STAT 515
Homework #8 WITH SOLUTIONS

1. The matrix exponential function is defined as

$$\exp\{M\} = \sum_{i=0}^{\infty} \frac{M^i}{i!}.$$

However, this definition does not provide a suitable method for calculating $\exp\{M\}$ for a given M. One simple alternative is to use one of the two formulas

$$\exp\{M\} = \lim_{n\to\infty} \left(I + \frac{M}{n}\right)^n = \lim_{n\to\infty} \left[\left(I - \frac{M}{n}\right)^{-1}\right]^n.$$

For the post office problem on HW #7, the rate matrix is given by

$$R = \begin{bmatrix} -8 & 8 & 0 & 0 & 0 \\ 12 & -20 & 8 & 0 & 0 \\ 0 & 12 & -20 & 8 & 0 \\ 0 & 0 & 12 & -20 & 8 \\ 0 & 0 & 0 & 12 & -12 \end{bmatrix},$$

where rates are in hours. Approximate the value of $\exp\{.5R\}$, which is the transition probability matrix for a time step of 30 minutes, using two methods:

(a) For successively larger powers of 2, i.e., $n = 2, 4, 8, \ldots$, find the value of $(I + .5R/n)^n$. Continue until the change in each entry is smaller than 10^{-5}. Report your final value of n and your final approximation of $\exp\{.5R\}$.

Solution: First, let's set up the R matrix:

```
> R <- matrix(c(-8, 12, 0, 0, 0, 8, -20, 12, 0, 0, 0, 8, -20,
+               12, 0, 0, 0, 8, -20, 12, 0, 0, 0, 8, -12),   5, 5)
```

Next, we'll use a loop to continue trying larger and larger n, starting with $n = 2$ and doubling it at every step, until the maximum change is smaller than 10^{-5}:

```
> lastAnswer <- matrix(0, 5, 5)
> finished <- FALSE
> k <- 0
> n <- 1
> while (!finished) {
+   k <- k+1
+   n <- n*2 # double the n
+   answer <- diag(5) + .5*R/n
+   for (j in 1:k) { # raise answer to the nth power by squaring repeatedly
+     answer <- answer %*% answer
+   }
+   finished <- all(abs(answer - lastAnswer) < 1e-5) # check for convergence
+   lastAnswer <- answer
+ }
> lastAnswer
          [,1]      [,2]      [,3]       [,4]       [,5]
[1,] 0.4176125 0.2665951 0.1623088 0.09580071 0.05768283
[2,] 0.3998927 0.2611831 0.1668329 0.10513201 0.06695930
[3,] 0.3651949 0.2502494 0.1754179 0.12357079 0.08556707
[4,] 0.3233274 0.2365470 0.1853562 0.14607047 0.10869892
[5,] 0.2920193 0.2259876 0.1925259 0.16304838 0.12641878
> n
[1] 32768
```

So the desired accuracy required $n = 2^{15} = 32{,}768$.

(b) Repeat the same procedure as in part (a) but use $[(I - .5R/n)^{-1}]^n$ instead.

Solution: This requires only minor modifications to the code above:

```
> lastAnswer <- matrix(0, 5, 5)
> finished <- FALSE
> k <- 0
> n <- 1
> while (!finished) {
+   k <- k+1
+   n <- n*2 # double the n
+   answer <- solve(diag(5) - .5*R/n)
+   for (j in 1:k) { # raise answer to the nth power by squaring repeatedly
+     answer <- answer %*% answer
+   }
+   finished <- all(abs(answer - lastAnswer) < 1e-5) # check for convergence
+   lastAnswer <- answer
+ }
> lastAnswer
          [,1]      [,2]      [,3]       [,4]       [,5]
[1,] 0.4176172 0.2665964 0.1623075 0.09579832 0.05768059
[2,] 0.3998946 0.2611839 0.1668326 0.10513091 0.06695803
[3,] 0.3651919 0.2502489 0.1754190 0.12357216 0.08556805
[4,] 0.3233193 0.2365445 0.1853582 0.14607473 0.10870316
[5,] 0.2920080 0.2259833 0.1925281 0.16305474 0.12642582
> n
[1] 32768
```

This example also required $n = 2^{15} = 32{,}768$.

(c) Use the `expm` function in R or Matlab to evaluate $\exp\{.5R\}$ and compare with the two approximations you obtained.

In R, you will have to install and load the package called `Matrix`*. Do this using* `install.packages("Matrix")` *and then* `library(Matrix)`.

Solution:

```
> library(Matrix)
> expm(.5*R)
5 x 5 Matrix of class "dgeMatrix"
          [,1]      [,2]      [,3]       [,4]       [,5]
[1,] 0.4176149 0.2665957 0.1623082 0.09579952 0.05768171
[2,] 0.3998936 0.2611835 0.1668328 0.10513146 0.06695866
[3,] 0.3651934 0.2502491 0.1754184 0.12357147 0.08556756
[4,] 0.3233234 0.2365458 0.1853572 0.14607260 0.10870104
[5,] 0.2920136 0.2259855 0.1925270 0.16305156 0.12642230
```

The answers in parts (a) and (b) are very close to the `expm` answer.

