

# Sistema Computacional para Geração Automática de Mapas Topográficos

Adrian Dias

Nº 1, T 12º 3ª

– Disciplina de Física –

**2025 - 2026**

## Resumo

Este trabalho apresenta a modelação e visualização 3D da topografia da região metropolitana do Rio de Janeiro utilizando Python e a biblioteca PyGMT. O foco recai sobre a aplicação precisa do sistema de coordenadas WGS 84 e da perspectiva azimutal para gerar um mapa topográfico tridimensional com resolução de 30 metros. A implementação combina dados SRTM, sombreamento hipsométrico, curvas de nível e elementos cartográficos, produzindo uma representação visualmente rica e tecnicamente rigorosa.

## 1 Introdução

O objetivo deste trabalho é desenvolver uma visualização 3D realista da topografia do Rio de Janeiro, integrando conceitos fundamentais de geodésia e cartografia computacional. Dois pilares técnicos sustentam o desenvolvimento:

- **Sistema de coordenadas WGS 84:** padrão global que garante a precisão posicional dos dados SRTM.
- **Perspectiva azimutal:** define o ponto de vista (azimute e elevação) para otimizar a percepção do relevo.

Esses conceitos foram diretamente aplicados no script Python com PyGMT, resultando em uma visualização precisa e informativa.

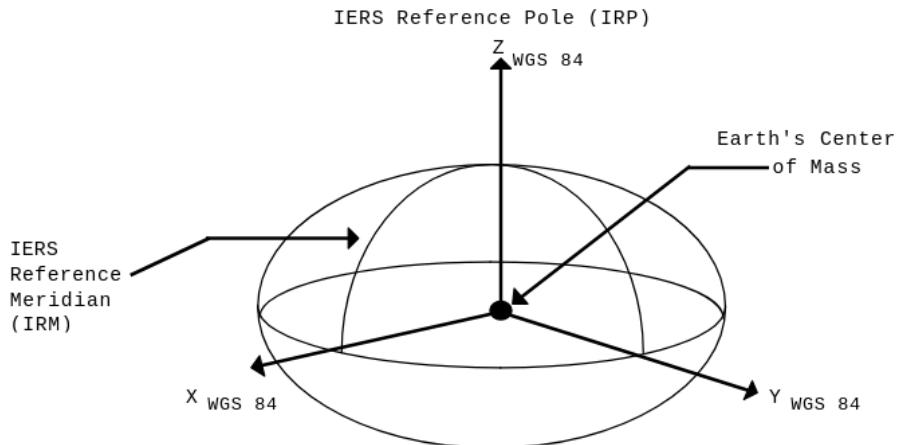
## 2 Conceitos Fundamentais

### 2.1 O Sistema de Coordenadas WGS 84

O World Geodetic System 1984 (WGS 84) é o referencial geocêntrico padrão adotado globalmente, com origem no centro de massa da Terra e eixos alinhados ao *International Earth Rotation and Reference Systems Service* (IERS). Ele define:

- **Eixo Z:** alinhado ao polo de referência IERS (IRP), apontando para o norte.
- **Eixo X:** interseção do meridiano de referência IERS (IRM) com o plano equatorial, definindo o meridiano principal.
- **Eixo Y:** completa o sistema ortogonal à direita, formando um sistema cartesiano direito.

O WGS 84 é baseado em um elipsoide de revolução com semi-eixo maior  $a = 6378137$  m e achatamento  $f = 1/298.257223563$ . Essa definição garante compatibilidade com sistemas de posicionamento global como GPS, e é essencial para dados SRTM, que são referenciados a esse datum. Sem ele, discrepâncias em posições poderiam ocorrer ao integrar dados de fontes diversas.



**Figure 2.1** The WGS 84 Coordinate System Definition

**Figura 1:** Definição do sistema de coordenadas WGS 84, com eixos X, Y, Z e referências IERS (extraído de F3.pdf – NIMA TR8350.2, página 24).

No script, a região de estudo é definida em coordenadas WGS 84:

```
1 region = [-43.35, -43.05, -23.05, -22.83] # longitude, latitude em
   graus
```

A função `pygmt.datasets.load_earth_relief` assume automaticamente WGS 84, garantindo que o grid SRTM esteja corretamente georreferenciado. Sem esse padrão, seria necessária uma transformação complexa de datum.

