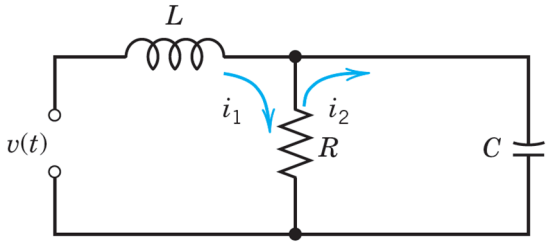
Read all of the following information before starting the exam:

- The exam is open book, open notes, open Python documentation, open internet, etc.
- You **MAY NOT** use any form of technology to communicate with, send to, or receive information from another person (e.g., classmates, other instructors, anonymous or known persons on the internet). HOWEVER, you are **encouraged** to submit written questions to the professor or TA by email and/or have a private help session with the instructor through ZOOM.
- **MODULES/PACKAGES:** You may now use any Python packages you wish. However, you must follow specific instructions even if a package/library may be the easier way. You may use/reuse code (with proper attribution; e.g., "this function is modified from Dr. Smay's.py file" or "this import is from my HW1 file")
- **COMMENTS/DOCUMENTATION:** **ALL** of your functions (especially constructors for classes) should use `docstrings` and other comments inside the function as necessary.
- **SUBMISSION:** You must place all your .py files in a single folder called EX2SP22 and submit the zipped folder as a single .zip file to Canvas titled EX2SP22.zip.
- **GRADING:** When we grade your assignment, we will run your program with those given numerical values, looking for correct answers. Then we will change the numerical values (including changing the SIZES of the arrays) and look for correct answers for those modified values as well. We will only use numerical values, array sizes and functions that make sense. We will not be testing your program to see how it handles bad data. As you have seen in your homework, comments/documentation will count for about 20% of the grade on a problem.

1. (30 points) Write a python program that graphs the currents i_1, i_2 and voltage across C (v_c) for 10 seconds for the following RLC network:

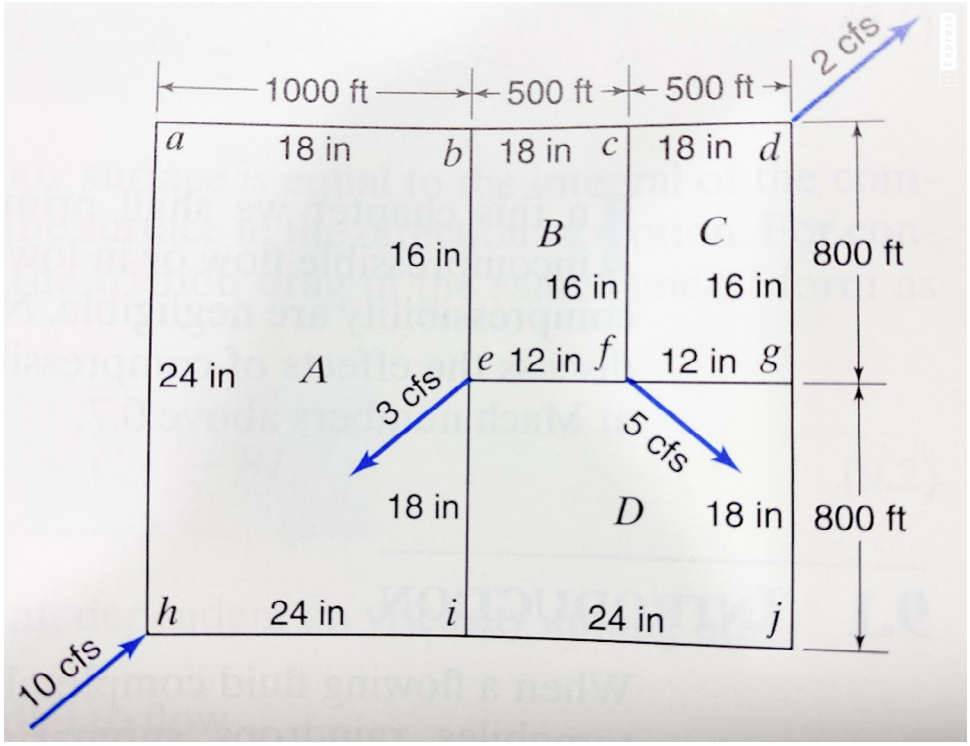


Notes:

