UNIVERDADE FEDERAL DO PARÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM GEOFÍSICA
Prof. Cícero Régis
FORTRAN

1 – Logaritmos

Se conhecemos o logaritmo de um número **N** na base **b**, podemos calcular o seu logaritmo na base **a** através da fórmula.

$$\log_a(x)=\frac{\log_b(x)}{\log_b(a)}$$

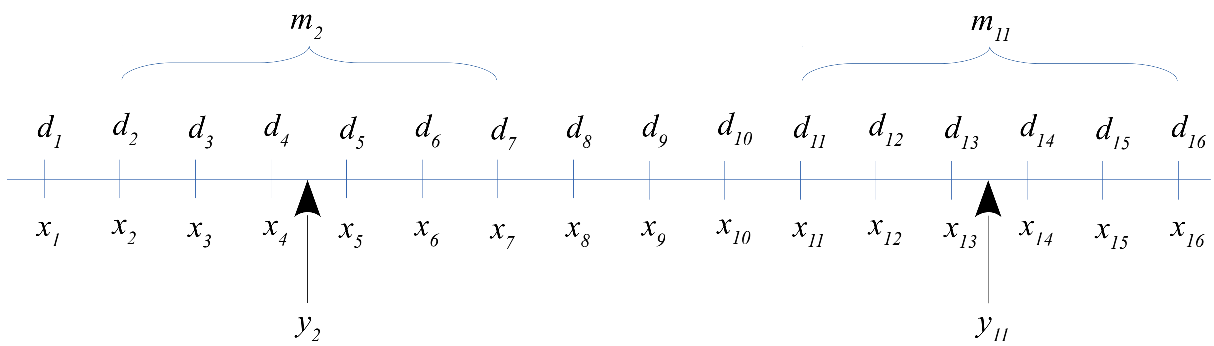
Escreva uma função que calcule o logaritmo de um número ***x*** em uma base ***a***, qualquer.

2 – Média móvel.

Considere um vetor de dados ***d*** de tamanho total N, no qual os dados são contaminados com ruído aleatório de média zero. Cada valor deste vetor está associado a uma coordenada *x*. O processo de suavização conhecido como "média móvel" é uma espécie de filtragem dos dados pelo cálculo dos valores médios de uma sequência de sub-conjuntos do vetor. O processo consiste nos seguintes passsos:

- definir o tamanho de uma janela *w*, ou seja, o tamanho do sub-conjunto do vetor;
- calcular a média m_1 dos valores dos dados no intervalo da janela, começando da primeira posição do vetor original;
- atribuir o valor da média calculado no passo 2 à coordenada (y_1) do centro do intervalo e salvar estes dois valores: o da média e o da coordenada;
- repetir os passos 2 e 3 para o intervalo começando na segunda posição do vetor original, depois começando da terceira posição, e assim por diante até que a janela móvel atinja o fim do vetor;

Por exemplo, na figura abaixo estão ilustradas duas posições de uma janela *w* de largura 6, em um conjunto de 16 valores. A primeira posição gera o segundo valor (m_2) em uma sequência de médias móveis, o qual é atribuído à coordenada y_2. A segunda posição da janela de filtragem ilustrada gera o décimo-primeiro e último valor de média móvel, gerando o par (y_{11}, m_{11}).



Nesta tarefa você irá programar a suavização pela média móvel, Para isso, realize o que se pede nos seguintes passos:

1- Escreva uma `function` para calcular a média de um dado vetor de dados reais. A `function` deve ter como entrada apenas o vetor de dados e o número de elementos do vetor.

2- Escreva um programa para ler os dados do arquivo `dados_ruido.dat`. Este arquivo contém duas colunas de valores. Na primeira estão as coordenadas *x* e na segunda estão os valores dos dados *d* contaminados com ruído, num total de 1257 pontos igualmente espaçados.

3- Defina um valor para a largura *w* da janela de filtragem e escreva o código para calcular a

sequência de valores filtrados juntamente com suas coordenadas, usando a `function` que você criou para calcular as médias, tendo como vetor de entrada apenas a parte do vetor de dados abrangida pela janela de filtragem em cada posição. Experimente com uma janela de 50 pontos.
4- Crie um novo arquivo chamado `dados_filtrados.dat`, contendo os valores de coordenadas *y* e de valores filtrados *m*, que você calculou.
5- Para observar o resultado, construa no MATLAB, um gráfico mostrando os dados suavizados juntamente com os dados originais.
6- Observe o resultado da suavização com diferentes tamanhos de janelas.

