Key:
Green: Correct
Red: Problem has no solution if red in “Number” problem, otherwise this just means the given column didn’t return a value or a “good” value
Yellow: Problem solvable is yellow in “Cond #” column, but not to machine precision. If in “Notes” column, means problem is solvable but not fast enough
Black: Values are probably wrong, need to be recomputed
Blue: Function is unimplemented and/or values are uncomputed

Number	Continuous? Smooth? Where?	Other Observations	Preferred quadrature method	Integration Value	Interval	Cond #	Notes
1	Yes, yes everywhere	Should be very easy	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	1.718281828459045	[0, 1]	1.5819767068693264 24385	CC, p=270
2	No, not at x=.3	Constant on two separate intervals, derivative 0	Adaptive Quadrature/Simpson’s Rule	0.7000000000000000	[0, 1]	1.43	AS- tol=10^(-16)
3	No, not at x=0	N/A	Adaptive Quadrature/Simpson’s Rule	0.3785300171241612	[0, 1]	2.22	AS recurses infinitely. CC converges very slowly, accurate to about 11 digits with 100000 points. Not sure what to do on this one QUADGK COMES CLOSEST
4	No, not at x= -1, this should be continuous on the interval given though	CONDITION NUMBER UNDERFLOWS IN DENOMINATOR -> ILL CONDITIONED PROBLEM	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	-3.906250000000000*10^-15	[0, 10^(-12)]	BLOWS UP TO INFINITY	SEE IF YOU CAN BEAT UNDERFLOW
5	Yes, yes, everywhere	Spikes at x=.4 Possible cancellation errors? Looking at graph should show why this one is continuous but not easy with gauss due to spike	Adaptive Quadrature/Simpson’s Rule?	0.1727205034243749 (PRETTY SURE THIS IS WRONG) 0.1717416237343111	[0, 1]	1.3 * 10^(-6)	AS, tol = 10^(-10) should work in theory but doesn’t. Are we sure our integral value here is correct?? CC also seems to think the second digit is a 6 not a 7. Yeah just checked on

							wolfram and the value should be closer to .163102. Weird tho since both CC and AS give values around .163494 QUAD and SIMPSON GIVE EXACT SAME VALUE => THAT IS CORRECT (MAYBE??)
6	Yes, yes, everywhere	Should be easy	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	1.564396444069050	[-1, 1]	.32	CC, p=270
7	Yes, but with noise	Noise, do a least squares	Least Squares	1.692603132845149 (give or take .05 due to choice of epsilon)	[-1, 1]	.32	QUAD OR QUADGK MIGHT BE CLOSE WHO KNOWS
8	Yes, yes, everywhere	Should be easy	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	1.000000000000000	[0, 10]	6.67*10^(-107)	CC, p=270
9	Yes, yes, everywhere	Possible cancellation errors in numerator Possible overflow errors in numerator and denominator. Should probably rewrite/simplify Simplifies to: (E^x)^(-1/10) - (E^x)^(9/10)/E^(11 x)	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	9.900536099712276	[0, 100]	.00046	CC, p=270, had to rewrite into (exp(x)).^(-1/10) - (exp(x)).^(9/10)./(exp(11*x)) first. Weird that we got this to machine precision, condition number may be off
10	Yes, yes, everywhere	Should be very easy	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	1.833333333333333	[0, 1]	1.6363636363636363 63636363636	CC, p=5000
11	No, not where e^x = .5, 1.5, 2.5, 3.5… 27.5	Multiple discontinuities, Will be pretty tough Looking at graph helps a lot	Adaptive Quadrature/Simpson's Rule	17.66446061011451	[0, 3]	3.4??	AS loses 13 digits, CC loses at least 10, looks like errorestimate in AS is giving some kind of 0 error like a cancellation error, AS does not appear to be working properly; continues running even when it gives errorestimate = 0. If true value is 17.66 etc. then quadgk comes closest. Else,

