Applied Numerical Computing for Scientists and Engineers

Computational Assignment 3

Electronic submission via Bitbucket and course website.
Assignment weight: 10%.

The purpose of this assignment is to give you practice with using built-in functions and library routines for numerical methods (specifically ODE solvers) in MATLAB and Python. Exercise 1 uses the files you generate in Exercises 2–3. The six files generated in this assignment must be submitted via your Bitbucket repository (MUST use these filenames):

1. `solve_ODEs_CA3.m`
2. `solve_ODEs_CA3.html`
3. `solve_ODEs_CA3.py`
4. `solve_ODEs_CA3.ipynb`
5. `solve_ODEs_CA3.tex`
6. `solve_ODEs_CA3.pdf`

You are encouraged to submit all the extra files that are generated during this assignment. The above six are the ones that will be graded.

1. Git (2%)

Use your Bitbucket repository named firstname_lastname_applnumcomp for version control with the .m and .html files associated with Exercise 2 and the .py, .ipynb, .tex, and .pdf files associated with Exercise 3. Create a subfolder called "CA 3". Work on Exercises 2–3 of this assignment in the "CA 3" subfolder. There should be at least one commit of each required file and at least three total commits for this assignment with comments that briefly explain states of progress on the assignment, e.g., "outline of MATLAB function for Exercise 2". The last commit you wish to submit for a grade must have the commit -m message "assignment 3 submission". Everything between the assignment 2 submission commit and assignment 3 submission commit will be evaluated.

For the course website Computational Assignment 3 submission, use the text box to enter the web address for the commit that corresponds to the submission. For example, `https://bitbucket.org/ashleefv/ashleefordversyptapplnumcomp/commits/d8390344f1b0ef0faed7db84c01ffce95c2f0423`. This is simply to have a clear time stamp for your submission. **Submissions and/or final commits after the deadline submitted on the due date will receive maximum of half credit for the assignment**.

2. MATLAB (49%)

This Exercise has two parts: writing a .m file in MATLAB and generating a .html file through the publish feature.

2.1. .m file (40%)

Write a MATLAB function titled `solve_ODEs_CA3.m` that takes no input.

2.1.1. Define the system of ODEs

Within `solve_ODEs_CA3.m`, write another function `ODEs_CA3` (may be nested or not) to define a system of ordinary differential equations for $\frac{dF_A}{dV}$, $\frac{dF_B}{dV}$, $\frac{dF_C}{dV}$, and $\frac{dT}{dV}$ that describes the molar flow rates of species A, B, and C in mol/s and temperature in K in a non-isothermal plug-flow reactor. The reactions are at steady-state but vary spatially along the volume of the reactor, hence V is the independent variable here instead of time. The equations and all of the parameters and initial conditions are in the document titled "Computational Assignment 3 Equations.pdf". Equations E12-5.3–E12-5.6 should be defined within `ODEs_CA3` along with E12-5.7–E12-5.13 and the formulas for k_{1A} and k_{2A}. These are all highlighted in yellow in Computational Assignment 3

Equations.pdf. Note: the ordering in the example problem in Computational Assignment 3 Equations.pdf didn't matter for the software program used there, but for MATLAB and Python, the order does matter. Your function `ODEs_CA3` should be structured similarly to `system_of_ODEs.m` from Computational Assignment 2. You can find a sample solution to Computational Assignment 2 and a sample Python analog in the course Bitbucket repository ashleefv/checlassfa18 as `system_of_ODEs.m` and `system_of_ODEs.py`.

In this assignment, you should allow for all of the listed constant parameter values to be adjustable. Use the parameter names in the formulas rather than explicitly typing the given numbers into the formulas. To clarify, the following examples are acceptable

```
E1overR = 4000; % ok to supply a value when defining a ...
    parameter name
k1A = 10*exp(E1overR*(1/300-1/T)) % Good: parameter name ...
    is used in the formula for k1A instead of hard-coding ...
    a value for E1overR
```

but the following is not be acceptable

```
k1A = 10*exp(4000*(1/300-1/T)) % Bad: parameter name is ...
    not used in the formula for k1A and a value of 4000 ...
    for E1overR has been hard-coded making it challenging ...
    to update E1overR
```

The parameter values may be supplied in the ODE solver call and passed to the ODE function, i.e., `ODEs_CA3(V,F_and_T,parameter1,parameter2, ...)`. Alternatively, the parameters may be supplied before the ODE solver call and then the ODE function can be a nested function of the form `ODEs_CA3(V,F_and_T)`. You are not required to supply default values for this assignment. Do not use global or other similar keywords to define the parameters.

2.1.2. Call ODE solver to solve system of ODEs

Within `solve_ODEs_CA3.m`, use `ode45` to solve the system of ODEs defined by `ODEs_CA3`. V should be evaluated between 0 and 1 at increments of 0.01. The initial conditions are given in Table E12-5.1 in Computational Assignment 3 Equations.pdf. Default values for the optional inputs for `ode45` should be used, therefore, `options` does not need to be specified.

