BPL_IEC_validation script with FMPy ver 0.3.15

The key library FMPy ver 0.3.15 is installed.

After the installation a small application BPL_IEC_validation is loaded and run. You can continue with this example if you like.

```
!lsb_release -a # Actual VM Ubuntu version used by Google
```

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
```

```
%env PYTHONPATH=
```

```
env: PYTHONPATH=
```

```
!wget https://repo.anaconda.com/miniconda/Miniconda3-py310_23.1.0-1-Linux-x86_64.sh
!chmod +x Miniconda3-py310_23.1.0-1-Linux-x86_64.sh
!bash ./Miniconda3-py310_23.1.0-1-Linux-x86_64.sh -b -f -p /usr/local
import sys
sys.path.append('/usr/local/lib/python3.10/site-packages/')
```

```
--2024-03-08 20:14:14--  https://repo.anaconda.com/miniconda/Miniconda3-py310_23.1.0-1-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.130.3, 104.16.131.3, 2606:4700::6810:8303, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.130.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 74403966 (71M) [application/x-sh]
Saving to: ‘Miniconda3-py310_23.1.0-1-Linux-x86_64.sh’

Miniconda3-py310_23 100%[===================>]  70.96M   205MB/s    in 0.3s

2024-03-08 20:14:15 (205 MB/s) - ‘Miniconda3-py310_23.1.0-1-Linux-x86_64.sh’ saved [74403966/74403966]

PREFIX=/usr/local
Unpacking payload ...

Installing base environment...


Downloading and Extracting Packages


Downloading and Extracting Packages

Preparing transaction: done
Executing transaction: done
installation finished.
```

```
!conda update -n base -c defaults conda --yes
```

```
    Preparing transaction: done
    Verifying transaction: done
    Executing transaction: done
```

```
!conda --version
!python --version
```

```
    conda 24.1.2
    Python 3.10.13
```

```
!conda install -c conda-forge fmpy --yes # Install the key package
```

```
    Preparing transaction: done
    Verifying transaction: done
    Executing transaction: done
```

```
!conda install matplotlib --yes
```

```
    Channels:
     - defaults
     - conda-forge
    Platform: linux-64
    Collecting package metadata (repodata.json): done
    Solving environment: done

    ## Package Plan ##

      environment location: /usr/local

      added / updated specs:
        - matplotlib


    The following packages will be downloaded:

        package                    |            build
        ---------------------------|-----------------
        matplotlib-3.8.0           |  py310h06a4308_0           8 KB
        matplotlib-base-3.8.0      |  py310h1128e8f_0         6.8 MB
        pyparsing-3.0.9            |  py310h06a4308_0         153 KB
        ------------------------------------------------------------
                                               Total:         7.0 MB

    The following NEW packages will be INSTALLED:

      matplotlib         pkgs/main/linux-64::matplotlib-3.8.0-py310h06a4308_0

    The following packages will be UPDATED:

      matplotlib-base    conda-forge::matplotlib-base-3.5.2-py~ --> pkgs/main::matplotlib-base-3.8.0-py310h1128e8f_0

    The following packages will be SUPERSEDED by a higher-priority channel:

      certifi            conda-forge/noarch::certifi-2024.2.2-~ --> pkgs/main/linux-64::certifi-2024.2.2-py310h06a4308_0
      conda              conda-forge::conda-24.1.2-py310hff520~ --> pkgs/main::conda-24.1.2-py310h06a4308_0
      pyparsing          conda-forge/noarch::pyparsing-3.1.2-p~ --> pkgs/main/linux-64::pyparsing-3.0.9-py310h06a4308_0



    Downloading and Extracting Packages:
    matplotlib-base-3.8. | 6.8 MB    | :   0% 0/1 [00:00<?, ?it/s]
    pyparsing-3.0.9      | 153 KB    | :   0% 0/1 [00:00<?, ?it/s]

    matplotlib-base-3.8. | 6.8 MB    | :   2% 0.018342538281959999/1 [00:00<00:05,  5.49s/it]
    pyparsing-3.0.9      | 153 KB    | :  10% 0.10471552197977783/1 [00:00<00:00,  1.02s/it]

    matplotlib-3.8.0     | 8 KB      | : 100% 1.0/1 [00:00<00:00,  9.52it/s]

    matplotlib-3.8.0     | 8 KB      | : 100% 1.0/1 [00:00<00:00,  9.52it/s]




