Open in Colab

BPL_CHO_Fedbatch script with PyFMI

The key library PyFMI is installed.

After the installation a small application BPL_CHO_Fedbatch is loaded and run. You can continue with this example if you like.

In [1]:
```
!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
```

In [2]:
```
%env PYTHONPATH=
```

```
env: PYTHONPATH=
```

In [3]:
```
!python --version
```

```
Python 3.11.11
```

In [4]:
```
!wget https://repo.anaconda.com/miniconda/Miniconda3-py311_24.11.1-0-Linux-x86_64.s
!chmod +x Miniconda3-py311_24.11.1-0-Linux-x86_64.sh
!bash ./Miniconda3-py311_24.11.1-0-Linux-x86_64.sh -b -f -p /usr/local
import sys
sys.path.append('/usr/local/lib/python3.11/site-packages/')
```

```
--2025-03-26 16:27:01--  https://repo.anaconda.com/miniconda/Miniconda3-py311_24.11.
1-0-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.32.241, 104.16.191.158, 26
06:4700::6810:bf9e, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.32.241|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 145900576 (139M) [application/octet-stream]
Saving to: ‘Miniconda3-py311_24.11.1-0-Linux-x86_64.sh’

Miniconda3-py311_24 100%[===================>] 139.14M   104MB/s    in 1.3s

2025-03-26 16:27:02 (104 MB/s) - ‘Miniconda3-py311_24.11.1-0-Linux-x86_64.sh’ saved
[145900576/145900576]

PREFIX=/usr/local
Unpacking payload ...

Installing base environment...

Preparing transaction: ...working... done
Executing transaction: ...working... done
installation finished.
```

In [5]:
```
!conda update -n base -c defaults conda --yes
```

```
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): - ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\
▒▒| ▒▒/ ▒▒done
Solving environment: \ ▒▒| ▒▒done

## Package Plan ##

  environment location: /usr/local

  added / updated specs:
    - conda


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2025.2.25  |       h06a4308_0         129 KB
    certifi-2025.1.31          |  py311h06a4308_0         163 KB
    openssl-3.0.16             |       h5eee18b_0         5.2 MB
    ------------------------------------------------------------
                                           Total:         5.5 MB

The following packages will be UPDATED:

  ca-certificates                    2024.11.26-h06a4308_0 --> 2025.2.25-h06a4308_0
  certifi                          2024.8.30-py311h06a4308_0 --> 2025.1.31-py311h06a4
308_0
  openssl                             3.0.15-h5eee18b_0 --> 3.0.16-h5eee18b_0



Downloading and Extracting Packages:
openssl-3.0.16       | 5.2 MB    | :   0% 0/1 [00:00<?, ?it/s]
certifi-2025.1.31    | 163 KB    | :   0% 0/1 [00:00<?, ?it/s]

ca-certificates-2025 | 129 KB    | :   0% 0/1 [00:00<?, ?it/s]
openssl-3.0.16       | 5.2 MB    | :   3% 0.026843345551458574/1 [00:00<00:03,  3.78
s/it]

ca-certificates-2025 | 129 KB    | :  25% 0.24763646531593148/1 [00:00<00:00,  2.40i
t/s]
certifi-2025.1.31    | 163 KB    | : 100% 1.0/1 [00:00<00:00,  8.08it/s]
certifi-2025.1.31    | 163 KB    | : 100% 1.0/1 [00:00<00:00,  8.08it/s]

ca-certificates-2025 | 129 KB    | : 100% 1.0/1 [00:00<00:00,  2.40it/s]




Preparing transaction: - ▒▒done
Verifying transaction: | ▒▒/ ▒▒- ▒▒done
Executing transaction: | ▒▒done
```

In [6]:
```
!conda --version
!python --version
```

```
conda 24.11.1
Python 3.11.11
```

