BPL_TEST2_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 22.04.4 LTS
Release:        22.04
Codename:       jammy
```

```
!python --version
```

```
Python 3.11.11
```

```
!pip install fmpy
```

```
Collecting fmpy
  Downloading FMPy-0.3.22-py3-none-any.whl.metadata (1.9 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.11/dist-packages (from fmpy) (25.1.0)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.11/dist-packages (from fmpy) (3.1.5)
Collecting lark (from fmpy)
  Downloading lark-1.2.2-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.11/dist-packages (from fmpy) (5.3.1)
Requirement already satisfied: msgpack in /usr/local/lib/python3.11/dist-packages (from fmpy) (1.1.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from fmpy) (1.26.4)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2->fr
Downloading FMPy-0.3.22-py3-none-any.whl (4.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.9/4.9 MB 19.3 MB/s eta 0:00:00
Downloading lark-1.2.2-py3-none-any.whl (111 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 111.0/111.0 kB 2.5 MB/s eta 0:00:00
Installing collected packages: lark, fmpy
Successfully installed fmpy-0.3.22 lark-1.2.2
```

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

```
Collecting optlang
  Downloading optlang-1.8.3-py2.py3-none-any.whl.metadata (8.2 kB)
Collecting swiglpk>=5.0.12 (from optlang)
  Downloading swiglpk-5.0.12-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.5 kB)
Requirement already satisfied: sympy>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from optlang) (1.
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from sympy>=
Downloading optlang-1.8.3-py2.py3-none-any.whl (141 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 141.8/141.8 kB 5.1 MB/s eta 0:00:00
Downloading swiglpk-5.0.12-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 26.1 MB/s eta 0:00:00
Installing collected packages: swiglpk, optlang
Successfully installed optlang-1.8.3 swiglpk-5.0.12
```

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/BPL_YEAST_COB_Batch
```

```
Cloning into 'BPL_YEAST_COB_Batch'...
```

```
%cd BPL_YEAST_COB_Batch
```

```
/content/BPL_YEAST_COB_Batch
```

```
run -i BPL_YEAST_COB_Batch_fmpy_explore.py
```

```
Linux - run FMU pre-comiled OpenModelica

    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]
```

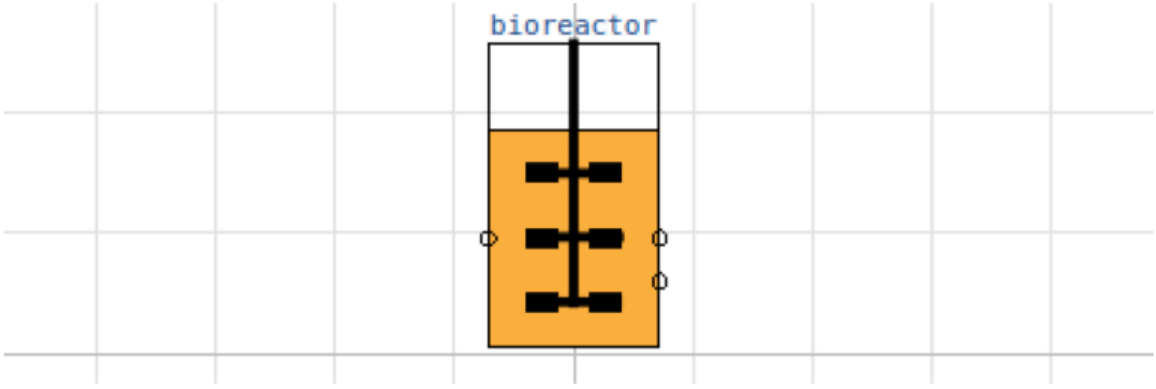
```
import warnings
warnings.filterwarnings("ignore")
```

BPL_YEAST_COB_Batch - demo

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

```
process_diagram()
```

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

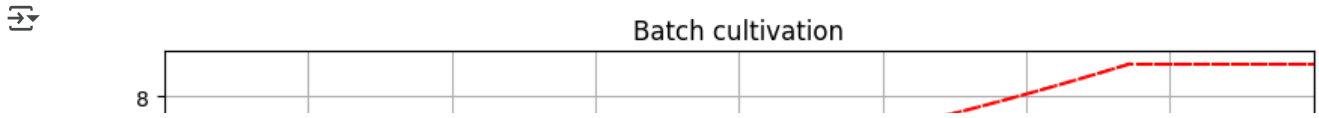



```
describe('culture'); print(); #describe('liquidphase')                                    # Pump schedule param
```

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

```
# 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.


    # Initialization
    V_start=1.0
    init(V_start=V_start, VX_start=V_start*2.0, VG_start=V_start*10, VE_start=V_start*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]']
        par(mum=mum_opt, qGr=qGr_opt, qEr=qEr_opt, qO2=qO2_opt)
        simu(t_samp, 'cont', options=opts_fast)
```




```
system_info()
```

```
System information
 -OS: Linux
 -Python: 3.11.11
 -Scipy: not installed in the notebook
 -FMPy: 0.3.22
 -FMU by: OpenModelica Compiler OpenModelica 1.25.0~dev-133-ga5470be
 -FMI: 2.0
 -Type: ME
 -Name: BPL_YEAST_COB.Batch
 -Generated: 2024-11-08T08:21:20Z
 -MSL: 3.2.3
 -Description: Bioprocess Library version 2.3.0
 -Interaction: FMU-explore for FMPy version 1.0.1
```

```
!pip show optlang
```

```
Name: optlang
Version: 1.8.3
Summary: Formulate optimization problems using sympy expressions and solve them using interfaces to third-
Home-page: https://github.com/opencobra/optlang
Author: Nikolaus Sonnenschein
Author-email: niko.sonnenschein@gmail.com
License: Apache-2.0
Location: /usr/local/lib/python3.11/dist-packages
Requires: swiglpk, sympy
Required-by:
```

Start coding or generate with AI.