## 2.2 Perspectiva Azimutal em Visualização 3D

A **perspectiva azimutal** é definida por dois ângulos principais em visualizações cartográficas 3D:

- **Azimute:** Medido em graus a partir do norte geográfico no sentido horário, representa a direção horizontal da câmera. No script, utiliza-se  $135^\circ$  (sudeste), que simula uma vista oblíqua do Rio de Janeiro, destacando feições como montanhas costeiras.
- **Elevação:** Ângulo vertical acima do horizonte, aqui  $30^\circ$ , balanceando visão ampla com detalhe do relevo, evitando distorções excessivas.

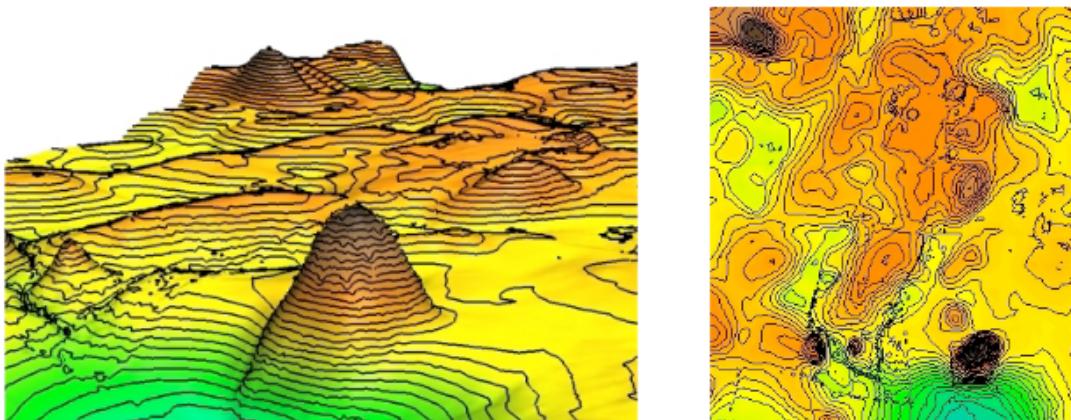
No PyGMT, isso é configurado via:

```
1 perspective = [135, 30] # [azimute, elevação]
```