2. Problem 3 in homework #7 described two video game machines at an amusement park. For video game i, each period when it is being used is exponentially distributed with mean $1/\alpha_i$ hours and each period when it is not being used is exponentially distributed with mean $1/\beta_i$ hours, independent of the other machine. Furthermore, $\alpha_1 = 2$, $\alpha_2 = 3$, $\beta_1 = 5$, and $\beta_2 = 6$.

(a) If neither machine is in use when the park opens at 8:00am, find the probability that both machines are in use at 9:30am.

Solution: To find the answer, we can find the probability transition matrix $P(1.5)$, which is given by the matrix exponential $\exp\{1.5R\}$:

```
> R <- matrix(c(-11, 3, 2, 0, 6, -8, 0, 2, 5, 0, -8, 3, 0, 5, 6, -5), 4, 4)
> expm(1.5*R)
```

```
4 x 4 Matrix of class "dgeMatrix"
           [,1]      [,2]      [,3]      [,4]
[1,] 0.09524491 0.1904890 0.2380893 0.4761767
[2,] 0.09524452 0.1904894 0.2380884 0.4761777
[3,] 0.09523573 0.1904707 0.2380985 0.4761951
[4,] 0.09523534 0.1904711 0.2380975 0.4761960
```

The answer is the probability that starting in the first state at time zero, the chain is in the fourth state at time 1.5:

```
> expm(1.5*R)[1,4]
[1] 0.4761767
```

(b) Simulate 10,000 realizations of the Markov chain and give a 95% confidence interval for the probability in part (a) based on your simulation. Does your empirical estimate agree with the theoretical value?

Solution: I will make some minor changes to the code used in homework #7, problem 3(c):

```
> n <- 10000
> finalState <- rep(0, n)
> maxTime <- 1.5 # This is the cutoff time.
> for (count in 1:n) {
+   currentTime <- 0
+   currentState <- 1 # Assume that we always start in state 1 at time 0
+   finished <- FALSE # We'll set this to TRUE when it's time to stop.
+   while (!finished) {
+     deltaTime <- rexp(1, rate = -R[currentState, currentState])
+     if (currentTime + deltaTime > maxTime) {
+       # Now we need to finish this chain and declare the final state decided
+       finalState[count] <- currentState
+       deltaTime <- maxTime - currentTime
+       finished <- TRUE
+     }
+     currentTime <- currentTime + deltaTime
+     possibleMoves <- (1:4)[-currentState]
+     currentState <- sample(possibleMoves, 1, prob=R[currentState,possibleMoves])
+   }
+ }
> table(finalState) / n
finalState
     1      2      3      4
0.0986 0.1886 0.2488 0.4640
```

Notice how close the four probabilities are to the first row (really, all rows) of the matrix found in part (a). Here is a 95% confidence interval for the probability of ending in the fourth state:

```
> phat <- sum(finalState==4) / n
> phat + c(-1.96, 1.96) * sqrt(phat * (1-phat) / n)
[1] 0.4542254 0.4737746
```

3. Suppose that X_1, X_2, X_3, X_4 are i.i.d. from a uniform$(0,1)$ distribution. Let $S = X_1 + X_2 + X_3 + X_4$.

(a) Find $P(S < 1)$ exactly using a four-dimensional integral. (Hint: This is not too difficult.)

Solution: The joint density on the four-dimensional hypercube $(0,1)^4$ is just the constant 1.

The probability of $S < 1$ is found by integrating this density over the region where $S < 1$:

$$\begin{aligned}
\int_0^1 \int_0^{1-w} \int_0^{1-w-x} \int_0^{1-w-x-y} dz\,dy\,dx\,dw &= \int_0^1 \int_0^{1-w} \int_0^{1-w-x} (1-w-x-y)\,dy\,dx\,dw \\
&= \frac{1}{2} \int_0^1 \int_0^{1-w} \frac{1}{2}(1-w-x)^2\,dx\,dw \\
&= \frac{1}{6} \int_0^1 (1-w)^3\,dw \\
&= \frac{1}{24}.
\end{aligned}$$

(b) Now consider $P(S < 1.5)$. This is much more difficult to find analytically. Instead, use Monte Carlo simulation to approximate this probability. Give a 99% confidence interval for the true probability, and use a large enough sample so that your interval is no wider than 0.01. Report the sample size you used in addition to the interval.

Solution: Since we are finding a proportion, and the standard deviation of a sample proportion is not more than $1/\sqrt{4n}$, we could take n large enough so that $2.58/\sqrt{4n} < .005$ since the 99% confidence interval is $\hat{p} \pm 2.58 \times$ (standard error). This leads to n larger than about 66,000. Let's use 100,000 just for a nice round number:

```
> n <- 1e6
> x <- matrix(runif(4*n), ncol=4)
> phat <- sum(rowSums(x)<1.5) / n
> phat + c(-2.58, 2.58) * sqrt(phat * (1-phat) / n)
[1] 0.1995289 0.2015951
```

So the 99% confidence interval obtained from a sample of size 100,000 has a width of much less than 0.01 (the width is about 0.002).

(c) The central limit theorem approximation to $P(S < 1.5)$ is $P(Y < 1.5)$, where Y is a normal random variable with the same mean and variance as S. Based on your answer to part (b), how good does the central limit theorem approximation appear in this case?

Solution: Since X_i has mean 1/2 and variance 1/12, S has mean 2 and variance 1/3. Thus, the central limit theorem approximation is

```
> pnorm(1.5, mean=2, sd=1/sqrt(3))
[1] 0.1932381
```