    Preparing transaction: done
    Verifying transaction: done
    Executing transaction: done
```

Now specific installation and the run simulations. Start with connecting to Github. Then upload the two files:

- FMU - BPL_IEC_operation_linux_om_me.fmu
- Setup-file - BPL_IEC_operation_fmpy_explore.py

```
%%bash
git clone https://github.com/janpeter19/BPL_IEC_validation
```

```
    Cloning into 'BPL_IEC_validation'...
```

```
%cd BPL_IEC_validation
```

```
    /content/BPL_IEC_validation
```

BPL IEC validation

Author: Jan Peter Axelsson

Here I try to reproduce somre results from Jonas Månssons master thesis report "Control of chromatography comlumn in production scale", TFRT-5599.

The interaction withe the model has changed slightly so that linear flow rate LF = F/area is used. The column volume is now specified in terms of (cross-section) area times height. Thus by changing area the scale up works fine and what you do in practice. Further the time unit changed from seconds to minutes.

The model has 5 states as listed below, 3 states in liquid mobile phase and 2 states for the gel for bound proteins. In a few of Jonas figures also bound ions of the buffer are plotted but I do not do that. Bound ions of the buffer can be calculated from the difference of total binding capaciety Q_{av} and bound protein and bound protein antagonist, see section 5.1 in the report.

The molecular weights listed below are not used in the simulations. They just give typical values and the molecular weight for bound protein and antagonist protein is just arbitrary here.

Parameters of the model in general as well as time scale are all arbitrary and focus is on qualitative aspects of the model.

The height of the column is 20 cm which is a common size in the industry.

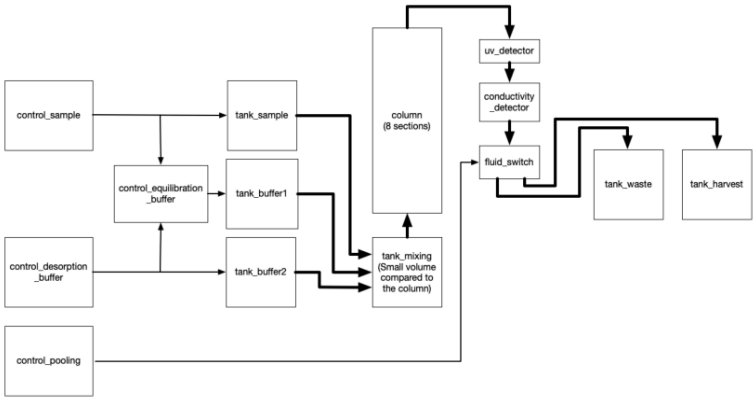
```
run -i BPL_IEC_fmpy_explore.py
```

```
    Linux - run FMU pre-comiled OpenModelica 1.21.0

    Model for bioreactor has been setup. Key commands:
     - par()       - change of parameters and initial values
     - init()      - change initial values only
     - simu()      - simulate and plot
     - newplot()   - make a new plot
     - show()      - show plot from previous simulation
     - disp()      - display parameters and initial values from the last simulation
     - describe()  - describe culture, broth, parameters, variables with values/units

    Note that both disp() and describe() takes values from the last simulation
    and the command process_diagram() brings up the main configuration

    Brief information about a command by help(), eg help(simu)
    Key system information is listed with the command system_info()
```

```
plt.rcParams['figure.figsize'] = [30/2.54, 24/2.54]
```

```
# The process diagram is made outside Modelica to illustrate the configuration
process_diagram()
```

