Computação II — Laboratório 7

Entregue todos os métodos em um arquivo chamado `lab7.py`. Critérios de avaliação: Principalmente (75%) o código funciona, os métodos fazem o que foi pedido, parâmetros de entrada e valores de retorno são corretos e em forma correta. Adicionalmente (25%) o código é legível, eficaz, o mais simples possível.

Daqui para adiante `np` significa `numpy` – a biblioteca a ser usada nesta lista, portanto: `import numpy as np`.

1. (3 pontos) Cria a função chamada `senoPositivo` com parâmetros de entrada `a,b,n`. A função deve ciar um `np.ndarray x` de `n` pontos do intervalo `[a,b]` e retornar os elementos de `x` cujos seno é positivo. É proibido usar os laçõs de repetição. Exemplo de chamada:

```
>>> senoPositivo(0,3*np.pi,6)
array([1.88495559, 7.53982237, 9.42477796])
```

2. (3 pontos) Cria a função chamada `polinomio` cujo parâmetro de entrada é um `np.ndarray` unidimensional (um vetor) e um número real `z`. O valor de retorno é volor do polinômio $p(x)$ no ponto `z`, onde

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x = \sum_{k=0}^{N} a_k x^k$$

   e a_0, a_1, a_N são elementos do `np.ndarray`. Utilize so métodos da biblioteca `numpy` para calcular o polinômio (e.g., `np.cumprod`). É proibido usar os laçõs de repetição.

3. (4 pontos) Cria a função chamada `ortogonal` que recebe uma `np.ndarray` bidimensional (uma matriz) e retorna o valor booleno `True` caso a matriz é ortogonal e `False` caso a matriz não é ortogonal. Lembrando que a matriz é ortogonal se ela é quadrada e sua matriz transposta coincide com sua matriz inversa, iso é, uma matriz M é orgogonal se $M^\top M = I$, onde I é a matriz identidade. Dica: use os métodos `np.transpose` e `np.allclose`. Uns exemplos de chamada:

```
>>> ortogonal(np.array([1,2,3]))
False
>>> ortogonal(np.eye(N = 2, M =3))
False
>>> ortogonal(1/3*np.array([[2,-2,1],[1,2,2],[2,1,-2]]))
True
```