Homework 4. Due Friday Sept. 27 by 10am in my mailbox

Always provide complete, well written solutions.

The solution **Problems 2 and 3** have to be typed in Latex, while the rest of the solutions can be handwritten.

Always submit printouts of the output of your codes - either as figures, or as tables, or as numbers - and always provide complete description of these outputs. Moreover your matlab codes have to be submitted via ELMS.

1. Ref. [1], Chapter 4, Problem 11 (a,b,c,d,f) (pages 100-101).

   We wish to solve the *boundary value problem* for the differential equation

$$\frac{1}{2}\partial_x^2 u = f(x) \quad \text{for } 0 < x < 1, \tag{1}$$

   with *boundary conditions*

$$u(0) = u(1) = 0\ . \tag{2}$$

   We *discretize* the interval $[0,1]$ using a uniform *grid* of points $x_j = j\Delta x$ with a grid size given by $n\Delta x = 1$, for a given $n \in \mathbb{N}$, $n > 1$. We consider the $n-1$ unknowns, U_j, for $j = 1, \dots, n-1$, that are approximations to the corresponding $u(x_j)$. If we use a second order approximation to $\frac{1}{2}\partial_x^2 u$, we get discrete equations

$$\frac{1}{2}\frac{1}{\Delta x^2}\left(U_{j+1} - 2U_j + U_{j-1}\right) = F_j\ , \tag{3}$$

   where $F_j := f(x_j)$. Together with boundary conditions $U_0 = U_n = 0$, this is a system of $n-1$ linear equations for the vector $U = (U_1, \dots, U_{n-1})^*$ that we write as $AU = F$.

   (a) Check that there are $n-1$ distinct eigenvectors of A having the form $r_{kj} = \sin(k\pi x_j)$. Here r_{kj} is the j component of eigenvector r_k. Note that $r_{k,j+1} = \sin(k\pi x_{j+1}) = \sin(k\pi(x_j + \Delta x))$, which can be evaluated in terms of r_{kj} using trigonometric identities.

   (b) Use the eigenvalue information from part (a) to show that $\|A^{-1}\|_{l^2} \to 2/\pi^2$ as $n \to \infty$ and $\kappa_{l^2}(A) = O(n^2)$ (in the informal sense) as $n \to \infty$.

   (c) Suppose $\widetilde{U}_j = u(x_j)$ where $u(x)$ is the exact but unknown solution of (1)-(2). In order to be consistent with the L^2 integral norm of functions, $\|u\|_{L^2}^2 = \int_{x=0}^1 u^2(x)dx$, we adjust the definition of the l^2 discrete norm and consider instead the adjusted norm defined by $\|U\|_{\Delta x}^2 = \Delta x \sum_{k=1}^n U_j^2$. Show that if $u(x)$ is smooth then the *residual*[1], $R = A\widetilde{U} - F$, satisfies $\|R\|_{\Delta x} = O(\Delta x^2) = O(1/n^2)$.

   (d) What does this mean in terms of $\|R\|_{l^2}$?

   (e) Show that $A\left(U - \widetilde{U}\right) = R$. Use part (b) to show that $\left\|U - \widetilde{U}\right\|_{\Delta x} = O(\Delta x^2)$.

   (f) Write a program in Matlab to solve (3). The matrix A is symmetric and *tridiagonal* (has non-zeros only on three *diagonals*: the *main diagonal*, and the immediate sub and super diagonals). Do a convergence study to show that the results are second order accurate. Submission on ELMS: You will include the function file and the driver file.

[1] Residual refers to the extent to which equations are not satisfied. Here, the equation is $AU = F$, which $\widetilde{U}$ does not satisfy, so $R = A\widetilde{U} - F$ is the residual.