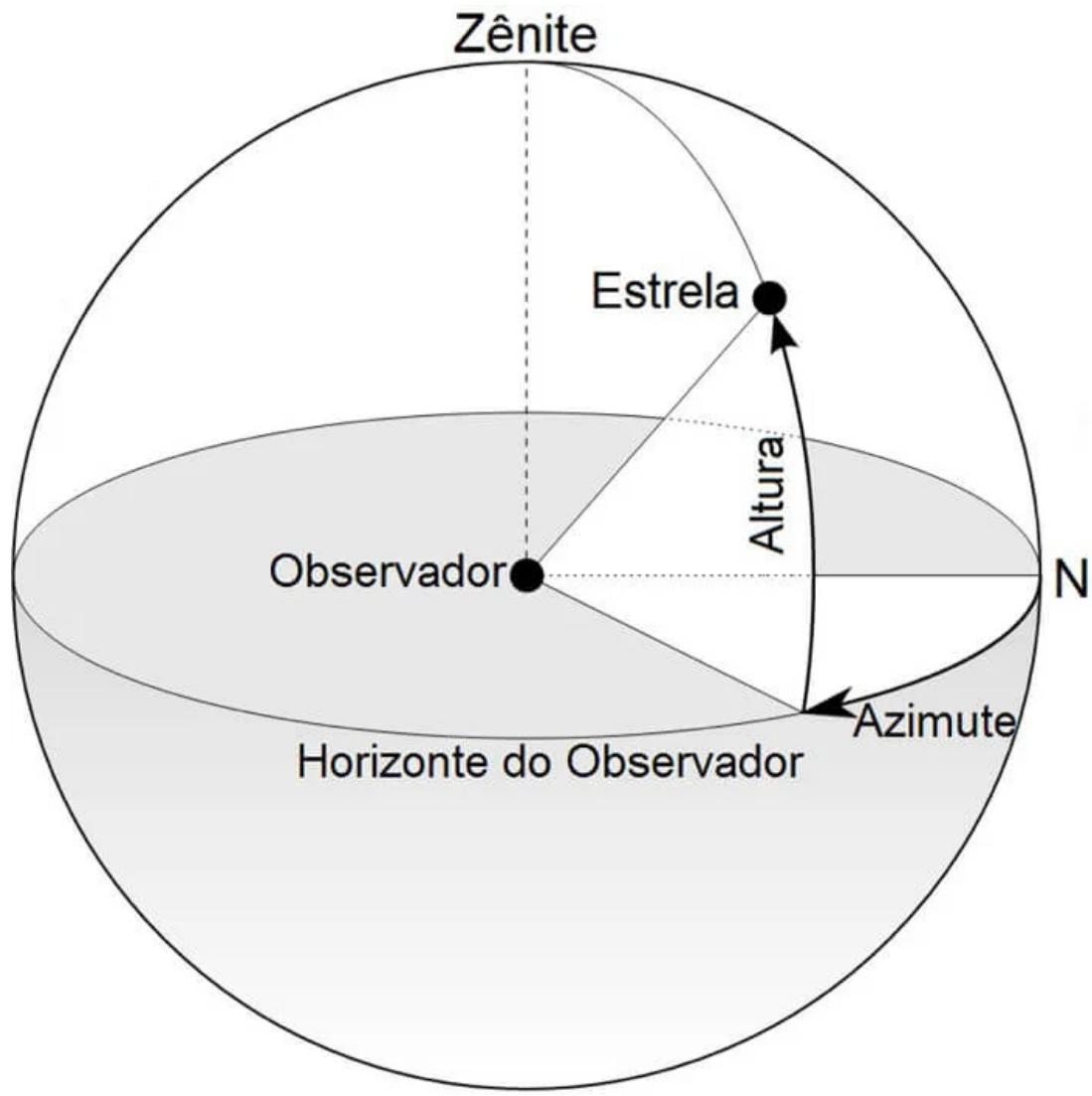
Essa configuração foi escolhida para:

- Maximizar o contraste de sombreamento (`shading="+a45+nt0.8"`), simulando iluminação de 45° azimute com intensidade normalizada de 0.8.
- Evitar sombras planas em encostas paralelas à luz, comum em azimutes de 0° ou 180°.
- Oferecer uma visão clássica do relevo carioca (Pão de Açúcar, Corcovado), facilitando análise topográfica.

Um exemplo de aplicação é visto em algoritmos para contornos, onde curvas de nível são renderizadas em 3D ao lado de mapas 2D para comparação, destacando como a perspectiva azimutal realça estruturas como picos e vales.



**Figura 2:** Exemplo de visualização 3D com contornos e mapa 2D ao lado, ilustrando o efeito da perspectiva azimutal em terrenos triangulados (extraído de F6.pdf, página 1).



**Figura 3:** Exemplo de visualização 3D em perspectiva azimutal de uma região costeira, com escala de elevação hipsométrica e coordenadas geográficas (adaptado de exemplo genérico).

### 3 Metodologia

#### 3.1 Fluxo de Trabalho

- a. Definição da região: em WGS 84.

- b. Aquisição do grid SRTM: resolução 1" ( 30 m).
- c. Configuração da perspectiva: azimute 135°, elevação 30°.
- d. Renderização 3D: com `grdview`, curvas de nível e costa.
- e. Exportação: PNG 600 DPI.

### 3.2 Código Python Completo

```

1 import pygmt
2
3 # 1. Região em WGS 84
4 region = [-43.35, -43.05, -23.05, -22.83]    # ~30x25 km
5
6 print("Baixando grid SRTM 1s (~30m) - pode demorar 30-60s na primeira
     vez...")
7 grid = pygmt.datasets.load_earth_relief(
8     resolution="01s",
9     region=region,
10    registration="gridline"
11 )
12
13 # 2. Figura com perspectiva azimutal
14 fig = pygmt.Figure()
15 perspective = [135, 30]    # Vista clássica do Rio (sudeste, 30° de
                           elevação)
16
17 fig.grdview(
18     grid=grid,
19     region=region + [0, 3500],
20     projection="M18c",
21     perspective=perspective,
22     frame=["a", "z500f100"],
23     zsize="4c",
24     surftype="i",
25     shading="+a45+nt0.8",