```
    No processDiagram.png file in the FMU, but try the file on disk.
```




```
describe('chromatography'); print(); #describe('liquidphase')
```

```
    Ion exchange chromatorgraphy controlled with varying salt-concentration. The pH is kept constant.
```

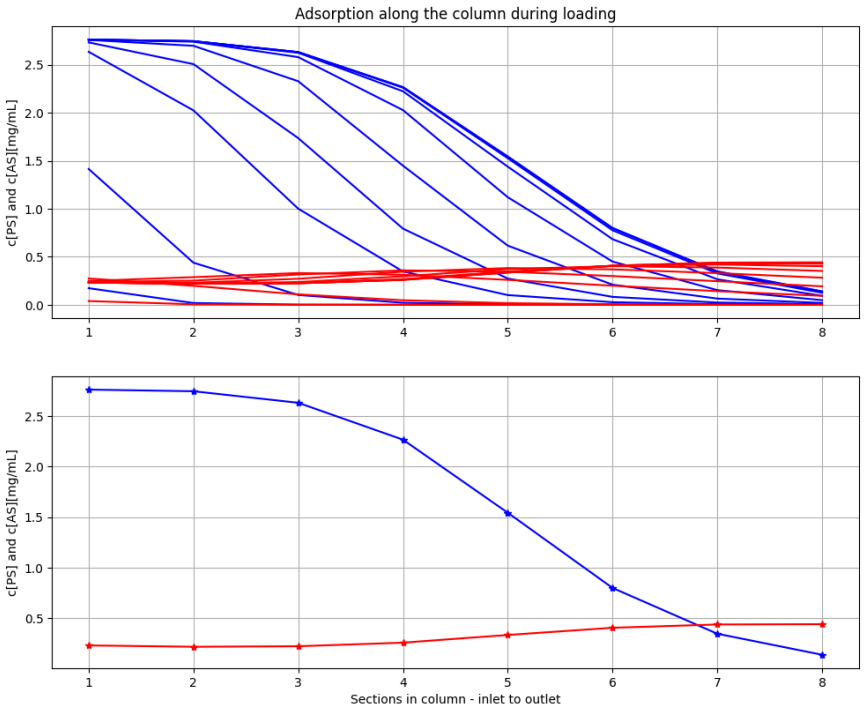
Loading or adsorption

The parameter notation and values are the same as in the referred report. However the flow rate is here denoted F while q in the report. The column is diveded in n=8 sections and set at compilation time. The values are arbitrarly chosen in the report and the focus is on qualitative aspects of the model.

The simplified model describe only the column in terms of volume and does not distinguish a high column with a small diameter from a lower with larger diameter.

The parameters k1, k2, k3, k4 and Q_av are given relative volume and with increased column volume a larger capacity is thus obtained.

```
# Loading of the column - try to reproduce Jonas figure 13.
newplot(title='Adsorption along the column during loading', plotType='Loading')

# Sample
par(P_in=1.0, A_in=1.0, E_in=0)

# Column properties
par(k1=0.3, k2=0.05, k3=0.05, k4=0.3, Q_av=3.0)
par(height=20, diameter=0.714)
par(x_m=0.3)

# Operation
par(E_in_desorption_buffer=8)
par(LFR=12)
par(scale_volume=False)
par(start_adsorption=0, stop_adsorption=50)
par(start_desorption=150, stationary_desorption=450)
par(start_pooling=220, stop_pooling=450)

# Simulation
simu(100)
```



The results are the same af Figure 13 in [1].

```
# We just check that we had the same volume flow rate as Jonas
describe('F')
```

