In [22]:
```
pip install cobra
```

```
Requirement already satisfied: cobra in /Users/chf/anaconda3/lib/python3.1
1/site-packages (0.30.0)
Requirement already satisfied: appdirs~=1.4 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from cobra) (1.4.4)
Requirement already satisfied: depinfo~=2.2 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from cobra) (2.2.0)
Requirement already satisfied: diskcache~=5.0 in /Users/chf/anaconda3/lib/p
ython3.11/site-packages (from cobra) (5.6.3)
Requirement already satisfied: future in /Users/chf/anaconda3/lib/python3.1
1/site-packages (from cobra) (0.18.3)
Requirement already satisfied: httpx~=0.24 in /Users/chf/anaconda3/lib/pyth
on3.11/site-packages (from cobra) (0.28.0)
Requirement already satisfied: numpy>=1.13 in /Users/chf/anaconda3/lib/pyth
on3.11/site-packages (from cobra) (1.23.5)
Requirement already satisfied: optlang~=1.8 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from cobra) (1.8.3)
Requirement already satisfied: pandas<3.0,>=1.0 in /Users/chf/anaconda3/li
b/python3.11/site-packages (from cobra) (1.5.3)
Requirement already satisfied: pydantic>=1.6 in /Users/chf/anaconda3/lib/py
thon3.11/site-packages (from cobra) (2.10.2)
Requirement already satisfied: python-libsbml~=5.19 in /Users/chf/anaconda
3/lib/python3.11/site-packages (from cobra) (5.21.0)
Requirement already satisfied: rich>=8.0 in /Users/chf/anaconda3/lib/python
3.11/site-packages (from cobra) (13.9.4)
Requirement already satisfied: ruamel.yaml~=0.16 in /Users/chf/anaconda3/li
b/python3.11/site-packages (from cobra) (0.17.21)
Requirement already satisfied: swiglpk in /Users/chf/anaconda3/lib/python3.
11/site-packages (from cobra) (5.0.13)
Requirement already satisfied: anyio in /Users/chf/anaconda3/lib/python3.1
1/site-packages (from httpx~=0.24->cobra) (3.5.0)
Requirement already satisfied: certifi in /Users/chf/anaconda3/lib/python3.
11/site-packages (from httpx~=0.24->cobra) (2024.8.30)
Requirement already satisfied: httpcore==1.* in /Users/chf/anaconda3/lib/py
thon3.11/site-packages (from httpx~=0.24->cobra) (1.0.7)
Requirement already satisfied: idna in /Users/chf/anaconda3/lib/python3.11/
site-packages (from httpx~=0.24->cobra) (3.4)
Requirement already satisfied: h11<0.15,>=0.13 in /Users/chf/anaconda3/lib/
python3.11/site-packages (from httpcore==1.*->httpx~=0.24->cobra) (0.14.0)
Requirement already satisfied: sympy>=1.12.0 in /Users/chf/anaconda3/lib/py
thon3.11/site-packages (from optlang~=1.8->cobra) (1.14.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/chf/anacond
a3/lib/python3.11/site-packages (from pandas<3.0,>=1.0->cobra) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from pandas<3.0,>=1.0->cobra) (2022.7)
Requirement already satisfied: annotated-types>=0.6.0 in /Users/chf/anacond
a3/lib/python3.11/site-packages (from pydantic>=1.6->cobra) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.1 in /Users/chf/anaconda
3/lib/python3.11/site-packages (from pydantic>=1.6->cobra) (2.27.1)
Requirement already satisfied: typing-extensions>=4.12.2 in /Users/chf/anac
onda3/lib/python3.11/site-packages (from pydantic>=1.6->cobra) (4.12.2)
Requirement already satisfied: six>=1.5 in /Users/chf/anaconda3/lib/python
3.11/site-packages (from python-dateutil>=2.8.1->pandas<3.0,>=1.0->cobra)
(1.16.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in /Users/chf/anaconda
3/lib/python3.11/site-packages (from rich>=8.0->cobra) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /Users/chf/anacon
da3/lib/python3.11/site-packages (from rich>=8.0->cobra) (2.19.2)
Requirement already satisfied: mdurl~=0.1 in /Users/chf/anaconda3/lib/pytho
n3.11/site-packages (from markdown-it-py>=2.2.0->rich>=8.0->cobra) (0.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /Users/chf/anaconda3/l
ib/python3.11/site-packages (from sympy>=1.12.0->optlang~=1.8->cobra) (1.3.