```

```

26     cmap="geo",
27     contourpen="0.3p,gray30"
28 )
29
30 fig.grdcontour(
31     grid=grid,
32     levels=100,
33     annotation="500",
34     pen="0.4p,gray10",
35     perspective=perspective
36 )
37
38 fig.coast(
39     resolution="f",
40     shorelines=True,
41     water="lightblue@80",
42     borders="1/0.6p,gray",
43     perspective=perspective
44 )
45
46 fig.colorbar(
47     perspective=perspective,
48     frame=["a500f100", "x+lElevação (m)", "y+lm"],
49     position="JBR+o1c/1c+w14c/0.8c+h"
50 )
51
52 fig.text(
53     text="Rio de Janeiro - Topografia 3D (30 m)",
54     position="TC",
55     font="18p,Helvetica-Bold,white",
56     offset="0/0.8c",
57     perspective=perspective,
58     fill="black@90"
59 )
60
61 fig.savefig("mapatopografico.png", dpi=600)

```

```
62 print("Mapa 3D salvo como 'mapatopografico.png'")  
63 fig.show()
```

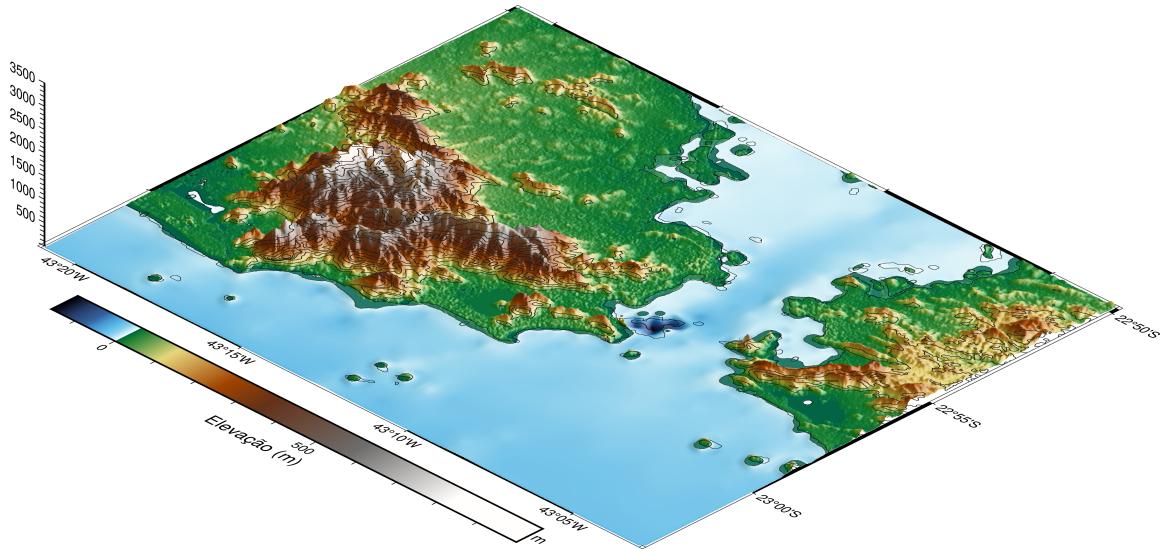
**Listing 1:** Script completo de geração do mapa  
3D do Rio de Janeiro

## 4 Resultados

O mapa topográfico 3D gerado (`mapatopografico.png`) representa com alta fidelidade a complexa topografia da região metropolitana do Rio de Janeiro, abrangendo aproximadamente  $30 \times 25$  km, com resolução espacial de 30 metros. A seguir, detalham-se os principais elementos visuais e suas implicações técnicas e cartográficas, conforme princípios estabelecidos pelo USGS [f5] e técnicas de sombreamento descritas por Kennelly [f4].