In [7]:
```
!conda config --set channel_priority strict
```

In [8]:
```
!conda install -c conda-forge pyfmi --yes # Install the key package
```

```
Channels:
 - conda-forge
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): - ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\
▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒|
▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/
▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒-
▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒- ▒▒\ ▒▒| ▒▒/ ▒▒done
Solving environment: \ ▒▒| ▒▒/ ▒▒done

## Package Plan ##

  environment location: /usr/local

  added / updated specs:
    - pyfmi


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    _x86_64-microarch-level-3  |      2_broadwell           8 KB  conda-forge
    assimulo-3.6.0             |  py311h083bc19_0         1.1 MB  conda-forge
    certifi-2025.1.31          |     pyhd8ed1ab_0         159 KB  conda-forge
    conda-25.1.1               |  py311h38be061_1         1.1 MB  conda-forge
    fmilib-2.4.1               |       hac33072_1         383 KB  conda-forge
    gmp-6.3.0                  |       hac33072_2         449 KB  conda-forge
    libamd-3.3.3               | haaf9dc3_7100102          49 KB  conda-forge
    libblas-3.9.0              |31_h59b9bed_openblas          16 KB  conda-forge
    libbtf-2.3.2               | h32481e8_7100102          27 KB  conda-forge
    libcamd-3.3.3              | h32481e8_7100102          46 KB  conda-forge
    libcblas-3.9.0             |31_he106b2a_openblas          16 KB  conda-forge
    libccolamd-3.3.4           | h32481e8_7100102          42 KB  conda-forge
    libcholmod-5.3.1           | h59ddab4_7100102         1.1 MB  conda-forge
    libcolamd-3.3.4            | h32481e8_7100102          33 KB  conda-forge
    libcxsparse-4.4.1          | h32481e8_7100102         118 KB  conda-forge
    libgcc-14.2.0              |       h767d61c_2         828 KB  conda-forge
    libgcc-ng-14.2.0           |       h69a702a_2          52 KB  conda-forge
    libgfortran-14.2.0         |       h69a702a_2          52 KB  conda-forge
    libgfortran-ng-14.2.0      |       h69a702a_2          53 KB  conda-forge
    libgfortran5-14.2.0        |       hf1ad2bd_2         1.4 MB  conda-forge
    libgomp-14.2.0             |       h767d61c_2         449 KB  conda-forge
    libklu-2.3.5               | hf24d653_7100102         142 KB  conda-forge
    liblapack-3.9.0            |31_h7ac8fdf_openblas          16 KB  conda-forge
    libldl-3.3.2               | h32481e8_7100102          24 KB  conda-forge
    libopenblas-0.3.29         |pthreads_h94d23a6_0         5.6 MB  conda-forge
    libparu-1.0.0              | h17147ab_7100102          91 KB  conda-forge
    librbio-4.3.4              | h32481e8_7100102          47 KB  conda-forge
    libspex-3.2.3              | had10066_7100102          79 KB  conda-forge
    libspqr-4.3.4              | h852d39f_7100102         213 KB  conda-forge
    libstdcxx-14.2.0           |       h8f9b012_2         3.7 MB  conda-forge
    libstdcxx-ng-14.2.0        |       h4852527_2          53 KB  conda-forge
    libsuitesparseconfig-7.10.1| h92d6892_7100102          42 KB  conda-forge
    libumfpack-6.3.5           | heb53515_7100102         424 KB  conda-forge
```

```
    metis-5.1.0                |    hd0bcaf9_1007         3.7 MB  conda-forge
    mpfr-4.2.1                 |       h90cbb55_3         620 KB  conda-forge
    numpy-2.2.4                |  py311h5d046bc_0         8.6 MB  conda-forge
    openssl-3.4.1              |       h7b32b05_0         2.8 MB  conda-forge
    pyfmi-2.16.3               |  py311h9f3472d_0         5.2 MB  conda-forge
    python_abi-3.11            |          2_cp311           5 KB  conda-forge
    scipy-1.15.2               |  py311h8f841c2_0        16.4 MB  conda-forge
    suitesparse-7.10.1         | ha0f6916_7100102          12 KB  conda-forge
    sundials-7.1.1             |       ha52427a_0         907 KB  conda-forge
    ------------------------------------------------------------
                                           Total:        56.1 MB

The following NEW packages will be INSTALLED:

  _x86_64-microarch~ conda-forge/noarch::_x86_64-microarch-level-3-2_broadwell
  assimulo           conda-forge/linux-64::assimulo-3.6.0-py311h083bc19_0
  fmilib             conda-forge/linux-64::fmilib-2.4.1-hac33072_1
  gmp                conda-forge/linux-64::gmp-6.3.0-hac33072_2
  libamd             conda-forge/linux-64::libamd-3.3.3-haaf9dc3_7100102
  libblas            conda-forge/linux-64::libblas-3.9.0-31_h59b9bed_openblas
  libbtf             conda-forge/linux-64::libbtf-2.3.2-h32481e8_7100102
  libcamd            conda-forge/linux-64::libcamd-3.3.3-h32481e8_7100102
  libcblas           conda-forge/linux-64::libcblas-3.9.0-31_he106b2a_openblas
  libccolamd         conda-forge/linux-64::libccolamd-3.3.4-h32481e8_7100102
  libcholmod         conda-forge/linux-64::libcholmod-5.3.1-h59ddab4_7100102
  libcolamd          conda-forge/linux-64::libcolamd-3.3.4-h32481e8_7100102
  libcxsparse        conda-forge/linux-64::libcxsparse-4.4.1-h32481e8_7100102
  libgcc             conda-forge/linux-64::libgcc-14.2.0-h767d61c_2
  libgfortran        conda-forge/linux-64::libgfortran-14.2.0-h69a702a_2
  libgfortran-ng     conda-forge/linux-64::libgfortran-ng-14.2.0-h69a702a_2
  libgfortran5       conda-forge/linux-64::libgfortran5-14.2.0-hf1ad2bd_2
  libklu             conda-forge/linux-64::libklu-2.3.5-hf24d653_7100102
  liblapack          conda-forge/linux-64::liblapack-3.9.0-31_h7ac8fdf_openblas
  libldl             conda-forge/linux-64::libldl-3.3.2-h32481e8_7100102
  libopenblas        conda-forge/linux-64::libopenblas-0.3.29-pthreads_h94d23a6_0
  libparu            conda-forge/linux-64::libparu-1.0.0-h17147ab_7100102
  librbio            conda-forge/linux-64::librbio-4.3.4-h32481e8_7100102
  libspex            conda-forge/linux-64::libspex-3.2.3-had10066_7100102
  libspqr            conda-forge/linux-64::libspqr-4.3.4-h852d39f_7100102
  libstdcxx          conda-forge/linux-64::libstdcxx-14.2.0-h8f9b012_2
  libsuitesparsecon~ conda-forge/linux-64::libsuitesparseconfig-7.10.1-h92d6892_7100
102
  libumfpack         conda-forge/linux-64::libumfpack-6.3.5-heb53515_7100102
  metis              conda-forge/linux-64::metis-5.1.0-hd0bcaf9_1007
  mpfr               conda-forge/linux-64::mpfr-4.2.1-h90cbb55_3
  numpy              conda-forge/linux-64::numpy-2.2.4-py311h5d046bc_0
  pyfmi              conda-forge/linux-64::pyfmi-2.16.3-py311h9f3472d_0
  python_abi         conda-forge/linux-64::python_abi-3.11-2_cp311
  scipy              conda-forge/linux-64::scipy-1.15.2-py311h8f841c2_0
  suitesparse        conda-forge/linux-64::suitesparse-7.10.1-ha0f6916_7100102
  sundials           conda-forge/linux-64::sundials-7.1.1-ha52427a_0

The following packages will be UPDATED:

  conda              pkgs/main::conda-24.11.1-py311h06a430~ --> conda-forge::conda-2
5.1.1-py311h38be061_1
```

```
  libgcc-ng          pkgs/main::libgcc-ng-11.2.0-h1234567_1 --> conda-forge::libgcc-
ng-14.2.0-h69a702a_2
  libgomp              pkgs/main::libgomp-11.2.0-h1234567_1 --> conda-forge::libgomp
-14.2.0-h767d61c_2
  libstdcxx-ng       pkgs/main::libstdcxx-ng-11.2.0-h12345~ --> conda-forge::libstdc
xx-ng-14.2.0-h4852527_2
  openssl              pkgs/main::openssl-3.0.16-h5eee18b_0 --> conda-forge::openssl
-3.4.1-h7b32b05_0

The following packages will be SUPERSEDED by a higher-priority channel:

  certifi            pkgs/main/linux-64::certifi-2025.1.31~ --> conda-forge/noarch::
certifi-2025.1.31-pyhd8ed1ab_0



Downloading and Extracting Packages:
scipy-1.15.2         | 16.4 MB   | :   0% 0/1 [00:00<?, ?it/s]
numpy-2.2.4          | 8.6 MB    | :   0% 0/1 [00:00<?, ?it/s]

libopenblas-0.3.29   | 5.6 MB    | :   0% 0/1 [00:00<?, ?it/s]


pyfmi-2.16.3         | 5.2 MB    | :   0% 0/1 [00:00<?, ?it/s]



metis-5.1.0          | 3.7 MB    | :   0% 0/1 [00:00<?, ?it/s]



libstdcxx-14.2.0     | 3.7 MB    | :   0% 0/1 [00:00<?, ?it/s]




openssl-3.4.1        | 2.8 MB    | :   0% 0/1 [00:00<?, ?it/s]




libgfortran5-14.2.0  | 1.4 MB    | :   0% 0/1 [00:00<?, ?it/s]




conda-25.1.1         | 1.1 MB    | :   0% 0/1 [00:00<?, ?it/s]
```

```
assimulo-3.6.0       | 1.1 MB    | :   0% 0/1 [00:00<?, ?it/s]

libcholmod-5.3.1     | 1.1 MB    | :   0% 0/1 [00:00<?, ?it/s]

sundials-7.1.1       | 907 KB    | :   0% 0/1 [00:00<?, ?it/s]

libgcc-14.2.0        | 828 KB    | :   0% 0/1 [00:00<?, ?it/s]

mpfr-4.2.1           | 620 KB    | :   0% 0/1 [00:00<?, ?it/s]
```

```
gmp-6.3.0            | 449 KB    | :   0% 0/1 [00:00<?, ?it/s]

libgomp-14.2.0       | 449 KB    | :   0% 0/1 [00:00<?, ?it/s]

libumfpack-6.3.5     | 424 KB    | :   0% 0/1 [00:00<?, ?it/s]
```

```
fmilib-2.4.1         | 383 KB    | :   0% 0/1 [00:00<?, ?it/s]

libspqr-4.3.4        | 213 KB    | :   0% 0/1 [00:00<?, ?it/s]

 ... (more hidden) ...

pyfmi-2.16.3         | 5.2 MB    | :   1% 0.005967906113297332/1 [00:00<00:18, 18.24
s/it]

metis-5.1.0          | 3.7 MB    | :   0% 0.004175799528999174/1 [00:00<00:25, 26.02
s/it]
scipy-1.15.2         | 16.4 MB   | :   0% 0.0009529389827073913/1 [00:00<02:46, 166.
48s/it]

pyfmi-2.16.3         | 5.2 MB    | :  64% 0.6415499071794631/1 [00:00<00:00,  3.62i
t/s]
```

```
metis-5.1.0          | 3.7 MB    | :  74% 0.7391165166328538/1 [00:00<00:00,  4.19i
t/s]
scipy-1.15.2         | 16.4 MB   | :  10% 0.09719977623615392/1 [00:00<00:01,  2.20
s/it]



metis-5.1.0          | 3.7 MB    | : 100% 1.0/1 [00:00<00:00,  4.19it/s]

libopenblas-0.3.29   | 5.6 MB    | :   0% 0.0027679004637044184/1 [00:00<01:47, 107.
77s/it]


pyfmi-2.16.3         | 5.2 MB    | : 100% 1.0/1 [00:00<00:00,  3.62it/s]



scipy-1.15.2         | 16.4 MB   | :  20% 0.20011718636855216/1 [00:00<00:01,  1.49
s/it]




openssl-3.4.1        | 2.8 MB    | :   1% 0.005574104907757137 6/1 [00:00<01:04, 64.9
8s/it]

libopenblas-0.3.29   | 5.6 MB    | :   1% 0.011071601854817674/1 [00:00<00:30, 30.76
s/it]



libstdcxx-14.2.0     | 3.7 MB    | : 100% 1.0/1 [00:00<00:00,  2.69it/s]



scipy-1.15.2         | 16.4 MB   | :  37% 0.36783444732505305/1 [00:00<00:00,  1.06
s/it]
numpy-2.2.4          | 8.6 MB    | : 100% 1.0/1 [00:00<00:00,  2.15it/s]
numpy-2.2.4          | 8.6 MB    | : 100% 1.0/1 [00:00<00:00,  2.15it/s]



openssl-3.4.1        | 2.8 MB    | : 100% 1.0/1 [00:00<00:00,  2.56it/s]
```

```
openssl-3.4.1        | 2.8 MB    | : 100% 1.0/1 [00:00<00:00,  2.56it/s]

libopenblas-0.3.29   | 5.6 MB    | :   5% 0.04982220834667953/1 [00:00<00:06,  7.24
s/it]

libgfortran5-14.2.0  | 1.4 MB    | :   1% 0.011206734985068174/1 [00:00<00:45, 46.39
s/it]

assimulo-3.6.0       | 1.1 MB    | :   1% 0.014703493605362324/1 [00:00<00:37, 37.82
s/it]

scipy-1.15.2         | 16.4 MB   | :  59% 0.5946339252094122/1 [00:00<00:00,  1.32i
t/s]

libopenblas-0.3.29   | 5.6 MB    | :  25% 0.2518789421971021/1 [00:00<00:01,  1.42s/
it]

assimulo-3.6.0       | 1.1 MB    | : 100% 1.0/1 [00:00<00:00, 37.82s/it]

conda-25.1.1         | 1.1 MB    | : 100% 1.0/1 [00:00<00:00,  2.00it/s]
```

```
conda-25.1.1          | 1.1 MB    | : 100% 1.0/1 [00:00<00:00,  2.00it/s]

libgfortran5-14.2.0   | 1.4 MB    | : 100% 1.0/1 [00:00<00:00,  1.95it/s]

scipy-1.15.2          | 16.4 MB   | :  76% 0.7566335522696687/1 [00:00<00:00,  1.41i
t/s]
libopenblas-0.3.29    | 5.6 MB    | :  75% 0.747333125200193/1 [00:00<00:00,  1.98it/
s]

libgcc-14.2.0         | 828 KB    | :   2% 0.01932337522187561/1 [00:00<00:36, 37.53
s/it]

libcholmod-5.3.1      | 1.1 MB    | :   1% 0.014870549794649543/1 [00:00<00:48, 49.42
s/it]

sundials-7.1.1        | 907 KB    | :   2% 0.01763373830085844/1 [00:00<00:40, 41.72
```

