▾ BPL_YEAST_COB_Batch script with PyFMI

The key library PyFMI is installed.

After the installation a small application BPL_YEAST_COB_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 22.04.2 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/')

    --2023-08-15 19:18:41--  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   184MB/s    in 0.4s

    2023-08-15 19:18:41 (184 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 23.7.2
    Python 3.10.12
```

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

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

```
!pip install optlang
```

```
    Collecting optlang
      Downloading optlang-1.7.0-py2.py3-none-any.whl (138 kB)
                                    —————— 138.3/138.3 kB 5.8 MB/s eta 0:00:00
    Collecting swiglpk>=5.0.8
      Downloading swiglpk-5.0.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
                                    —————— 2.3/2.3 MB 20.4 MB/s eta 0:00:00
    Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/site-packages (from optlang) (1.16.0)
    Collecting sympy>=1.12.0
      Downloading sympy-1.12-py3-none-any.whl (5.7 MB)
                                    —————— 5.7/5.7 MB 79.0 MB/s eta 0:00:00
    Collecting mpmath>=0.19
      Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
                                    —————— 536.2/536.2 kB 45.5 MB/s eta 0:00:00
    Installing collected packages: swiglpk, mpmath, sympy, optlang
    Successfully installed mpmath-1.3.0 optlang-1.7.0 swiglpk-5.0.8 sympy-1.12
    WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the sys
```

▾ Notes YEAST_COB_Batch constraint-based approach

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

- FMU - BPL_YEAST_AIR_Fedbatch_linux_jm_cs.fmu
- Setup-file - BPL_YEAST_AIR_Fedbatch_explore

```
%%bash
git clone https://github.com/janpeter19/CONF_2023_10_MODELICA15
```

```
    Cloning into 'CONF_2023_10_MODELICA15'...
```

```
%cd CONF_2023_10_MODELICA15
```

```
    /content/CONF_2023_10_MODELICA15
```

```
run -i BPL_YEAST_COB_Batch_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

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

```
plt.rcParams['figure.figsize'] = [20/2.54, 16/2.54]
```

▾ Try using LP in each step

```python
from optlang import Model, Variable, Constraint, Objective


# Define culture constraint-based model
def culture(G, E):

    # LP calculation of the optimal qGr, qEr based on G and E values

    # - parameters
    qO2max = 6.9e-3; kog = 2.3; koe = 1.6; YGr = 3.5; YEr = 1.32;
    alpha = 0.01; beta = 1.0

    # - transfer data from dynamic reactor model to static LP model
    qGr_opt = Variable('qGr_opt', lb=0)
    qEr_opt = Variable('qEr_opt', lb=0)

    # - LP model constraint and objective
    mu_max = Objective(YGr*qGr_opt + YEr*qEr_opt, direction='max')
    qO2lim = Constraint(kog*qGr_opt + koe*qEr_opt, ub=qO2max)
    qGlim = Constraint(qGr_opt, ub=alpha*max(0,G))
    qElim = Constraint(qEr_opt, ub=beta*max(0,E))

    # - put together the LP model
    yeast_model = Model(name='Yeast bottleneck model')
    yeast_model.objective = mu_max
    yeast_model.add(qO2lim)
    yeast_model.add(qGlim)
    yeast_model.add(qElim)

    # - do LP optimization
    yeast_model.optimize()

    return (yeast_model.objective.value, yeast_model.variables.qGr_opt.primal, yeast_model.variables.qEr_opt.primal,

# Initialization
V_0=1.0
init(V_0=V_0, VX_0=V_0*2.0, VG_0=V_0*10, VE_0=3.0)


# Loop of simulations
t_final = 8.0
t_samp = 0.0333
n_samp = t_final/t_samp + 1


# Simulate n sample steps
newplot(title='Batch cultivation', plotType='TimeSeries2')
ax1.set_xlim([0, t_final]); ax2.set_xlim([0, t_final]); ax3.set_xlim([0, t_final])

simu(t_samp, options=opts_fast)
for i in range(int(n_samp)):
    (mum_opt, qGr_opt, qEr_opt, qO2_opt) = culture(sim_res['bioreactor.c[2]'][-1], sim_res['bioreactor.c[3]'][-1])
    par(mum=mum_opt, qGr=qGr_opt, qEr=qEr_opt, qO2=qO2_opt)
    simu(t_samp, 'cont', options=opts_fast)
```
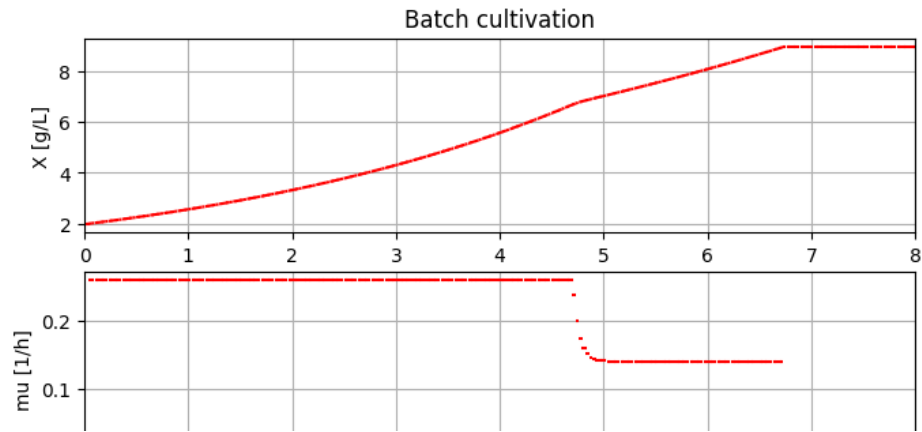
Batch cultivation


```
system_info()
```

```
System information
 -OS: Linux
 -Python: 3.10.12
 -Scipy: not installed in the notebook
 -PyFMI: 2.11.0
 -FMU by: OpenModelica Compiler OpenModelica 1.21.0
 -FMI: 2.0
 -Type: FMUModelME2
 -Name: BPL_YEAST_COB.Batch
 -Generated: 2023-05-31T09:43:28Z
 -MSL: 3.2.3
 -Description: Bioprocess Library version 2.1.1
 -Interaction: FMU-explore version 0.9.7
```

```
!conda list optlang
```

```
# packages in environment at /usr/local:
#
# Name                    Version                   Build  Channel
optlang                   1.7.0                    pypi_0    pypi
```

✓ 1s   completed at 21:21   ● ✕