This short Python program

```
********
a, b,c,n = 0,0,1,0
while n <30:
print a,b,c
a, b,c, n = c +.5*b, a, .5*b , n+1
**********
```

starts with an input (row) vector, or "seed" $I_0 = (a, b, c)$ and outputs, iteratively, $I_{k+1} = HI_k$ where H is the transition matrix we worked out last class. I seeded it with $I_0 = (1, 0, 0)$ and ran it for 25 iterations:

```
1 0 0
0.0 1 0.0
0.5 0.0 0.5
0.5 0.5 0.0
0.25 0.5 0.25
0.5 0.25 0.25
0.375 0.5 0.125
0.375 0.375 0.25
0.4375 0.375 0.1875
0.375 0.4375 0.1875
0.40625 0.375 0.21875
0.40625 0.40625 0.1875
0.390625 0.40625 0.203125
0.40625 0.390625 0.203125
0.3984375 0.40625 0.1953125
0.3984375 0.3984375 0.203125
0.40234375 0.3984375 0.19921875
0.3984375 0.40234375 0.19921875
0.400390625 0.3984375 0.201171875
0.400390625 0.400390625 0.19921875
0.3994140625 0.400390625 0.2001953125
0.400390625 0.3994140625 0.2001953125
0.39990234375 0.400390625 0.19970703125
0.39990234375 0.39990234375 0.2001953125
0.400146484375 0.39990234375 0.199951171875
```

so that you can see the convergence to $I_* = (.4, .4, .2)$.

Theory tells us that the rate of convergence of the iteration to the fixed point I_* is geometric in the ratio $|\lambda_2|$, where λ_2 is second largest eigenvalue (in magnitude). This rate of convergence means that we expect "$I_k \sim I_* + \lambda_2^k$(error) with (error) $= O(1)$, or more precisely:

$$|I_k - I_*|/|I_{k+1} - I_*| = |\lambda_2|^k + o(k)$$

been that if we computed