s/it]

libgcc-14.2.0 | 828 KB | : 100% 1.0/1 [00:00<00:00, 37.53s/it]

scipy-1.15.2 | 16.4 MB | : 97% 0.9691389454134169/1 [00:00<00:00, 1.57i
t/s]

libcholmod-5.3.1 | 1.1 MB | : 100% 1.0/1 [00:00<00:00, 49.42s/it]

mpfr-4.2.1 | 620 KB | : 3% 0.025811696239942908/1 [00:00<00:32, 33.24
s/it]

```
gmp-6.3.0            | 449 KB    | :   4% 0.03561313321233331/1 [00:00<00:23, 24.46
s/it]

libgomp-14.2.0       | 449 KB    | :   4% 0.03562807972826631/1 [00:00<00:23, 24.62
s/it]

metis-5.1.0          | 3.7 MB    | : 100% 1.0/1 [00:00<00:00,  4.19it/s]

libgomp-14.2.0       | 449 KB    | : 100% 1.0/1 [00:00<00:00, 24.62s/it]
```

```
gmp-6.3.0            | 449 KB    | : 100% 1.0/1 [00:00<00:00, 24.46s/it]

mpfr-4.2.1           | 620 KB    | : 100% 1.0/1 [00:00<00:00, 33.24s/it]

libspqr-4.3.4        | 213 KB    | :   8% 0.07503068271326775/1 [00:00<00:11, 12.75
s/it]

libspqr-4.3.4        | 213 KB    | : 100% 1.0/1 [00:00<00:00, 12.75s/it]
```

```
libumfpack-6.3.5     | 424 KB    | :   4% 0.037731330084655984/1 [00:00<00:24, 25.50
s/it]
```

```
libumfpack-6.3.5     | 424 KB    | : 100% 1.0/1 [00:00<00:00, 25.50s/it]
libopenblas-0.3.29   | 5.6 MB    | : 100% 1.0/1 [00:01<00:00,  1.35it/s]
libopenblas-0.3.29   | 5.6 MB    | : 100% 1.0/1 [00:01<00:00,  1.35it/s]
pyfmi-2.16.3         | 5.2 MB    | : 100% 1.0/1 [00:01<00:00,  3.62it/s]
```

```
 ... (more hidden) ...
```

```
 ... (more hidden) ...
```

```
fmilib-2.4.1         | 383 KB    | :   4% 0.04180391656566945/1 [00:01<00:23, 24.65
s/it]
```

```
scipy-1.15.2         | 16.4 MB   | : 100% 1.0/1 [00:01<00:00,  1.57it/s]
```

```
libstdcxx-14.2.0     | 3.7 MB    | : 100% 1.0/1 [00:01<00:00,  2.69it/s]

openssl-3.4.1        | 2.8 MB    | : 100% 1.0/1 [00:01<00:00,  2.56it/s]

assimulo-3.6.0       | 1.1 MB    | : 100% 1.0/1 [00:02<00:00,  1.88s/it]

assimulo-3.6.0       | 1.1 MB    | : 100% 1.0/1 [00:02<00:00,  1.88s/it]

conda-25.1.1         | 1.1 MB    | : 100% 1.0/1 [00:02<00:00,  2.00it/s]

libgfortran5-14.2.0  | 1.4 MB    | : 100% 1.0/1 [00:02<00:00,  1.95it/s]

libgcc-14.2.0        | 828 KB    | : 100% 1.0/1 [00:02<00:00,  2.70s/it]
```

```
libgcc-14.2.0        | 828 KB    | : 100% 1.0/1 [00:02<00:00,  2.70s/it]
```

```
sundials-7.1.1       | 907 KB    | : 100% 1.0/1 [00:03<00:00,  2.96s/it]
```


```
sundials-7.1.1       | 907 KB    | : 100% 1.0/1 [00:03<00:00,  2.96s/it]
```


```
libcholmod-5.3.1     | 1.1 MB    | : 100% 1.0/1 [00:03<00:00,  3.04s/it]
```


```
libcholmod-5.3.1     | 1.1 MB    | : 100% 1.0/1 [00:03<00:00,  3.04s/it]
```


```
numpy-2.2.4          | 8.6 MB    | : 100% 1.0/1 [00:03<00:00,  2.15it/s]
```

```
gmp-6.3.0            | 449 KB    | : 100% 1.0/1 [00:03<00:00,  3.08s/it]
```


```
gmp-6.3.0            | 449 KB    | : 100% 1.0/1 [00:03<00:00,  3.08s/it]
```


```
libgomp-14.2.0       | 449 KB    | : 100% 1.0/1 [00:03<00:00,  3.12s/it]
```


```
libgomp-14.2.0       | 449 KB    | : 100% 1.0/1 [00:03<00:00,  3.12s/it]
```

```
libspqr-4.3.4        | 213 KB    | : 100% 1.0/1 [00:03<00:00,  3.17s/it]
```


```
libspqr-4.3.4        | 213 KB    | : 100% 1.0/1 [00:03<00:00,  3.17s/it]
```


```
mpfr-4.2.1           | 620 KB    | : 100% 1.0/1 [00:03<00:00,  3.18s/it]
```

```
mpfr-4.2.1           | 620 KB    | : 100% 1.0/1 [00:03<00:00,  3.18s/it]

libumfpack-6.3.5     | 424 KB    | : 100% 1.0/1 [00:03<00:00,  3.18s/it]

libumfpack-6.3.5     | 424 KB    | : 100% 1.0/1 [00:03<00:00,  3.18s/it]

 ... (more hidden) ...
```

```
 ... (more hidden) ...
```

```
fmilib-2.4.1         | 383 KB    | : 100% 1.0/1 [00:03<00:00,  3.29s/it]
```


```
fmilib-2.4.1         | 383 KB    | : 100% 1.0/1 [00:03<00:00,  3.29s/it]
```


```
scipy-1.15.2         | 16.4 MB   | : 100% 1.0/1 [00:04<00:00,  1.57it/s]
```

```
Preparing transaction: - ▯▯\ ▯▯done
Verifying transaction: / ▯▯- ▯▯\ ▯▯| ▯▯/ ▯▯done
Executing transaction: \ ▯▯| ▯▯/ ▯▯- ▯▯\ ▯▯| ▯▯/ ▯▯- ▯▯\ ▯▯| ▯▯/ ▯▯- ▯▯\ ▯▯| ▯▯/ ▯▯-
▯▯\ ▯▯done
```

BPL_CHO_Fedbatch setup

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

- FMU - BPL_CHO_Fedbatch_linux_om_me.fmu
- Setup-file - BPL_CHO_Fedbatch_explore

In [9]:
```
%%bash
git clone https://github.com/janpeter19/BPL_CHO_Fedbatch
```

```
Cloning into 'BPL_CHO_Fedbatch'...
```

In [10]:
```
%cd BPL_CHO_Fedbatch
```

```
/content/BPL_CHO_Fedbatch
```

BPL_CHO_Fedbatch - demo

Author: Jan Peter Axelsson

This notebook deals with CHO fedbatch cultivation and recombinant protein production is included. First we make a check of the model by comparing a simulation result with the corresponding published diagram. We take a look at viable and dead cells and introduce cell lysis and negative effects on viable cell growth. Then we take a closer look at the start-up strategy to keep the by-product formation low. After that we investigate a whole cultivation and see the impact of feeding strategy on both cell growth and protein production where a trade-off is needed in this case.

The model used takes its inspiration from the microbial bottleneck models as described in the original papers [1] and [2] and reformulated and studied in [3]. The laboratory cultures used for model validation in [1] did produce mAb (against part of IgG) but no mAb-data was presented. The paper focus on viable and non-viable cell concentrations only. The original model is expanded with a state for lysed cells coming from dead cells [5, 7]. Further the lysed cell material is described as having a toxic negative effect on viable cell growth rate [5]. The character of this lysed cell material is further described in [6]. The original model is in section 6 further expanded with the classical empirical Luedeking-Piret model recombinant protein production, see chapter 5 in [7]. In this way can get more insight into choice of feeding profile.

The dead cell measurements presented in [1] is difficult and prone to errors, personal communication [6].

Interaction with the compiled model as FMU is mainly through the simplified commands: par(), init(), newplot(), simu() etc. The last simulation is always available in the workspace and called 'sim_res'. The command describe() brings mainly up description infomration from the actual Modelica code from the FMU but is complemented with information given in the dedicated Python setup-file.

The idea here is to demonstrate how simulations and varyiing conditions can provide some process insight that can support the experimental work. I hope that at the end of this session you are ready to formulate your own questions you want to address with simulations - and you can just go on in this notebook! Just press the field "+Code" in the upper left part of notebook interface and you get a new "cell" where you write your own code. You can copy and paste from cells above using ctrl-c and ctrl-v as usual and edit the cell. When your are ready to execute the cell just press the "play button" to the left in the cell or press shift-enter as in "ordinary" Jupyter notebooks.

After a session you may want to save your own notebook. That you can do on your Google Drive account and I refer to Colab instructions for how to do this. It is easy.

Good luck!

In [11]:
```
run -i BPL_CHO_Fedbatch_explore.py
```

```
Linux - run FMU pre-compiled OpenModelica

Model for the process 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()
```

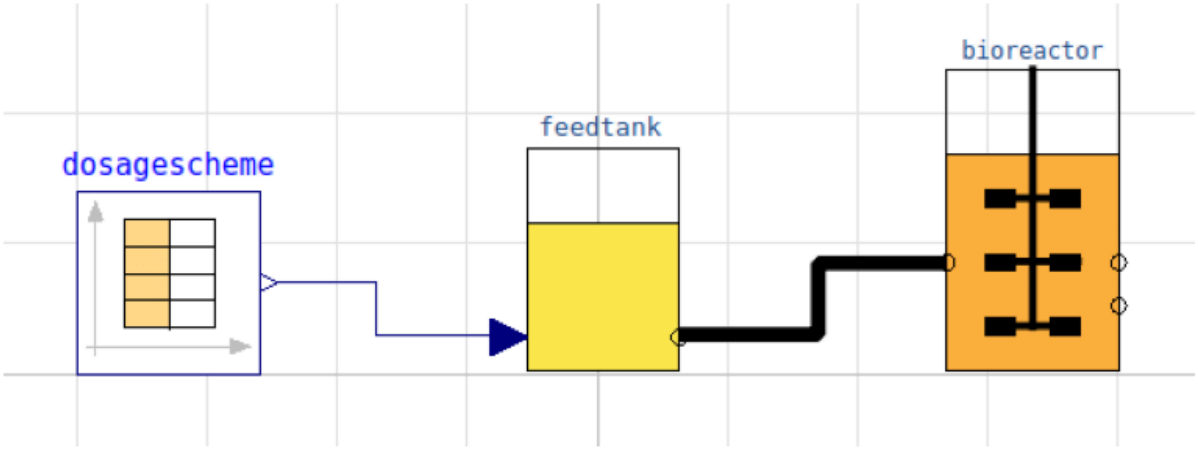
In [12]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = [25/2.54, 20/2.54]
```

1 About the process model

We can get information about the process and liquid phase by the command describe(). Here is no gas-phase included. This command can also be used to bring up information about a specific variable or parameter. However, you should use describe() after a simulation to get the valued used during the simulation.

In [13]:
```
process_diagram()
```

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



In [14]:
```
describe('culture'); print(); #describe('liquidphase')
```

```
Reactor culture CHO-MAb - cell line HB-58 American Culture Collection ATCC
```

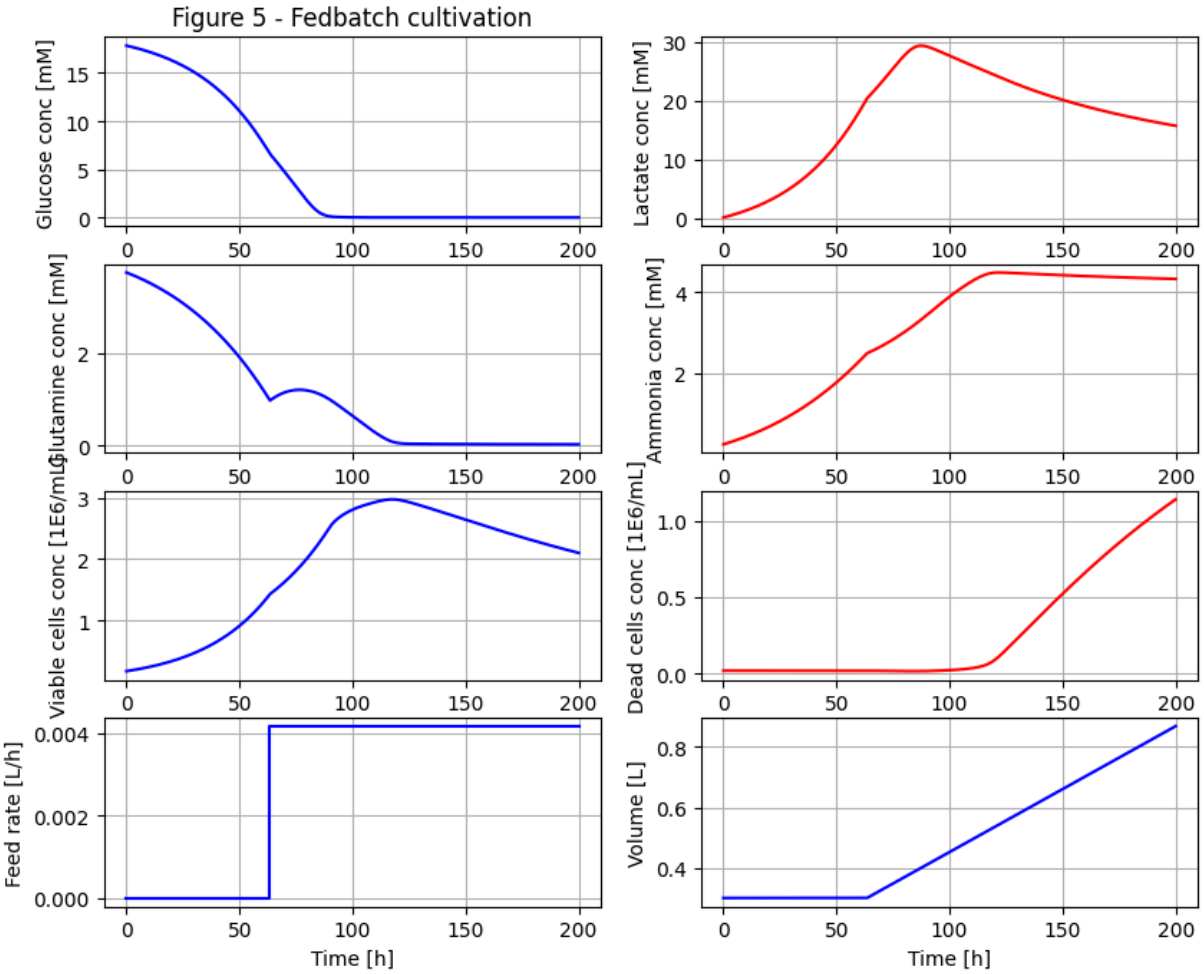
The molecular weight of the recombinant protein (MAb) is somewhat arbitrarly chosen and the value not used in the simulations.

2 Simulation reproducing the original paper

The simulation below reproduce diagrams in Figure 5 in the original paper. There are several simulation in the paper showing how well the model describe different experiments and here I just choose one of them.

In [15]:
```
# Data from Table 1 and 2 for experiment 4 shown in Figure 5 in paper [1]
# -culture parameters taken from Table 5 identified parameters for cultures 1,2,and

# Initial process conditions
V_start=0.30
init(V_start=V_start, VXv_start=V_start*0.172, VXd_start=V_start*0.020)
init(VG_start=V_start*17.83, VGn_start=V_start*3.74, VL_start=V_start*0.12, VN_star

# Feeding
Feed=0.1/24
par(G_in=15, Gn_in=9.3)
par(t0=0, F0=0, t1=63.5, F1=Feed, t2=300, F2=Feed)
par(t3=1003, t4=1004, t5=1005, t6=1006)

