EE2021: Computer Tools for Electrical Engineering

Lab 7

Fall 2018

In this lab session, you will write use cell arrays to implement a numerical method for computing the largest eigenvalue and a corresponding eigenvector for a square matrix. Furthermore, by testing your program with 2×2 and 3×3 matrices, you will visualize the convergence of the algorithm in 2D and 3D.

In mathematics, given a diagonalizable matrix A, *power iteration* is an iterative method for finding the largest (in magnitude) eigenvalue λ and an eigenvector v corresponding to that eigenvalue.

The method is summarized as follows:

1. Generate a random vector v_1 with unit length ($||v_1|| = 1$) and choose a small positive number Δ. Compute $\lambda_1 = \dfrac{v_1^T A v_1}{v_1^T v_1}$ and set $k = 1$.
2. Compute $v_{k+1} = \dfrac{Av_k}{||Av_k||}$ and $\lambda_{k+1} = \dfrac{v_{k+1}^T A v_{k+1}}{v_{k+1}^T v_{k+1}}$.
3. If $||v_{k+1} - v_k|| > \Delta$, increase k and repeat Step 2. Otherwise, proceed to step 4.
4. The largest eigenvalue is approximately equal to λ_{k+1} and an eigenvector corresponding to the largest eigenvalue is v_{k+1}.

In order to implement the power iteration and see its convergence visually,

1. Define a function `poweriter` which accepts an array `A` and a double `delta` as the input arguments and returns the approximations of the largest eigenvalue `eigMax` and corresponding eigenvector `eigvMax` as outputs.
2. Define `eigMax` and `eigvMax` as cell arrays and fill them using the power iteration summarized above.
3. Create a script and test your function with the matrices $\begin{bmatrix} -2 & 2 \\ 30 & 4 \end{bmatrix}$ and $\begin{bmatrix} -20 & 20 & 30 \\ 4 & 5 & 0 \\ -70 & 0 & 75 \end{bmatrix}$.
   Plot the eigenvector approximations in 2D and 3D and observe the step by step convergence. Furthermore, plot the eigenvalue approximations as a function of k.

Post-lab exercise: Enhance the visualization by the methods discussed in Section 7.3 of your textbook.