UNIVERSIDADE FEDERAL DO PARÁ
FACULDADE DE GEOFÍSICA

PROGRAMAÇÃO COMPUTACIONAL
Prof. Cícero Régis

1 Eliminação gaussiana

Agora que você já tem uma rotina para realizar a fatoração LU de uma matriz e uma para resolver um sistema de equações lineares, vamos utilizá-las para resolver outros problemas importantes usando eliminação gaussiana.

Inicialmente vamos considerar uma matriz quadrada $\mathbf{A}_{(N\times N)}$, que pode ser singular ou não, e um sistema de equações lineares $\mathbf{A}\,\mathbf{x} = \mathbf{b}$.

1.1 Eliminação com pivoteamento parcial

Construa uma rotina para realizar a eliminação na matriz e resolver o sistema de equações realizando trocas de linhas sempre que um número abaixo da diagonal principal for maior (em módulo) do que o elemento na diagonal. O maior número na coluna da diagonal principal para baixo passa a ser o pivô. Não esqueça de realizar a mesma permutação entre linhas no vetor **b**.

Aproveite e inclua a verificação sobre a singularidade da matriz. Se o maior número na coluna, da diagonal principal até a última linha for menor do que um certo número ϵ pequeno, pare o processo e retorne um código de erro para o programa principal. O valor do ϵ vai depender da precisão dos números reais com que você estiver trabalhando.

1.2 Fatoração LU com trocas de linha

Para programar a fatoração com pivoteamento parcial, o processo é semelhante ao caso anterior: primeiro realiza a troca de linhas, depois vai armazenando os multiplicadores nas posições correspondentes aos números eliminados. Entretanto, para que se possa resolver o sistema posteriormente é necessário guardar as informações sobre quais trocas foram realizadas, para que também possam ser feitas no vetor **b**. Faça da seguinte maneira:

- Crie um vetor de números inteiros **p** com a sequência de valores de 1 a N. Este vetor guardará as permutações feitas durante a fatoração.
- Implemente o processo de eliminação incluindo as permutações de linhas. Toda vez que realizar uma troca de linhas, troque também as posições correspondentes no vetor **p**. No final os valores dos elementos do vetor **p** indicam as posições corretas de cada elemento no vetor **b** original.
- Depois de cada troca, continue o processo normalmente, armazenando os multiplicadores nas posições correspondentes aos números que são eliminados.
- Na etapa de solução do sistema, use a informação no vetor de permutações para reorganizar o vetor que receberá a solução: se estiver fazendo em um vetor **x** diferente do vetor de entrada **b**, basta atribuir os elementos do **b** ao **x** usando a informação das permutações contidas no **p**, assim:

```
do i = 1, N
     x(i) = b(p(i))
end do
```

  e em seguida proceder com as etapas de substituição direta e reversa normalmente. Por outro lado, se estiver fazendo a saída no mesmo vetor **x** de entrada, programe a lógica

correta para realizar as permutações no vetor (esta frase também pode ser reformulada como: *dê seu jeito*!).

1.3 Cálculo do determinante

As operações de eliminação NÃO alteram o determinante da matriz. Por isso, o determinante da **U** é exatamente o mesmo da **A**. Sendo assim, para calcular o determinante da matriz **A** basta realizar a eliminação e calcular o produto de todos os pivôs.

Repare que se incluir trocas de linhas existe um detalhe a mais: cada troca de linha implica uma mudança no sinal do determinante. Sendo assim, é preciso incluir um contador para guardar o número total de trocas. Se for um número par basta calcular o produto dos pivôs; já se o número de trocas for ímpar o determinante da **A** será o negativo do produto dos pivôs.

1.4 Cálculo da matriz inversa

A inversa de uma matriz **A**, representada pelo símbolo $\mathbf{A^{-1}}$, é aquela que satisfaz a igualdade

$$\mathbf{A\,A^{-1} = A^{-1}A = I},$$

em que **I** é a matriz identidade.