# Simulation
newplot(title='Figure 5 - Fedbatch cultivation')
simu(200)
```

```
Could not find cannot import name 'dopri5' from 'assimulo.lib' (/usr/local/lib/pytho
n3.11/site-packages/assimulo/lib/__init__.py)
Could not find cannot import name 'rodas' from 'assimulo.lib' (/usr/local/lib/python
3.11/site-packages/assimulo/lib/__init__.py)
Could not find cannot import name 'odassl' from 'assimulo.lib' (/usr/local/lib/pytho
n3.11/site-packages/assimulo/lib/__init__.py)
Could not find ODEPACK functions.
Could not find RADAR5
Could not find GLIMDA.
```



Comment: The simulation results looks very similar to the published diagram Figure 5 in [1]. The model pass this quality check.

3 Extending the model with cell lysis

A common experience in fedbatch cultivation of CHO is a slow decrease of viable cell number after the peak concentration is reached. This can be described in terms of accumulation of various toxic material during cultivation [5] and further characterized in [6].

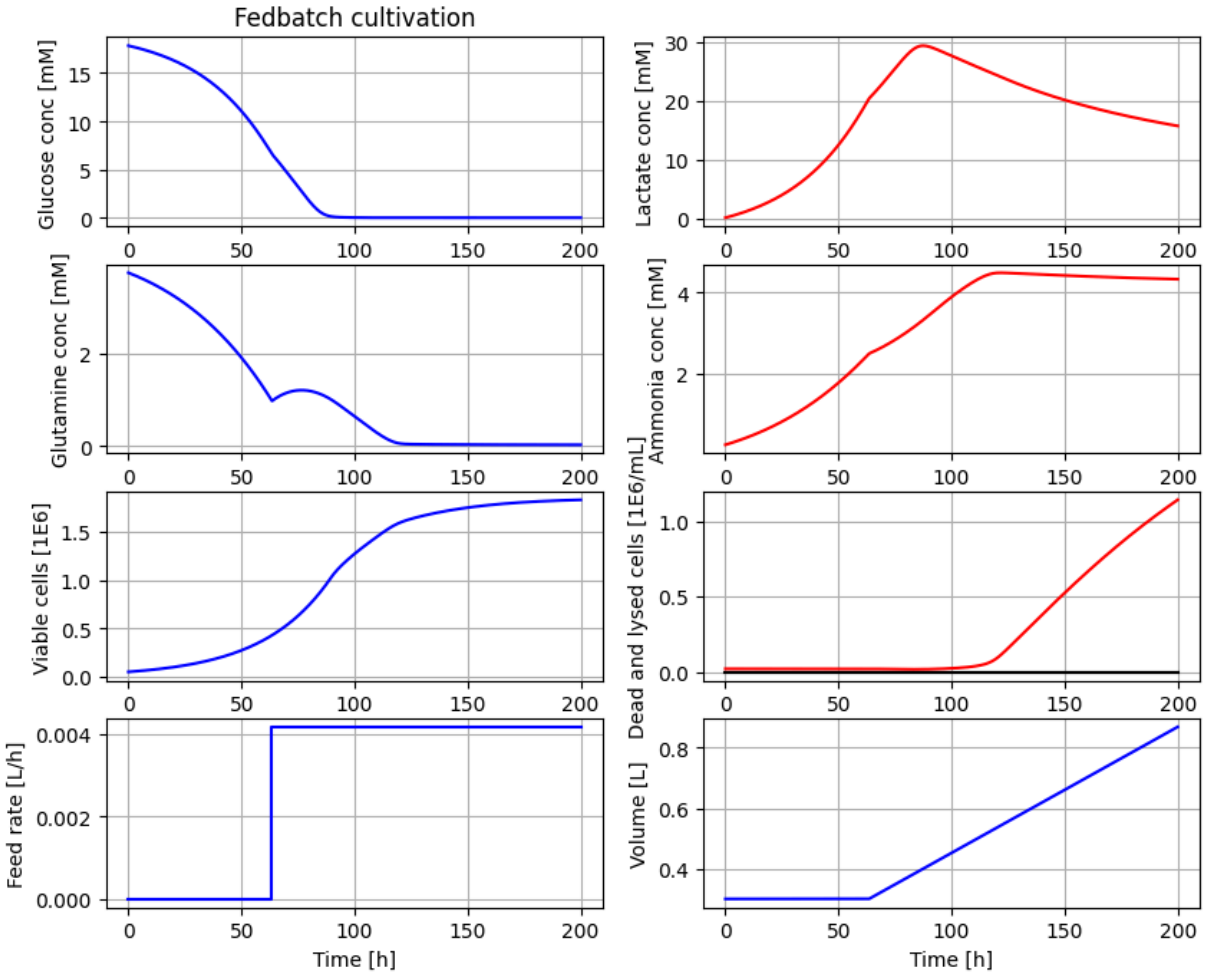
The orginal model [1] does include cell death but not that dead cell concentration has any effect on the culture. The decrease in the viable cell concentration Xv in the previous simulation is actually just an effect of an increased volume of the reactor broth, i.e. a dilution effect. By instead plotting viable cell number VXv instead of Xv this fact is obvious.

The model can easily be extended by including a state for lysed cells VXl that is a variable that reflect the dead cells that have degraded into molecules. This lysis process is briefly outlined on page 195 in [7] and further discussed in [5, 6]. A measure Xl is released of cellular content like DNA or some enzyme like LDH [5].

Important is that the lysed material Xl has a negative effect of viable cell growth and here modelled as an increase in cell death rate. Here just added as a linear term k_toxix*Xl added to the original function of mu_d(G, Gn). The Xl may have a toxic effect also on recombinant protein production but not further studied here.

Here we take a look att the impact of typical values of lysis and toxicity on the original simulation above.

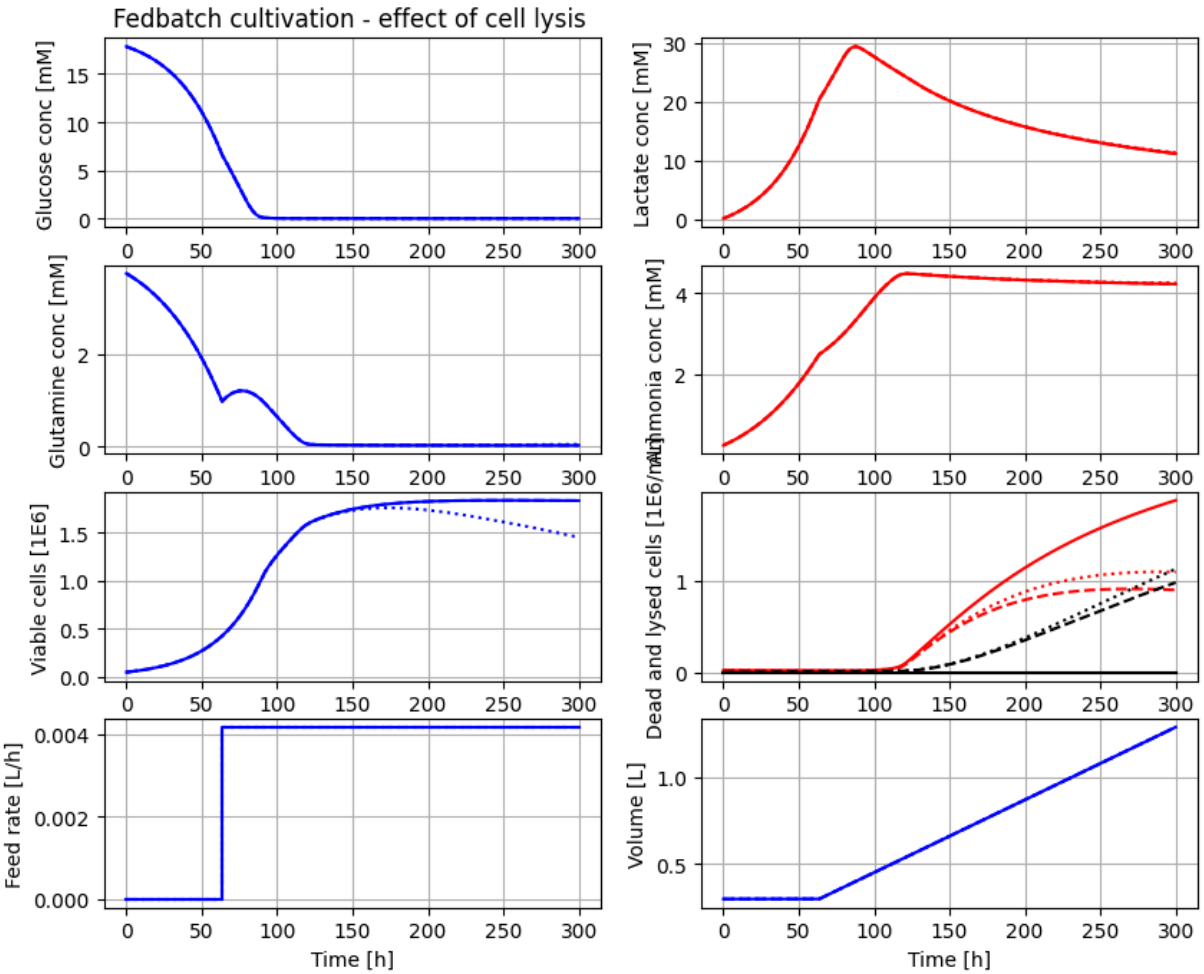
In [16]:
```
newplot(plotType='TimeSeries2')
show()
```



Comment Note Viable cell number VXv reach a plateu, i.e. does not decrease.

In [17]:
```
# Simulation
newplot(title='Fedbatch cultivation - effect of cell lysis ', plotType='TimeSeries2

for value in [0, 0.01]:
   par(k_lysis_d=value, k_toxic=0)
   simu(300)