3 – Lançamento de um dado

Escreva uma função para simular o lançamento de um dado, usando a sub-rotina implícita RANDOM_NUMBER. A função, do tipo `integer`, tem lista de argumentos vazia e sua saída é um número inteiro entre 1 e 6. Teste sua função para verificar se os seis números tem a mesma probabilidade no lançamento do dado: execute a função alguns milhares vezes, conte as ocorrências de cada número e determine a probabilidade de cada um. Se seu dado não é "viciado" cada número tem que aparecer aproximadamente 1/6 das vezes.

4 – Craps

Neste exercício você vai usar sua function de lançamento de dado para programar uma versão simples do jogo chamado CRAPS.

Craps é um jogo de dados muito comum nos cassinos (desculpem, mas não sei o nome desse jogo em português). Nele um jogador joga contra a "casa". As regras são as seguintes: O jogador lança dois dados. Se o resultado dos dados somar 7 ou 11 na primeira jogada, o jogador ganha. Se a soma for 2, 3 ou 12 na primeira jogada ("craps") o jogador perde, ou seja, ganha a casa. Se a soma for 4, 5, 6, 8, 9 ou 10 na primeira jogada, então aquele valor se torna o "ponto" do jogador. Para ganhar, o jogador deve continuar lançando os dados até que seu ponto apareça novamente. O jogador perde se aparecer um 7 antes de aparecer seu ponto novamente.
a) Implemente as regras do jogo em um programa chamado craps, usando sua função de simulação do lançamento do dado. O programa deve terminar exibindo na tela a frase "O jogador ganhou!" ou "O jogador perde."
b) Modifique o programa do jogo de dados para incluir apostas. Transforme o programa que joga uma partida de "craps" em uma `function`. Inicialize a variável **deposito** com 1000 reais. Peça ao jogador para fazer uma aposta. Use um laço while para verificar que a **aposta** é menor ou igual ao **deposito** e, se não for, peça ao jogador para apostar outro valor até um valor de **aposta** válido seja escrito. Feita a aposta, execute o jogo uma vez. Se o jogador ganhar, some ao **deposito** o valor da **aposta** e mostre o novo valor do **deposito**. Se o jogador perder, subtraia do **deposito** o valor da **aposta,** mostre o novo valor do **deposito** e verifique se o **deposito** se tornou zero e se o for exiba a mensagem "Lamento, você faliu."

5- Integração numérica com trapézios

Neste problema você irá implementar um processo para acelerar a convergência da rotina de integração numérica usando trapézios, que você já programou.

O processo é baseado no fato de que o termo principal do erro cometido ao estimar a integral com N trapézios é uma função de $(1/N^2)$ e de termos subsequentes com expoentes pares de N. Isto sugere a seguinte análise: Suponha que avaliamos a integral com N intervalos, obtendo um resultado S_N, e novamente com $2N$ intervalos, obtendo o resultado S_{2N}. O termo principal do erro na segunda avaliação será ¼ do valor daquele na primeira avaliação. Então a combinação

$$S = \frac{4}{3}S_{2N} - \frac{1}{3}S_N$$

irá cancelar o termo principal do erro. Como não há termo de erro de ordem $1/N^3$, o erro restante será de ordem $1/N^4$.

Programe sua subrotina de integração aplicando a fórmula nos valores de duas avaliações seguidas e testando a convergência no resultado S. Verifique que agora são necessários menos intervalos para atingir a convergência desejada.