Vamos implementar uma rotina para gerar a inversa de uma matriz usando eliminação gaussiana. Para começar, repare que na expressão $\mathbf{A\,A^{-1} = I}$ o produto da matriz **A** por cada coluna da inversa resulta numa coluna da matriz identidade. Por exemplo, se a primeira coluna da inversa é o vetor $\mathbf{c_1}$ e a segunda for o vetor $\mathbf{c_2}$, então

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{bmatrix} \begin{bmatrix} | \\ \mathbf{c_1} \\ | \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{bmatrix} \begin{bmatrix} | \\ \mathbf{c_2} \\ | \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix},$$

e assim por diante.

Isto nos dá a dica de que para calcular a inversa basta resolver N sistemas de equações com a mesma matriz de coeficientes, ou seja, você pode implementar usando as rotinas de que já dispõe.

Programe usando a decomposição LU, de modo que só é preciso fatorar a matriz uma única vez e depois realizar N etapas de substituição direta e reversa. Resolva com a rotina que gera a solução no mesmo vetor de entrada, para economizar memória. Você pode simplesmente chamar N vezes a rotina de solução que você já tem, cada vez dando de entrada uma coluna da identidade, ou programar uma rotina específica que realize as operações em todas as colunas da matriz identidade no mesmo laço.

Teste sua rotina com estas matrizes:

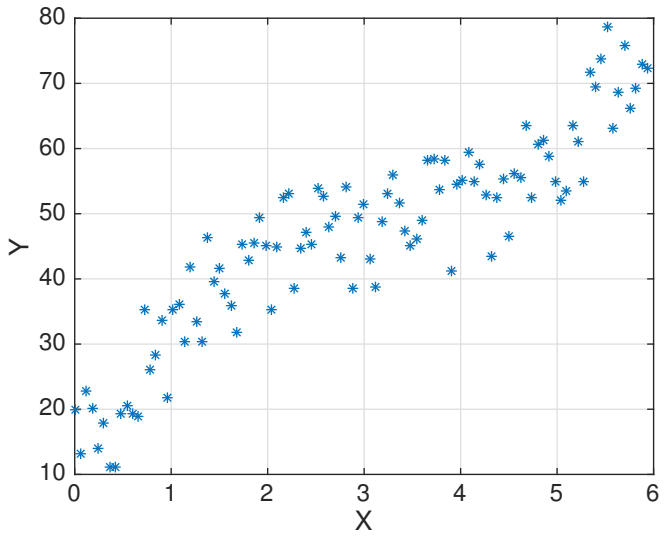
$$\mathbf{A} = \begin{bmatrix} 2 & 5 & 2 \\ 3 & 0 & 3 \\ -4 & 3 & -3 \end{bmatrix} \qquad \mathbf{A^{-1}} = \begin{bmatrix} 3/5 & -7/5 & -1 \\ 1/5 & -2/15 & 0 \\ -3/5 & 26/15 & 1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix} \qquad \mathbf{B^{-1}} = \begin{bmatrix} -3/2 & 5/2 & -2 \\ 1/2 & -1/2 & 1 \\ 1/2 & -1/2 & 0 \end{bmatrix}$$

Depois, teste com um sistema grande gerado aleatoriamente para verificar o tempo necessário para calcular a inversa.

2 Ajuste de dados com polinômios

O arquivo `dados_para_mq.dat` contém as coordenadas dos 100 pontos mostrados na figura.



Neste problema, você irá encontrar duas funções para ajustar estes pontos por mínimos quadrados: uma reta e um polinômio de terceiro grau. Depois determinará qual das duas proporciona o melhor ajuste. Para isto, construa um programa para realizar as seguintes tarefas:

