Applied Numerical Computing for Scientists and Engineers

Computational Assignment 4

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

The purpose of this assignment is to give you practice with parameter estimation of dynamic models using MATLAB and Python. Exercise 1 uses the files you generate in Exercises 2–3. The eight files generated in this assignment must be submitted via your Bitbucket repository (MUST use these filenames):

1. `param_estim_3lump.m`
2. `param_estim_3lump.html`
3. `param_estim_4lump.m`
4. `param_estim_4lump.html`
5. `param_estim_3lump.py`
6. `param_estim_4lump.py`
7. `param_estim_3lump.ipynb`
8. `param_estim_4lump.ipynb`

For the course website Computational Assignment 4 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.

0.1. Problem statement

A fluid catalytic cracker or FCC unit (Figure 1) is an industrial process used to convert heavy petroleum refining intermediates into more valuable lighter products via catalytic reactions. The FCC process involves many chemical components and reactions. It is common to gather sets of component chemical species by molecular weight into subgroups referred to as "lumps" instead of considering each component explicitly. Two of the most common kinetic models for the process are called the three-lump model (Figure 2) and the four-lump model (Figure 3).

Figure 1: Photograph of a fluid catalytic cracker.

0.1.1. Three-lump model

The three-lump model involves three subgroups: VGO (y_1), gasoline (y_2), and the sum of gas and coke (y_3). The reaction network for the three-lump model is shown in Figure 2. The equations that describe the three-lump model are

$$\frac{dy_1}{dt} = -(k_1 + k_3)y_1^2 \tag{1}$$

$$\frac{dy_2}{dt} = k_1 y_1^2 - k_2 y_2 \tag{2}$$

$$\frac{dy_3}{dt} = k_3 y_1^2 + k_2 y_2 \tag{3}$$

There are three parameters in the three-lump model: k_1, k_2, and k_3. y_i denotes weight fraction of lump i. Conversion is defined as $1 - y_1$. Note that the initial conditions are $y_1(0) = 1$, $y_2(0) = 0$, and $y_3(0) = 0$.

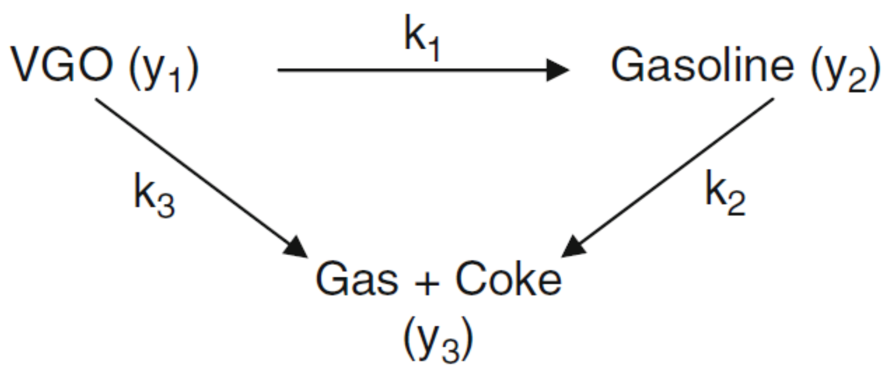

Figure 2: Three-lump model reaction network.

0.1.2. Four-lump model

The four-lump model involves four subgroups: VGO (y_1), gasoline (y_2), gas (y_3), and coke (y_4). The reaction network for the four-lump model is shown in Figure 3. The equations that describe the four-lump model are

$$\frac{dy_1}{dt} = -(k_{12} + k_{13} + k_{14})y_1^2 \tag{4}$$

$$\frac{dy_2}{dt} = k_{12} y_1^2 - k_{23} y_2 - k_{24} y_2 \tag{5}$$

$$\frac{dy_3}{dt} = k_{13}y_1^2 + k_{23}y_2 \tag{6}$$

$$\frac{dy_4}{dt} = k_{14}y_1^2 + k_{24}y_2 \tag{7}$$

There are five parameters in the four-lump model: k_{12}, k_{13}, k_{14}, k_{23}, and k_{24}. y_i denotes weight fraction of lump i. Conversion is defined as $1 - y_1$. Note that the initial conditions are $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 0$, and $y_4(0) = 0$.

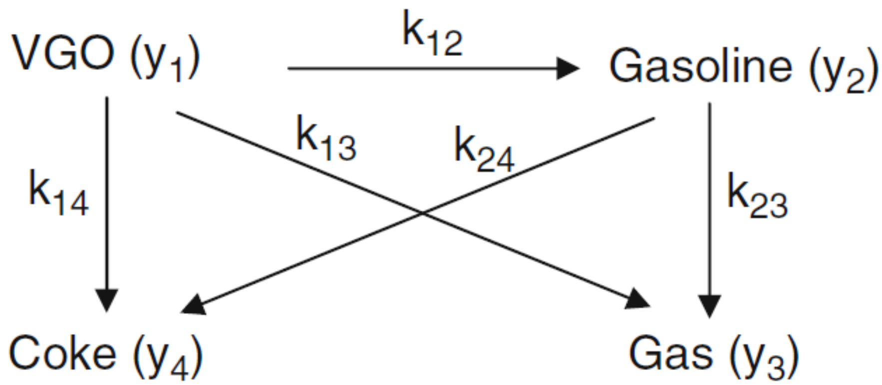

Figure 3: Four-lump model reaction network.

0.1.3. Data

The following data have been reported in the literature and converted to a convenient form for use in this assignment.

0.1.4. Expected results

The fitted parameters should yield predicted curves that agree with Figures 4 and 5. Some slight deviations are allowable as the parameter estimation procedures required here differ from those used in the original work.

Table 1: Data for FCC process.

Time (h)	Conversion	Weight Fraction VGO	Gasoline	Gas	Coke
1/60	0.4926	0.5074	0.3767	0.0885	0.0274
1/30	0.6204	0.3796	0.4385	0.136	0.0459
1/20	0.7118	0.2882	0.4865	0.1681	0.0572
1/10	0.8238	0.1762	0.5416	0.2108	0.0714

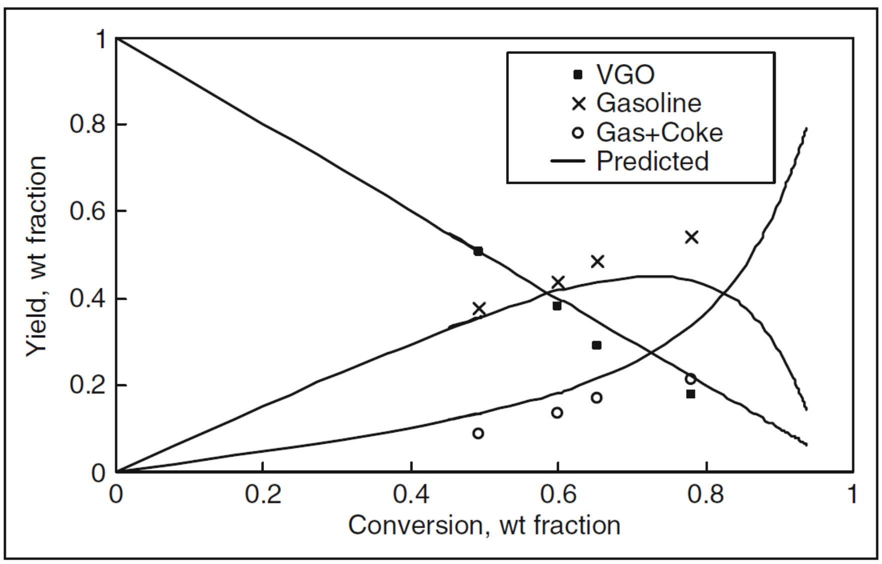

Figure 4: Data and fitted curve solution for three-lump model.

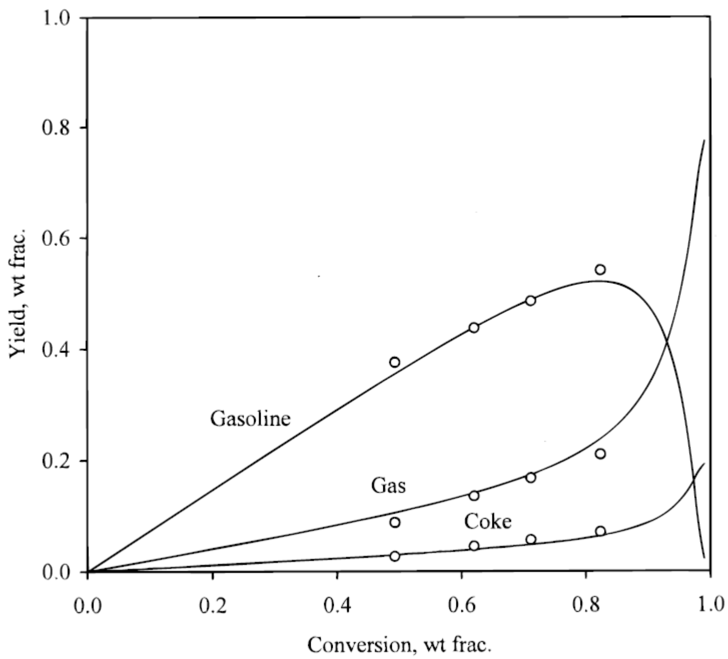

Figure 5: Data and fitted curve solution for four-lump model.

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 4". Work on Exercises 2–3 of this assignment in the "CA 4" 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 4 submission". Everything between the assignment 3 submission commit and assignment 4 submission commit will be evaluated.

2. MATLAB (49%)

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

2.1. .m file (40%)

1. Write two MATLAB functions: `param_estim_3lump.m` and `param_estim_4lump.m`.
2. In `param_estim_3lump.m`, use `lsqcurvefit` to estimate the parameters for the three-lump model using the data in Table 1.
3. In `param_estim_4lump.m`, use `lsqcurvefit` to estimate the parameters for the four-lump model using the data in Table 1.
4. Have `lsqcurvefit` return the norm of the residuals (resnorm) as well as the parameter values.
5. The ODEs should be solved using `ode23s`.
6. Print the values for the parameters to the output screen labeled with their names (i.e., don't just let `params` output to the screen but have it label `k_1 = params(1)`, etc.).

7. Create two plots of the output: y_i vs. t and y_i vs. conversion. Each plot should include the data and the corresponding fitted curves along with a legend labeling each curve or set of points. The second plot should be similar to Figure 4 for the three-lump model and Figure 5 for the four-lump model.

You will probably want to use `L12example1.m` and `L12example2.m` for getting started.

2.2. .html file (9%)

Use the publish feature in MATLAB to generate html files for `param_estim_3lump.m` and `param_estim_4lump.m`. 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 semicolon

3. Python (49%)

This Exercise has two parts: writing two .py files in Python and generating two .ipynb files to document the results of both models.

3.1. .py files (40%)

1. Write two Python scripts: `param_estim_3lump.py` and `param_estim_4lump.py`.

2. In `param_estim_3lump.py`, use `curve_fit` to estimate the parameters for the three-lump model using the data in Table 1.

3. In `param_estim_4lump.py`, use `curve_fit` to estimate the parameters for the four-lump model using the data in Table 1.

4. Have `curve_fit` return the covariance array (pcov) as well as the parameter values.

5. The ODEs should be solved using `odeint`.

6. Print the values for the parameters to the output screen labeled with their names (i.e., don't just let `params` output to the screen but have it label `k_1 = params[0]`, etc.).

7. Create two plots of the output: y_i vs. t and y_i vs. conversion. Each plot should include the data and the corresponding fitted curves along with a legend labeling each curve or set of points. The second plot should be similar to Figure 4 for the three-lump model and Figure 5 for the four-lump model.

You will probably want to use `L13example1.py` and `L13example2.py` for getting started.

3.2. .ipynb files (9%)

The purpose of this part of the assignment is to create documentation for the Python files using Jupyter. You should complete this task after you have completed Section 3.1. Create two Jupyter notebooks titled `param_estim_3lump.ipynb` and `param_estim_4lump.ipynb`. The following steps should be repeated to document `param_estim_3lump.py` and `param_estim_4lump.py`, respectively, by Jupyter notebooks:

- Level 1 Markdown heading (using # symbol) with text with the name of the corresponding Python code (e.g., `param_estim_3lump.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 the system of ODEs in the three-lump or four-lump model. 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 `param_estim_3lump.py` or `param_estim_4lump.py`.
- Jupyter cell of Python 3 code that
  - runs the .py file.
  - prints the parameter values for the model
  - generate the two figures of the output from the Python simulations for the model.
- 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.