HW10 (10.1 – 10.4)   Due Friday March 29 – On paper *AND* Online

10.1 Fundamentals: Integration Error

4 pts

Pretend I did a Taylor Series analysis for both of the following two quadrature methods, and came up with the given error expressions. That is, each expression indicates how the exact integral I_{exact} is related to the quadrature method I_{approx} over just *ONE* panel (*not* the whole [a,b] range).

(i) Method 1:
$$I_{\substack{exact\\ \text{one panel}}} = I_{\substack{approx\\ \text{one panel}}} + \frac{h^2 f'''(x_2)}{2} + \frac{h^3 f''''(x_2)}{12} + \frac{h^4 f'''''(x_2)}{60} - \cdots$$

(ii) Method 2:
$$I_{\substack{exact\\ \text{one panel}}} = I_{\substack{approx\\ \text{one panel}}} - \frac{h^4 f'(x_2)}{9} - \frac{h^5 f''(x_2)}{81} - \frac{h^6 f'''(x_2)}{263} - \cdots$$

For *each* method …

(a) Apply the definition for **precision** of a method to determine the precision of the method.

(b) Explain what is the **error order** with *h* for the method over the ***entire*** integration range (generally consisting of many intervals & panels).

Remember: you can't just write down the precision or order; you have to **justify** it mathematically.

10.2 Computational Integration: Simpson's method (By-Hand)

4 pts

Remember the "Failure Rate" integral in HW9, for which you calculated the integral *by-hand* using the midpoint and trapezoid methods? You evaluated:

$$F(70) = \int_0^{70} f(x)\,dx \quad \text{where} \quad f(x) = \lambda e^{-\lambda x} \quad \text{for } \lambda = 0.01$$

(a) Repeat the integration *by-hand* using the same n = 8 equally-spaced nodes, except this time using the **Simpson's method**. (*Hint*: for this value of n do you need a combo of 1/3- and 3/8-methods??) And (I know this is a pain), because the Simpson's method is so accurate, you have to do all calculations accurate to **8** decimals (pretty much all the decimals on your calculator).

(b) Given that the integral has the exact (analytic) answer $F(t) = 1 - e^{-\lambda t}$, calculate the **percent (relative) error** in calculating $F(70)$ for the Simpson's method, and compare that to the errors in HW9 from the Midpoint and Trapezoid rules.

Show all your work on paper, especially the values you're using for **all the nodes**, and do each calculation accurate to at least 8 decimals.

10.3 Application: MATLAB's Built-in "integral" function

2 pts

Use the built-in function `integral` to solve the following integral:
$$X = \int_2^{100} \frac{\left(10\sqrt{t}\right)\ln(3t)}{t^3 + 10}\,dt$$
(I have no idea what it might mean, it just looks nasty.)

Once you get it working, write out **ON PAPER**:

1. Exactly how you called `integral` (*i.e.* your command that starts with: `X = integral` …)
2. The function you had to write for the "integrand" that integral has to call.
3. The output value for *X* (rounded to **4** decimals).

Don't submit anything to Carmen – I should be able to see exactly everything you did **on paper**.

HW10 (10.1 – 10.4) Due Friday March 29 – On paper *AND* Online

10.4 Computational Integration: MATLAB (Simpson's method, and Built-in "integral")

5 pts

Go back to the MATLAB code you wrote in 9.4 to solve for the failure rate *F*(70), but now you're going to add procedures to: use the Simpson's method, and use the built-in "integral" function.

Start with your working script from HW9.4 (or "steal" my posted code if yours didn't work perfectly), and rename it **HW10_4.m**. Now add the following functionality:

a) Create a *NEW* function: `Isimp = `**`Simpson`**`(a,b,n)` that outputs the integral of `Fun(x)` over [*a,b*] using *n* nodes with the Simpsons combined -1/3 and -3/8 rules. Make your life easier by assuming the number of nodes *n* will always be *even*.

b) In addition to calling the `Midpoint` and `Trapezoid` functions, have your HW10_4 script call the *new* `Simpson` function to evaluate *F*. Change your script so that the number of nodes over which you run these three functions is now: *n* = [2 8 50 500 5000 50000].
(I made them all even now, to be consistent with your assumption for the Simpson method.)

c) Uses the exact value $F(70) = 1 - e^{(-0.01)(70)}$ to calculate the **percent** (relative) **error** in the approximations of *F* from ALL three of your own methods (midpoint, trapezoid, Simpsons) for all the nodes *n* above.

d) Plots the **absolute value** of the % error for all three methods as a function of **interval size**, *h*. Please make each line different, so the **midpoint** method shows **circles** connected by **black** lines, the **Trapezoid** method shows **triangles** connected by **green** lines, and the **Simpson's** method shows **squares** connected by **blue** lines. Be sure to use the *loglog* plotting command to allow details to be seen even when the error gets very small. So, use something like this plot command:

```
loglog(h,PctErrMid,'ok-', h,PctErrTrap,'^g--', h,PctErrSimp,'sb-.')
```

Save this plot in pdf form as **HW10_4.pdf**

e) Uses the built-in function `integral` to solve for F(70) directly (using your `Fun.m` for *f*(*x*)).

That's it!

Please submit the following things **online** in the Carmen HW10 location:

- Your new function `Simpson.m`, new script `HW10_4.m`, and plot `HW10_4.pdf`.
- Answers to the following questions in the *comment* section:
  1. From the `integral` command in (e): What is the approximate **value** of F(70) to 8 decimals? What is the **percent error** in this result?
  2. What is the (approximate) **slope** of the Simpsons line from your log-log plot? (That should be consistent with the "order" of the method, right?)
  3. From your plot, estimate how many **nodes** (*n*) it would take for your trapezoid method to give the same error for this problem as using the built-in `integral` command.