

✓ BPL_TEST2_Batch script with FMPy ver 0.3.15

The key library FMPy ver 0.3.15 is installed.

After the installation a small application BPL_TEST2_Batch 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-py39_23.1.0-1-Linux-x86_64.sh
!chmod +x Miniconda3-py39_23.1.0-1-Linux-x86_64.sh
!bash ./Miniconda3-py39_23.1.0-1-Linux-x86_64.sh -b -f -p /usr/local
import sys
sys.path.append('/usr/local/lib/python3.9/site-packages/')
```

Show hidden output

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

Show hidden output

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

conda 23.11.0
Python 3.9.18
```

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

Show hidden output

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

Show hidden output

```
#!conda install scipy --yes
```

```
#!conda install xlrd --yes
```

```
#!conda install openpyxl --yes
```

✓ BPL_TEST2_Batch setup

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

- FMU - BPL_TEST2_Batch_linux_om_me.fmu
- Setup-file - BPL_TEST2_Batch_fmpy_explore.py

```
%bash
git clone https://github.com/janpeter19/BPL_TEST2_Batch

Cloning into 'BPL_TEST2_Batch'...

%cd BPL_TEST2_Batch

/content/BPL_TEST2_Batch

run -i BPL_TEST2_Batch_fmpy_explore.py
```

Linux – run FMU pre-compiled 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()

```
%matplotlib inline
plt.rcParams['figure.figsize'] = [25/2.54, 20/2.54]
```

```
import warnings
warnings.filterwarnings("ignore")
```

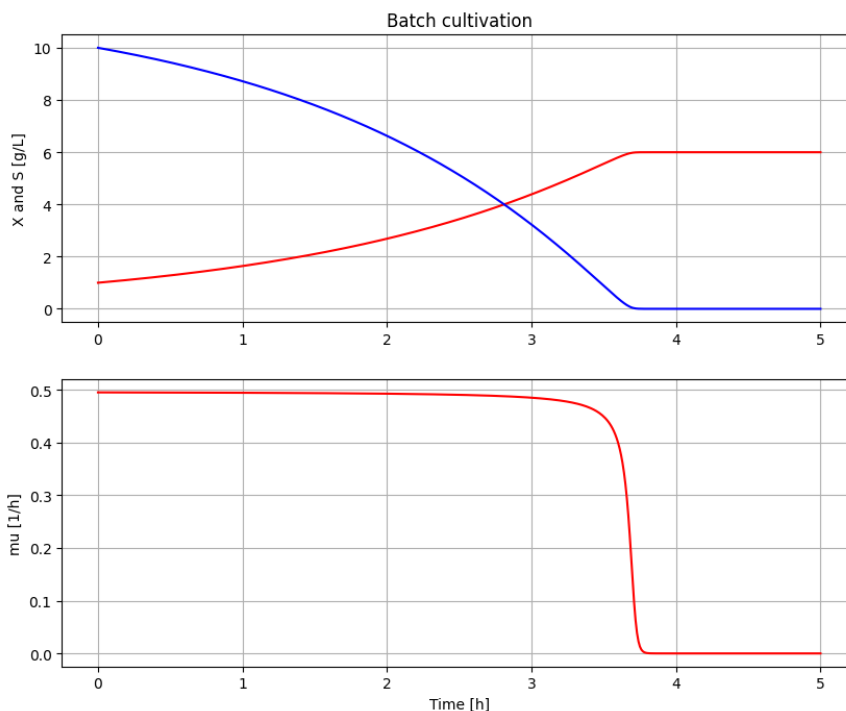
✓ BPL_TEST2_Batch - demo

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

Pump schedule parameter

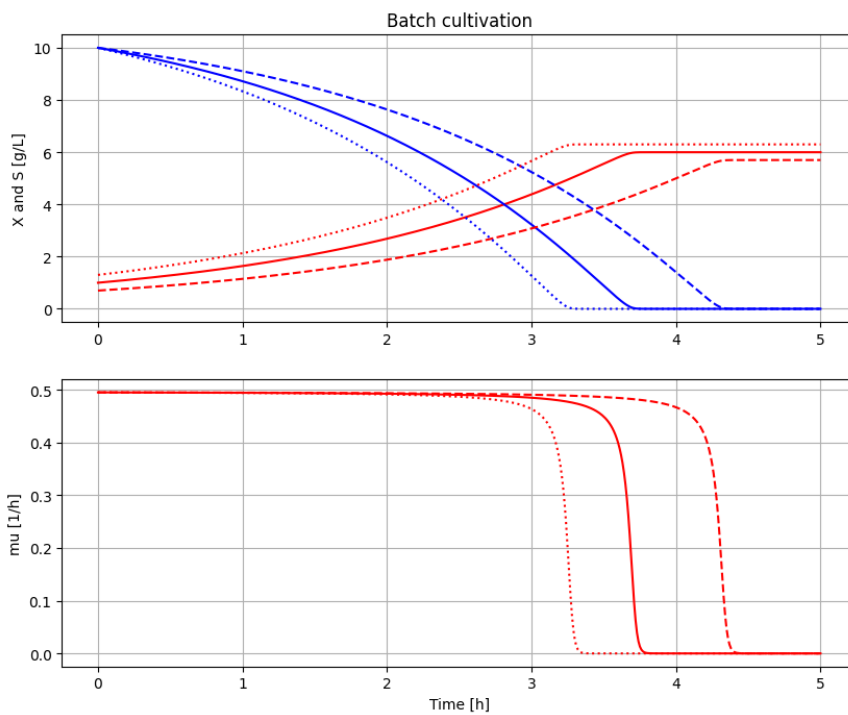
Simplified text book model – only substrate S and cell concentration X

```
# Simulation with default values of the process
newplot(plotType='TimeSeries')
simu()
```



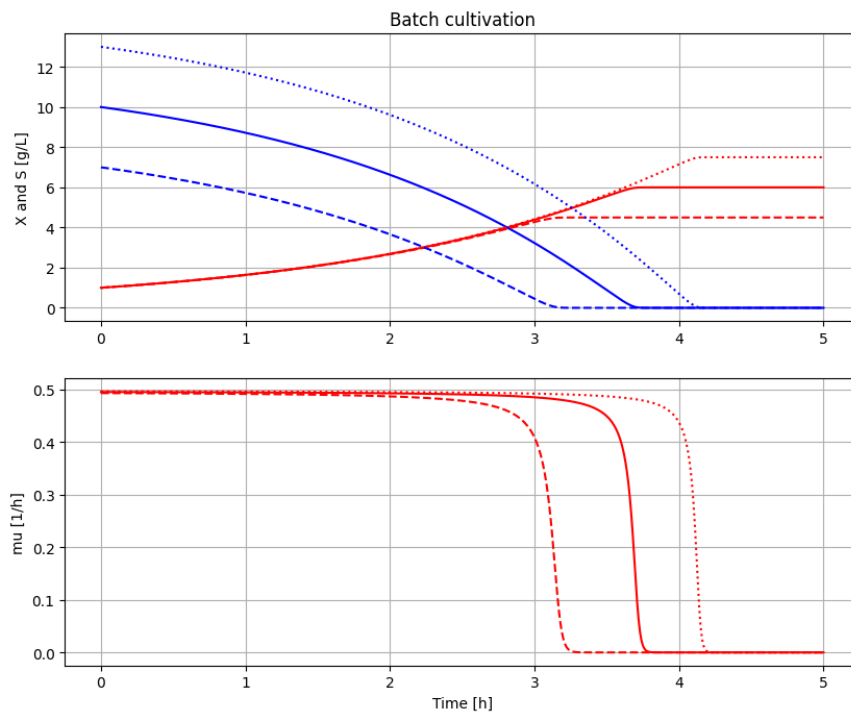
```
# Simulation were initial value of biomass VX_0 is varied
newplot(plotType='TimeSeries')
for value in [1.0, 0.7, 1.3]: init(VX_0=value); simu(5)
```

```
# Restore default value of VX_0
init(VX_0=1.0)
```



```
# Simulation were initial value of substrate VS_0 is varied
newplot(plotType='TimeSeries')
for value in [10, 7, 13]: init(VS_0=value); simu(5)
```

```
# Restore default value of VS_0
init(VS_0=10)
```



```
# Simulation where metabolism is changed after 3 hours
newplot(plotType='TimeSeries')
simu(5)

simu(3)
par(Y=0.4, qSmax=1.0/(0.4/0.5)); simu(2, 'cont')

# Restore default value of Y and qSmax
par(Y=0.5, qSmax=1.0)
```