```
    Column ackumulated volume flow : 0.08 [ mL ]
```

```
describe('V')
```

```
    Column volume total - derived : 8.008 [ mL ]
```

```
model_get('column.x_m')
```

```
    0.3
```

```
model_get('column.V_m')
```

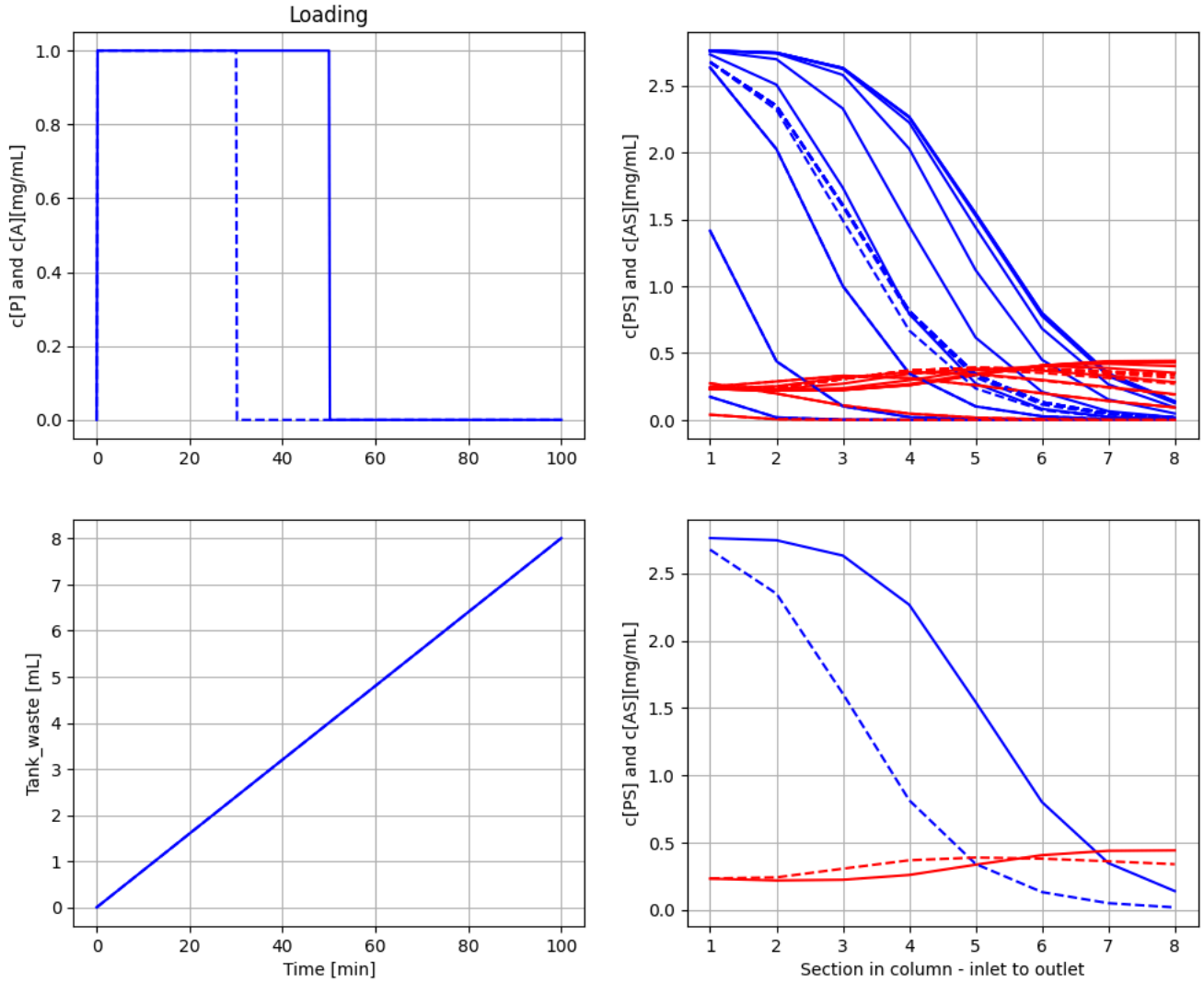
```
    2.4023570526441924
```

```
#describe('column.n')
```

```
model_get('column.column_section[1].V_m')
```

```
    0.30029463158052405
```

```
# Impact of shorter time for loading and then less material
newplot(title='Loading', plotType='Loading-combined')
show()

