Create Rubric

15 points

Create your rubric now or come back to it later. You can also make edits to your rubric while g

Q1 Problem 1

10 points

This quiz is open-notes and you are permitted to use software like python, Wolfram Alpha, desmost, etc. You are not allowed to search for the solution, to use stackexchange, Chegg, etc.

Consider the following code for obtaining an approximation to the function $f(x) = 1/(1+16x^2)$:

```
n = 100
k = 5
x = np.linspace(-1,1,n)
y = 1/(1+16*x**2)
A = np.zeros((n,k+1))
for i in range(k+1):
    A[:,i] = np.cos(i*np.pi*x)

c = np.linalg.lstsq(A,y)[0]
```

Q1.1 (a)

1 point

Describe in words the columns of A

1 +1.0 credit

2 +1.0 should be explicit

3 +0.0
no credit

+ Add Rubric Iter

Q1.2

4 points

Describe in words what the code is doing.

1 -0.0
Correct

2 -1.0
not polynomial

3 -0.5
not linear system /

4 -1.0
too vague

5 -3.0
not 5 points

6 -3.0
not chebyshev noc

+ Add Rubric Iter

Q1.3

3 points

Describe how to obtain an approximation to the function $f(x) = 1/(1 + 16x^2)$ from the output of the code.

1	+3.0	Correct
2	+1.5	using the x for the equally spaced po
3	+3.0	this stil gives a dis
4	+1.5	not chebyshev pol
5	+2.0	c0 not divided by 2
6	+0.0	does not explain h

+ Add Rubric Iter

Q1.4

2 points

Will this approach work for approximating $f(x) = np.exp(-x)$? Why or why not?

1 +2.0
Correct

2 +2.0
function on [-1,1] (although not with problem didn't exp on [-1,1], so full cr

3 +1.0
insufficient justifu

4 +0.0
Incorrect

+ Add Rubric Iter

Q2 Problem 2

5 points

The n-th entry of `t1` and `t2` are the flop counts from two different algorithms run on an input of size n.

```
t1 = np.array([    4,     7,    16,    37,    76,   139,
         751,  1024,  1357,  1756,  2227,  2776,  3409,
        5872,  6901,  8044,  9307, 10696, 12217, 13876,
       19741, 22012, 24451])
t2 = np.array([    0,    97,   394,   891,  1588,  2485,
        8073,  9970, 12067, 14364, 16861, 19558, 22455,
       32346, 36043, 39940, 44037, 48334, 52831, 57528,
       72819, 78316, 84013])
```

1 -0.0
correct

2 -0.0
full credit - overly

3 -0.0
how do you know

One of the algorithms requires $O(n^2)$ flops and the other requires $O(n^3)$ flops. Determine which is which and explain your solution:

4	-1.0
	correct answer, bu
5	-3.0
	incorrect

+ Add Rubric Iter