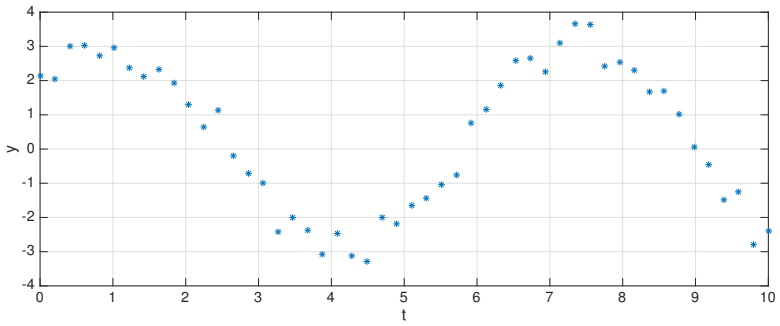
1. Ler o arquivo de dados.
2. Usar as rotinas que você já possui para calcular os coeficientes de um polinômio de grau g que ajusta os dados por mínimos quadrados.
3. Gerar um vetor δy contendo os valores das diferenças entre os dados originais e os valores calculados com o polinômio gerado no passo anterior nas mesmas abscissas.
4. Use sua `function` para calcular o desvio padrão dos valores no vetor δy, supondo que os desvios são aleatórios com distribuição uniforme.
5. Escreva um arquivo de saída contendo os valores dos coeficientes do polinômio e o valor do desvio padrão do vetor δy. Dê um nome para o arquivo de saída que indique o grau do polinômio usado.

Use o programa que você criou para testar dois casos diferentes: Primeiro ajuste os dados com uma reta e crie o arquivo de saída chamado `P1.out`. Em seguida, faça o ajuste com um polinômio de terceiro grau, criando o arquivo de saída com o nome `P3.out`.

Observe qual das duas funções melhor ajusta os dados.

3 Ajuste de dados com uma senóide

Em alguns problemas, os dados observados podem ser descritos por uma função senoidal. Imagine que os dados mostrados na figura representam um conjunto de medidas feitas em um intervalo de tempo.



O problema é ajustar este conjunto de dados com uma senóide na forma

$$y(t) = Y_0 \,\mathrm{sen}(\omega t + \phi).$$

Acontece que este problema é não linear e não dá para aplicar mínimos quadrados para estimar todos os parâmetros da função. Porém, se a frequência angular ω for conhecida, o problema se torna linear da seguinte maneira: Primeiro, reescreva a função aplicando a fórmula para o seno de uma soma:

$$\begin{aligned} y(t) &= Y_0 \Big[\mathrm{sen}(\omega t)\cos(\phi) + \mathrm{sen}(\phi)\cos(\omega t)\Big], \\ y(t) &= Y_0 \cos(\phi)\mathrm{sen}(\omega t) + Y_0 \mathrm{sen}(\phi)\cos(\omega t). \end{aligned}$$

Agora, defina os parâmetros p_1 e p_2:

$$\begin{aligned} p_1 &= Y_0 \cos(\phi), \\ p_2 &= Y_0 \,\mathrm{sen}(\phi). \end{aligned}$$

Então, se existem N pontos para ser ajustados podemos montar um sistema linear fazendo

$$\begin{bmatrix} \mathrm{sen}(\omega t_1) & \cos(\omega t_1) \\ \mathrm{sen}(\omega t_2) & \cos(\omega t_2) \\ \vdots & \vdots \\ \mathrm{sen}(\omega t_N) & \cos(\omega t_N) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_N) \end{bmatrix}.$$

Agora é só encontrar a solução de mínimos quadrados para este sistema sobredeterminado e depois calcular Y_0 e ϕ a partir dos valores de p_1 e p_2:

$$Y_0 = \sqrt{p_1^2 + p_2^2}\,, \qquad\qquad \phi = \arctan\left(\frac{p_2}{p_1}\right).$$

Escreva um programa para realizar esta tarefa e teste com os dados no arquivo `dados_seno.dat`, que estão mostrados na figura. Para estes dados o valor da frequência angular é $\omega = 1$. Não esqueça que a função intrínseca para o arco tangente no Fortran é `atan`.

Construa o gráfico da função obtida com os valores encontrados para ver o ajuste conseguido. Os dados foram gerados com os valores $Y_0 = 3, \phi = 0.5$. Verifique se os valores estimados estão próximos destes.

