em z (Figura 2), e o gradiente horizontal total (Figura 3). Estes filtros foram demonstrados pelo uso de uma anomalia fictícia disponibilizada junto com os algoritmos no repositório (arquivo de entrada ‘davi.csv’), que contém valores de 0 a 1. O uso da forma geométrica, neste caso, tem o objetivo de exibir o resultado do filtro de forma desconectada do contexto geológico, a fim de demonstrar o efeito primário deste tipo de filtro de realce.

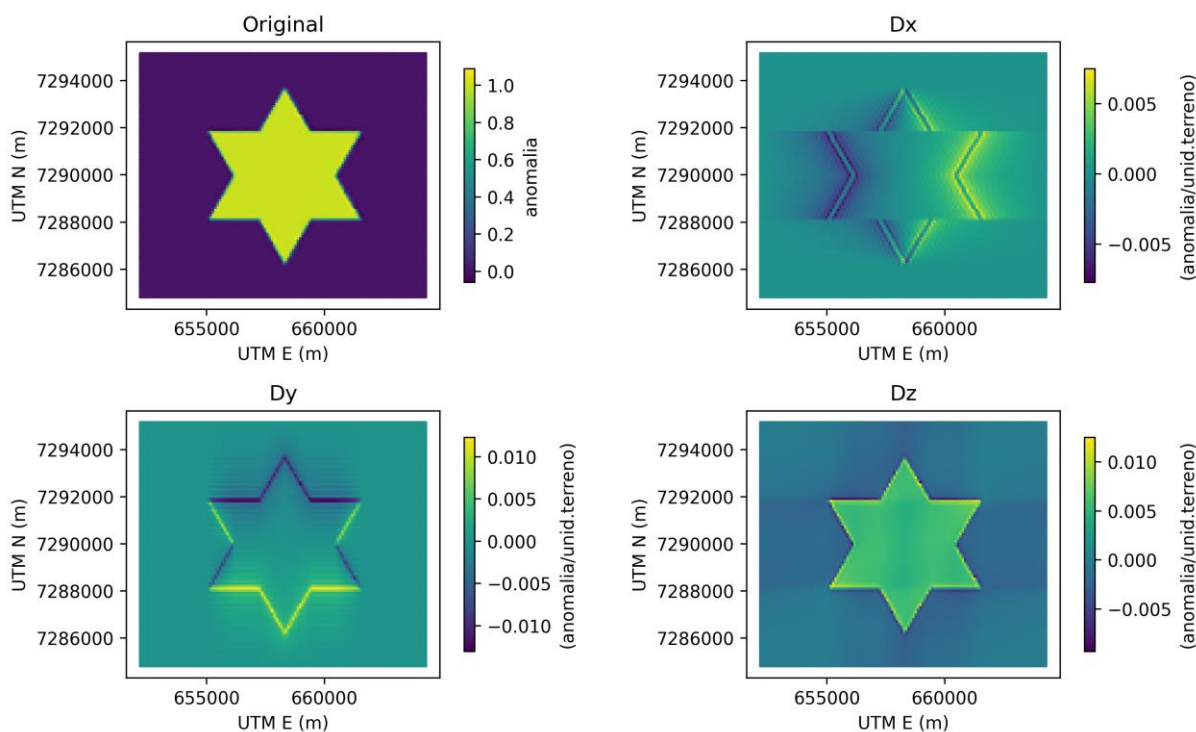


Figura 2 - Mapa de saída dos algoritmos de derivadas nas direções x (mapa Dx), y (mapa Dy) e z (mapa Dz), executados pela função Derivadas().

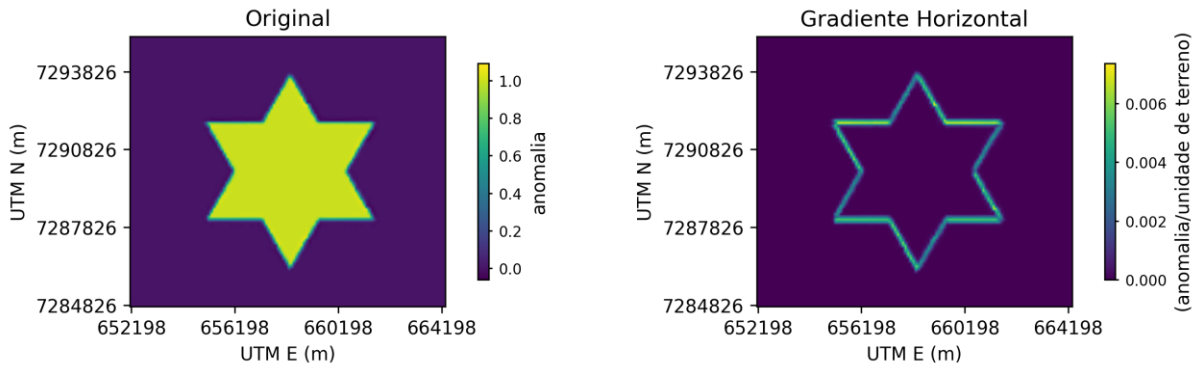


Figura 3 – Mapa de saída do algoritmo do filtro de gradiente horizontal, executado pela função GradHoriz().