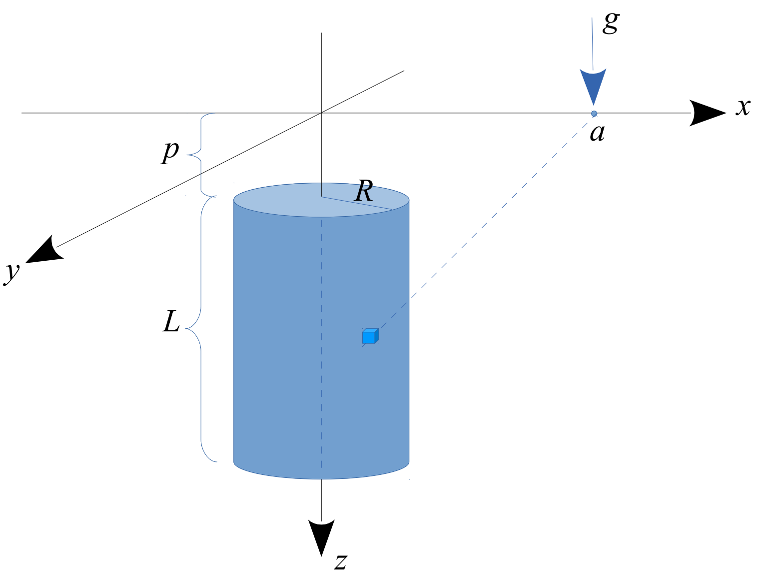
A análise que leva à formula acima está no livro Numerical Recipes: The Art of Scientific Programming, cujas edições antigas podem ser baixadas livremente na página

http://numerical.recipes/com/storefront.html

6- Gravimetria

Nesta questão, você irá modelar um levantamento gravimétrico sobre um alvo que simula uma intrusão de formato cilíndrico em um meio encaixante uniforme.

Usaremos o sistema de coordenadas mostrado na figura:



Você deve calcular a componente vertical da variação do campo gravimétrico devido a um cilindro vertical de raio R, na posição a sobre o eixo x. A densidade do meio encaixante (ρ_0) é uniforme. Em um ponto distante do cilindro a aceleração da gravidade vertical é g_0. Tomando este valor como referência para o cálculo da variação $\Delta g = g - g_0$ do campo na superfície, temos:

$$\Delta g = G\Delta\rho \int_{-R}^{R} \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \int_{p}^{p+L} \frac{z}{[(a-x)^2+y^2+z^2]^{3/2}} dz\, dy\, dx$$

Esta integral é resolvida em três etapas:

- Primeiro, a integral em *dz*:

$$I_z = \int_p^{p+L} \frac{z}{[(a-x)^2+y^2+z^2]^{3/2}} dz = \frac{1}{[(a-x)^2+y^2+p^2]^{1/2}} - \frac{1}{[(a-x)^2+y^2+(p+L)^2]^{1/2}} .$$

- Depois, a integral em *dy*:

$$I_y = \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} I_z(x,y) \; dy$$

$$I_y = \ln\left(\frac{(L+p)^2+(x-a)^2}{\left[\sqrt{R^2-x^2}+\sqrt{(L+p)^2+R^2+a^2-2xa}\right]^2}\right) - \ln\left(\frac{p^2+(x-a)^2}{\left[\sqrt{R^2-x^2}+\sqrt{p^2+R^2+a^2-2xa}\right]^2}\right)$$

- Finalmente, a integral em *dx*,

$$I_x = \int_{-R}^{R} I_y(x) \; dx ,$$

que deve ser calculada numericamente.

Sua tarefa é escrever um programa para simular uma linha de dados gravimétricos gerados pelo modelo de intrusão cilíndrica, acompanhando o eixo *x*. Faça seguindo as seguintes instruções:

- Construa uma function que recebe a variável de integração (*x*) e retorne o valor da função I_y. Esta function deve usar os valores dos parâmetros geométricos do modelo (*R, p, L, a*) como variáveis globais, que devem ser declaradas no mesmo módulo que contém a function.
- Use sua rotina para calcular integrais com trapézios para avaliar a integral I_x, fazendo subdivisões do intervalo de integração. Faça o teste de convergência com um valor de 10^{-4}.
- O programa principal deve ter um laço para variar a posição de medida (*a*) de -10*R* até 10*R*.
- Calcule os valores de Δg normalizados por $G\Delta\rho$, de modo que seu programa irá retornar apenas o valor da integral tripla.
- Salve os valores calculados e as coordenadas em um arquivo para ser lido pelo MATLAB para visualização.

Como exemplo, para um cilindro com *R* = 10, *L* = 100 e *p* = 1, a integral dá os seguintes valores:

a	I_x
0.0000	53.759
1.0000	53.604
5.0000	49.705
10.000	34.014
15.000	19.206
50.000	3.530