4 Ajuste de dados gravimétricos

4.1 Problema direto: simulação de um levantamento gravimétrico sobre alvos cilíndricos

A primeira tarefa para este estudo é criar uma função ou subrotina para simular uma medida do campo gravimétrico de um cilindro vertical enterrado em um meio homogêneo. O dado simulado é a variação Δg do campo gravimétrico em relação ao do meio encaixante homogêneo. Vamos simplificar fazendo Δg normalizado pela constante de gravitação universal G e representando o dado simplesmente por g. O contraste de densidade entre o material do cilindro e o do meio é representado por $\Delta\rho$, de modo que se a densidade do cilindro for maior do que a do meio, $\Delta\rho$ será um número positivo, se o cilindro for menos denso do que o meio o contraste será um numero negativo e se as densidades forem iguais, $\Delta\rho$ será zero.

Você já tem um programa que realiza este cálculo considerando que o eixo do cilindro é vertical e coincidente com o eixo z. Agora, construa um subprograma que tenha como uma das entradas a coordenada $x_c \neq 0$ do eixo do cilindro, ainda vertical e no plano (x, z), de modo que você possa simular uma linha de um levantamento gravimétrico sobre o cilindro localizado em qualquer coordenada. As entradas deste subprograma são o raio (R) e o comprimento (L) do cilindro, a profundidade $(p > 0)$ do topo, a coordenada x_c do centro do cilindro e a coordenada x do ponto de medida.

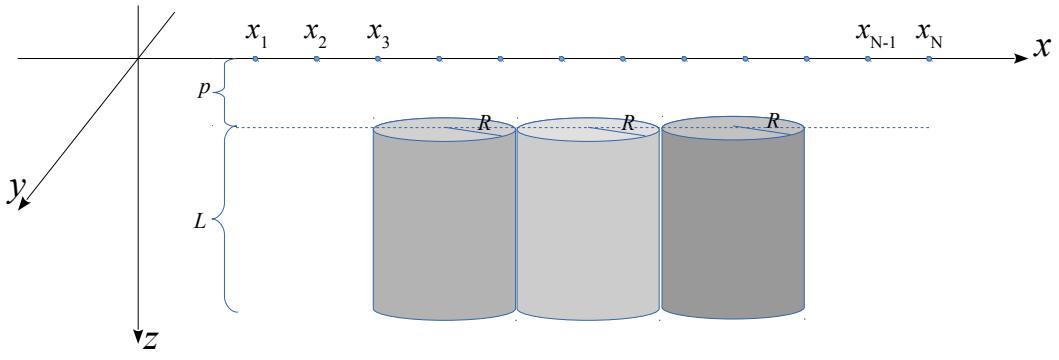
Conforme você já sabe, o problema exige a avaliação de uma integral tripla no volume do cilindro. Dados os parâmetros geométricos do modelo, o valor da integral é uma função $f(x)$ da coordenada x do ponto de observação. O dado simulado será, então

$$g = \frac{\Delta g(x)}{G} = \Delta\rho\, f(x)$$

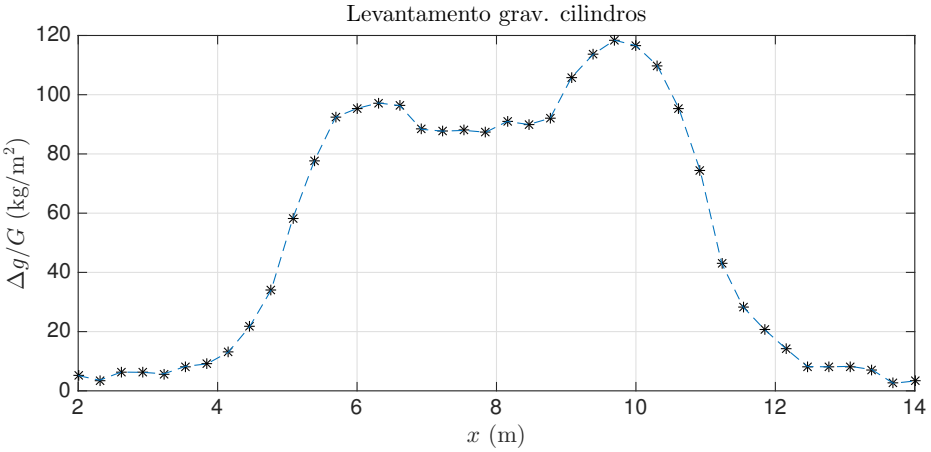
Você poderá usar sua função ou subrotina para simular um levantamento passando sobre vários cilindros diferentes, simplesmente executando o subprograma várias vezes, com os parâmetros de cada cilindro e somando os valores calculados na mesma coordenada. Preste atenção para não posicionar dois cilindros com uma distância entre seus centros menor do que $2R$, para que não haja sobreposição.