4.4. - Transformações para dados magnetométricos - Redução ao polo e Amplitude do Sinal Analítico

O algoritmo do RTP foi adaptado de Blakely (1996), mantendo a lógica original. Os parâmetros que levam a palavra “theta” no nome das variáveis do algoritmo são referentes aos parâmetros Θ (Theta maiúsculo) das equações 7 e 8. Para este trabalho, utilizamos o RTP para baixas latitudes com a modificação aplicada em MacLeod et al. (1993), que utiliza uma pseudo-inclinação (I'). Pela teoria, o valor (I') pode ser definido pelo usuário, devendo apenas ser maior do que a inclinação do campo (I). Como o complemento do ângulo de inclinação

(c) pode ser usado como o valor de I' , neste trabalho colocamos este valor como padrão.

O código utilizado para o RTP tem base no sistema cartesiano, e a adaptação para algoritmo fica como no quadro que se segue. A função `DirCos()`, cosseno direcional, fornece os parâmetros de projeção nos eixos (x, y, z). Já a função `Kvalue()`, retorna os valores de k_x e k_y , com base no cálculo de k .

Para o filtro RTP, são utilizados dois exemplos. O primeiro, é um dado do Sítio Controlado da UFPR (Bongiolo et al., 2022), com uma anomalia de declinação e inclinação -30 e -20, respectivamente, onde foi possível conseguir um bom resultado do filtro RTP implementado como no modo original deste filtro (Figura 4).

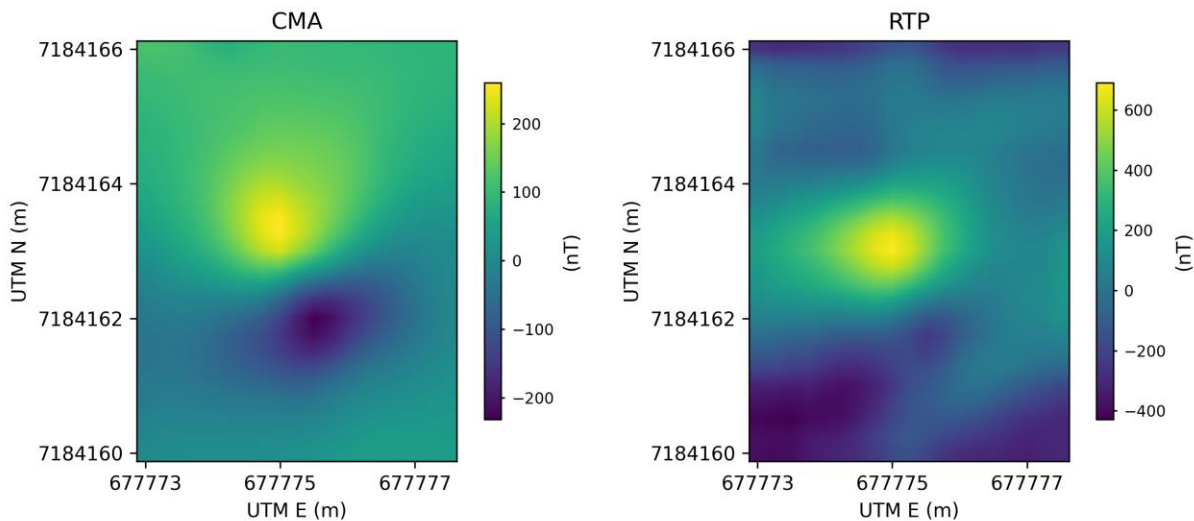


Figura 4 – Exemplos da aplicação do filtro de redução ao polo, para inclinação=30 (fonte enterrada, Sítio Controlado UFPR).

Já o segundo exemplo é uma anomalia com declinação e inclinação equivalentes a um ponto na região da margem equatorial brasileira (inclinação=11°, declinação=-18°; Figura 5.a e Figura 5.b) onde o RTP clássico mostra um desempenho menos eficiente (Figura 5.c) do que no exemplo do sítio controlado, devido à baixa latitude magnética (inclinação <20°). A anomalia de baixa latitude foi gerada no programa GRAV MAG PRISM (Bongiolo et al., 2013), para um campo de intensidade 30000nT. O filtro RTP modificado para baixas latitudes apresenta maior estabilidade (Figura 5.d). Mas este tipo de anomalia, localizada na região do equador geomagnético, dificilmente é centrada sobre a fonte de forma satisfatória como filtro RTP, sendo este um objeto de estudo constante de diversos geocientistas.

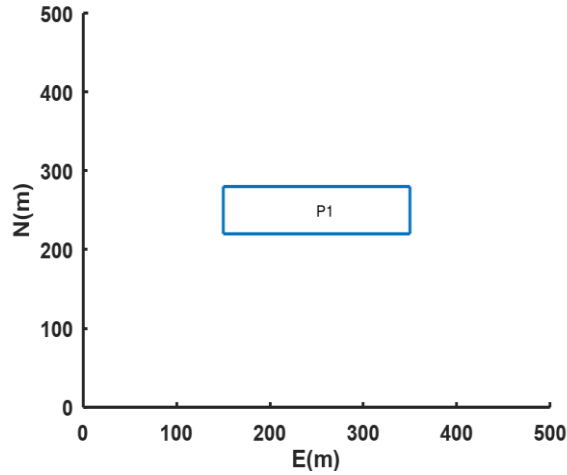
O outro filtro presente no repositório é o ASA, que também busca centralizar a feição anômala positiva sobre o corpo magnetizado (Figura 5.e).