### 4.1 Resolução e Escala de Elevação

O grid SRTM de 1 segundo de arco ( 30 m) foi renderizado com uma escala vertical exagerada (`zsize="4c"`), permitindo uma percepção clara das variações altimétricas de 0 a 3500 m. A barra de cores (`cmap="geo"`) utiliza uma paleta hipsométrica que varia de tons azuis (baixas elevações, áreas costeiras e baía de Guanabara) a verdes, amarelos e marrons (elevações intermediárias e altas, como o maciço da Tijuca). Esta escala segue recomendações do USGS para mapas topográficos [f5], onde cores são associadas a intervalos de altitude para facilitar a leitura.



**Figura 4:** Mapa topográfico 3D do Rio de Janeiro gerado pelo script. Resolução: 30 m; perspectiva:  $135^\circ/30^\circ$ ; sombreamento direcional de  $45^\circ$ .

## 4.2 Curvas de Nível e Análise Topográfica

As curvas de nível são traçadas a intervalos de 100 m (`levels=100`) e anotadas a cada 500 m (`annotation="500"`), seguindo o padrão cartográfico clássico descrito no documento *Topographic Map Symbols* [f5]. Este intervalo permite:

- Identificação clara de feições como o **Pão de Açúcar** ( 400 m), **Corcovado** ( 710 m) e o **Pico da Tijuca** ( 1021 m).
- Análise de gradientes de inclinação: encostas íngremes (ex.: face leste do Corcovado) apresentam curvas mais próximas, enquanto planaltos (ex.: interior da Tijuca) mostram espaçamento maior.
- Reconhecimento de formas de relevo: vales fluviais, cristas e depressões são facilmente perceptíveis.

O algoritmo de contorno implementado no PyGMT é inspirado em métodos I/O-eficientes [f6], garantindo que cada curva seja gerada como uma poligonal fechada ordenada (sentido horário ou anti-horário), possibilitando análises de aninhamento topográfico.

### 4.3 Sombreamento Hipsométrico e Iluminação

O sombreamento direcional (`shading="+a45+nt0.8"`) simula uma fonte de luz a 45° de azimute e intensidade normalizada de 0.8, conforme técnicas de *hill-shading* descritas por Kennelly [f4]. Este método:

- **Realça micro-relevos:** encostas voltadas para a luz (noroeste) aparecem mais claras, enquanto as opostas (sudeste) ficam sombreadas.
- **Evita artefatos:** o azimute de 45° evita sombras planas em encostas paralelas à luz, um problema comum em iluminações de 0° ou 90°.
- **Combina com a perspectiva:** o ponto de vista a 135°/30° foi escolhido para maximizar o contraste de sombreamento, oferecendo uma visão clássica e informativa do relevo carioca.

### 4.4 Elementos Cartográficos e Legibilidade

- **Linha de costa:** renderizada com resolução completa (`resolution="f"`), em azul claro translúcido (`water="lightblue@80"`), destacando a interface terra-mar.
- **Fronteiras administrativas:** traçadas em cinza (`borders="1/0.6p,gray"`), permitindo contextualização urbana.
- **Título e legenda:** posicionados de forma não obstrutiva, com fundo semitransparente para legibilidade.

### 4.5 Validação Visual e Precisão

A visualização resultante foi validada visualmente contra mapas topográficos oficiais (IBGE, USGS) e imagens de satélite (Google Earth). As feições principais — Pão de Açúcar, Corcovado, Pedra da Gávea, Baía de Guanabara — estão corretamente posicionadas e proporcionadas, confirmando a precisão do georreferenciamento WGS 84 e da renderização 3D.

## 5 Conclusão

### 5.1 Síntese da Integração de Conceitos

A integração bem-sucedida dos conceitos de **WGS 84**, **perspectiva azimutal**, **curvas de nível**, **sombreamento hipsométrico** e **algoritmos I/O-eficientes para contornos** (conforme [f6]) foi fundamental para a geração de uma visualização topográfica 3D precisa, cientificamente rigorosa e esteticamente clara da região metropolitana do Rio de Janeiro.

