Math 328 FINAL REVIEW

1. week 4 (5d): If you get stuck trying to find if a sequence coverges, you can often program a quick python script to determine the convergence. For instance, the script

```
for k in range(1,20):
    print (k,3**k/k**3)
```

   indicates that this sequence divergest to $\infty$.

2. Week 5 (1a): Do the algebra by starting with the root problem $f(x) = 0$ and turning it into a fixed point problem. Or start with the fixed point problem and turn it into the root problem. When you do, you will show that the problems are equivalent.

3. Week 5 (6). You can use the IVT to verify that there is a solution. Then you must find an interval around the fixed point $u^*$ so that $|g'(u)| < 1$ in that interval. You can do this using calculus by making pen and paper computations as in the hw solutions. If you get stuck at any point you can try to graph $f(x)$ to show there is a root or $g'(u)$ to show that $g'(u)$ is below 1 on an interval. Here is how I use python to "view" a bound on $g'(u)$

```
from pylab import plot,show, grid
from numpy import linspace

def gPrime(u):
    return 1.5*u*(3*u**2 +3)**(-0.75)

u=linspace(1,2,50)
y=gPrime(u)
plot(u,y)
grid()
show()
```

   You should also be able to approximate solutions and discuss the error to this problem using Newton's method, the secant method, and bisection search.

4. Given $A = \begin{bmatrix} -2 & -3 \\ 6 & 7 \end{bmatrix}$. Use 8 iterations of the power method with $u^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ to approximate $\lambda_1$, the larger eigenvalue and its corresponding eigenvector $v_1$.

```
from numpy import array, dot
from math import sqrt

T = array([[-2,-3], [6,7]],float)
u = array([1,0],float)  # seed

for k in range(8):
    u1 = u[0]
    u = dot(T,u) #update u
    lambda1 = u[0]/u1
    print(k+1,u,lambda1)

print (-65534/4**8, 131070/4**8)
```

   ANSWER: $\lambda_8 = 4.00036625565, v_8 = (-0.999969482421875, 1.999969482421875)$

5. Let $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$.

   (a) Factor $A$ as $QR$

   (b) Apply two iterations of the QR Algorithm to find $A_2$, when $A_0 = A$.

   (c) Continue the algorithm to $A_{100}$ to approximate the three eigenvalues.

```
import numpy as np
from numpy import array,dot
from math import sqrt

A = np.array([[2,-1,0],[-1,2,-1],[0,-1,2]],float)
I = np.identity(3)
#v = np.array([1,0,-1],float)
#print(dot(A,v)) # check eigenvector with eigenvalue 2

def q_r(A,Q):
    R = np.zeros((3,3))
    for k in range(3):
        for t in range(k):
            R[t,k] = dot(A[:,t],A[:,k])
            A[:,k] -= R[t,k]*A[:,t]
        R[k,k] = sqrt(dot(A[:,k],A[:,k]))
        A[:,k] /= R[k,k]
    print(A,R)
    return(dot(R,A), dot(Q,A))

q_r(A,I)

for k in range(0):
    print(k,A)
    A,I = q_r(A,I)

#print(A)
#print(I)
```

ANSWER:

(a) $Q = \begin{bmatrix} 0.89442719 & 0.35856858 & 0.26726124 \\ -0.4472136 & 0.71713717 & 0.53452248 \\ 0. & -0.5976143 & 0.80178373 \end{bmatrix}, R = \begin{bmatrix} 2.23606798 & -1.78885438 & 0.4472136 \\ 0. & 1.67332005 & -1.91236577 \\ 0. & 0. & 1.06904497 \end{bmatrix}$

(b) $\begin{bmatrix} 3.14285714e+00 & -5.59397149e-01 & -1.66533454e-16 \\ -5.59397149e-01 & 2.24844720e+00 & -1.87847556e-01 \\ 0.00000000e+00 & -1.87847556e-01 & 6.08695652e-01 \end{bmatrix}$

(c) $A^{(100)} = \begin{bmatrix} 3.41421356e+00 & -7.58437937e-17 & 4.35740192e-17 \\ -6.96146361e-24 & 2.00000000e+00 & -3.96507854e-16 \\ 0.00000000e+00 & -2.74581324e-54 & 5.85786438e-01 \end{bmatrix}$ implies that the second column $\begin{bmatrix} 7.07106781e-01 \\ 3.13530871e-17 \\ -7.07106781e-01 \end{bmatrix}$ of the matrix $Q_1Q_2Q_3 \ldots Q_{99}Q_{100}$ is an eigenvector of A with eigenvalue approximately 2. From this I guess that the eigenvector is $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$.

6. Exercise 5.2 Newman.
7. week 8 (10)
8. week 10 (13)
9. week 13 (3)
10. week 14 (1)
11. week 15 (2)
12. week 15 (10)
13. Given the IVP: $u' = F(t, u), u(0) = u_0$.
    (a) If you approximate $u(0.8)$ using Euler's method with stepsize $h = 0.05$ and then approximate $u(0.8)$ again with stepsize $h = 0.025$ what do you expect to be the relation of the errors in these computations. Explain.
    (b) If you approximate $u(0.8)$ using RK4 method with stepsize $h = 0.05$ and then approximate $u(0.8)$ again with stepsize $h = 0.025$ what do you expect to be the relation of the errors in these computations.