2. Ref. [1], Chapter 4, Problem 12 (pages 101-102).

   This exercise explores conditioning of the non-symmetric eigenvalue problem. It shows that although the problem of computing the fundamental solution is well-conditioned, computing it using eigenvalues and eigenvectors can be an unstable algorithm because the problem of computing eigenvalues and eigenvectors is ill-conditioned. For parameters $0 < \lambda < \mu$, there is a *Markov chain transition rate* matrix, A, whose entries are, taking j and k to run from 1 to n:

   - $a_{jk} = 0$ if $|j-k| > 1$,
   - $a_{j,j-1} = \mu$ if $2 \leq j \leq n-1$,
   - $a_{jj} = -(\lambda+\mu)$ if $2 \leq j \leq n-1$,
   - $a_{j,j+1} = \lambda$ if $2 \leq j \leq n-1$,

   as well as

   - $a_{11} = -\lambda$,
   - $a_{12} = \lambda$,
   - $a_{n,n} = -\mu$,
   - $a_{n,n-1} = \mu$.

   This matrix describes a continuous time Markov process with a random walker whose position at time t is the integer $X(t)$. Transitions $X \to X+1$ happen with rate λ and transitions $X \to X-1$ have rate μ. The transitions $0 \to -1$ and $n-1 \to n$ are not allowed. This is the *M/M/1 queue* used in operations research to model queues ($X(t)$ is the number of customers in the queue at time t, λ is the rate of arrival of new customers, μ is the service rate. A customer arrival is an $X \to X+1$ transition.). For each t, we can consider the row vector $p(t) = (p_1(t), \ldots, p_n(t))$ where $p_j(t) = \mathrm{Prob}(X(t) = j)$. These probabilities satisfy the differential equation $\dot{p} = \frac{d}{dt}p = pA$. The solution can be written in terms of the *fundamental solution*, $S(t)$, which in an $n \times n$ matrix that satisfies $\dot{S} = SA$, $S(0) = I$.

   (a) Show that if $\dot{S} = SA$, $S(0) = I$, then $p(t) = p(0)S(t)$.

   (b) The *matrix exponential* may be defined through the Taylor series $\exp(B) = \sum_{k=0}^{\infty} \frac{1}{k!}B^k$. Use matrix norms and the fact that $\left\|B^k\right\| \leq \|B\|^k$ to show that the infinite sum of matrices converges.

   (c) Show that the fundamental solution is given by $S(t) = \exp(tA)$. To do this, it is enough to show that $\exp(tA)$ satisfies the differential equation $\frac{d}{dt}\exp(tA) = \exp(tA)A$ using the infinite series, and show $\exp(0A) = I$.

   (d) Suppose $A = R\Lambda L$ is the eigenvalue and eigenvector decomposition of A, show that $\exp(tA) = R\exp(t\Lambda)L$, and that $\exp(t\Lambda)$ is the obvious diagonal matrix.

   (e) Use the Matlab function `[R,Lam] = eig(A);` to calculate the eigenvalues and right eigenvector matrix of A. Let r_k be the k^{th} column of R. For $k = 1, \ldots, n$, print r_k, Ar_k, $\lambda_k r_k$, and $\|\lambda_k r_k - Ar_k\|$ (you choose the norm). Mathematically, one of the eigenvectors is a multiple of the vector $\mathbf{1}$ defined in part h. The corresponding eigenvalue is $\lambda = 0$. The computed eigenvalue is not exactly zero. Take $n = 4$ for this, but do not hard wire $n = 4$ into the Matlab code.

   (f) Let $L = R^{-1}$, which can be computed in Matlab using `L=R^(-1);`. Let l_k be the k^{th} row of L, check that the l_k are left eigenvectors of A as in part e. Corresponding to $\lambda = 0$ is a left eigenvector that is a multiple of p_∞ from part h. Check this.

(g) Write a function in Matlab to calculate $S(t)$ using the eigenvalues and eigenvectors of A as above. Explain your choice of input/output for this function. Compare the results to those obtained using the Matlab built-in function `S = expm(t*A);`. Use the values $\lambda = 1$, $\mu = 4$, $t = 1$, and n ranging from $n = 4$ to $n = 80$. Compare the two computed $\widehat{S}(t)$ (one using eigenvalues, the other just using `expm`) using the l^1 matrix norm. Use the Matlab routine `cond(R)` to compute the condition number of the eigenvector matrix, R. Print three columns of numbers, n, error, condition number. Comment on the quantitative relation between the error and the condition number. SUBMISSION ON ELMS: You will include the function file and the driver file.