0)
Requirement already satisfied: sniffio>=1.1 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from anyio->httpx~=0.24->cobra) (1.3.1)
```

```
Requirement already satisfied: cobra in /Users/chf/anaconda3/lib/python3.1
1/site-packages (0.30.0)
Requirement already satisfied: appdirs~=1.4 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from cobra) (1.4.4)
Requirement already satisfied: depinfo~=2.2 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from cobra) (2.2.0)
Requirement already satisfied: diskcache~=5.0 in /Users/chf/anaconda3/lib/p
ython3.11/site-packages (from cobra) (5.6.3)
Requirement already satisfied: future in /Users/chf/anaconda3/lib/python3.1
1/site-packages (from cobra) (0.18.3)
Requirement already satisfied: httpx~=0.24 in /Users/chf/anaconda3/lib/pyth
on3.11/site-packages (from cobra) (0.28.0)
Requirement already satisfied: numpy>=1.13 in /Users/chf/anaconda3/lib/pyth
on3.11/site-packages (from cobra) (1.23.5)
Requirement already satisfied: optlang~=1.8 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from cobra) (1.8.3)
Requirement already satisfied: pandas<3.0,>=1.0 in /Users/chf/anaconda3/li
b/python3.11/site-packages (from cobra) (1.5.3)
Requirement already satisfied: pydantic>=1.6 in /Users/chf/anaconda3/lib/py
thon3.11/site-packages (from cobra) (2.10.2)
Requirement already satisfied: python-libsbml~=5.19 in /Users/chf/anaconda
3/lib/python3.11/site-packages (from cobra) (5.21.0)
Requirement already satisfied: rich>=8.0 in /Users/chf/anaconda3/lib/python
3.11/site-packages (from cobra) (13.9.4)
Requirement already satisfied: ruamel.yaml~=0.16 in /Users/chf/anaconda3/li
b/python3.11/site-packages (from cobra) (0.17.21)
Requirement already satisfied: swiglpk in /Users/chf/anaconda3/lib/python3.
11/site-packages (from cobra) (5.0.13)
Requirement already satisfied: anyio in /Users/chf/anaconda3/lib/python3.1
1/site-packages (from httpx~=0.24->cobra) (3.5.0)
Requirement already satisfied: certifi in /Users/chf/anaconda3/lib/python3.
11/site-packages (from httpx~=0.24->cobra) (2024.8.30)
Requirement already satisfied: httpcore==1.* in /Users/chf/anaconda3/lib/py
thon3.11/site-packages (from httpx~=0.24->cobra) (1.0.7)
Requirement already satisfied: idna in /Users/chf/anaconda3/lib/python3.11/
site-packages (from httpx~=0.24->cobra) (3.4)
Requirement already satisfied: h11<0.15,>=0.13 in /Users/chf/anaconda3/lib/
python3.11/site-packages (from httpcore==1.*->httpx~=0.24->cobra) (0.14.0)
Requirement already satisfied: sympy>=1.12.0 in /Users/chf/anaconda3/lib/py
thon3.11/site-packages (from optlang~=1.8->cobra) (1.14.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/chf/anacond
a3/lib/python3.11/site-packages (from pandas<3.0,>=1.0->cobra) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from pandas<3.0,>=1.0->cobra) (2022.7)
Requirement already satisfied: annotated-types>=0.6.0 in /Users/chf/anacond
a3/lib/python3.11/site-packages (from pydantic>=1.6->cobra) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.1 in /Users/chf/anaconda
3/lib/python3.11/site-packages (from pydantic>=1.6->cobra) (2.27.1)
Requirement already satisfied: typing-extensions>=4.12.2 in /Users/chf/anac
onda3/lib/python3.11/site-packages (from pydantic>=1.6->cobra) (4.12.2)
Requirement already satisfied: six>=1.5 in /Users/chf/anaconda3/lib/python
3.11/site-packages (from python-dateutil>=2.8.1->pandas<3.0,>=1.0->cobra)
(1.16.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in /Users/chf/anaconda
3/lib/python3.11/site-packages (from rich>=8.0->cobra) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /Users/chf/anacon
da3/lib/python3.11/site-packages (from rich>=8.0->cobra) (2.19.2)
Requirement already satisfied: mdurl~=0.1 in /Users/chf/anaconda3/lib/pytho
n3.11/site-packages (from markdown-it-py>=2.2.0->rich>=8.0->cobra) (0.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /Users/chf/anaconda3/l
ib/python3.11/site-packages (from sympy>=1.12.0->optlang~=1.8->cobra) (1.3.
