Math 471 Numerical Methods

Homework 5. Due: Aug. 6

Problems requiring coding are marked with $*$.

Please do some editing to minimize the pages of your print-outs. Thanks!

1. (Multivariable interpolation). Suppose we want to interpolation a two-variable function $f(x, y)$ using a quadratic polynomial

$$p(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy.$$

   (a) How many data points are required for a unique solution of (a_0, $a_1, ...,$ a_5)?

   (b) Write the linear system used for finding the interpolation. Explain what condition has be imposed to guarantee the unique existence of such interpolation.

2. * Consider an n-by-n tridiagonal matrix $A = \begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots \\ -1 & 2 & -1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & -1 & 2 & -1 \\ \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}$. We know that all its eigenvalues are real since A is real symmetric.

   (a) Prove by contradication that all eigenvalues of A are strictly positive. (An old trick: assume $A\vec{v} = \lambda\vec{v}$ with $\lambda \leq 0$; pick the largest absolute value among the entries of $\vec{v}$ and go from there).

   (b) * Implement the inverse power method in a subroutine that takes the size n as the input and outputs

   - the **smallest** eigenvalue λ_{min};
   - the number of iterations;
   - the associated eigenvector $\vec{v}_{min}$ so that $\|\vec{v}_{min}\|_2 = 1$.

   Use $\dfrac{(1, 2, 3, ..., n)^T}{\sqrt{1^2 + 2^2 + 3^2 + ... + n^2}}$ as the initial guess (which can be generated by Matlab code `x0=[1:n]'; x0=x0/norm(x0,2);`). Set error tolerance to be $= 10^{-12}$, that is, implement $\|\vec{x}_{k+1} - \vec{x}_k\| < 10^{-12}$ as the stopping condition. Here, $\vec{x}_k$ denotes the *normalized* eigenvector from the k-th iteration.

(c) * Test the above code for $n = 200$ so that it yields $\lambda_{min} = 0.000244286......$ Then, test $n = 400, 800, 1600$. Use `format long` in Matlab so it displays about 16 significant digits. Use the `plot(...)` routine in Matlab to graph the corresponding $\vec{v}_{min}$ for these n's. What kind of shape do you observe in these plots?

(d) Create a table with two columns: column 1: n; column 2: number of iterations. Describe what you observe and explain the factors that affect the number of iterations.

(e) If we take advantage of the sparsity of matrix A, which part of your code can be improved? Describe your idea with the $O(\cdot)$ language but do NOT implement it.

3. * Let $\vec{v}_{min}$ be the eigenvector associated with the smallest eigenvalue obtained from the previous problem. Then, use the **deflation** strategy (§4.3) to find the second smallest eigenvalue λ_{2min} and associated eigenvector $\vec{v}_{2min}$ for $n = 400$ with error tolerance 10^{-12}. Make sure that, in each iteration, normalization is performed *after* $\vec{v}_{min}$ is removed. Plot the associated eigenvector $\vec{v}_{2min}$ and describe what you observe.

4. Let the dominant eigenvalue of A be of multiplicity 2 and positive. Then, there exists two associated eigenvectors $\vec{v}$ and $\vec{w}$ that are also linearly independent. Also, let $\vec{u}$ be the eigenvector associated with the second dominant eigenvalue. If we perform a power method with initial guess $\vec{x}_0 = a\vec{v} + b\vec{w} + c\vec{u}$ without any round-off error, prove that

$$\lim_{k\to\infty} \vec{x}_k = \frac{a\vec{v} + b\vec{w}}{\|a\vec{v} + b\vec{w}\|}$$

5. Let $B = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & -1 \end{pmatrix}$. It's 3 eigenvalues are -4.046833261023624, -0.204465546618958, 7.251298807642582. Someone implemented the shifted inverse power method with the speeding-up of Rayleigh Quotient. If the program started with an initial guess of eigenvector $\vec{v} = \begin{pmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix}$ and therefore initial shift $\dfrac{\vec{v}^T B \vec{v}}{\vec{v}^T \vec{v}} = 7$, the program converges to the eigenvalue 7.251298807642582 and an associated eigenvector. If

the initial guess is $\vec{w} = \begin{pmatrix} 1/\sqrt{3} \\ -1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix}$, what eigenvalue can we expect the program to converge to? Give a theoretical explanation (coding is optional).