par(k_lysis_d=0.01, k_toxic=0.007)
simu(300)
```

```
# Restore default values
par(k_lysis_d=0, k_toxic=0)
```

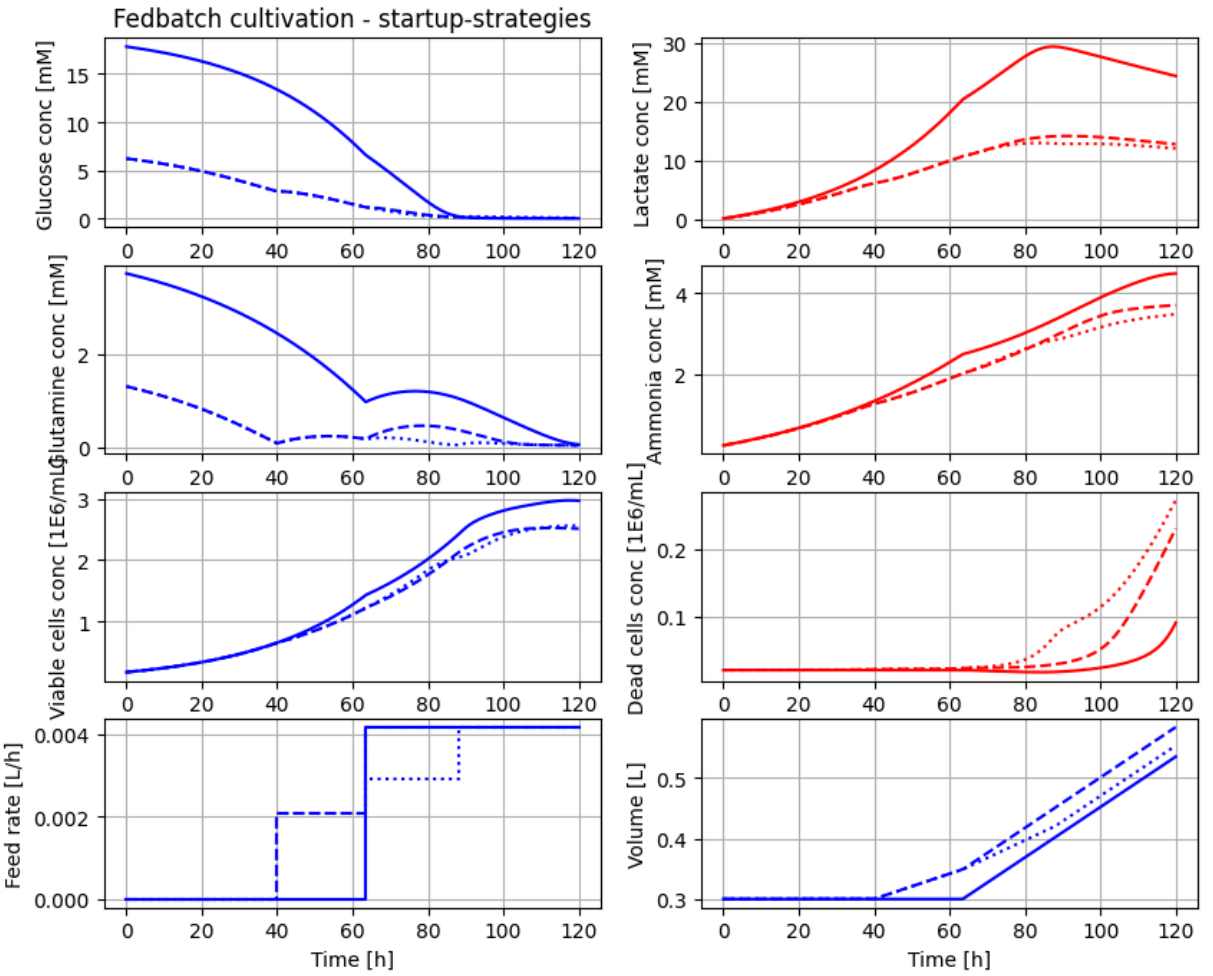


Comment Note that just including lysis only decrease the number of dead cells (dashed line). If we also include the toxic effect of the lysed material then we get a decrease in viable cell number (dotted line). To illustarate the effect simulations are run 300 hours and a typical length for industrial recombinant protein production. A decrease of viable cell number with about 20 percent during the last part of the cultivation is rather typical.

4 Simulation of different start-up feeding strategies

In [18]:
```
# Figur 5
V_start=0.30
init(V_start=V_start, VXv_start=V_start*0.172, VXd_start=V_start*0.020)
init(VG_start=V_start*17.83, VGn_start=V_start*3.74, VL_start=V_start*0.12, VN_star

# Feeding
Feed=0.1/24
par(G_in=15, Gn_in=9.3)
par(t0=0, F0=0, t1=63.5, F1=Feed, t2=300, F2=Feed)

newplot(title='Fedbatch cultivation - startup-strategies')
simu(120)
```

```
init(VG_start=0.35*V_start*17.83, VGn_start=0.35*V_start*3.74)
par(t0=0, F0=0, t1=40.0, F1=0.5*Feed, t2=63.5, F2=Feed,t3=300, F3=Feed)
simu(120)

init(VG_start=0.35*V_start*17.83, VGn_start=0.35*V_start*3.74)
par(t0=0, F0=0, t1=40.0, F1=0.5*Feed, t2=63.5, F2=0.7*Feed, t3=88.0, F3=Feed, t4=30
simu(120)

# Reset time table to avoid problems below
par(t1=1001, t2=1002, t3=1003, t4=1004, t5=1005, t6=1006)
```



Comment: We see that starting the feed a day erlier at lower rate and then increase decreaes lactate formation to half, while the cell conentation is just slightlly lower. With a more careful design of the feedprofile the ammonia formation can be decreased more than shown here.

5 Simulation of optimal feed profile for cell growth

At the end of the original paper section 5 in [1], the derived model is used to find an optimal feeding profile for high final cell concentration. It is stated that protein productivity is assumed to be mainly positively growth associated and therefore optimization of cell

concentration is very similar to optimization of protein product. The optimization of feed profile is done with differnt structures of the feed profile. All of them have a start-time and all of them has a fixed amount of substrate and concentrations in the media are also the same.

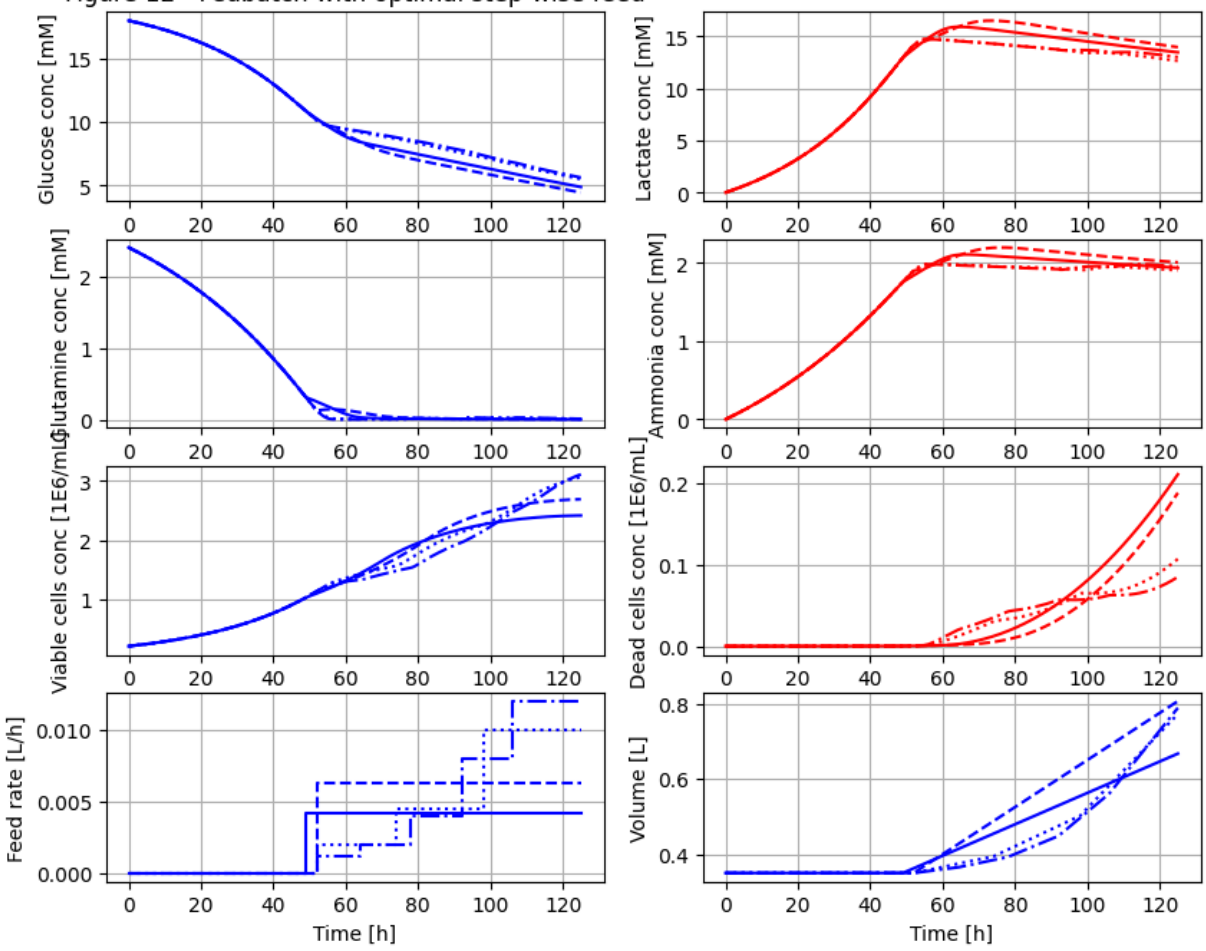
- The first optimzation is for a feed profile similar to the experimental, i.e. after start the feed rate remains constant throughout the cultivation. Thus the start time and the actual feed rate are optimized. The result was that the start time was about the same as expermentally but the feed rate was 50% higher, see Figure 7 and Figure 10 in [1].
- The second optimization is for a feed profile with not just one increase but three steps of increase of feed rate. The results is a somwehat higher final cell concentration, see Figure 11.
- The third optimization is for a feed profile with five steps of increase of feed rate. The results is a slightly higher final cell concentration than for three steps, see Figure 12.
- The fourth optimization is for a feed profie with continuous exponential increase of the reed rate. The result is a bit higher final concentration than the previous with five steps, see Figure 13 but not shown in the figure below.

Below we just show the results of the original experimetnal cultivation, compared with results from three and five steps. It is possible to do the optimization in Python with the FMU, but we save that for a future notebook.

In [19]:
```
# Culture parameters taken from Table 5 identified parameters for cultures 1,2,and