# Simulation with changed parameter t2
par(stop_adsorption=30); simu(100)

# Reset changed parameter
par(stop_adsorption=50)
```



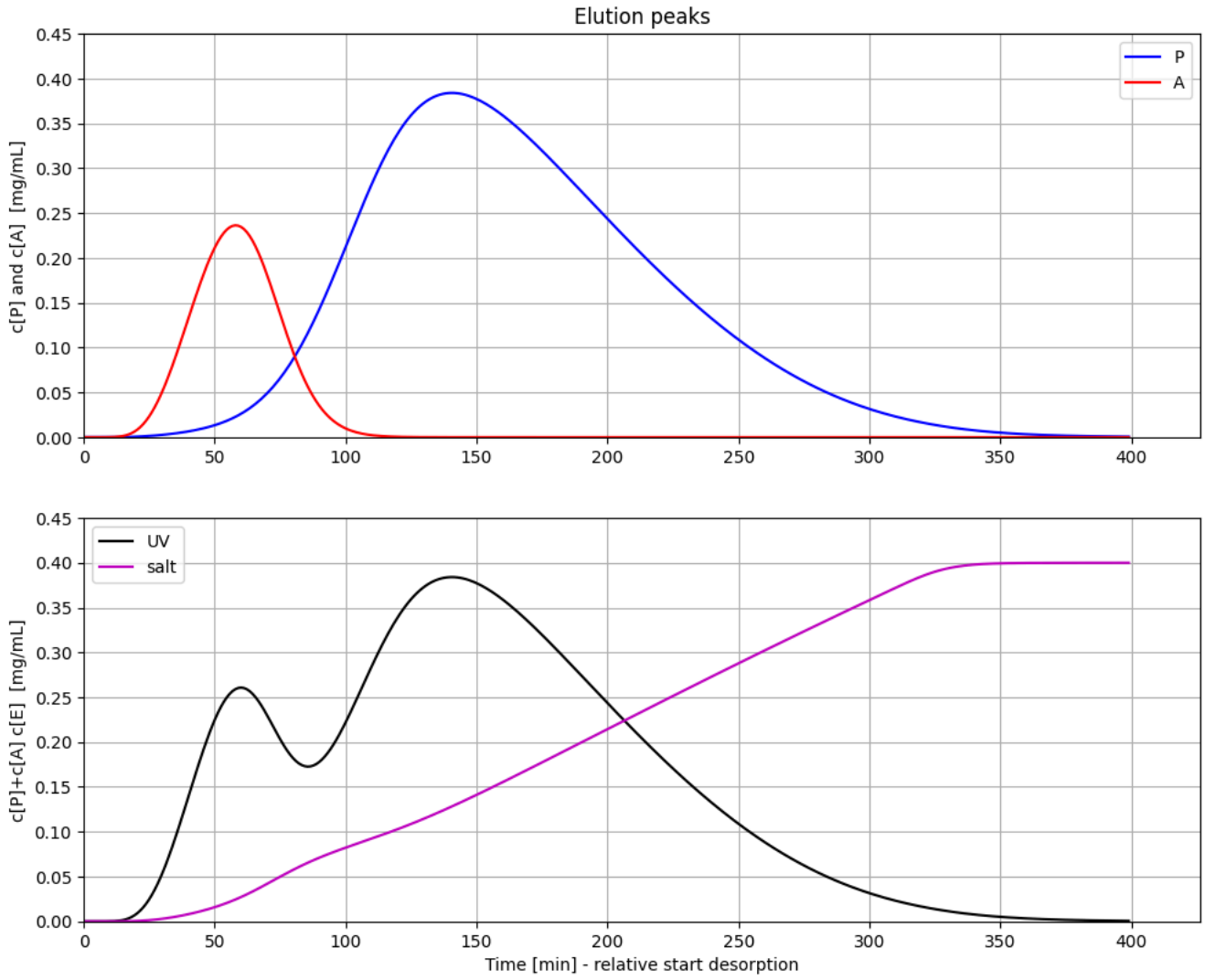
To the left the inlet loading over time. To the right upper concentration along the column at different times and in steady states finally To the right lower concentrations along the colunn in steady state.

We see that a shorter time and then less material makes less of the column capacity used.

Note that the flow through the column is constant despite change from sample to just buffer 1, and shown in how the volume of the waste tank increase with time.

Elution or desorption

