˅ BPL_TEST2_Batch script with FMPy

The key library FMPy 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-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/')
```

```
    --2024-03-07 08:51:14--  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::6810:8203, ...
    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   245MB/s    in 0.3s

    2024-03-07 08:51:15 (245 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-3.8.0     | 8 KB       |  :    0% 0/1 [00:00<?, ?it/s]
        pyparsing-3.0.9      | 153 KB     |  :   94% 0.9424396978180005/1 [00:00<00:00,  9.40it/s]

        matplotlib-base-3.8. | 6.8 MB     |  :    0% 0.0022928172852449986/1 [00:00<00:49, 49.39s/it]

        matplotlib-3.8.0     | 8 KB       |  : 100% 1.0/1 [00:00<00:00,  9.23it/s]



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

#!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")
```
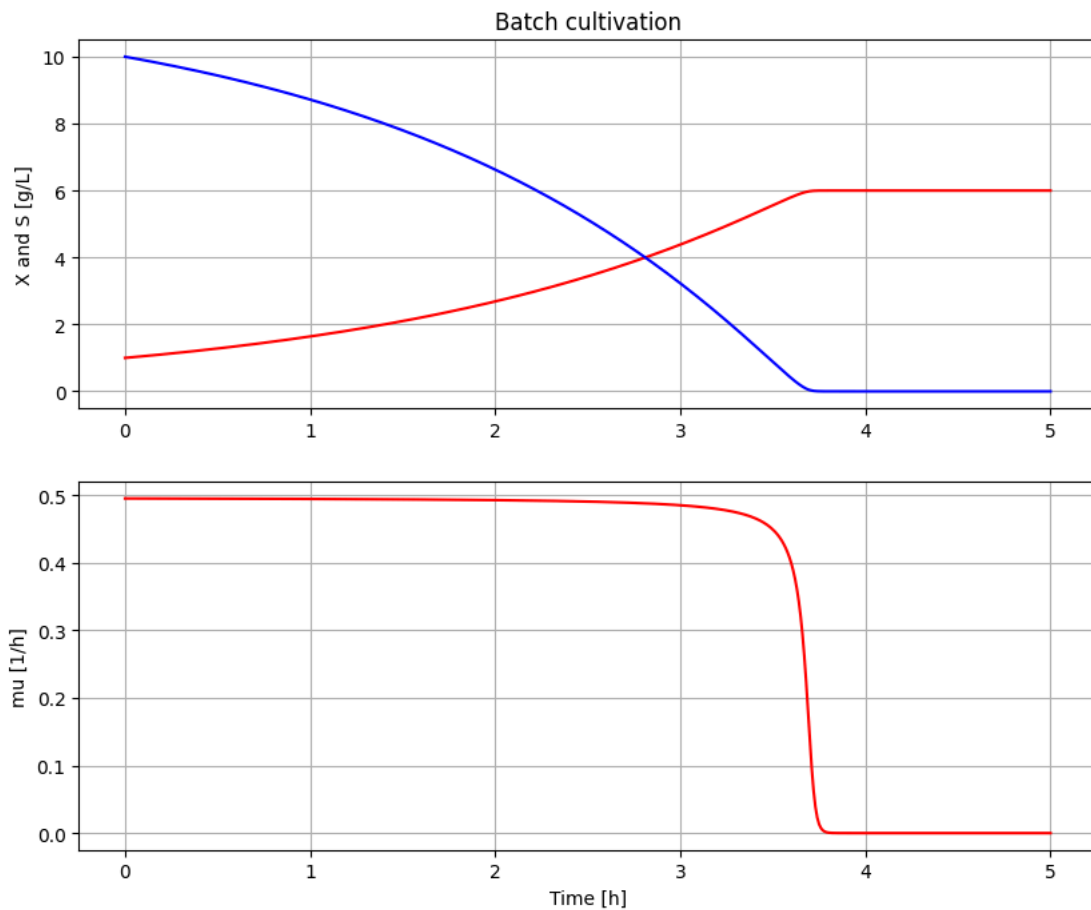
⌄ BPL_TEST2_Batch - demo