4.2 Problema inverso: determinação de densidades a partir do dado gravimétrico

Três tanques cilíndricos iguais, com raio $R = 1$ m e comprimento $L = 2$ m, foram usados para armazenar três materiais de densidades ρ_j diferentes. Os tanques estão dispostos lado a lado, com os eixos nas coordenadas 6 m, 8 m e 10 m, e enterrados na mesma profundidade $p = 10$ cm, como ilustrado na figura



Para determinar as densidades dos materiais sem romper os tanques, é realizado um levantamento gravimétrico em uma linha passando sobre os centros dos cilindros. Os valores de variação Δg do campo gravimétrico medido em relação àquele gerado pelo meio encaixante são normalizados pela constante de gravitação universal G e listados no arquivo `grav_cilindros.dat`. O resultado, incluindo um certo nível de ruído, está mostrado no gráfico.



Sua tarefa é gerar, a partir dos dados medidos, uma estimativa dos contrastes entre as densidades dos três cilindros e a do meio encaixante. Para isto, escreva um programa que use o subprograma que você criou na primeira parte do problema para realizar os seguintes passos:

- Monte um sistema de equações lineares tendo como coeficientes os contrastes de densidades $\Delta\rho_1$, $\Delta\rho_2$ e $\Delta\rho_3$ a partir das observações, ou seja: se a função que dá o campo vertical do cilindro j for f_j, então um dado sintético g na coordenada x_i é escrita como

$$g(x_i) = \Delta\rho_1\, f_1(x_i) + \Delta\rho_2\, f_2(x_i) + \Delta\rho_3\, f_3(x_i),$$

  de modo que para N observações o sistema será

$$\begin{aligned}
\Delta\rho_1\, f_1(x_1) + \Delta\rho_2\, f_2(x_1) + \Delta\rho_3\, f_3(x_1) &= g(x_1) \\
\Delta\rho_1\, f_1(x_2) + \Delta\rho_2\, f_2(x_2) + \Delta\rho_3\, f_3(x_2) &= g(x_2) \\
&\vdots \\
\Delta\rho_1\, f_1(x_N) + \Delta\rho_2\, f_2(x_N) + \Delta\rho_3\, f_3(x_N) &= g(x_N)
\end{aligned}$$

  ou, em notação matricial:

$$\begin{bmatrix} f_1(x_1) & f_2(x_1) & f_3(x_1) \\ f_1(x_2) & f_2(x_2) & f_3(x_2) \\ \vdots & \vdots & \vdots \\ f_1(x_N) & f_2(x_N) & f_3(x_N) \end{bmatrix} \begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \end{bmatrix} = \begin{bmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_N) \end{bmatrix}$$

- Encontre a solução de mínimos quadrados para este sistema sobredeterminado ($N \times 3$).
- Como sempre, construa o gráfico dos dados gerados por sua solução mostrando o ajuste com os dados observados.

Os dados foram gerados com os seguintes valores de contrastes de densidade, em kg/m^3:

$$\begin{aligned}
\Delta\rho_1 &= 20, \\
\Delta\rho_2 &= 15, \\
\Delta\rho_3 &= 25.
\end{aligned}$$

Verifique se os valores que você encontrou estão próximos destes.