

## ✓ 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 22.04.3 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/')
```

```
➤ --2024-11-08 07:26:35-- https://repo.anaconda.com/miniconda/Miniconda3-py310_23.1.0-1-Linux-x86_64.sh
  Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.191.158, 104.16.32.241, 2606:4700::6810:bf9e, ...
  Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.191.158|: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 76.7MB/s in 0.9s
```

```
2024-11-08 07:26:36 (76.7 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 23.1.0  
   Python 3.10.15
```

```
!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-1.8.2-py2.py3-none-any.whl (141 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 141.8/141.8 kB 4.2 MB/s eta 0:00:00
Collecting swiglpk>=5.0.8
  Downloading swiglpk-5.0.10-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 38.7 MB/s eta 0:00:00
Collecting sympy>=1.12.0
  Downloading sympy-1.13.3-py3-none-any.whl (6.2 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 79.6 MB/s eta 0:00:00
Collecting mpmath<1.4,>=1.1.0
  Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 536.2/536.2 kB 42.9 MB/s eta 0:00:00
Installing collected packages: swiglpk, mpmath, sympy, optlang
Successfully installed mpmath-1.3.0 optlang-1.8.2 swiglpk-5.0.10 sympy-1.13.3
```

## ✓ 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/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_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
```

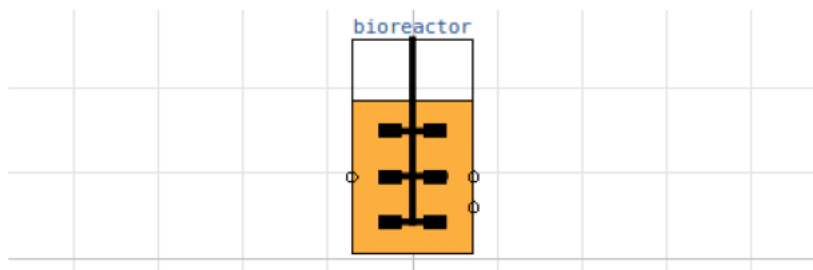
Note that both disp() and describe() takes values from the last simulation and the command process\_diagram() brings up the main configuration

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

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

```
process_diagram()
```

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



## ✓ Try using LP in each step

```

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
    q02max = 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')
    q02lim = Constraint(kog*qGr_opt + koe*qEr_opt, ub=q02max)
    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(q02lim)
    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, q02lim)

# 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)

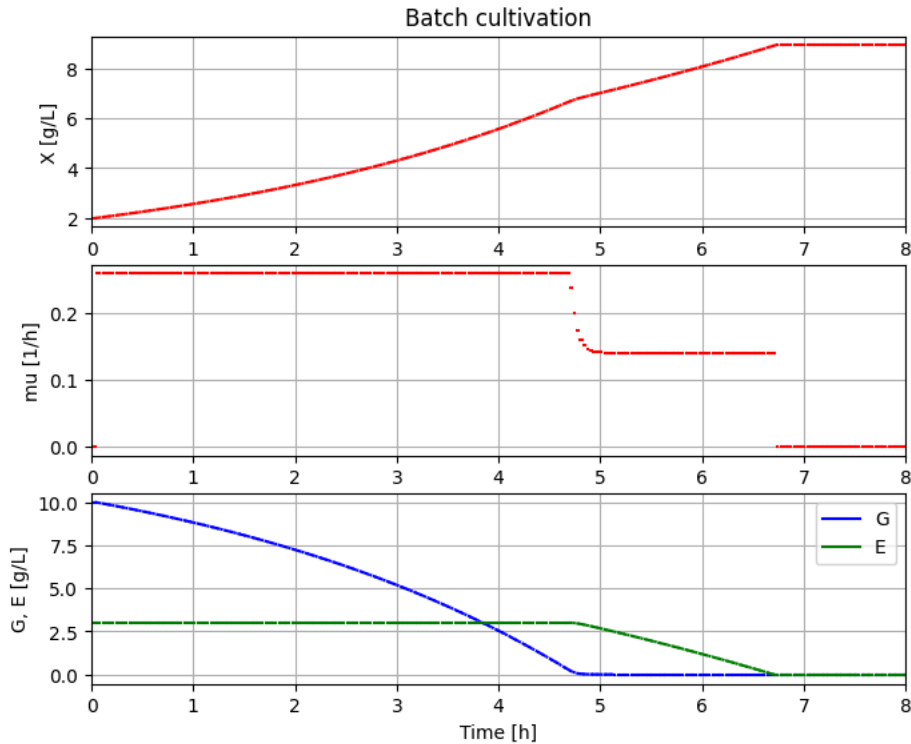
# Loop of simulations
t_final = 8.0
t_samp = 0.0333
n_samp = t_final/t_samp + 1

# 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, q02_opt) = culture(sim_res['bioreactor.c[2]'][-1], sim_res['bioreactor.c[3]'][-1])
    par(mum=mum_opt, qGr=qGr_opt, qEr=qEr_opt, q02=q02_opt)
    simu(t_samp, 'cont', options=opts_fast)

```

Could not find cannot import name 'dopri5' from 'assimulo.lib' (/usr/local/lib/python3.10/site-packages/assimulo/lib/\_\_i  
 Could not find cannot import name 'rodas' from 'assimulo.lib' (/usr/local/lib/python3.10/site-packages/assimulo/lib/\_\_i  
 Could not find cannot import name 'odassl' from 'assimulo.lib' (/usr/local/lib/python3.10/site-packages/assimulo/lib/\_\_i  
 Could not find ODEPACK functions.  
 Could not find RADAR5  
 Could not find GLIMDA.



system\_info()



System information  
 -OS: Linux  
 -Python: 3.10.12  
 -Scipy: not installed in the notebook  
 -PyFMI: 2.14.0  
 -FMU by: OpenModelica Compiler OpenModelica 1.25.0~dev-133-ga5470be  
 -FMI: 2.0  
 -Type: FMUModelME2  
 -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 version 1.0.0

!conda list optlang