```
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()
```
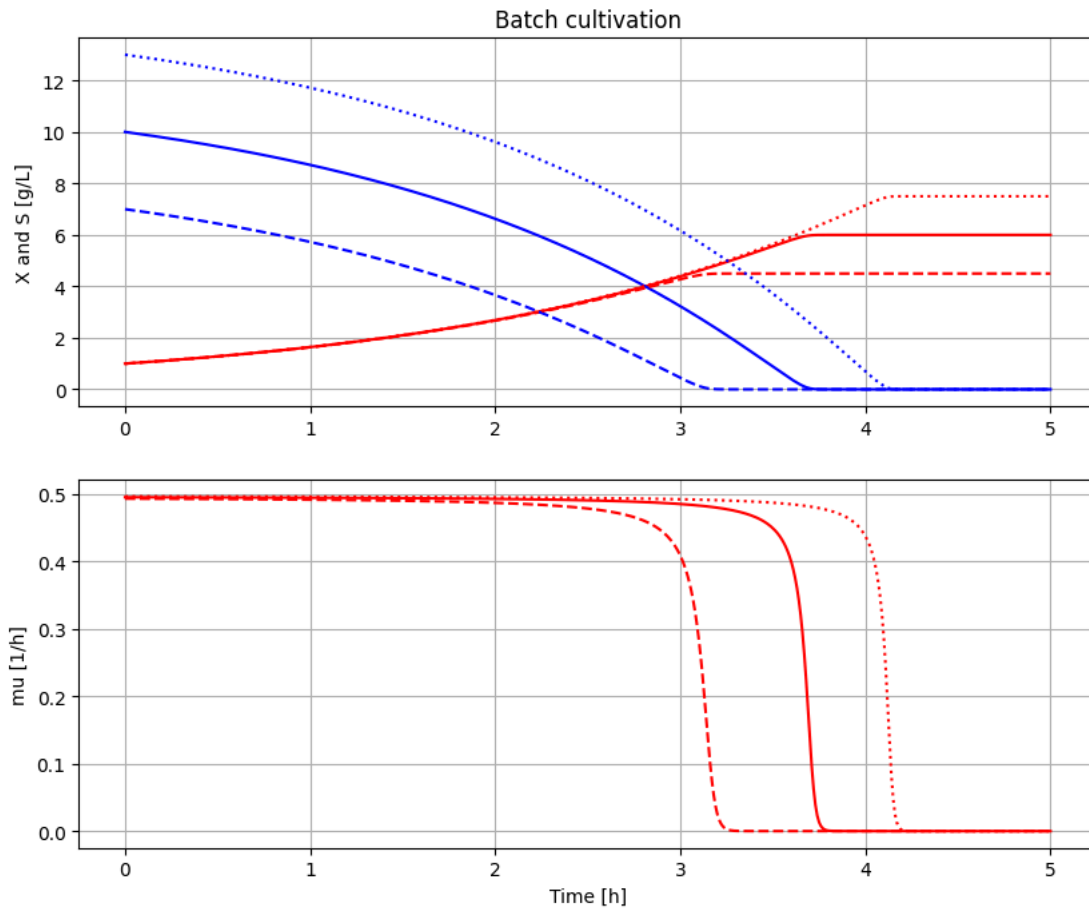
Batch cultivation


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

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

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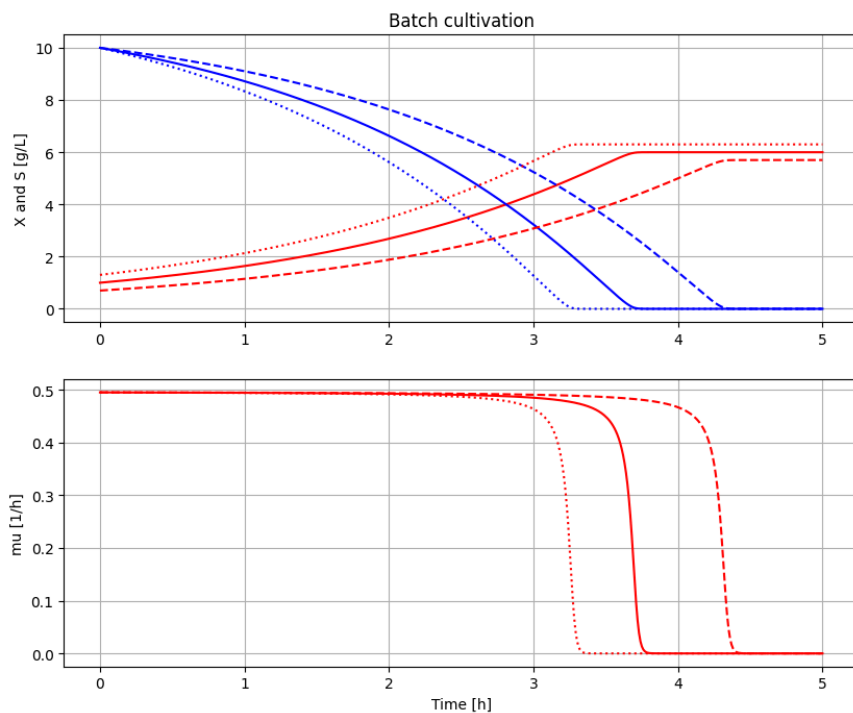
```
# Simulation were initial value of substrate VS_start is varied
newplot(plotType='TimeSeries')
for value in [10, 7, 13]: init(VS_start=value); simu(5)

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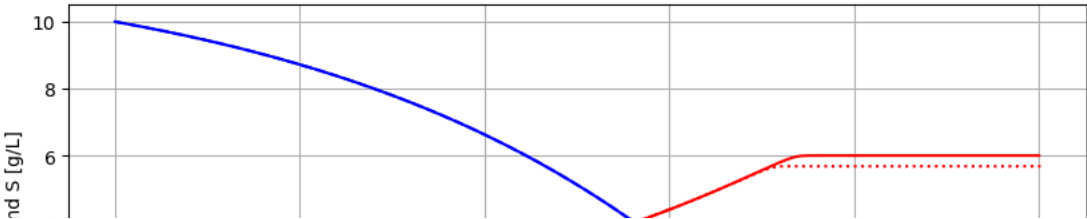

Batch cultivation

```
# 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)
```

Batch cultivation


```
disp('culture')

    Y : 0.4
    qSmax : 1.25
    Ks : 0.1
```

```
# Growth rate variable at the end of the cultivation
describe('mu')

    Cell specific growth rate variable : -0.0 [ 1/h ]
```

```
describe('parts')

    ['bioreactor', 'bioreactor.culture']
```

```
describe('MSL')

    MSL: 3.2.3 – used components: none
```

```
system_info()

    System information
     -OS: Linux
     -Python: 3.10.12
```