Este filtro mostra um resultado melhor do que os demais filtros RTP neste exemplo.

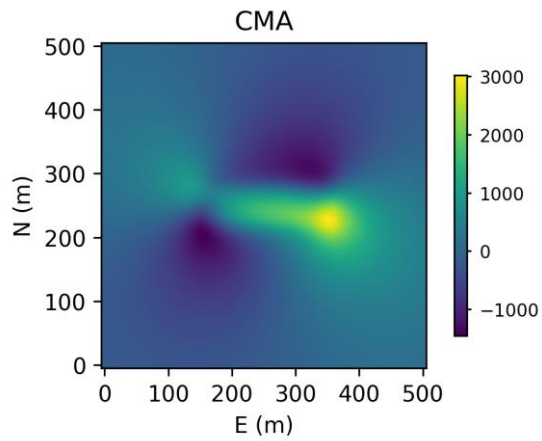
4.5. - Transformação de dados gravimétricos: Anomalia Bouguer

O cálculo da Anomalia Bouguer foi realizado utilizando funções prontas, originadas do projeto “Fatiando a Terra” (Uieda et al., 2013). Deste modo, a nossa contribuição para este algoritmo é limitada à apresentação da técnica com fins didáticos, mas sem a necessidade de adaptações. O dado utilizado para tal fim foi a anomalia ar-livre (Figura 6.a) disponibilizada em Sandwell et al. (2014; resolução espacial de 2x2km), recortada na região leste do estado do Paraná. Para o cálculo do efeito gravimétrico da topografia, utilizamos como referência de terreno o modelo digital SRTM3 (Figura 6.b), de resolução espacial de 1x1km, mas amostrado para células de 2x2km (mesmas posições da matriz da anomalia ar-livre).

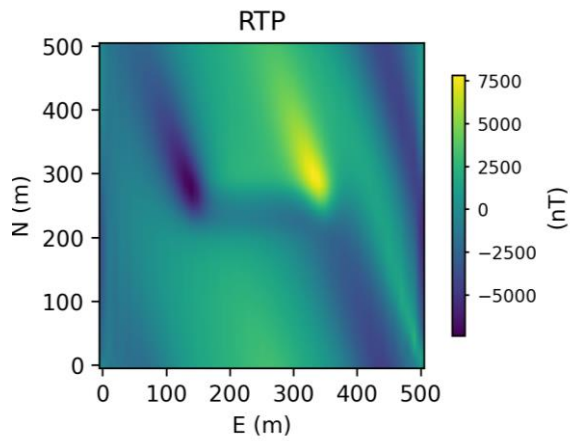
(a) Fonte da anomalia no equador magnético (Inclinação = 11)



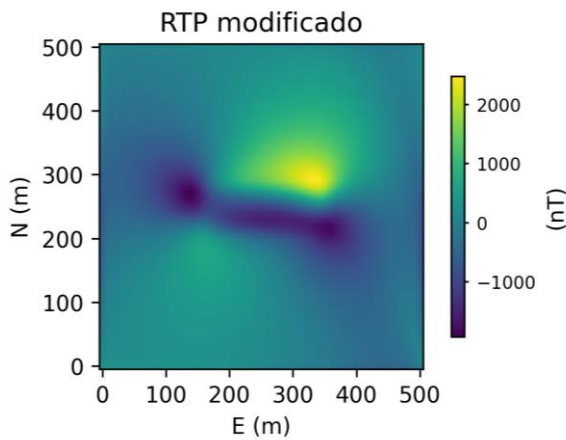
(b)



(c)



(d)



(e)

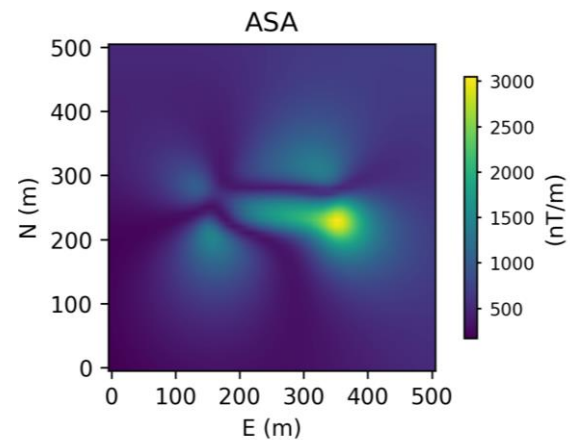


Figura 5 – Exemplo da aplicação do filtro de redução ao polo para um vetor de magnetização em baixa latitude magnética (<math><20^\circ</math>). Em (a), o prisma original em mapa. A anomalia original está em (b), o resultado do filtro RTP clássico em (c), e do filtro RTP melhorado em (d).

