Instituto Tecnológico de Aeronáutica – ITA
Matemática Computacional – CCI-22
Laboratório 11 – Determinação de Autovalores e Autovetores

Professor: Marcos Ricardo Omena de Albuquerque Maximo

9 de junho de 2019

1 Tarefas

1.1 Implementação

Implementar as seguintes funções em MATLAB (cada uma em um arquivo .m separado):

1. `[v, lambda] = MetodoPotencias(A, x0, epsilon, maxIteracoes)`: determina o maior autovalor λ da matriz $\mathbf{A} \in \mathbb{R}^{N\times N}$ e seu autovetor normalizado associado $\mathbf{v} \in \mathbb{R}^N$, usando $\mathbf{x}_0 \in \mathbb{R}^N$ como chute inicial, ε como tolerância relativa e `maxIteracoes` como número máximo de iterações. Caso $\mathbf{x}_0$, ε ou `maxIteracoes` não sejam fornecidos, a função considera $\mathbf{x}_0 = [1, 1, \ldots, 1]^T$, $\varepsilon = 10^{-3}$ e `maxIteracoes` = 10000. Na k-ésima iteração, o critério de parada termina o método se:

$$\frac{\max_{1\leq i\leq N} \left| v_i^{(k)} - v_i^{(k-1)} \right|}{\max_{1\leq i\leq N} \left| v_i^{(k)} \right|} < \varepsilon \tag{1}$$

2. `[V, D] = MetodoJacobi(A, epsilon, maxIteracoes)`: determina todos os autovalores e autovetores da matriz simétrica $\mathbf{A} \in \mathbb{R}^{N\times N}$, usando ε como tolerância sobre os elementos fora da diagonal principal e `maxIteracoes` como número máximo de iterações. Caso ε ou `maxIteracoes` não sejam fornecidos, a função usa $\varepsilon = 10^{-3}$ e `maxIteracoes` = 10000. Os retornos da função são a matriz dos autovetores normalizados $\mathbf{V} \in \mathbb{R}^{N\times N}$ e uma matriz diagonal $\mathbf{D} \in \mathbf{R}^{N\times N}$ com os autovalores. O critério de parada termina o método quando:

$$\max_{1\leq i,j\leq N,\ i>j} |a_{i,j}| < \varepsilon \tag{2}$$

3. `[Q, R] = DecomposicaoQR(A)`: realiza decomposição QR na matriz $\mathbf{A} \in \mathbb{R}^{N\times N}$ de modo que $\mathbf{A} = \mathbf{Q}\cdot\mathbf{R}$, em que $\mathbf{Q} \in \mathbb{R}^{N\times N}$ é ortogonal e $\mathbf{R} \in \mathbb{R}^{N\times N}$ é triangular superior. Para realizar a decomposição, a função usa rotações de Givens.

4. `T = AlgoritmoQR(A, epsilon, maxIteracoes)`: realiza o algoritmo QR para determinação de autovalores da matriz quadrada $\mathbf{A} \in \mathbb{R}^{N\times N}$, utilizando ε como tolerância para os elementos abaixo da diagonal principal e `maxIteracoes` como número máximo de iterações. Caso ε ou `maxIteracoes` não sejam fornecidos, a função usa $\varepsilon = 10^{-3}$ e `maxIteracoes` $= 10000$. O retorno da função é a matriz triangular superior $\mathbf{T} \in \mathbb{R}^{N\times N}$, resultante do algoritmo QR e que possui os autovalores na diagonal principal. O critério de parada termina o método quando:

$$\max_{1\leq i,j\leq N,\ i>j} |a_{i,j}| < \varepsilon \tag{3}$$

1.2 Análise

1. Considere as seguintes matrizes quadradas (presentes no arquivo `analise1.mat`):

$$\mathbf{A}_1 = \begin{bmatrix} 3 & 0 & 1 \\ 2 & 2 & 2 \\ 4 & 2 & 5 \end{bmatrix} \tag{4}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 8 & 9 \\ 7 & 7 & 3 \\ 5 & 9 & 7 \end{bmatrix} \tag{5}$$

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 7,5 & 7 \\ 7,5 & 7 & 6 \\ 7 & 6 & 7 \end{bmatrix} \tag{6}$$

$$\mathbf{A}_4 = \begin{bmatrix} 2 & 2,5 & 8 & 3,5 \\ 2,5 & 9 & 7 & 7,5 \\ 8 & 7 & 6 & 1 \\ 3,5 & 7,5 & 1 & 6 \end{bmatrix} \tag{7}$$

$$\mathbf{A}_5 = \begin{bmatrix} 2 & 5 & 9 & 2 \\ 0 & 9 & 8 & 9 \\ 7 & 6 & 6 & 0 \\ 5 & 6 & 2 & 5 \end{bmatrix} \tag{8}$$

$$\mathbf{A}_6 = \begin{bmatrix} 2 & 3 & 8 & 6 \\ 3 & 2 & 3 & 3 \\ 8 & 3 & 8 & 5 \\ 6 & 3 & 5 & 4 \end{bmatrix} \tag{9}$$