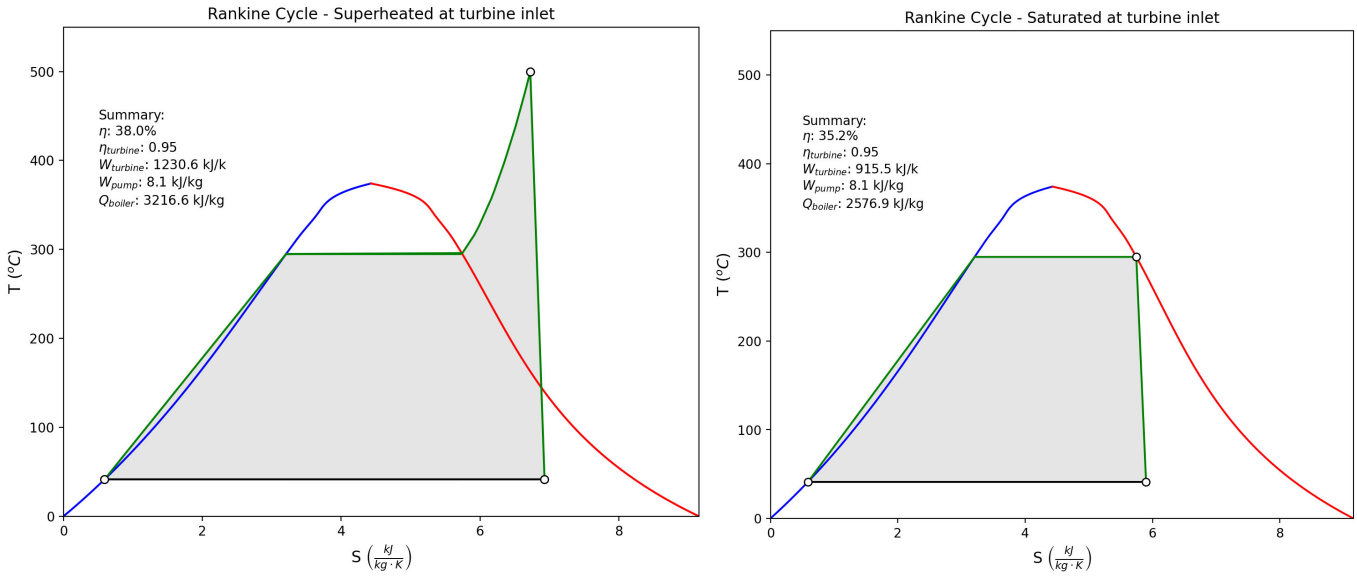
i. You must write your system of differential equations in state form and solve using `odeint`.
ii. R=10Ω, L=20H, C=0.05F, $v(t)=20\cdot\sin(20\cdot t)$
iii. Your plot should use only black and white lines with $i_1(t)$ as a solid line, $i_2(t)$ as a dashed line and $v_C(t)$ as a dotted line. You should plot $v_c(t)$ on the right vertical axis ant $i_1(t)$ and $i_2(t)$ on the left vertical axis. All axes should be labeled and a legend placed on the graph to identify the various lines.

2. (35 points) You have analyzed a pipe network with sprinklers in homework 6. Modify your program, appropriately, to analyze the pipe network below and output a nicely formatted report for the: ***i) flow rate and direction in each pipe, ii) the head loss in each pipe (in inches of water), and iii) the pressure at each node in psi with the knowledge that the node h pressure is 80 psi.***

Notes: The 12" and 16" pipes are cast iron (roughness = 0.00085 ft) while the 18" and 24" pipes are concrete (roughness = 0.003 ft). Minor losses may be ignored. cfs stands for cubic feet per second. Room temperature water is the fluid ($\mu=20.50\times10^{-6}$ lb·s/ft^2, $\gamma=62.3$ lb/ft^3). Name your file EX2_2.py.



3. (35 points) Use pyplot to graph your Rankine cycle from homework 6 part 2 on a T-S diagram. The graphs should look like the following graphs (including title, axes labels, and summary notation): (we are connecting pump exit to saturated liquid at p_{high} with a straight line)



Notes: i) You should modify your rankine class to include a value for isentropic turbine efficiency ($\eta_{turbine}$) and include $\eta_{turbine}$ in your calculation for cycle efficiency. ii) You should include the plotting function as a method of the rankine class. And, iii) You should write a python program called EX2_3.py that, has only the following code:

```
from rankineFile import rankine  # rankineFile is where your rankine class is.

def main():
    '''
    A test program for rankine power cycles.
    R1 is a rankine cycle object that is instantiated for turbine inlet of saturated vapor.
    R2 is a rankine cycle object that is instantiated for turbine inlet of superheated vapor.
    :return: none
    '''
    R1=rankine(p_high=8000, p_low=8, eff_turbine=0.95, name='Rankine Cycle - Saturated at Turbine Inlet')
    R1.calc_efficiency()
    R1.plot_cycle_TS()

    R2=rankine(p_high=8000, p_low=8, t_high=500, eff_turbine=0.95, name='Rankine Cycle - Superheated at Turbine Inlet')
    R2.calc_efficiency()
     R2.plot_cycle_TS()

 if __name__==”__main__”:
    main()
```