ASSIGNMENT 2

Due October 5, 2004, before 11:00 am

Problem 2

This problem is a modified version of Computer Problem 4.14 on p. 213 of Heath's book.

(a) The matrix exponential function of an $n \times n$ matrix $\mathbf{A}$ is defined by the infinite series

$$\exp(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{1}{k!}(\mathbf{A})^k = \mathbf{I} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 \cdots + \frac{1}{k!}\mathbf{A}^k \cdots.$$

Write a Matlab program to evaluate $\exp(\mathbf{A})$ using the foregoing series definition.

(b) An alternate way to compute the matrix exponential uses the result

$$\mathbf{AX} = \mathbf{XD},$$

in section 4.2.2 so that $\mathbf{A}$ has the eigenvalue-eigenvector decomposition

$$\mathbf{A} = \mathbf{XDX}^{-1}.$$

Therefore

$$\begin{aligned}\exp(\mathbf{A}) &= \sum_{k=0}^{\infty} \frac{1}{k!}(\mathbf{XDX}^{-1})^k = \sum_{k=0}^{\infty} \frac{1}{k!}(\mathbf{XDX}^{-1}\mathbf{XDX}^{-1} \cdots \mathbf{XDX}^{-1}\mathbf{XDX}^{-1}) \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{XD^kX}^{-1} = \mathbf{X}\left(\sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{D^k}\right)\mathbf{X}^{-1} = \mathbf{X}\exp(\mathbf{D})\mathbf{X}^{-1},\end{aligned}$$

where the diagonal matrix $\exp(\mathbf{D})$ is given by $\mathrm{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n})$. In the case where $\mathbf{A}$ is symmetric (Hermitian), $\mathbf{X}^{-1}$ is simply given by $\mathbf{X}^T$ ($\mathbf{X}^H$). Write a program to evaluate $\exp(\mathbf{A})$ using this method.

Test both methods using each of the following test matrices:

$$\mathbf{A}_1 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \qquad \mathbf{A}_1 = \begin{bmatrix} -49 & 24 \\ -64 & 31 \end{bmatrix}.$$

Also test using an $n \times n$ matrix with randomly generated real elements. You should also try to run you programs using progressively large matrix sizes. What happens when n is large? Consider the relative errors in each case by comparing with the results from using Matlab's built-in library routine `expm` for computing the matrix exponential. [Note that this routine is different than the Matlab function `exp`, which carries out an element-wise exponentiation.] The runtime for 2×2 matrices is too short to be determined accurately. However one can get an accurate measure when n is large. Use Matlab's `tic` and `toc` functions to time the 3 different ways of computing the matrix exponential. Which of the two methods [(a) or (b)] is more accurate and robust? Try to explain why.