```
# Elution of the column
newplot(title='Elution peaks', plotType='Elution')

# Sample
par(P_in=1, A_in=1.0, E_in=0)

# Operation
par(E_in_desorption_buffer=8)
par(LFR=12.0, start_adsorption=0, stop_adsorption=50, start_desorption=150, stationary_desorption=450)

# Simulation
simu(550)
```



The results are the same af Figure 14 in [1].

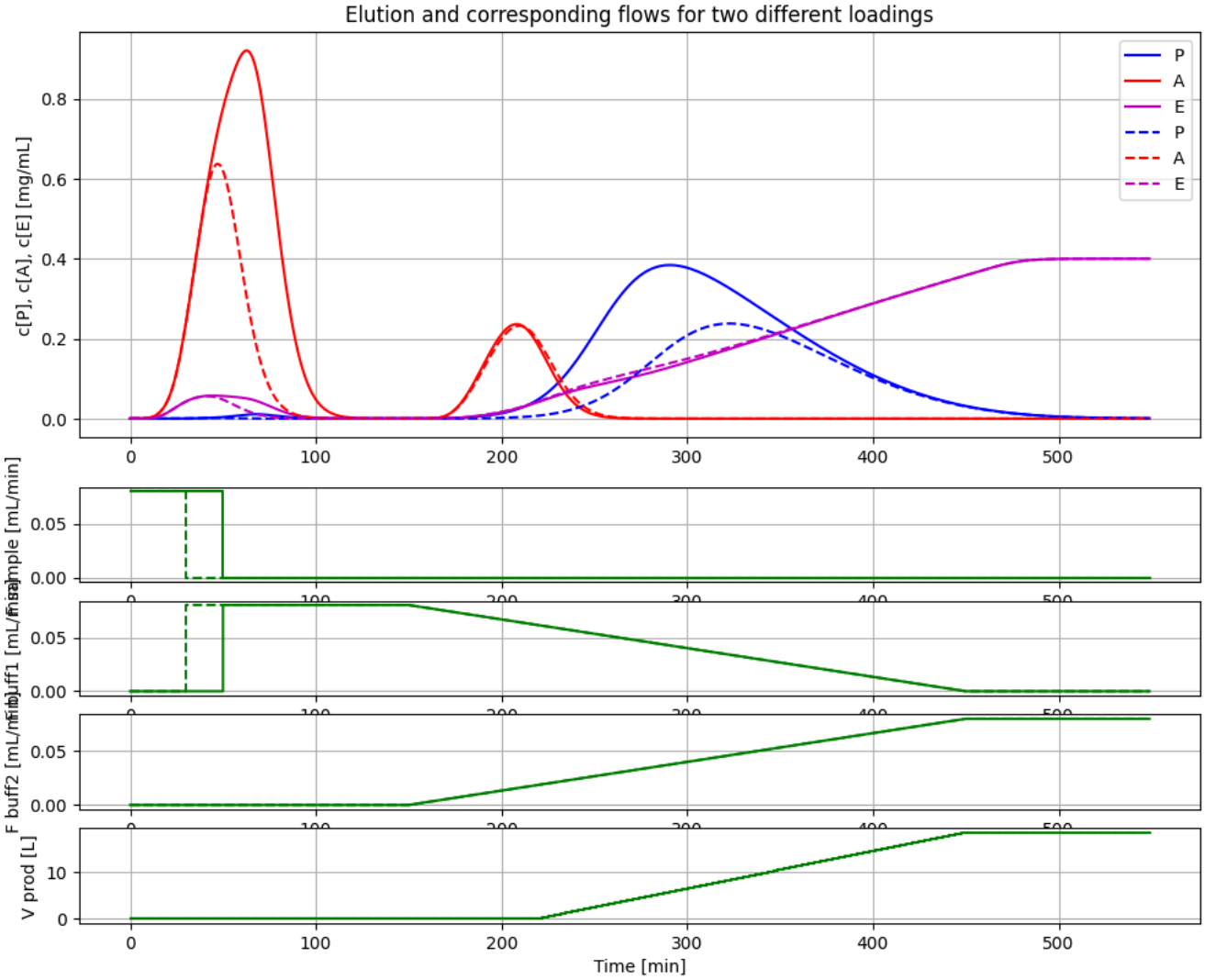
The upper diagrams shows the column outlet concentrations of P and A over time.

The lower diagram shows the sum (or possibly the UV signal) at column outlet as well as the salt concentration. We have some separation between the two peaks.

Note that the salt concentration deviates slightly from the linear increse between time 50 to 100. This is due to ion interaction with P and A in the column. This is phenomenon can also be seen in real data. The ion-salt concentration is scaled with factor 0.05 to get comparable concentrations to P and A.

I have here simulated time 150 of adsorbtion and then started elution. Here is time counted as zero at time of start of elutions. Not sure how long Jonas simulated to get steady state before he did elution.

