Daniel Perry (dperry@cs.utah.edu)
cs6210 / Fall 2005
Homework 5

1)
I implemented newtons algorithm, and it seemed to work perfectly for almost any situation (I tried values of P (p-th root) from 2 to 30, and values of A from 2 to 15). In every case it came up with the exact same answer as the math.h function pow(), or came within 10e-7 of it (that was the difference between iterations I used to stop iterating).

Some results:

P (power)	***A***	***A^(1/P)***	***iterations***
2	2	1.4142135	5
18	8	1.122462	9
20	14	1.1410545	13
29	2	1.0241896	5
29	14	1.0952712	13
30	132	1.1767544	50

2)
a)Analysis is attached to back (handwritten paper).
summary:
g1 – diverges
g2 – converges
g3 – converges
g4 - converges

b) By implementing the four fixed point problems I found the implementation matched my analysis. I found that g1 diverged, g2 would converge, g3 would converge about half the time of g2, and g4 would converge just under half the time of g3.
Results:

x0	***g1 iterations***	***g2 iterations***	***g3 iterations***	***g4 iterations***
5	diverges	45	20	7
10	diverges	46	20	8
20	diverges	46	20	9

3)
In this implementation I found that both types of sampling (a-uniform, b-chebyshev's zeros), had similar results with the max difference between the interpolation and the original function with values of j very small. However as j increased, the uniform

Daniel Perry (dperry@cs.utah.edu)
cs6210 / Fall 2005
Homework 5

sampling slowly increased in max difference, while the chebyshev zeros sampled implementation didn't increase in difference as much.

Results.

j	***max diff w/ Uniform***	***max diff w/ Chebyshev***
3	0.646154	0.828912
4	0.707014	1.09565
6	0.432692	0.846387
8	0.247359	0.703027
10	0.298402	0.570512
16	2.09904	0.264316

I've attached my code as well. It can also be found at http://www.cs.utah.edu/~dperry/classes/cs6210/.