Usando os métodos das potências e das potências inversas, determine os maiores e menores autovalores de cada uma dessas matrizes, respectivamente, assim como os autovetores normalizados (i.e. com norma unitária) associados. Então, utilize o método de Jacobi para determinar todos os autovalores e autovetores normalizados das matrizes que são simétricas. Também, utilize o algoritmo QR para determinar os autovalores de todas as matrizes. Finalmente, monte uma tabela comparando os resultados encontrados. Uma sugestão de modelo de tabela é apresentada na Tabela 1 (repetir uma tabela para os autovalores e outra para os autovetores). No

Método	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$
Potências						
Potências Inversas						
Jacobi						
QR						

Tabela 1: Modelo de tabela para autovalores ou autovetores da questão 1 da Análise.

caso de não ser possível aplicar o método de Jacobi, deixe a célula em questão em branco. Utilize $\varepsilon = 10^{-3}$ e `maxIteracoes` = 10000 para todos os métodos. Além disso, utilize $\mathbf{x}_0 = [1, 1, \ldots, 1]^T$ para os métodos de potências. Comente os resultados obtidos, comparando os valores de autovetores e autovalores obtidos pelos métodos.

2. Uma aplicação interessante de determinação numérica de autovalores é que podemos usar estes métodos para determinar todas as raízes de um polinômio sem a necessidade de prover estimativas iniciais para essas raízes para o método numérico. Inclusive, este é o método utilizado pela função `roots` do MATLAB. Assim, seja um polinômio definido por:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \tag{10}$$

Definimos a matriz companheira de $p(x)$ como:

$$\mathbf{P} = \begin{bmatrix} -a_{n-1}/a_n & -a_{n-2}/a_n & -a_{n-3}/a_n & \cdots & -a_1/a_n & -a_0/a_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{11}$$

O polinômio característico de $\mathbf{P}$ é:

$$p(\lambda) = \det(\mathbf{P} - \lambda I) \Rightarrow p(\lambda) = \lambda^n + \frac{a_{n-1}}{a_n}\lambda^{n-1} + \cdots + \frac{a_1}{a_n}\lambda + \frac{a_0}{a_n} \tag{12}$$

Portanto, os autovalores de $\mathbf{P}$ são iguais às raizes de $p(x)$! Assim, podemos transformar o problema de determinar todos as raízes de um polinômio para um problema de determinar todos os autovalores de uma matriz, problema para o qual há algoritmos eficientes conhecidos. No caso do polinômio ter apenas raízes reais, podemos usar o algoritmo QR conforme aprendido no curso. Dessa forma, utilize o método aqui descrito para determinar todas as raízes do polinômio:

$$p(x) = 14x^6 + 119x^5 + 282x^4 + 127x^3 - 101x^2 - 72x - 9 \tag{13}$$

Explique no relatório o procedimento que utilizou no MATLAB para resolver este item.

2 Instruções

- A primeira etapa do processo de correção consistirá em submeter as funções implementadas a vários casos de teste de forma automatizada. Assim, os cabeçalhos das funções devem ser seguidos **rigorosamente**. Arquivos .m com os nomes destas funções e os cabeçalhos já implementados foram fornecidos juntamente com este roteiro. Dê preferência a implementar seu laboratório a partir destes arquivos .m fornecidos para evitar erros.

- **Não** é permitido o uso de funções ou comandos prontos do MATLAB que realizem toda a funcionalidade atribuída a uma certa função. Entretanto, o uso destas funções para verificação das implementações realizadas é encorajado. Em caso de dúvida quanto à permissão de uso de alguma função ou comando, recomenda-se consultar o professor.

- Não é necessário reimplementar métodos de laboratórios anteriores. Assim, funções implementadas em laboratórios anteriores podem ser utilizadas. Caso prefira, também é permitido utilizar funções equivalentes do MATLAB.

- Não é necessário se preocupar com verificação dos dados de entrada.

- Os arquivos .m implementados devem ser entregues juntamente com um relatório.

- No relatório, não é necessário demonstrar que as funções implementadas funcionam corretamente (isto será verificado separadamente). Basta incluir resultados e conclusões relativos à **Análise**.

- Para facilitar a correção da Análise, inclua os gráficos diretamente no relatório. Nos gráficos, coloque títulos, legendas e nomes nos eixos.

3 Dicas:

- Para carregar o arquivo com as matrizes da Análise 1, basta usar o comando `load('analise1.mat')`.

- A função `[V, D] = eig(A)` do MATLAB determina todos os autovetores e autovalores de uma matriz $\mathbf{A}$. Esta função é útil para testar as implementações deste laboratório.

- A função `r = roots(p)` do MATLAB determina todas as raízes do polinômio definido pelo vetor $\mathbf{p}$.

- A função `diag(A)` retorna a diagonal principal da matriz $\mathbf{A}$.

- Para criar gráficos com alta qualidade em formato PNG para inclusão em arquivos do Microsoft Word, utilize o comando: `print -dpng -r300 grafico.png`.

- Se utilizar LaTeX, dê preferência para incluir gráficos em formato vetorizado. No Linux, utilizando `pdflatex`, você pode gerar um gráfico em formato EPS usando "`print -depsc2 grafico.eps`" e depois convertê-lo para PDF usando o comando de terminal "`epstopdf grafico.eps`". O arquivo PDF é aceito pelo `pdflatex`.