0)
Requirement already satisfied: sniffio>=1.1 in /Users/chf/anaconda3/lib/pyt
hon3.11/site-packages (from anyio->httpx~=0.24->cobra) (1.3.1)
```

```
Note: you may need to restart the kernel to use updated packages.
Note: you may need to restart the kernel to use updated packages.
```

In [23]:
```
import math
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import cobra
```

In [24]:
```
# load the models
fruit_model = cobra.io.read_sbml_model("wtfruit.xml")
veg_model = cobra.io.read_sbml_model("Supp_File_2.xml")

# load the solvers
veg_model.solver = 'glpk'
fruit_model.solver = 'glpk'
```

```
Model does not contain SBML fbc package information.
SBML package 'layout' not supported by cobrapy, information is not parsed
Model does not contain SBML fbc package information.
SBML package 'layout' not supported by cobrapy, information is not parsed
SBML package 'render' not supported by cobrapy, information is not parsed
SBML package 'render' not supported by cobrapy, information is not parsed
Adding exchange reaction EX_Suc_in with default bounds for boundary metabol
ite: Suc_in.
Adding exchange reaction EX_Gln_in with default bounds for boundary metabol
ite: Gln_in.
Adding exchange reaction EX_NO3 with default bounds for boundary metabolit
e: NO3.
Adding exchange reaction EX_CW with default bounds for boundary metabolite:
CW.
Adding exchange reaction EX_DAG with default bounds for boundary metabolit
e: DAG.
Adding exchange reaction EX_PROTEIN with default bounds for boundary metabo
lite: PROTEIN.
Adding exchange reaction EX_STARCH with default bounds for boundary metabol
ite: STARCH.
Adding exchange reaction EX_Glu_ac with default bounds for boundary metabol
ite: Glu_ac.
Adding exchange reaction EX_Asp_ac with default bounds for boundary metabol
ite: Asp_ac.
Adding exchange reaction EX_Suc_in with default bounds for boundary metabol
ite: Suc_in.
Adding exchange reaction EX_Gln_in with default bounds for boundary metabol
ite: Gln_in.
Adding exchange reaction EX_NO3 with default bounds for boundary metabolit
e: NO3.
Adding exchange reaction EX_CW with default bounds for boundary metabolite:
CW.
Adding exchange reaction EX_DAG with default bounds for boundary metabolit
e: DAG.
Adding exchange reaction EX_PROTEIN with default bounds for boundary metabo
lite: PROTEIN.
Adding exchange reaction EX_STARCH with default bounds for boundary metabol
ite: STARCH.
Adding exchange reaction EX_Glu_ac with default bounds for boundary metabol
ite: Glu_ac.
Adding exchange reaction EX_Asp_ac with default bounds for boundary metabol
ite: Asp_ac.
Adding exchange reaction EX_Ala_ac with default bounds for boundary metabol
ite: Ala_ac.
Adding exchange reaction EX_Ser_ac with default bounds for boundary metabol
ite: Ser_ac.
Adding exchange reaction EX_Cit_ac with default bounds for boundary metabol
ite: Cit_ac.
Adding exchange reaction EX_Mal_ac with default bounds for boundary metabol
ite: Mal_ac.
Adding exchange reaction EX_Ala_ac with default bounds for boundary metabol
ite: Ala_ac.
Adding exchange reaction EX_Ser_ac with default bounds for boundary metabol
ite: Ser_ac.
Adding exchange reaction EX_Cit_ac with default bounds for boundary metabol
ite: Cit_ac.
Adding exchange reaction EX_Mal_ac with default bounds for boundary metabol
ite: Mal_ac.
Adding exchange reaction EX_Glc_ac with default bounds for boundary metabol
ite: Glc_ac.