No método utilizado, o efeito gravimétrico do relevo (Figura 6.c) é obtido pelas estimativas de efeito de prismas retangulares em cada célula do dado, com a base em zero ou na menor altitude da área, e o topo determinado pelo valor do topo (altitude SRTM) subtraído o valor da base. A anomalia da topografia é obtida de uma camada composta por estes prismas, e com o valor de densidade determinado pelo usuário. Neste caso, utilizamos o valor de densidade para cobertura sedimentar, 2000 kg/m³. No exemplo,

a Anomalia Bouguer resultante (Figura 6.d) destaca uma estrutura positiva orientada a NW, correspondente a uma estrutura clássica da geologia do Brasil, o Arco de Ponta Grossa. Nota-se ainda que há efeitos gravimétricos da topografia no mapa final, devido ao contraste aplicado ser único e, portanto, não ser eficiente para representar todos os tipos de rochas da superfície.

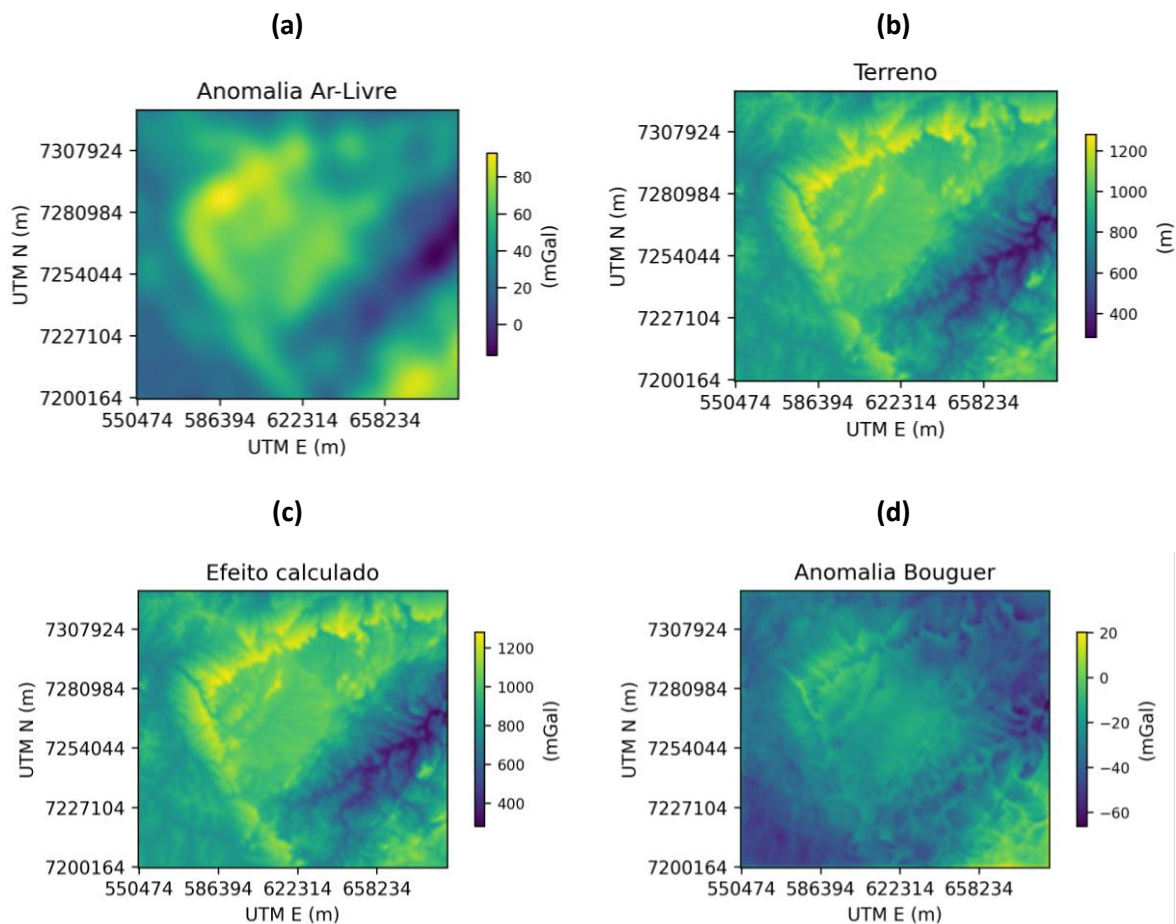


Figura 6 – Anomalia gravimétrica Bouguer. Os dados de Ar-Livre de Sandwell et al. (2014) em (a) e o modelo digital de terreno SRTM3 (1x1km) em (b) são dados de entrada para resultar no efeito gravimétrico exibido em (c). O efeito gravimétrico retirado da Anomalia de Ar-Livre resulta na Anomalia Bouguer (d).

5.0 - CONSIDERAÇÕES FINAIS

O material desenvolvido neste trabalho serve de apoio para ensino e pesquisa geocientífica envolvendo dados gravimétricos e magnetométricos, no tema de tratamento e filtragem de anomalias para dados terrestres, aéreos e de satélites. No caso particular do uso com fins didáticos no curso de Geologia, o trabalho atende à demanda de expor os alunos ao ambiente de programação, visto que esta é uma ferramenta essencial para diversos tipos de trabalhos geocientíficos.