# Data chosen
V_start=0.35
init(V_start=V_start, VXv_start=V_start*0.20, VXd_start=V_start*0.0)
init(VG_start=V_start*18.0, VGn_start=V_start*2.4, VL_start=V_start*0, VN_start=V_s

# Feeding n=1 - experimental and lower feed rate
par(G_in=15, Gn_in=4.0)
par(t0=0, F0=0, t1=49, F1=0.00417)
par(t2=1002, t3=1003, t4=1004, t5=1005)

# Simulation
newplot(title='Figure 12 - Fedbatch with optimal step-wise feed')
simu(125)

# Feeding n=1
par(G_in=15, Gn_in=4.0)
par(t0=0, F0=0, t1=52, F1=0.00625)
par(t2=1002, t3=1003, t4=1004, t5=1005, t6=1006)

# Simulation
simu(125)

# Feeding n=3
```

```
par(G_in=15, Gn_in=4.0)
par(t0=0, F0=0, t1=52, F1=0.002, t2=74, F2=0.0045, t3=98.0, F3=0.010)
par(t4=99.0, F4=0.010, t5=106, F5=0.010, t6=150, F6=0.010)

# Simulation
simu(125)

# Feeding n=5
par(G_in=15, Gn_in=4.0)
par(t0=0, F0=0, t1=52, F1=0.0012, t2=64, F2=0.0020, t3=78.0, F3=0.0040)
par(t4=92.0, F4=0.0080, t5=106, F5=0.012, t6=150, F6=0.012)

# Simulation
simu(125)

# Reset feeding parameters since the table need time in strict increasing value
par(t3=1004, t4=1005, t5=1006, t6=1007)
```

Figure 12 - Fedbatch with optimal step-wise feed



Comment: We see that that already the better tuned constant feed rate (dahsed) compared to the experimental (solid) gives higher final cell concnetration.

Breaking up the constant feed rate in three (dotted) and five (dash-dotted) steps with a more gradual increase of the feed rate gives even higher final cell concentration. The difference between n=3 and n=5 is small. The change to continuous exponential feed is even smaller and not shown here.

The results shown here are similar to what is presented in Table 7 in [1] but our simulation are slightly longer and here are small differences in the final cel concentration too. The qualitative result is the same though. The difference we see to the result in the original paper is most likely due to the fact that we here use the full model with 17 parameters while in the paper they have reduced the model to 15 parameters for the optimization work.

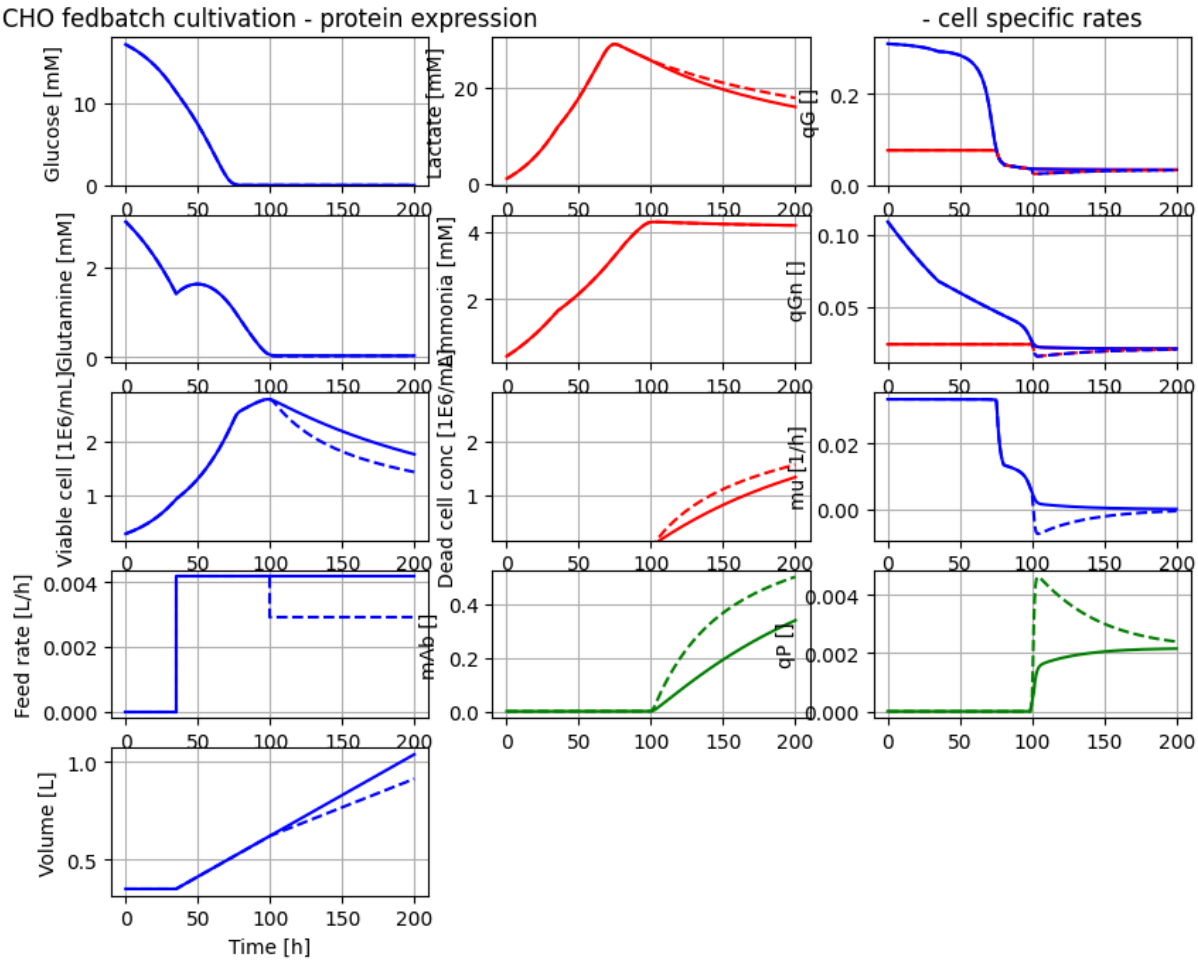
6 Simulation of different feed profiles to increase recombinant protein production

In this section we take a closer look at recombinant protein production. The original model is extended with the empirical model for specific protein production, see chapter 5 in [4]

$$q_P = \alpha \cdot \mu + \beta$$

Here we choose a negative value of growth-associated protein production production α while keeping the non-growth associated β positive. The culture produced recombinant protein in the form of monoclonal antibodies for a specific IgG1 molecule,see section 2 in [1]. However, no experimental results were given. The only information we have is that feed rate was kept constant at a low level during fedbatch production and this choice indicates that the the growth-associated protein production is negative. The consequence of this observation for the feed profile we take a look at there by simulation.

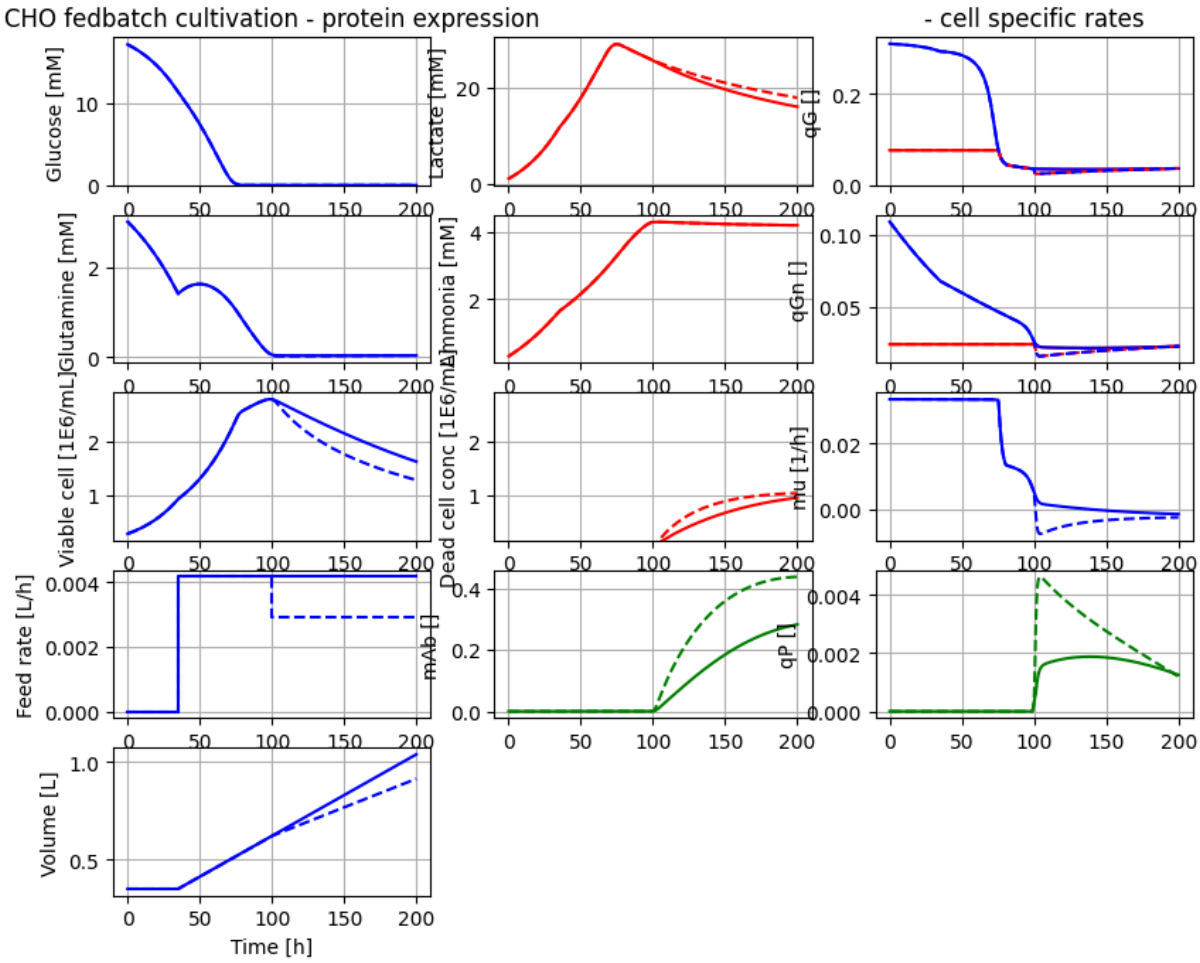
In [20]:
```
# Slide 3
newplot('CHO fedbatch cultivation - protein expression', plotType='Textbook_3')