Adding exchange reaction EX_Glc_ac with default bounds for boundary metabol
ite: Glc_ac.
Adding exchange reaction EX_Fru_ac with default bounds for boundary metabol
ite: Fru_ac.
```

```
Adding exchange reaction EX_Suc_ac with default bounds for boundary metabol
ite: Suc_ac.
Adding exchange reaction EX_DNA_RNA with default bounds for boundary metabo
lite: DNA_RNA.
Adding exchange reaction EX_Fru_ac with default bounds for boundary metabol
ite: Fru_ac.
Adding exchange reaction EX_Suc_ac with default bounds for boundary metabol
ite: Suc_ac.
Adding exchange reaction EX_DNA_RNA with default bounds for boundary metabo
lite: DNA_RNA.
Adding exchange reaction EX_Pi with default bounds for boundary metabolite:
Pi.
Adding exchange reaction EX_Pi with default bounds for boundary metabolite:
Pi.
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtpi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtpi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtpi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgapdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtpi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgapdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgapdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgapdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Veno>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Veno>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Veno>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Veno>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vidh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vidh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vidh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vkgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vkgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vidh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vkgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vkgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VFum>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VFum>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VFum>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VFum>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vmdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vmdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vmdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vmdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vepi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vepi>'
```

```
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vriso>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vepi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vepi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vriso>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vriso>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vriso>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgm>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgm>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgm>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgm>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsusy>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsusy>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Valaat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Valaat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaspat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsusy>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsusy>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Valaat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Valaat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaspat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaspat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vserat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vserat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaspat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vserat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vserat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vadk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vadk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VSuc_up
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VSuc_up>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VGln_up
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VGln_up>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vadk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vadk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VSuc_up
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VSuc_up>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VGln_up
```

```
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VGln_up>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfbp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfbp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfbp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfbp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepc>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepck>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepck>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vscol>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vscol>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vme>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vme>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepc>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepck>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepck>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vscol>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vscol>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vme>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vme>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vg6pdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vg6pdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vrbco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vrbco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vagpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vagpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vugpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vugpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vinv>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vinv>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsps>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsps>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vspase>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vspase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vg6pdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vg6pdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vrbco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vrbco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vagpase
>'
```

```
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vagpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vugpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vugpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vinv>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vinv>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsps>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsps>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vspase>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vspase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgogat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgogat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnr>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnr>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfas>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfas>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Velong>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Velong>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vglyc3P
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vglyc3P>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vdag>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vdag>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcw>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgogat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgogat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnr>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnr>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfas>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfas>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Velong>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Velong>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vglyc3P
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vglyc3P>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vdag>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vdag>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcw>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcw>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Glu
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Glu>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Asp
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Asp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ala
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ala>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ser
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ser>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vprotein
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vprotein
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_cit
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_cit>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_mal
>'
```

```
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_mal>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_glc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_glc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_fru
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_fru>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_suc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_suc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vss>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vss>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsd>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsd>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj3>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj3>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcw>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Glu
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Glu>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Asp
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Asp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ala
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ala>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ser
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ser>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vprotein
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vprotein
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_cit
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_cit>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_mal
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_mal>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_glc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_glc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_fru
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_fru>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_suc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_suc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vss>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vss>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsd>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsd>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj3>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj3>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnga_ATP
m>'
```

```
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnga_ATPm
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vppi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vppi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnucleo
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnucleo>'
No objective coefficients in model. Unclear what should be optimized
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnga_ATP
m>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnga_ATPm
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vppi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vppi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnucleo
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnucleo>'
No objective coefficients in model. Unclear what should be optimized
Missing flux bounds on reactions set to default bounds.As best practise and
to avoid confusion flux bounds should be set explicitly on all reactions.
Missing flux bounds on reactions set to default bounds.As best practise and
to avoid confusion flux bounds should be set explicitly on all reactions.
Adding exchange reaction EX_biomass_stem_b with default bounds for boundary
metabolite: biomass_stem_b.
Adding exchange reaction EX_orn_b with default bounds for boundary metaboli
te: orn_b.
Adding exchange reaction EX_biomass_stem_b with default bounds for boundary
metabolite: biomass_stem_b.
Adding exchange reaction EX_orn_b with default bounds for boundary metaboli
te: orn_b.
Adding exchange reaction EX_mg2_b with default bounds for boundary metaboli
te: mg2_b.