O conjunto de algoritmos disponibilizado em repositório público, e no formato de execução na nuvem, proporciona o aprendizado em formato prático e acessível, não apenas para alunos, mas também para o público em geral. Apesar dos notebooks estarem voltados para uso online, também podem ser facilmente adaptados para uso local, mediante a instalação de programas e bibliotecas necessários para o bom funcionamento dos códigos.

Adicionalmente, ainda que o aluno não esteja motivado a aprender programação em si, o acesso aos algoritmos permitirá maior intimidade com o processo de transformação dos dados, além de estimular o pensamento computacional destes alunos. Assim como um cliente de restaurante se sente diferente ao comer olhando o cozinheiro preparar o prato, o aluno tem a oportunidade de observar quais as formulações e ferramentas estão por trás da geração dos mapas de anomalias geofísicas que ele irá utilizar em seus estudos. Com este método, esperamos que seja estabelecido um vínculo entre os alunos e o produto gerado do tratamento de seus dados para fins acadêmicos ou profissionais, e que seja incentivado o uso da geofísica nos trabalhos relativos a outras disciplinas das geociências, como mapeamento geológico e prospecção geoquímica, por exemplo.

Agradecimentos

As autoras agradecem ao apoio do Departamento de Geologia, Setor de Ciências da Terra (UFPR), e da equipe e alunos do LPGA-UFPR. O desenvolvimento dos

algoritmos foi beneficiado por sugestões do Geólogo Júlio C. S. O. Lyrio (Petrobras), e discussões construtivas com o Físico Erico S. Szameitat (Polícia Científica do Estado do Paraná, PCP/PR). O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, L. Szameitat.

6.0. - REFERÊNCIAS

BLAKELY, R. J., & SIMPSON, R. W. 1986. Approximating edges of source bodies from magnetic or gravity anomalies. *Geophysics*, 51(7), 1494-1498.

BLAKELY, R.J. Potential theory in gravity and magnetic applications: 1996. Cambridge University Press.

BONGIOLO, A.B.S.; DE SOUZA, J.; FERREIRA, F.J.F.; DE CASTRO, L.G. 2013. GRAV MAG PRISM: A Matlab/Octave program to generate gravity and magnetic anomalies due to rectangular prismatic bodies. *Brazilian Journal of Geophysics*, 31(3), 347-363.

BONGIOLO, A. B. S., S., SZAMEITAT, L. S. A., STEVANATO, R., CANATA, R. E., PEREIRA, H. G., & ANTONELLI, F. 2022. Geophysical surveying for the detection of ferrous-based objects: possibilities for depth estimative combining analytic signal and vertical integral of the anomalous magnetic field. *Brazilian Journal of Geophysics*, 40(1), 43-52.

CORDELL, L.; GRAUCH, V. Mapping Basement magnetization zones from aeromagnetic data in the San Juan Basin, New México, 1985. In: W. Hinze, ed., *Utility of regional gravity and magnetic maps*: SEG 181-197.

EVJEN, H.M. 1936. The place of vertical gradient in gravitational interpretations. *Geophysics*, 1, 127-136.

GUNN, P. J. 1975. Linear transformations of gravity and magnetic fields. *Geophysical Prospecting*, 23,300-312.

JONES, E., OLIPHANT, T., & PETERSON, P. (2001). SciPy: Open source scientific tools for Python. <http://www.scipy.org/>.

KEAREY P., BROOKS M., HILL I. 2009. Geofísica de Exploração. São Paulo, Oficina de textos, 429p.

HANSEN, R. O., & PAWLOWSKI, R. S. 1989. Reduction to the pole at low latitudes by Wiener filtering. *Geophysics*, 54(12), 1607-1613.

LI, X. 2006. Understanding 3D analytic signal amplitude. *Geophysics*, 71(2), L13-L16.

LI, X., 2008. Magnetic reduction-to-the-pole at low latitudes: Observations and considerations. *The Leading Edge*, 27, 990-1002.

LOWRIE W. 2007. Fundamentals of Geophysics. Cambridge University Press. Cambridge.

MACLEOD, I. N., JONES, K., & DAI, T. F. 1993. 3-D analytic signal in the interpretation of total magnetic field data at low magnetic latitudes. *Exploration Geophysics*, 24(4), 679-688.

NABIGHIAN M.N. 1972. The analytic signal of two-dimensional magnetic bodies with polygonal cross-section: its properties and use for automated anomaly interpretation. *Geophysics*, 37, 507-517.

ROEST, W.R., VERHOEF, V. & PILKINGTON, M. 1992. Magnetic interpretation using the 3-D analytic signal. *Geophysics*, 57, 116-125.

SANDWELL, D. T., MÜLLER, R. D., SMITH, W. H., GARCIA, E., & FRANCIS, R. 2014. New global marine gravity model from CryoSat-2 and Jason-1 reveals buried tectonic structure. *Science*, 346(6205), 65-67.

UIEDA, L., OLIVEIRA JR, V. C., & BARBOSA, V. C. 2013. Modeling the Earth with Fatiando a Terra. In *Proceedings of the 12th Python in Science Conference*, June, 96-103.

VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., HABERLAND, M., REDDY, T., COURNAPEAU, D., BUROVSKI, E., PETERSON, P., WECKESSER, W., BRIGHT, J., VAN DER WALT, S. J., BRETT, M., WILSON, J., MILLMAN, K. J., MAYOROV, N., NELSON, A. R. J., JONES, E., KERN, R., LARSON, E., CAREY, C. J., POLAT, İ., FENG, Y., MOORE, E. W., VANDERPLAS, J., LAXALDE, D., PERKTOLD, J., CIMRMAN, R., HENRIKSEN, I., QUINTERO, E. A., HARRIS, C. R., ARCHIBALD, A. M., RIBEIRO, A. H., PEDREGOSA, F., VAN MULBREGT, P. & SCIPY 1.0 CONTRIBUTORS. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261-272.

Submetido em 08/07/2023

Aceito em 18/01/2024

APÊNDICE

IMPORTAÇÕES NECESSÁRIAS

Bibliotecas nativas

- numpy, importar como np;
- scipy.fftpack;
- math, importar radians, sin, cos, sqrt;
- matplotlib.pyplot, importar como plt;
- pandas, importar como pd.

Bibliotecas “Fatiando a Terra” (apenas para o cálculo da anomalia Bouguer)

- verde, importar como vd;
- harmonica, importar como hm.

FUNÇÕES IMPLEMENTADAS EM PYTHON PARA TRATAMENTO DE DADOS

Função Kvalue()

```
def Kvalue(i, j, nx, ny, dkx, dky):
```

```
    """
```

```
    Parametros
```

```
    -----
```

```
    i, j(int):
```

```
        posicoes, ou indices de varredura
```

```
    nx, ny (int):
```

```
        número de amostras em x e y
```

```
    dkx, dky (int):
```

```
        distância em x e y em radianos
```

```
    """
```

```
    nyqx = nx / 2 + 1
```

```
    nyqy = ny / 2 + 1
```

```
    kx = float()
```

```
    ky = float()
```

```
    if j <= nyqx:
```

```
        kx = (j-1) * dkx
```

```
    else:
```

```
        kx = (j-nx-1) * dkx
```

```
    if l <= nyqy:
```

```
        ky = (i-1) * dky
```

```
    else:
```

```
        ky = (i-ny-1) * dky
```

```
    return kx, ky
```

Função Derivada()

```
def Derivada(vetor, axis, nx, ny, dkx, dky):
```

```
    """
```

```
    Parametros
```

```
    -----
```

```
    grid (2d array):
```

```
        array de duas dimensões com o campo a ser filtrado
```

```
    axis (string):
```

```
        'dx', 'dy' ou 'dz'
```

```
    nx, ny (int):
```

```
        numero de amostras em x e y
```

```
    dx, dy (int):
```

```
        distancia entre amostras em x e y
```

```
    """
```

```
    # FFT, necessita de um grid de 2 dimensões como entrada
```

```
    grid = np.reshape(vetor, (ny, nx)) #de 1D para 2D
```

```
    gridfft = scipy.fftpack.fft2(grid)
```

```
    # Grid – de matriz para vetor
```

```
    gridfft = np.reshape(gridfft, ny*nx)
```

```
    # Cria um vetor complexo
```

```
    gridfft_filt
```

```
=
```

```
    np.zeros(ny*nx).astype(complex)
```

```
    if axis == 'z':
```

```
        for j in range(1, ny+1):
```

```
            for l in range(1, nx+1):
```

```
                ij = (j-1) * nx + l
```

```
                kx, ky = Kvalue(l, j, nx, ny, dkx, dky)
```

```
                k = sqrt(kx**2 + ky**2)
```

```
                gridfft_filt[ij-1] = gridfft[ij-1]*k**1
```

```
    #filtro
```

```
    elif axis == 'y':
```

```
        for j in range(1, ny+1):
```

```
            for l in range(1, nx+1):
```

```
                ij = (j-1) * nx + l
```

```
                kx, ky = Kvalue(l, j, nx, ny, dkx, dky)
```

```
                k = kx
```

```
                gridfft_filt[ij-1]
```

```
=
```

```
                gridfft[ij-
```

```
1]*k*complex(0, 1)
```

```
    elif axis == 'x':
```

```
        for j in range(1, ny+1):
```

```
            for i in range(1, nx+1):
```

```
                ij = (j-1) * nx + i
```

```
                kx, ky = Kvalue(i, j, nx, ny, dkx, dky)
```

```
                k = ky
```

```
                gridfft_filt[ij-1]
```

```
=
```

```
                gridfft[ij-
```

```
1]*k*complex(0, 1)
```

```
    else:
```

```
        return print ('Tipo não suportado: ', axis)
```

```
#iFFT, entrada como matriz 2D
gridfft_filt = np.reshape(gridfft_filt, (ny, nx))
gridfft_filt = scipy.fftpack.ifft2(gridfft_filt)

#retorna para 1D, para saída como vetor
vetor_out = np.reshape(gridfft_filt.real,
(ny*nx))

return vetor_out
```

Função GradHoriz()

```
def GradHoriz(vetor, nx, ny, dx, dy):
'''
Parametros
-----
vetor (1d array):
array de uma dimensão contendo o
campo total
nx, ny (int):
numero de amostras em x e y
dx, dy (int):
distancia entre amostras em x e y
'''
dx2 = 2*dx
dy2 = 2*dy
vetor_GH = np.ones(nx*ny)
for j in range(1, ny+1):
jm1 = j - 1
if jm1 < 1: #condição de borda
jm1 = 1
jp1 = j + 1
if jp1 > ny: #condição de borda
jp1 = ny
for i in range(1, nx+1):
im1 = i - 1
if im1 < 1: #condição de borda
im1 = 1
ip1 = i + 1
if ip1 > nx: #condição de borda
ip1 = nx
#index(i,j,nx) = (j-1) * nx + i #função index
em Blakely (1996)
indexip1 = (jp1-1) * nx + (i-1)
indexim1 = (jm1-1) * nx + (im1-1)
indexjp1 = (jp1-1) * nx + (i-1)
indexjm1 = (jm1-1) * nx + (i-1)
#Análise de valores no espaço
```

```
dfdx = (vetor[indexip1]-vetor[indexim1]) /
dx2
dfdy = (vetor[indexjp1]-vetor[indexjm1]) /
dy2
ij = (j-1) * nx + (i-1)
vetor_GH[ij] = sqrt(dfdx**2+dfdy**2)

return vetor_GH
```

Função DirCos()

```
def DirCos(i, d, az):
'''
Parametros
-----
i: inclinacao
d: declinacao
az: azimute
'''
a = cos(radians(i))*cos(radians(d)-
radians(az))
b = cos(radians(i))*sin(radians(d)-
radians(az))
c = sin(radians(i))
return a, b, c
```

Função NewVec()

```
def NewVec(grid, nx, ny, dx, dy, fi1, fd1, mi1,
md1, fi2, fd2, mi2, md2):
'''
NewVec(grid, nx, ny, dx, dy, fi1, fd1, mi1,
md1, fi2, fd2, mi2, md2)
Objetivo: simular mudanças na direção de
magnetização
Para redução ao polo, deve-se usar mi2 =
fi2 = 90
Parametros
-----
grid (2d array):
array de duas dimensões contendo o
campo total
nx, ny (int):
numero de amostras em x e y
dx, dy (int):
distancia entre amostras em x e y
mi1, md1 - (graus):
inclinacao e declinacao da
magnetizacao
fi1, fd1 (int ou float):
```

```

Magnetizacao do campo geomagnetico
atual (inclinacao, declinacao)
mi2, md2 (int ou float):
Nova direção de magnetizacao
(inclinacao, declinacao)
fi2, fd2 (int ou float):
Nova magnetizacao do campo
geomagnetico (inclinacao, declinacao)
'''

```

```

dkx = 2. * np.pi / (nx * dx)
dky = 2. * np.pi / (ny * dy)

```

```

#Cossenos direcionais
mx1, my1, mz1 = DirCos(mi1, md1, 0.) # i,d
do corpo, magnetização inicial
fx1, fy1, fz1 = DirCos(fi1, fd1, 0.) # i,d do
campo, inicial
mx2, my2, mz2 = DirCos(mi2, md2, 0.) # i,d
do corpo, final
fx2, fy2, fz2 = DirCos(fi2, fd2, 0.) # i,d do
campo, final

```

```

#Transformada de Fourier usando scipy
gridfft = scipy.fftpack.fft2(grid) # fft2 -
matriz 2D
gridfft = np.reshape(gridfft, nx*ny) # muda
a matriz fft para forma 1D

```

```

#Filtro de fase
for j in range (1, nx+1):
for i in range (1, ny+1):
ij = (j-1) * ny + i
if ij == 1:
cphase = 0.0
else:
kx, ky = Kvalue(i, j, nx, ny, dkx, dky) #
** funcao Kvalue
k = sqrt(kx**2 + ky**2)
thetam1 = complex(mz1, (kx*mx1 +
ky*my1) / k)
thetaf1 = complex(fz1, (kx*fx1 +
ky*fy1) / k)
thetam2 = complex(mz2, (kx*mx2 +
ky*my2) / k)
thetaf2 = complex(fz2, (kx*fx2 +
ky*fy2) / k)

```

```

cphase = thetam2 * thetaf2 /
(thetam1 * thetaf1)

```

```

# Filtra cada item do grid
gridfft[ij-1] = gridfft[ij-1]*cphase

```

```

#Fourier
gridfft = np.reshape(gridfft, (ny, nx)) #
retorna para formato 2D
gridiff = scipy.fftpack.ifft2(gridfft)
grid = gridiff.real #usamos apenas a parte
real

```

```

return grid

```

```

Função RTPL()

```

```

def RTPL(grid, nx, ny, dx, dy, mi1, md1, mi2,
md2):
'''

```

```

Alterado de NewVec(grid, nx, ny, dx, dy, fi1,
fd1, mi1, md1, fi2, fd2, mi2, md2)

```

```

Objetivo: simular mudanças na direção de
magnetização

```

```

Para redução ao polo, deve-se usar mi2 =
fi2 = 90

```

```

Parametros
-----

```

```

grid (2d array):
array de duas dimensões contendo o
campo total
nx, ny (int):
numero de amostras em x e y
dx, dy (int):
distancia entre amostras em x e y
mi1, md1 (int ou float):
inclinacao e declinacao da
magnetização, em graus
fi1, fd1 (int ou float):
magnetizacao do campo geomagnetico
atual (inclinacao, declinacao)
mi2, md2 (int ou float):
nova direção de magnetizacao
(inclinacao, declinacao)
fi2, fd2 (int ou float):
nova magnetizacao do campo
geomagnetico (inclinacao, declinacao)

```

Obs1: Poderíamos omitir os parâmetros de f ou de m , pois se assume que a remanência é negligenciável ($f == m$). No entanto, vamos manter a separação nomeando as variáveis m_i e f_i , e m_d e f_d , para facilitar a compreensão e a associação com as fórmulas do livro de Blakely (1996).