```
# More complete visualization of the elution phase and the different flows
newplot(title='Elution and corresponding flows for two different loadings', plotType='Elution-combined')
par(stop_adsorption=50); simu(550)

# Simulation with changed parameter t2
par(stop_adsorption=30); simu(550)

# Reset changed parameter
par(stop_adsorption=50)
```

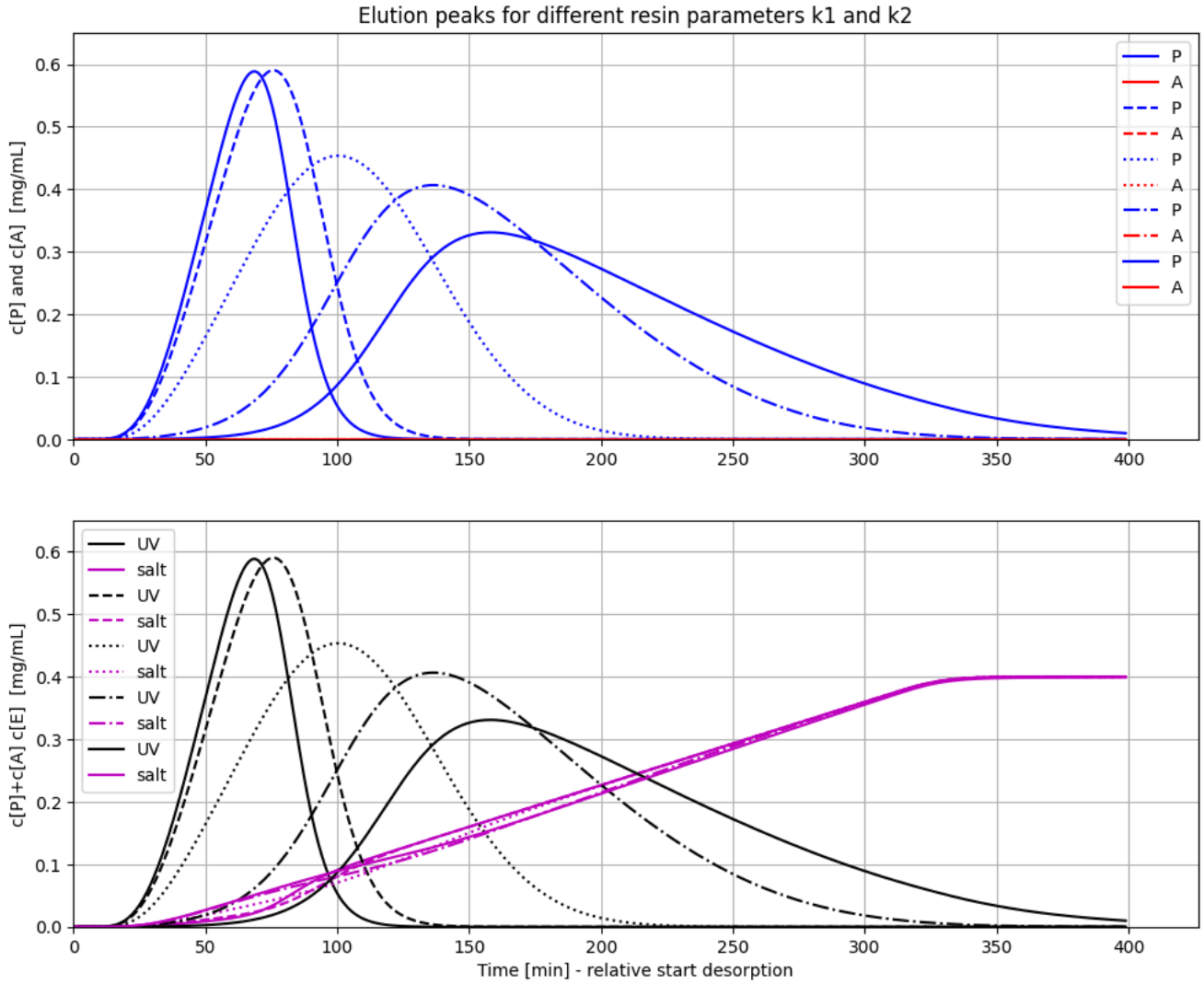


Here a diagram that shows the peaks at the outlet as shown in the previous diagram. Below the flow rates of the three differnet sorces. Here time is 0 at start of adsorbtion and elution starts at time 150.

Automatic pooling based on UV-mmeasurement is tested in another notebook.

Change of resin properties