Adding exchange reaction EX_mg2_b with default bounds for boundary metaboli
te: mg2_b.
Adding exchange reaction EX_5oxpro_b with default bounds for boundary metab
olite: 5oxpro_b.
Adding exchange reaction EX_5oxpro_b with default bounds for boundary metab
olite: 5oxpro_b.
Adding exchange reaction EX_his_L_b with default bounds for boundary metabo
lite: his_L_b.
Adding exchange reaction EX_his_L_b with default bounds for boundary metabo
lite: his_L_b.
Adding exchange reaction EX_na_b with default bounds for boundary metabolit
e: na_b.
Adding exchange reaction EX_na_b with default bounds for boundary metabolit
e: na_b.
Adding exchange reaction EX_cys_L_b with default bounds for boundary metabo
lite: cys_L_b.
Adding exchange reaction EX_cys_L_b with default bounds for boundary metabo
lite: cys_L_b.
Adding exchange reaction EX_starch300_b with default bounds for boundary me
tabolite: starch300_b.
Adding exchange reaction EX_starch300_b with default bounds for boundary me
tabolite: starch300_b.
Adding exchange reaction EX_ump_b with default bounds for boundary metaboli
te: ump_b.
Adding exchange reaction EX_ump_b with default bounds for boundary metaboli
te: ump_b.
Adding exchange reaction EX_coumarylalc_b with default bounds for boundary
metabolite: coumarylalc_b.
Adding exchange reaction EX_coumarylalc_b with default bounds for boundary
metabolite: coumarylalc_b.
Adding exchange reaction EX_succ_b with default bounds for boundary metabol
ite: succ_b.
```

```
Adding exchange reaction EX_succ_b with default bounds for boundary metabol
ite: succ_b.
Adding exchange reaction EX_xylan4_b with default bounds for boundary metab
olite: xylan4_b.
Adding exchange reaction EX_xylan4_b with default bounds for boundary metab
olite: xylan4_b.
Adding exchange reaction EX_etoh_b with default bounds for boundary metabol
ite: etoh_b.
Adding exchange reaction EX_etoh_b with default bounds for boundary metabol
ite: etoh_b.
Adding exchange reaction EX_val_L_b with default bounds for boundary metabo
lite: val_L_b.
Adding exchange reaction EX_val_L_b with default bounds for boundary metabo
lite: val_L_b.
Adding exchange reaction EX_pro_L_b with default bounds for boundary metabo
lite: pro_L_b.
Adding exchange reaction EX_pro_L_b with default bounds for boundary metabo
lite: pro_L_b.
Adding exchange reaction EX_biomass_leaf_b with default bounds for boundary
metabolite: biomass_leaf_b.
Adding exchange reaction EX_biomass_leaf_b with default bounds for boundary
metabolite: biomass_leaf_b.
Adding exchange reaction EX_asn_L_b with default bounds for boundary metabo
lite: asn_L_b.
Adding exchange reaction EX_asn_L_b with default bounds for boundary metabo
lite: asn_L_b.
Adding exchange reaction EX_met_L_b with default bounds for boundary metabo
lite: met_L_b.
Adding exchange reaction EX_met_L_b with default bounds for boundary metabo
lite: met_L_b.
Adding exchange reaction EX_phe_L_b with default bounds for boundary metabo
lite: phe_L_b.
Adding exchange reaction EX_phe_L_b with default bounds for boundary metabo
lite: phe_L_b.
Adding exchange reaction EX_confrl_b with default bounds for boundary metab
olite: confrl_b.
Adding exchange reaction EX_confrl_b with default bounds for boundary metab
olite: confrl_b.
Adding exchange reaction EX_atp_cost_b with default bounds for boundary met
abolite: atp_cost_b.
Adding exchange reaction EX_atp_cost_b with default bounds for boundary met
abolite: atp_cost_b.
Adding exchange reaction EX_tyr_L_b with default bounds for boundary metabo
lite: tyr_L_b.
Adding exchange reaction EX_tyr_L_b with default bounds for boundary metabo
lite: tyr_L_b.
Adding exchange reaction EX_fe2_b with default bounds for boundary metaboli
te: fe2_b.
Adding exchange reaction EX_fe2_b with default bounds for boundary metaboli
te: fe2_b.
