Python Científico

Andre Nepomuceno

October 4, 2021

Exercícios - Matplotlib Avançado

Os arquivos de dados estão disponíveis AQUI.

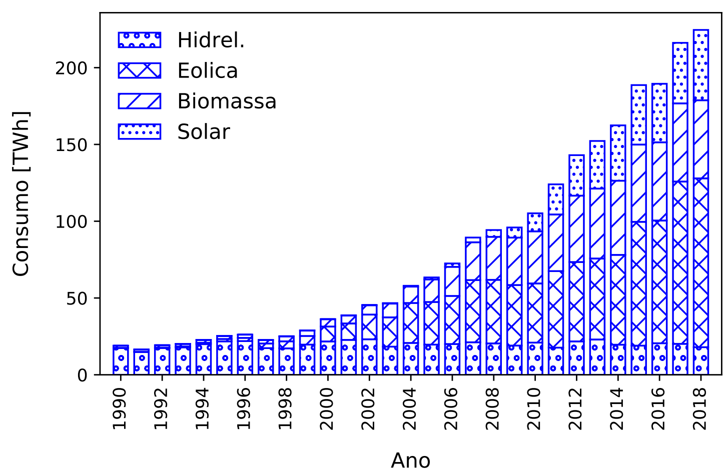
1.1 A radiância espectral de um corpo negro a temperatura T (em Kelvin) em função do comprimento de onda λ é dada pela lei de Planck:

$$B(\lambda) = \frac{2hc^2}{\lambda^5}\frac{1}{\exp(hc/\lambda k_B T) - 1}.$$

Escreva um programa para fazer o gráfico de $B(\lambda)$ entre $\lambda = 100$ e $\lambda = 4000$ nm para três valores de T: 3000 K, 4000 k e 5000 K. Use NumPy para calcular $B(\lambda)$ e modifique o range do eixo x para (0,4000). Acrescente legendas ao gráfico e altere o tamanho padrão dos *labels* dos eixos. Valores das constantes: $h = 6.626$ x 10^{-34} Js, c = 2.998 x 10^8 m/s e $k_B = 1.381$ x 10^{-23} J/K.

1.2 Os arquivos `pib_paises.tsv`, `imc_homens.tsv`, `populacao_total.tsv` e `continentes.tsv` contém, respectivamente, o PIB dos países, o índice de massa corporal dos homens (em kg/m^2), a população de cada país e o continente ao qual o país pertence. Utilize esses dados para produzir um gráfico de dispersão (*scatter plot*) do IMC *versus* PIB, com o tamanho dos pontos representando a população, e as cores os continentes. Para melhor visualização, normalize o tamanho da população de cada país para 2000 pts^2 por bilhão de habitantes, (pop/1.e9)*2000, e utilize escala log para o eixo do PIB (`ax.set_xscale('log')`). Perceba que nesses arquivos existem dados perdidos. Sugestão: use um dicionário para associar cores ao continentes.

1.3 A função `pyplot.bar()` contém, detre outros, os argumentos `bottom`, que dá a coordenada y da **base** da barra, `height`, que é a altura da barra, e `hatch`, que são opções para hachurar a barra (opções disponíveis: ’/’, ’; ’|’, ’-’, ’+’, ’x’, ’o’, ’O’, ’.’, ’*’). Utilize essas opções para produzir um gráfico de barras das diferentes fontes de energia utilizas na Alemanha entre 1990 e 2018 com os dados disponíveis no arquivo `fontes_energia_alemanha.txt`. Para produzir a legenda, crie um lista que guardará as informações do gráfico de barra de cada fonte, uma lista com o nome das fontes, e use o comando `ax.legend(barra, fontes)`. Você deve produzir um plot semelhante ao da figura abaixo. Os *labels* do eixo x são rotacionados com o comando `plt.xticks(rotation=90)`.



1.4 Faça um gráfico polar, no intervalo $0 \leq \theta \leq 24\pi$, da curva da borboleta, dada pela seguinte equação:

$$r = e^{\operatorname{sen}\theta} - 2\cos 4\theta + \operatorname{sen}^5\left(\frac{2\,\theta - \pi}{24}\right)$$

1.5 Suponha que uma pedra caia numa piscina, produzindo ondas que se propagam a partir do ponto de impacto. Esse sistema pode ser modelado como ondas senoidais se propagando em círculos. Se o centro do círculo tem coordenadas (x_1, y_1), então a distância r_1 do centro a um ponto x, y é

$$r_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}$$

e o descolamento da onda pode ser escrito como

$$\psi_1 = \psi_0 \operatorname{sen} k r_1,$$

onde ψ_0 é a amplitude e k o número de onda, relacionado ao comprimento de onda λ por $k = 2\pi/\lambda$. Suponha que uma outra pedra caia na piscina no ponto x_2, y_2, produzindo também ondas circulares de mesma amplitude e comprimento de onda:

$$\psi_2 = \psi_0 \operatorname{sen} k r_2, \quad \text{com} \quad r_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2}.$$

A onda resultante da superposição é dada por

$$\psi(x, y) = \psi_0 \text{sen}\, kr_1 + \psi_0 \text{sen}\, kr_2.$$

Sendo $\lambda = 5$ cm, $\psi_0 = 1$ cm e assumindo que os centros dos círculos estão afastados por 20 cm, escreva um programa em Python para calcular $\psi(x, y)$ e use a função `plt.imshow()` para obter uma imagem do padrão de interferência das ondas numa área de 1 m^2 da piscina.

1.6 O conjunto de Mandelbrot é um fractal definido em termos de números complexos como descrito a seguir. Considere a equação

$$z' = z^2 + c$$

onde z é um número complexo e c uma *constante* complexa. Para um **dado valor** de c, começamos com z $= 0$ e iteramos a equação acima, ou seja, cada novo valor z' é reintroduzido na equação como o valor de z e o cálculo é repetido. Se o módulo do valor resultante $|z'|$ é maior do que 2, então o ponto c em questão, no plano complexo, **não** está no conjunto de Mandelbrot; caso contrário, está. Em princípio, para saber se um ponto c pertence ao conjunto, teríamos que iterar infinitas vezes para garantir que sempre $|z'| < 2$. Na prática, 100 iterações são suficientes, e se ao final $|z'| < 2$, temos um ponto pertencente ao conjunto. Escreva um programa para obter uma imagem do conjunto de Mandelbrot para os valores de $c = x + iy$ numa grade $N \times N$ na região $-2 \leq x \leq 2$ e $-2 \leq y \leq 2$. Os pontos de dentro do conjunto devem ser escuros, e os fora, claros. Para testar seu programa, comece com valor não muito grande de N, por exemplo, $N = 100$. Quando o programa estiver funcionando, aumente N para 500 ou 1000. Sugestões: 1) para obter a imagem, utilize a função `plt.contourf()` com a opções `plt.contourf(X,Y,Z,cmap = 'Greys')`. 2) Para melhor visualização, escolha os limites `plt.xlim(-2,1)`, `plt.ylim(-1.1,1.1)`. 3) Não confunda o array binário `Z` da entrada da função `plt.contourf(X,Y,Z)` com o número complexo z. Para mais detalhes sobre o conjunto de Mandelbrot, veja este Link.

1.7 Modifique ligeiramente o programa do exercício anterior para que o array `Z` na função `plt.contourf(Y,X,Z)` guarde o número de iterações do *loop* antes de $|z'| > 2$, ou o número máximo de interações se $|z'|$ nunca exceder 2. Utilize agora as opções `cmap = 'hot'` ou `cmap = 'jet'`, e contemple a beleza da figura formada.