```
# Elution of the column
newplot(title='Elution peaks for different resin parameters k1 and k2', plotType='Elution')

# Sample
par(P_in=1, A_in=0.0, E_in=0)

# Operation
par(E_in_desorption_buffer=8)
par(LFR=12.0, start_adsorption=0, stop_adsorption=50, start_desorption=150, stationary_desorption=450)

# Simulations
par(k1=0.05, k2=0.50); simu(550)
par(k1=0.05, k2=0.25); simu(550)
par(k1=0.05, k2=0.05); simu(550)
par(k1=0.25, k2=0.05); simu(550)
par(k1=0.50, k2=0.05); simu(550)

# Adjust diagrams
ax1.set_ylim([0, 0.65])
ax2.set_ylim([0, 0.65])
plt.show()
```



The results are the same af Figure 17 in [1].

Summary

Three important diagrams Figure 13, 14 and 17 in the original report [1] were reproduced and the implementation in Modelica used here is considered validated.

The model is now extended with an improved parametrization of the column that match the industrial practice.

Acknowledgement

The author thank Karl Johan Brink for sharing his know-how of chromatography operation. He has especially given input to how to parametrize the model in terms often used in the industrial practice.

References

1) Månsson, Jonas, "Control of chromatography comlumn in production scale", Master thesis TFRT-5599, Department of Automatic Control, LTH, Lund Sweden, 1998.

2) Pharmacia LKB Biotechnology. "Ion Exchange chromatography. Principles and Methods.", 3rd edition, 1991. 3) Jungbauer, Alois and Giorgio Carta, "Protein Chromatography: Process Development and Scale-Up", Wiley 2nd edition, 2020.

Appendix

```
describe('MSL')
```

```
    MSL: 3.2.3 - used components: RealInput, RealOutput, CombiTimeTable, Types
```

```
system_info()
```

```
    System information
     -OS: Linux
     -Python: 3.10.12
     -Scipy: not installed in the notebook
     -FMPy: 0.3.19
     -FMU by: OpenModelica Compiler OpenModelica 1.21.0
     -FMI: 2.0
     -Type: ME
     -Name: BPL_IEC.Column_system
     -Generated: 2024-03-08T10:56:20Z
     -MSL: 3.2.3
     -Description: Bioprocess Library version 2.1.2 prel
     -Interaction: FMU-explore for FMPy version 0.9.9
```