(h) Here we figure out which of the answers is correct. We first define the vector p_∞ as the invariant probability distribution, which is the left eigenvector of A normalized so that the sum of its entries is 1 that corresponds to the eigenvalue 0. You can check this:

$$\left[1, \frac{\lambda}{\mu}, \frac{\lambda^2}{\mu^2}, \ldots, \frac{\lambda^{n-1}}{\mu^{n-1}}\right] \begin{bmatrix} -\lambda & \lambda & 0 & \cdots & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \cdots \\ & \cdots & \cdots & \cdots & \\ \cdots & 0 & \mu & -(\lambda+\mu) & \lambda \\ & \cdots & 0 & \mu & -\mu \end{bmatrix} = [0, 0, 0, \ldots, 0].$$

Then normalize the eigenvector so that the sum of its entries is 1 and obtain

$$p_\infty = \frac{1 - \lambda/\mu}{1 - \lambda^n/\mu^n} \left[1, \frac{\lambda}{\mu}, \frac{\lambda^2}{\mu^2}, \ldots, \frac{\lambda^{n-1}}{\mu^{n-1}}\right].$$

In other words p_∞ is the row vector with $p_{\infty,j} = ((1-r)/(1-r^n))r^{j-1}$, with $r = \lambda/\mu$. Then you can use the known fact that $\lim_{t\to\infty} S(t) = S_\infty$ has the simple form $S_\infty = \mathbf{1}p_\infty$, where $\mathbf{1}$ is the column vector with all ones, that is to say that all rows of the matrix $S_\infty = \mathbf{1}p_\infty$ are equal p_∞. Take $t = 3 * n$ (which is close enough to $t = \infty$ for this purpose) and the same values of n and see which version of $S(t)$ is correct. What can you say about the stability of computing the matrix exponential using the ill conditioned eigenvalue/eigenvector problem?

3. Ref. [1], Chapter 4, Problem 13 (page 103).

This exercise explores eigenvalue and eigenvector perturbation theory for the matrix A defined in exercise 2. Let B be the $n \times n$ matrix with $b_{jk} = 0$ for all $(j,k) \in \mathbb{N}^2$, $1 \le j \le n$ and $1 \le k \le n$, except $b_{11} = -1$ and $b_{2,n} = 1$. Define $A(s) = A + sB$, so that $A(0) = A$ and $\frac{dA(s)}{ds} = B$ when $s = 0$.

(a) Write a function in Matlab to print the eigenvalues of $A(s)$. Explain your choice of input/output for this function. For $n = 20$, print the eigenvalues of $A(s)$ for $s = 0$ and $s = 0.1$. What does this say about the condition number of the eigenvalue eigenvector problem? All the eigenvalues of a real tridiagonal matrix are real[2] but that $A(s = 0.1)$ is not tridiagonal and its eigenvalues are not real. SUBMISSION ON ELMS: You will include the function file and the driver file.

(b) Use first order eigenvalue perturbation theory to calculate $\dot{\lambda}_k = \frac{d}{ds}\lambda_k$ when $s = 0$. What size s do you need for $\Delta\lambda_k$ to be accurately approximated by $s\dot{\lambda}_k$? Try $n = 5$ and $n = 20$. Note that first order perturbation theory always predicts that eigenvalues stay real, so $s = 0.1$ is much too large for $n = 20$.

[2] It is easy to see that if A is tridiagonal then there is a diagonal matrix, W, so that WAW^{-1} is symmetric. Therefore, A has the same eigenvalues as the symmetric matrix WAW^{-1}.

References

[1] Bindel and Goodman, Principles of scientific computing