⌄ BPL_YEAST_COB_Batch script with FMPy

The key library FMPy 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 22.04.3 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/')
```

Show hidden output

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

Show hidden output

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

Show hidden output

```
!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
```

```
# For some reason conda installation does not work for optlang
!pip install optlang
```

Show hidden output

⌄ BPL_YEAST_COB_Batch setup

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

- FMU - BPL_YEAST_COB_Batch_linux_om_me.fmu
- Setup-file - BPL_YEAST_COB_Batch_fmpy_explore.py

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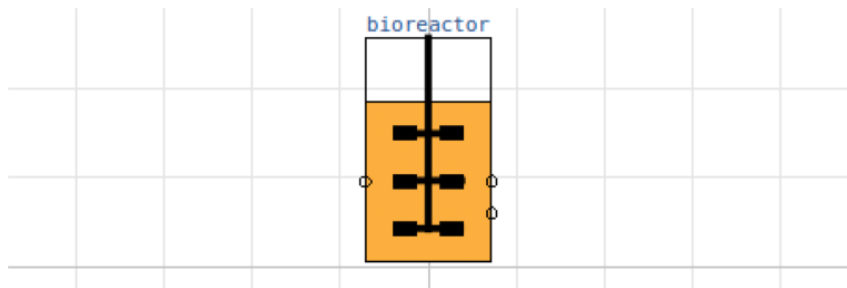
```
%matplotlib inline
plt.rcParams['figure.figsize'] = [25/2.54, 20/2.54]
```

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

∨ BPL_YEAST_COB_Batch - demo

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

```
process_diagram()
```



```
describe('culture'); print(); #describe('liquidphase')                           # Pump schedule parameter
```

```
    Saccharomyces cerevisae - default parameters for strain H1022
```

```python
# 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, qO2lim.


# 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)
```
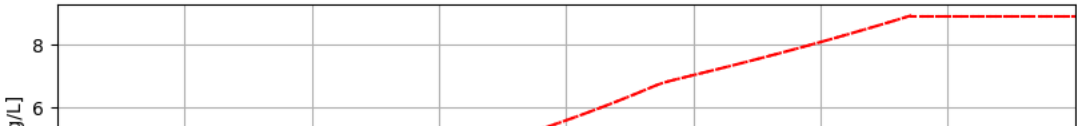
Batch cultivation


```
system_info()
```

System information
 —OS: Linux
 —Python: 3.10.12
 —Scipy: not installed in the notebook
 —FMPy: 0.3.19
 —FMU by: OpenModelica Compiler OpenModelica 1.21.0
 —FMI: 2.0
 —Type: ME
 —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 for FMPy version 0.9.8

```
!conda list optlang
```

packages in environment at /usr/local: