SME 0300 – Cálculo Numérico

9 de maio de 2023

Exercício 3: Sistemas de equações não-lineares e cálculo de autovalores (entrega: 17/05/23, apresente somente os códigos no formato .m)

1. Seja a função vetorial com as seguintes componentes:

$$\begin{cases} f_1(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 - 1 \\ f_2(x_1, x_2, x_3) = \sin(x_1) + x_2^3 + x_3 \\ f_3(x_1, x_2, x_3) = x_1^2 - 2x_2 + x_3 \end{cases} . \tag{1}$$

   (a) escreva uma função em Matlab/Octave que, dado um vetor de entrada $\vec{x}$, retorne um vetor de saída $\vec{f}$ correspondente à função acima.

   (b) encontre a solução do sistema $\vec{f}(\vec{x}) = \vec{0}$ usando a função do Matlab/Octave `sol = fsolve(@fun,x0))`, onde `sol` é a solução do sistema, `fun` é a função que você criou no item (a) e `x0` é o chute inicial, para `x0 = [1 1 1]`.

   (c) calcule o resíduo do resultado anterior (calculando a norma do vetor resultado de `fun` aplicada à solução `sol`).

   (d) escreva sua própria função em Matlab/Octave para a solução dos itens (b) e (c) utilizando o método de Newton, com critério de parada de `norm(deltax) < 1E-8`. Para isso, crie uma função `Jac` que retorne a matriz Jacobiana do problema para um dado $\vec{x}$ de entrada, e utilize as duas funções da seguinte forma:
   `function x = nonlinNewton(fun,Jac,x0,tol,nmax)` (na definição da função do Método de Newton)
   `x = nonlinNewton(@fun,@Jac,x0,1E-8,1000)` (na execução da função do Método de Newton).

2. Escreva uma função em Matlab/Octave que calcule a decomposição QR de uma matriz $A \in \mathbb{R}^{n\times n}$ utilizando o método de Gram-Schmidt **clássico**.

3. Escreva uma função em Matlab/Octave que calcule os autovalores de uma matriz $A \in \mathbb{R}^{n\times n}$ simétrica utilizando a decomposição QR e a função de Gram-Schmidt anterior. Cheque se seu código está correto comparando-o com o obtido pelo comando `[e]=eig(A)`, para a matriz A simétrica abaixo.

```
A = [15 2 3 4; 2 5 7 5; 3 7 9 6; 4 5 6 7]
```