Entregar dia 25 de maio de 2012

O método de Gauss-Seidel para resolução de sistemas lineares: Este é um método iterativo, isto é, a partir de um chute inicial $\mathbf{x}^{(0)}$ vamos produzindo outras estimativas $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \ldots, \mathbf{x}^{(k)}$ até que a estimativa esteja "boa", para tanto a sequência de vetores $\{\mathbf{x}^{(i)}\}$ deve convergir!
No método de Gauss-Seidel, a nova estimativa é produzida a partir da estimativa anterior segundo a equação;

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j>i} a_{ij} x_j^{(k)} - \sum_{j<i} a_{ij} x_j^{(k+1)} \right), \quad i = 1, 2, \ldots, n.$$

onde $\mathbf{x}^{(k)} = (x_1^{(k)}, \cdots, x_n^{(k)})$, e $A = (a_{ij}$ é a matriz de coeficientes.
Um dos critérios para a convergência do método é quando a matriz de coeficientes A do sistema tem diagonal dominante, isto é, para cada linha i da matriz A devemos ter a desigualdade:

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$$

A convergência não chega, em geral, até a solução. Então precisamos avaliar a que distância estamos da solução para um critériode parada. Podemos fazer isso avaliando o resíduo $\mathbf{r}^{(k)} = A\mathbf{x}^{(k)} - \mathbf{b}$, a cada nova estimativa. Usando a norma euclidiana de $\mathbf{r}^{(k)}$ por exemplo.
No seu programa em Python devemos ter o seguinte:
1. Uma função *Leia_A(arquivo.txt)* que leia a matriz de coefientes de uma arquivo e retorne uma matriz quadrada.
2. Uma função que leia o vetor $\mathbf{b}$ compatível com a dimensão de A também de um arquivo.
3. Uma função booleana que retorna *True* se a matriz A for diagonalmente dominante.
4. Uma função *GaussSeidel(A,b, N, eps)* que resolve o sistema $A\mathbf{x} = \mathbf{b}$, usando o método de Gauss-Seidel até que a norma euclidiana do resíduo seja menos que *eps*, ou o número de iterações tenha ultrapassado *N*.