NATIONAL UNIVERSITY OF SINGAPORE
Department of Mathematics

MA1506 Laboratory 3 (scilab)
Comments and Suggested Solutions
Semester II 2010/2011

Exercise 3

1. 716.25

2. 70. From the information available, $M = \begin{bmatrix} 0.7 & 0.2 & 0.3 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.4 \end{bmatrix}$. The 'brute force' way to compute the long run behaviour is to just calculate very large powers of M and compare the values of the entries.

```
--> M^(200) - M^(100)
--> M^(100)*[100; 100; 100]
```

In the first command, we can see that the difference between the entries in M^{200} and M^{100} are practically zero. So we are sufficiently confident that M^{100} is close enough to the long run value. Note that we need to take into account the initial number of cars at each location.

Alternatively, we can do things properly by diagonalizing M and taking limits.

```
--> [P D] =mtlb_eig(M)
D  =

    0.4414214    0      0
    0            1.     0
    0            0      0.1585786
 P  =

  - 0.8125199    0.7485692  - 0.2325878
    0.3365568    0.5346923  - 0.5615167
    0.4759631    0.3921077    0.7941045
--> D2= diag( [ 0 1 0 ])
--> P*D2*inv(P)*[100; 100; 100]
```

The **diag** command creates a diagonal matrix with the entries specified. It is clear that D2 is the limit of D^n as n gets large.

3. 104 million

4. Eigenvalues are -4, 3 and 3. The solution may look like complex numbers, but the imaginary parts are practically zero (of order 10^{-8} less than the real part,) so it is safe to drop them.

5. In this case we have complex eigenvalues $\pm i$ (real parts practically zero) and note that P is also a matrix with complex entries.

6.
```
--> C=[ 5 0 0; 1 5 0; 0 1 5]
--> [P D] = mtlb_eig(C)
--> det(P)
```

   In this case, we have the eigenvalue 5 repeated 3 times, and the determinant of P is approximately zero. Actually, matrix A is not diagonalizable and the determinant of P should be exactly zero. This slight error is due to the algorithm used by the **mtlb_eig** function.

7.
```
--> A=[5 6 2; 0 -1 -8; 1 0 -2]
--> poly(A,'s')
 ans  =


                 2   3
    36 - 15s - 2s + s


--> 36*eye(3,3)-15*A+ -2*A^2 +A^3
--> B= [ -2 -1 ; 5 2]
--> poly(B,'s')
 ans  =
                      2
    1 - 3.053D-16s + s


--> eye(2,2) + B^2  // second term is practically zero
--> C=[ 5 0 0; 1 5 0; 0 1 5]
--> poly(C,'s')
ans  =
                   2   3
  - 125 + 75s - 15s + s


-->-125*eye(3,3) + 75*C -15*C^2 + C^3
```

—The End—