							if true value is 17.78 etc.then quad comes closest
12	Yes, yes, everywhere	Should be easy	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	0.4000000000000000	[0, 1]	2.500000000000000	Works with fclencurt for high n values, CC, p=2000
13	No, not at x=0	N/A	Adaptive Quadrature/Simpson's Rule	2.000000000000000	[0, 1]	.5	AS appears not to converge (may not be properly written) Quadgk comes very close but loses 3 digits instead of 1
14	No, not continuous when x=3. Not smooth at 1 or 3.	N/A	Adaptive Quadrature/Simpson's Rule	7.500000000000000	[0, 5]	1.333	AS is off by approximately 10^(-16). Is this good? What does this mean? That is, AS gives 7.499999999999999. Whatever, quad works exactly for tol = 10^(-16)
15	Yes, yes, everywhere	Should be easy Should be similar to integral 6	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	0.8669729873399110	[0, 1]	.57	CC, p=270

t

Number	Continuous? Smooth? Where?	Other Observations	Preferred quadrature method	**Integration Value**	**Interval**	**Cond numb**	**Notes**
16	Not continuous at x=0	Not continuous Oscillates heavily.	Clenshaw Curtis	0.498986808693045	[0, 1]	0	CC, p=2000, QUADGK IS VERY CLOSE

		This is bad. Can't use either may need to combine and do a piecewise gauss					LOSES 2 DIGITS
17	No, not at x= -1	N/A	Adaptive Quadrature/Simpson's Rule	-1	[0, 1]	0	AS appears not to converge (may not be properly written). QUAD gets it for tol = 10^(-16)
18	Yes, yes, everywhere on its domain	N/A	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod		[-1, 1]		
19	Yes, yes, everywhere on its domain	N/A	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	-0.0002000200020002000	[-1, 1]	4999.5	CC, p=5000, have to take real part for some weird reason
20	Yes, yes, everywhere (it's a polynomial)	Computing the det as a polynomial is going to be hard	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod		[-1, 1]		
21	Discontinuous on x=0	N/A	Adaptive Quadrature/Simpson's Rule	0.3817732906760363	[-1, 1]	1.41	AS appears not to converge (may not be properly written) EVERYTHING GIVES -.6366 SOMETHING IS THE VALUE WRONG??? ALL OF THEM ARE VERY VERY CLOSE, CC, QUAD, and QUADGK agree up to the last digit
22	Yes, yes, everywhere	Numerator is very small compared to denominator May have to integrate first and then divide by constant	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	1.273239544735163*10^-9	[-1, 1]	0	CC, p=270
23	Yes, yes	Approaches 0 as x approaches infinity	Adaptive Quadrature/Simpson's	1.000000000000000	[0, infinity]		AS appears not to converge (may not be

		(quickly) (for x greater than 36.5 the function underflows below machine precision Will need something piecewise most likely	Rule??				properly written) RECHECK MAYBE WE CAN FIGURE OUT HOW TO BEAT THE UNDERFLOW
24	Discontinuous at x=0	N/A	Adaptive Quadrature/Simpson's Rule	.2375252341425132	[0,.5]	0	
25	Discontinuous at endpoints???	N/A	Adaptive Quadrature/Simpson's Rule		[0,1]		
26	Yes, yes, everywhere	N/A	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	-0.5818757877653272	[0, pi]	0	CC, p=270
27	Yes, yes, everywhere	N/A	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	2.925303491814363	[-1,1]	.93	CC, p=600
28	Discontinuous at x=0	INTEGRAL APPARENTLY DIVERGES	Adaptive Quadrature/Simpson's Rule	-124.8993729727460 (doesn't converge)	[0,1]		
29	Yes, yes, everywhere	Possible cancellation errors around x=0??	Clenshaw Curtis or Classical Gaussian/Gauss-Kronrod	1.047197551196598	[-.5,.5]	.55	CC, p=4600
30	Discontinuous at x=0	Also equals log(x)	Adaptive Quadrature/Simpson's Rule	-1.000000000000000	[0,1]	0	Quad gets it for tol =10^(-16)

16. Sin[100 Pi x]/(Pi x)
17. Log[x]
18. Cos[10000 ArcCos[x]]
19. Cos[100 ArcCos[x]]
20. ???
21. Piecewise[{{Sin[x], -1 <= x <= 0}, {Cos[x], 0 <= x <= 1}}]
22. Cos[Pi (x/2)]/10^9
23. E^(-x)
24. Log[2 x]/Log[x]

25. ???
26. Sin[5 Sin[x]]
27. E^x^2
28. Sin[x]/Log[x]
29. 1/Sqrt[1 - x^2]
30. Log[2 x] - Log[2]

SetPrecision[N[FullSimplify[Integrate[f9[x], {x, 0, 100}]]], 16]