Obs2: Fórmula da correção de fase para baixas latitudes em coord. polares:

```
theta = atan(ky/kx)
fase = (complex((sin(mi2)**2 - cos(mi2)**2),
-
(2*sin(mi2)*cos(mi2))))/complex( (sin(c)**2)+(
cos(c)**2)*(cos(md1-theta)**2) ) *
( (sin(mi1)**2)*(cos(mi1)**2)*(cos(md1-
theta)**2) ))
```

'''

```
fi1, fi2, fd1, fd2 = mi1, mi2, md1, md2
```

```
dkx = 2. * np.pi / (nx * dx)
```

```
dky = 2. * np.pi / (ny * dy)
```

```
#Transformada de Fourier usando scipy
gridfft = scipy.fftpack.fft2(grid) # fft2 -
matriz 2D
```

```
gridfft = np.reshape(gridfft, nx*ny) # muda
a matriz fft para forma 1D
```

```
#Filtro de fase
```

```
if mi1 < 0:
```

```
    c = -90-mi1
```

```
else:
```

```
    c = 90-mi1
```

```
print ("Complemento: ", c)
```

```
for j in range (1, nx+1):
```

```
    for i in range (1, ny+1):
```

```
        ij = (j-1) * ny + i
```

```
        if ij == 1:
```

```
            cphaseLL = 0.0
```

```
        else:
```

```
            kx, ky = Kvalue(i, j, nx, ny, dkx, dky) #
```

```
** funcao Kvalue
```

```
            k = sqrt(kx**2 + ky**2)
```

```
#Cossenos direcionais (projeção em
x, y, z dos vetores m e f)
```

```
mx1, my1, mz1 = DirCos(mi1, md1,
0.) # i,d do corpo, magnetização inicial
```

```
fx1, fy1, fz1 = DirCos(fi1, fd1, 0.) # i,d
do campo, magnetização inicial
```

```
mx2, my2, mz2 = DirCos(mi2, md2,
0.) # i,d do corpo, magnetização final
```

```
fx2, fy2, fz2 = DirCos(fi2, fd2, 0.) # i,d
do campo, magnetização final
```

```
thetam1 = complex(mz1, (kx*mx1 +
ky*my1) / k)
```

```
thetaf1 = complex(fz1, (kx*fx1 +
ky*fy1) / k)
```

```
thetam2 = complex(mz2, (kx*mx2 +
ky*my2) / k)
```

```
thetaf2 = complex(fz2, (kx*fx2 +
ky*fy2) / k)
```

```
#Condições de baixas latitudes,
trabalha com o complemento de 90
```

```
cx, cy, cz = DirCos(c, md1, 0.)
```

```
thetam3 = complex(mz1, (-kx*mx1 -
ky*my1) / k)
```

```
thetac1 = complex(cz, (kx*cx + ky*cy)
/ k)
```

```
thetac2 = complex(cz, (-kx*cx - ky*cy)
/ k)
```

```
cphase = thetam2 * thetaf2 /
(thetam1 * thetaf1)
```

```
cphaseLL = cphase * (thetam1 *
thetam3 / (thetac1 * thetac2))
```

```
# Filtra cada item do grid
```

```
gridfft[ij-1] = gridfft[ij-1]*cphaseLL
```

```
#Fourier – transformada inversa
```

```
gridfft = np.reshape(gridfft, (ny, nx)) #
retorna para formato 2D
```

```
gridiff = scipy.fftpack.ifft2(gridfft)
```

```
grid = gridiff.real #usamos apenas a parte
real
```

```
return grid
```

```
REPOSITÓRIO
```

```
Repositório público – GitHub:
```

```
https://github.com/lizam/ensino
```

```
Programas em formato Python notebook
(IPYNB).
```

- Filtro Derivada:
geof_filtros/geof_derivadas.ipynb
- Filtro Gradiente Horizontal:
geof_filtros/geof_gradientehoriz.ipynb
- Filtro RTP em latitude magnética alta:
geof_mag/geof_mag_rtp-
blakely_magsitio.ipynb
- Filtro RTP em baixa latitude:
geof_mag/geof_mag_rtp-
blakely_gravmagprism-baixaLAT.ipynb

- Filtro RTP adaptado para baixa latitude:
geof_mag/geof_mag_rtpL_gravmagprism-
baixaLAT.ipynb
- Filtro ASA:
geof_filtros/geof_mag_asa_gravmagprism-
baixaLAT.ipynb
- Anomalia Bouguer:
geof_grav/geof_grav_boug.ipynb