Problem 1

Solution:

Sample code in Octave.

```octave
function pi = markov_steadystate(Pmatrix, epsilon);

MAX = 10000;
flag = 0;

[N,M] = size(Pmatrix);
if (N == M)
    x_vectors = [];
    P = Pmatrix';
    i=1;
    while((i < MAX) && (!flag))
        abs_difference = 0;
        if (i == 1)
            x_prev = ones(M,1)*(1/M);
        else
            x_prev = x_vectors(:,i-1);
        end
        x_vectors(:,i) = P * x_prev;
        for j=1:M
            abs_difference += abs(x_vectors(j,i) - x_prev(j,1));
        end
        if (abs_difference < epsilon)
            flag = 1;
        end
        i++;
    end
    if (flag)
        pi = x_vectors(:,i-1);
    else
        error ('Halt: runaway execution');
    end
else
    error ('Pmatrix must be square.');
end
```
Problem 2
Solution:

1. The steady state probabilities are:

\[
\begin{pmatrix}
0.2143 \\
0.4286 \\
0.1020 \\
0.2551
\end{pmatrix}
\]

2. The \(x^{(k)}\) vectors are:

\[
\begin{align*}
    x^{(1)} &= (0.1250, 0.5500, 0.1000, 0.2250) \\
    x^{(3)} &= (0.1600, 0.5180, 0.1260, 0.1960) \\
    x^{(5)} &= (0.1766, 0.4917, 0.1238, 0.2080) \\
    x^{(10)} &= (0.2299, 0.4024, 0.0924, 0.2753) \\
    x^{(20)} &= (0.2170, 0.4241, 0.1004, 0.2586) \\
    x^{(50)} &= (0.2143, 0.4285, 0.1020, 0.2551) \\
    x^{(100)} &= (0.2143, 0.4286, 0.1020, 0.2551)
\end{align*}
\]

3. Figure 1 shows the plot.

![Plot of \(x^{(k)}_2\) at the \(k^{th}\) iteration.](image)

Figure 1: Plot \(x^{(k)}_2\) at the \(k^{th}\) iteration.
Problem 3

Solution:

As we can see in Figure 2, the values in the rank vector quickly converge to the same result as $k$ grows, regardless of the initialization vector.

![Figure 2: Plot $x_2^{(k)}$ at the $k^{th}$ iteration.](image)
Problem 4

Solution:
Sample code in Octave.

```octave
function sequence = markov_simulate(Pmatrix, ivector, N, rseed);

sequence = [];
flag = 0;
if (nargin > 3)
    rand(“seed”,rseed);
end
[L,M] = size(Pmatrix);
if ((L == M) && ([M 1] == size(ivector)))
    nums = rand(N);
    for i=1:N
        if (i == 1)
            P = ivector;
        else
            P = Pmatrix(sequence(i-1),:);
        end
        steps = cumsum(P);
        flag = 0;
        j=1;
        while((j <= M) && (!flag))
            if (nums(i) <= steps(j))
                sequence(i) = j;
                flag = 1;
            end
            j++;
        end
    end
else
    error (’Pmatrix must be square or check ivector size.’);
end
```

Problem 5

Solution:

1. The relative frequencies for $N = 20$ are:

```
[0.2000]
0.4500
0.1000
[0.2500]
```

2. The relative frequencies for $N = 100,000$ are:

```
[0.2137]
0.4273
0.1037
[0.2554]
```