Problem Set 8

Name: Jianxiang Fan
Email: jianxiang.fan@colorado.edu

1. (a)

$$
\begin{aligned}
w^T P &= [1\ 1\ \cdots\ 1] \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \\
&= [\sum_{i=1}^{n} p_{i1} \quad \sum_{i=1}^{n} p_{i2} \quad \cdots \quad \sum_{i=1}^{n} p_{in}] \\
&= [1\ 1\ \cdots\ 1] \\
&= w^T
\end{aligned}
$$

(b)

$$
\begin{aligned}
q_{k+1,1} &= q_{k,1}\, p_{11} + q_{k,2}\, p_{12} + q_{k,3}\, p_{13} \\
q_{k+1,2} &= q_{k,1}\, p_{21} + q_{k,2}\, p_{22} + q_{k,3}\, p_{23} \\
q_{k+1,3} &= q_{k,1}\, p_{31} + q_{k,2}\, p_{32} + q_{k,3}\, p_{33}
\end{aligned}
$$

That is

$$\mathbf{q}_{k+1} = P\mathbf{q}_k$$

(c) Since $\|\mathbf{q}_0\|_1 = 1$, and $\forall k \geq 1$

$$
\begin{aligned}
\|\mathbf{q}_{k+1}\|_1 &= |q_{k+1,1}| + |q_{k+1,2}| + |q_{k+1,3}| \\
&= q_{k,1}(p_{11} + p_{21} + p_{31}) + q_{k,2}(p_{12} + p_{22} + p_{32}) + q_{k,3}(p_{13} + p_{23} + p_{33}) \\
&= q_{k,1} + q_{k,2} + q_{k,3} \\
&= \|\mathbf{q}_k\|_1
\end{aligned}
$$

So $\forall k \geq 1$, $\|\mathbf{q}_k\| = 1$, there is no need to normalize the new vector.

(d) .

```
P = [0.63,0.18,0.14;0.26,0.65,0.31;0.11,0.17,0.55];
q0 = [1/3;1/3;1/3];
while 1
    q = P*q0;
    if norm(q-q0) < 1e-9
        break
    end
    q0 = q;
end
```

$\mathbf{q}$ converges to $\begin{bmatrix}0.3095097 & 0.4462493 & 0.2442410\end{bmatrix}^T$

2. (a) The algorithm may be terminated when we get a reasonable approximation for the eigenvector, which means

$$\|\mathbf{ev}_{new} - \mathbf{ev}_{old}\|_2 < \epsilon$$

(b) .

```
function [ew, ev2] = sipm(A, mu)
[n,~] = size(A);
ev1 = randn(n,1);
ev1 = ev1 ./ norm(ev1);
while 1
    ev2 = (A - mu * eye(n))\ev1;
    ev2 = ev2 ./ norm(ev2);
    ew = ev2'*A*ev2;
    if ev2(1) < 0
        ev2 = -ev2;
    end
    if norm(ev2 - ev1) < 1e-9
        break
    end
    ev1 = ev2;
end
end
```

(c) Eigenvalue: 0.475207973

Eigenvector: $\begin{bmatrix}0.807422688 \\ -0.298585457 \\ -0.508837231\end{bmatrix}$

Iterations: 23

(d) convergence rate: 0.3831365

(e) Yes. Using eig function in Matlab, we'll get the eigenvalues of P are $\lambda_1 = 1, \lambda_2 = 0.47520797, \lambda_3 = 0.35479203$. Based on the theory stated in class, the convergence rate is

$$\frac{|\mu - \lambda_2|}{|\mu - \lambda_3|} = \frac{|0.55 - 0.47520797|}{|0.55 - 0.35479203|} = 0.383140$$

We can see that the method stated in (d) gives a reasonable estimation.

3. When doing LU factorization on a $n \times n$ upper Hessenberg matrix, when iterating for column k, only row $k+1$ with nonzero $A_{k+1,k}$ need to be processed. Then the algorithm can be modified as below.

```
function [ A, p ] = hessLU( A )
[n,~] = size(A);
p = 1:n;
for k=1:n-1
    if A(p(k+1),k) > A(p(k), k)
        t = p(k+1);
        p(k+1) = p(k);
        p(k) = t;
    end
    A(p(k+1),k) = A(p(k+1),k)/A(p(k),k);
    for j=k+1:n
        A(p(k+1),j) = A(p(k+1),j) - A(p(k+1),k) * A(p(k),j);
    end
end
end
```

FLOPs:

$$\begin{aligned}&\sum_{k=1}^{n-1}\sum_{j=k+1}^{n} 2\\&= 2\sum_{k=1}^{n-1}(n-k)\\&= n(n-1)\\&= n^2 - n\end{aligned}$$

Without pivoting, L is lower bidiagonal with all 1's on the diagonal.

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ * & 1 & 0 & \cdots & 0 & 0 \\ 0 & * & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & * & 1 \end{bmatrix}$$

With pivoting, L is unit lower triangular with at most one nonzero element in each column besides the diagonal element.