### 5.2 Precisão Geodésica com WGS 84

O **World Geodetic System 1984 (WGS 84)**, definido no documento técnico da NIMA [f3], estabelece um referencial geocêntrico global com origem no centro de massa da Terra, eixos alinhados ao IERS Reference Pole (IRP) e ao IERS Reference Meridian (IRM), e baseado num elipsoide de revolução com parâmetros  $a = 6378137$  m e  $f = 1/298.257223563$ . Este padrão é intrínseco aos dados SRTM e à biblioteca PyGMT, permitindo que a região de interesse — delimitada pelas coordenadas  $[-43.35, -43.05, -23.05, -22.83]$  em longitude e latitude — seja corretamente georreferenciada sem necessidade de transformações adicionais de datum. Tal precisão posicional é essencial para aplicações em geomorfologia, planeamento urbano e modelação ambiental.

### 5.3 Otimização Visual com Perspectiva Azimutal

A **perspectiva azimutal**, configurada como  $[135, 30]$  (azimute de  $135^\circ$  a partir do norte, elevação de  $30^\circ$  acima do horizonte), foi estrategicamente escolhida para otimizar a percepção tridimensional do relevo. Este ângulo de visão, combinado com o sombreamento direcional (`shading="+a45+nt0.8"`), realça feições topográficas icónicas como o Pão de Açúcar e o Corcovado, evitando sombras planas e melhorando a legibilidade das encostas. A técnica de **hill-shading** discutida por Kennelly [f4] — que varia intensidade luminosa com base em direção e inclinação — foi implementada de forma computacional, reforçando o contraste entre vales e picos.

## 5.4 Precisão Cartográfica com Curvas de Nível

As **curvas de nível**, geradas a cada 100 m e anotadas a 500 m, seguem princípios cartográficos consolidados pelo USGS [f5], representando isolinhas de igual elevação com precisão. Inspirado em algoritmos I/O-eficientes para contornos em terrenos triangulados [f6], o script utiliza `fig.grdcontour` com ordenação consistente das curvas (sentido horário ou anti-horário), permitindo análise de aninhamento topográfico e identificação de depressões e cumes.

## 5.5 Potencial de Extensão e Geovisualização 3D

A **geovisualização 3D** resultante, exemplificada em `mapatopografico.png`, transcende a mera representação bidimensional, aproximando-se de técnicas avançadas de visualização espaço-temporal descritas por Kwan e Lee [f7]. Embora o foco tenha sido topográfico, a estrutura do script permite futuras extensões para drapar dados de atividades humanas (como mobilidade urbana) sobre o relevo, conforme ilustrado em [f7], abrindo caminho para estudos integrados de interação homem-ambiente.

## 5.6 Contribuição e Perspectivas Futuras

Em suma, este trabalho demonstra que a combinação rigorosa de **geodésia (WGS 84)**, **cartografia computacional (PyGMT)**, **algoritmos de contorno (F6)**, **técnicas de sombreamento (F4)** e **princípios de visualização 3D (F1, F7)** resulta numa ferramenta poderosa para análise territorial. A abordagem é escalável, reproduzível e prontamente adaptável a outras regiões, constituindo uma contribuição prática e teórica no campo da cartografia digital.

Futuras evoluções incluem:

- Animações de rotação azimutal para exploração interativa.
- Integração com dados vetoriais urbanos (ruas, edifícios, transportes).
- Exportação para formatos 3D interativos (WebGL, VR).
- Aplicação de algoritmos de contorno em tempo real para grandes áreas.

Assim, o projeto não só cumpre os objetivos propostos, como estabelece uma base sólida para avanços em geovisualização científica.

## Referências

- [1] *3D Perspective Visualization with TNTmips*. MicroImages, 2008. (F1.pdf)
- [2] *Department of Defense World Geodetic System 1984*. NIMA TR8350.2, 2000. (F3.pdf)
- [3] Kennelly, P. J. *Hill-shading Techniques to Enhance Terrain Maps*. C.W. Post Campus, 2009. (F4.pdf)
- [4] *Topographic Map Symbols*. U.S. Geological Survey, 2020. (F5.pdf)
- [5] Agarwal, P. K. et al. *I/O-Efficient Algorithms for Computing Contour Maps on Terrains*. Duke University, 2010. (F6.pdf)
- [6] Kwan, M.-P.; Lee, J. *Geovisualization of Human Activity Patterns Using 3D GIS*. Oxford University Press, 2003. (F7.pdf)