# Data from Table 1 and 2 for experiment 3
V_start=0.35
init(V_start=V_start, VXv_start=V_start*0.29, VXd_start=V_start*0.010)
init(VG_start=V_start*17.17, VGn_start=V_start*3.02, VL_start=V_start*1.12, VN_star

# Feeding
Feed=0.1/24
par(G_in=15, Gn_in=9.3)
par(t0=0, F0=0, t1=35, F1=Feed, t2=100, F2=Feed, t3=300, F3=Feed)

# Culture parameters
par(alpha=-1.0, beta=0.01)

# Simulation
simu(200)
par(t2=100, F2=0.7*Feed, t3=300, F3=0.7*Feed); simu(200)
par(F2=Feed, F3=Feed)
```



Comment: The simulation results show that actually a decease in the feed rate can lead to an increase in recombinant protein produced, although the cell concentration is a bit lower. This is a result due to the fact that growth-associated protein production here is set to a negative value. The main point is that the model can actually capture this phenomena.

In [21]:
```
# What about possible impact from cell lysis and toxicity

newplot('CHO fedbatch cultivation - protein expression', plotType='Textbook_3')

# Data from Table 1 and 2 for experiment 3
V_start=0.35
init(V_start=V_start, VXv_start=V_start*0.29, VXd_start=V_start*0.010)
init(VG_start=V_start*17.17, VGn_start=V_start*3.02, VL_start=V_start*1.12, VN_star

# Feeding
Feed=0.1/24
par(G_in=15, Gn_in=9.3)
par(t0=0, F0=0, t1=35, F1=Feed, t2=100, F2=Feed, t3=300, F3=Feed)

# Culture parameters
par(alpha=-1.0, beta=0.01)

par(k_lysis_d=0.01, k_toxic=0.007)
```

```
# Simulation
simu(200)
par(t2=100, F2=0.7*Feed, t3=300, F3=0.7*Feed); simu(200)
par(F2=Feed, F3=Feed)

# Reset parameters
par(k_lysis_d=0.0, k_toxic=0.0)
```



Comment: We see that the impact of cell lysis and toxicity on viable cell growth has a smaller impact of the result. The qualitative impact of descresing the feed rate to increase mAb-productiion still holds.

7 Summary

In short we have done the following:

- Section 2: The model was checked by comparing the simulation results with one of the published diagrams [1].
- Section 3: The original model was extended with a state for lysed cells and also included with modelling the toxicity of lysed cells increasing the cell death rate [5]. With typical values we could model a typical decrease in viable cell count during the later part of the cultivation.

- Section 4: The common startup-procedure with 3 days batch cultivation can be questioned. We found that by shorten it to 2 days, and giving smaller feed rate day 3, byproduct formation can be kept lower at the prize of just a bit lower cell concentration. Similar idea was shown in section 2.1 in [3].

- Section 5: In the original paper the experimental feeding strategy was to keep the substrate feed at a constant lower level. The authors made a point of that the optimal feeding strategy should be exponential for maximal cell production. This is an insight derived from the bottle-neck model and they showed that through simulation optimization [1]. However, there was no experimental support to confirm the results. The optimal cell growth feedprofile simulation was just reproduced here.

- Section 6: The model was further extended to include recombinant protein production. Here we do that with the empirical model that distinguish between growth-associated and non-growth-associated protein production, see chapter 5 in [7]. Now we can study optimization of recombinant protein production. For a class of CHO-processes the recombinant protein productivity is acutally negatively affected by cell growth. Simulation of the original model extended with such a protein production model shows that keeping the substrate feed rate constant as the cell culture grows, giving less and less feed per cell, actually can give higher protein production than an increaeing feed rate. Simulation confirms this idea. The results gives some possible background to why the constant feed rate was used experimetnally in the original paper [1].

8 References

[1] Amribt, Z., Niu, H. and Bogaerts P.: "Macroscopic modelling of overflow metabolism and model based optimization of hybridoma cell fed-batch cultures.", Biochem. Eng. Journal, 2013.

[2] Niu,H., Amribt, Z., Fickers, P., Tan, W. and Bogaerts P.: "Metabolic pathway analysis and reduction for mammalian cell cultures - towards macroscopic modelling", Chem. Eng. Science, 2013.

[3] Axelsson, J. P.: "Simplified model of CHO-cultivation in Bioproces Library for Modelica - some experience", conference paper 22nd NPCW Lyngby, Denmark, August 22-23, 2019.

[4] Bogaerts, P.: "Stated that dead cell measurements in Amribts work most likely had larger errors, and later work and publications with the same data sets omitted these dead cell data", met at DYCOPS-CAB in Trondheim, Norway, in june 2016.

[5] Kroll, P., Eilers, K., Fricke, J and Herwig C.: "Impact of cell lysis on the description of cell growth and death in cell culture", Eng. in Life Sci, 2017.

[6] Mulukutla, B. C., Kale, J., Kalomeris, T., Jaccobs, M., Hiller, G. W.: "Identification and control of novel growth inhibitors in fed-batch cultivation of Chinese hamster ovary cells.", Biotech. Bioeng., 2017.

[7] Hu, W-S: "Cell culture bioprocess engineering", 2nd edition, CRC Press, 2020.

Appendix

In [22]:
```
# List of components in the process setup and also a couple of other things like li
describe('parts')
```

```
['bioreactor', 'bioreactor.broth_decay', 'bioreactor.culture', 'dosagescheme', 'feed
tank']
```

In [23]:
```
describe('MSL')
```

```
MSL: 3.2.3 - used components: RealInput, RealOutput. CombiTimeTable, Types
```

In [24]:
```
system_info()
```

```
System information
 -OS: Linux
 -Python: 3.11.11
 -Scipy: not installed in the notebook
 -PyFMI: 2.16.3
 -FMU by: OpenModelica Compiler OpenModelica 1.25.0~dev-422-ge7d6d52
 -FMI: 2.0
 -Type: FMUModelME2
 -Name: BPL_CHO.Fedbatch
 -Generated: 2025-03-01T10:34:46Z
 -MSL: 3.2.3
 -Description: Bioprocess Library version 2.3.0
 -Interaction: FMU-explore version 1.0.0
```

In [24]:
```

```