Física Computacional I

Problemas para o capítulo 3

Problema 3.1: Plotando dados experimentais

No SIGAA pode ser encontrado um arquivo chamado `sunspots.txt` que contém o número observado de manchas solares para cada mês desde Janeiro de 1749. O arquivo contém duas colunas de números, a primeira sendo o mês e a segunda sendo o número de manchas solares.

a) Escreva um programa que lê os dados e faz um gráfico do número de manchas solares em função do tempo.

b) Modifique seu programa para mostrar apenas os primeiros 1000 pontos no gráfico.

c) Modifique seu programa ainda mais para calcular e plotar a média móvel dos dados, definida como

$$Y_k = \frac{1}{2r} \sum_{m=-r}^{r} y_{k+m}\,,$$

onde $r = 5$ neste caso (e os y_k são os números de manchas solares). Faça o programa plotar tanto os dados originais como a média móvel no mesmo gráfico, novamente mostrando apenas os 1000 primeiros pontos.

Problema 3.2: Plotando curvas

Apesar da função `plot` ser definida primariamente para gráficos do tipo xy, ela pode ser adaptada para outros tipo de plotagem também.

a) Faça um gráfico da curva *deltóide*, que é definida parametricamente pelas equações

$$x = 2\cos\theta + \cos 2\theta, \qquad y = 2\sin\theta - \sin 2\theta,$$

onde $0 \leq \theta < 2\pi$. Pegue um conjunto de valores de θ entre zero e 2π e calcule x e y para cada uma das equações acima, e em seguida plote y em função de x.

b) Levando esta abordagem um passo adiante, é possível fazer um gráfico polar $r = f(\theta)$ de uma função f calculando r para uma coleção de valores de θ e então convertendo r e θ para coordenadas cartesianas usando as equações padrão $x = r\cos\theta$, $y = r\sin\theta$. Use este método para fazer um gráfico da espiral de Galileu, $r = \theta^2$ para $0 \leq \theta \leq 10\pi$.

c) Usando o mesmo método, faça um gráfico polar da "função de Fey"

$$r = \mathrm{e}^{\cos\theta} - 2\cos 4\theta + \sin^5 \frac{\theta}{12}$$

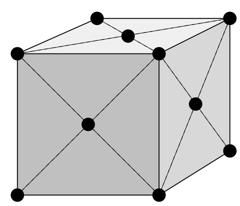
usando $0 \leq \theta \leq 24\pi$.

Problema 3.3: Há no SIGAA um arquivo chamado `stm.txt`, que contém uma grade de valores contendo medidas feitas em uma superfície (111) de silício usando um microscópio de tunelamento por varredura (STM). Um microscópio de tunelamento por varredura (STM) é um dispositivo que mede a forma da superfície em nível atômico, acompanhando o movimento de uma ponta aguda sobre uma superfície e medindo a corrente de tunelamento quântico em função da posição. O resultado final é uma grade de valores que representa a altura da superfície e o arquivo `stm.txt` contém uma destas grades de valores. Escreva um programa que lê os dados contidos no arquivo e faz um gráfico de densidade dos valores. Use as várias opções descritas no arquivo "Graficos.pdf" para preparar uma figura que mostra a estrutura da superfície de silício claramente.

Problema 3.4: Escreva um programa que faça o seguinte:

a) Um cristal de cloreto de sódio tem átomos de sódio e cloro arranjados em uma rede cúbica, só que os átomos de sódio e cloro se alternam na rede, de modo que cada sódio está cercado por seis cloros e cada cloro está cercado por seis sódios. Crie uma visualização da rede de cloreto de sódio usando duas cores diferentes para representar os dois tipos de átomos.

b) A rede cúbica de face centrada (fcc), que é a rede mais comum dentre os cristais que ocorrem naturalmente, consiste de uma rede cúbica com átomos posicionados não só nos vértices de cada cubo como também no centro de cada face:



Crie uma visualização de uma rede fcc com uma única espécie de átomo (como a que ocorre no ferro metálico, por exemplo).

Problema 3.5: Visualização do sistema solar

Os seis planetas mais internos do sistema solar giram em torno do Sol em órbitas mais ou menos circulares, que se localizam aproximadamente no mesmo plano. Aqui estão alguns parâmetros básicos:

Objeto	Raio do objeto (km)	Raio da órbita (milhões de km)	Período da órbita (dias)
Mercúrio	2440	57.9	88.0
Venus	6052	108.2	224.7
Terra	6371	149.6	365.3
Marte	3386	227.9	687.0
Júpiter	69173	778.5	4331.6
Saturno	57316	1433.4	10759.2
Sol	695500	–	–

Usando as funções fornecidas pelo pacote `vpython`, crie uma animação do sistema solar que mostre o seguinte:

a) O Sol e os planetas como esferas em suas posições aproximadas e com tamanhos proporcionais ao seu tamanho real. Como o raio dos planetas é pequeno comparado com as distâncias entre eles, represente os planetas usando esferas com um raio c_1 vezes maior do que os valores corretos, para que eles possam ser vistos claramente. Encontre um bom valor de c_1 que torne os planetas visíveis. Você também tem que encontrar um bom raio para o Sol. Escolha um valor que dê uma boa visualização. (Usar a mesma escala para o raio do Sol e dos planetas não funciona bem, porque o sol fica excessivamente grande. Use um valor que funcione.) Para ter realismo adicional, use cores diferentes para as esferas. Por exemplo, use azul para a Terra e amarelo para o Sol.

b) O movimento dos planetas em torno do Sol (fazendo as esferas que representam os planetas se mover). Por motivos práticos, gere uma animação que roda por um fator c_2 mais rápido que o tempo real. Encontre um bom valor de c_2 que torne o movimento das órbitas visível, mas não muito rápido. Faça uso da função `rate` para que a animação ande de maneira suave.

Dica: Você pode achar útil armazenar as variáveis das esferas que representam os planetas usando listas ou vetores.