2.1.3. Plot results

Within `solve_ODEs_CA3.m` after calling `ode45`, plot figure(1) T vs. V and figure(2) F_i vs. V for $i = A, B, C$. For figure(2), include a legend labeling F_A, F_B, and F_C. The line styles and colors should be different for F_A, F_B, and F_C. The resulting plots should match Figures E12-5.1 and E12-5.2 in Computational Assignment 3 Equations.pdf with xlabels, ylabels, and units. The coloring, line styles, and the legend placement may be different than those in the pdf.

2.2. .html file (9%)

Use the publish feature in MATLAB to generate an .html file and to evaluate the function at the default values. Your code and the .html output should demonstrate the following coding best practices:

- Write thorough comments and documentation
- Define the purpose of your function, the author, and all input and output parameters or variables at the top of the code
- Define the problem you are solving using comments
- Use descriptive variable names
- Indent code and use whitespace
- Avoid duplication: "Define once, and reuse often"
- Specify the units for physical quantities
- Suppress MATLAB printing to the command window with use of semi-colon

3. Python (49%)

This Exercise has three parts: writing a .py file in Python, documenting your Python output with a .ipynb using Jupyter, and generating a .tex file to describe the results.

3.1. .py file (40%)

Write a Python file titled `solve_ODEs_CA3.py`.

3.1.1. Define the system of ODEs

Within `solve_ODEs_CA3.py`, write a function `ODEs_CA3` to define a system of ordinary differential equations for $\frac{dF_A}{dV}$, $\frac{dF_B}{dV}$, $\frac{dF_C}{dV}$, and $\frac{dT}{dV}$. The instructions from Exercise 2 in Section 2.1.1 should be followed here except that you are now writing the function in Python. Your function `ODEs_CA3` inside of `solve_ODEs_CA3.py` should be structured similarly to `L9_odeint_example.py` from Lecture 9.

3.1.2. Call ODE solver to solve system of ODEs

Within `solve_ODEs_CA3.py`, use `scipy.integrate.odeint` to solve the system of ODEs defined by `ODEs_CA3`. V should be evaluated between 0 and 1 at increments of 0.01. The initial conditions are given in Table E12-5.1 in Computational Assignment 3 Equations.pdf. Default values for the optional inputs for `odeint` should be used.

3.1.3. Plot results

Use `import matplotlib.pyplot as plt` to access the plotting functions. Within `solve_ODEs_CA3.py` after calling `odeint`, plot figure(1) T vs. V and figure(2) F_i vs. V. For figure(2), include a legend labeling F_A, F_B, and F_C. The line styles and colors should be different for F_A, F_B, and F_C. The resulting plots should match Figures E12-5.1 and E12-5.2 in Computational Assignment 3 Equations.pdf with xlabels, ylabels, and units. The coloring, line styles, and the legend placement may be different than those in the pdf.

3.2. .ipynb file (4%)

The purpose of this part of the assignment is to create documentation for the `solve_ODEs_CA3.py` file using Jupyter. You should complete this task after you have completed Section 3.1. Create a Jupyter notebook titled `solve_ODEs_CA3.ipynb` that has the following features:

- Level 1 Markdown heading (using # symbol) with the text `solve_ODEs_CA3.py`
- Markdown text (no fancy formatting, just text) for the purpose of your function, the author, and all input and output parameters or variables at the top of the code.
- Markdown text to display the LaTeX equations for $\frac{dF_A}{dV}$, $\frac{dF_B}{dV}$, $\frac{dF_C}{dV}$, and $\frac{dT}{dV}$ found in Equations E12-5.3–E12-5.6 of Computational Assignment 3 Equations.pdf. You do not need to define all the parameter values or define all of the variables and parameters.
- Jupyter cells of Python 3 code for the lines of code and remaining documentation in `solve_ODEs_CA3.py`.
- Jupyter cell(s) of Python 3 code that run(s) the `solve_ODEs_CA3.py` and produces the two figures of the output from the Python simulations.
- Jupyter cell(s) of Markdown text that provide(s) a caption for each of the figures.
- All of the cells should be executed using Shift+Enter.

3.3. .tex file (5%)

The purpose of this part of the assignment is to create documentation for the `solve_ODEs_CA3.py` file using LaTeX. You should complete this task after you have completed Section 3.1. Use the built-in article class in LaTeX to create a document titled `solve_ODEs_CA3.tex` that has the following features:

- Two sections

  1. Section 1 titled Equations
     In this section, type the equations for $\frac{dF_A}{dV}$, $\frac{dF_B}{dV}$, $\frac{dF_C}{dV}$, and $\frac{dT}{dV}$ found in Equations E12-5.3–E12-5.6 of Computational Assignment 3 Equations.pdf. You do not need to define all the parameter values or define all of the variables and parameters. The four equations should be numbered.

  2. Section 2 titled Results
     Insert the two figures of the output from the Python simulations. Include a caption below each figure.

- a .pdf file generated from the .tex file (the figures must appear here)