Adding exchange reaction EX_nh4_b with default bounds for boundary metaboli
te: nh4_b.
Adding exchange reaction EX_nh4_b with default bounds for boundary metaboli
te: nh4_b.
Adding exchange reaction EX_ser_L_b with default bounds for boundary metabo
lite: ser_L_b.
Adding exchange reaction EX_ser_L_b with default bounds for boundary metabo
lite: ser_L_b.
Adding exchange reaction EX_sinapylalc_b with default bounds for boundary m
etabolite: sinapylalc_b.
Adding exchange reaction EX_sinapylalc_b with default bounds for boundary m
etabolite: sinapylalc_b.
Adding exchange reaction EX_gly_b with default bounds for boundary metaboli
te: gly_b.
```

```
Adding exchange reaction EX_gly_b with default bounds for boundary metaboli
te: gly_b.
Adding exchange reaction EX_ala_B_b with default bounds for boundary metabo
lite: ala_B_b.
Adding exchange reaction EX_ala_B_b with default bounds for boundary metabo
lite: ala_B_b.
Adding exchange reaction EX_o2_b with default bounds for boundary metabolit
e: o2_b.
Adding exchange reaction EX_o2_b with default bounds for boundary metabolit
e: o2_b.
Adding exchange reaction EX_glu_L_b with default bounds for boundary metabo
lite: glu_L_b.
Adding exchange reaction EX_glu_L_b with default bounds for boundary metabo
lite: glu_L_b.
Adding exchange reaction EX_gmp_b with default bounds for boundary metaboli
te: gmp_b.
Adding exchange reaction EX_gmp_b with default bounds for boundary metaboli
te: gmp_b.
Adding exchange reaction EX_glyclt_b with default bounds for boundary metab
olite: glyclt_b.
Adding exchange reaction EX_glyclt_b with default bounds for boundary metab
olite: glyclt_b.
Adding exchange reaction EX_biomass_root_b with default bounds for boundary
metabolite: biomass_root_b.
Adding exchange reaction EX_biomass_root_b with default bounds for boundary
metabolite: biomass_root_b.
Adding exchange reaction EX_fum_b with default bounds for boundary metaboli
te: fum_b.
Adding exchange reaction EX_fum_b with default bounds for boundary metaboli
te: fum_b.
Adding exchange reaction EX_thr_L_b with default bounds for boundary metabo
lite: thr_L_b.
Adding exchange reaction EX_thr_L_b with default bounds for boundary metabo
lite: thr_L_b.
Adding exchange reaction EX_nadh_work_b with default bounds for boundary me
tabolite: nadh_work_b.
Adding exchange reaction EX_nadh_work_b with default bounds for boundary me
tabolite: nadh_work_b.
Adding exchange reaction EX_asp_L_b with default bounds for boundary metabo
lite: asp_L_b.
Adding exchange reaction EX_asp_L_b with default bounds for boundary metabo
lite: asp_L_b.
Adding exchange reaction EX_h_b with default bounds for boundary metabolit
e: h_b.
Adding exchange reaction EX_h_b with default bounds for boundary metabolit
e: h_b.
Adding exchange reaction EX_cit_b with default bounds for boundary metaboli
te: cit_b.
Adding exchange reaction EX_cit_b with default bounds for boundary metaboli
te: cit_b.
Adding exchange reaction EX_g6p_b with default bounds for boundary metaboli
te: g6p_b.
Adding exchange reaction EX_g6p_b with default bounds for boundary metaboli
te: g6p_b.
Adding exchange reaction EX_cmp_b with default bounds for boundary metaboli
te: cmp_b.
Adding exchange reaction EX_cmp_b with default bounds for boundary metaboli
te: cmp_b.
Adding exchange reaction EX_co2_b with default bounds for boundary metaboli
te: co2_b.
Adding exchange reaction EX_co2_b with default bounds for boundary metaboli
te: co2_b.
Adding exchange reaction EX_h2o_b with default bounds for boundary metaboli
te: h2o_b.
```

```
Adding exchange reaction EX_h2o_b with default bounds for boundary metaboli
te: h2o_b.
Adding exchange reaction EX_b14glucan_b with default bounds for boundary me
tabolite: b14glucan_b.
Adding exchange reaction EX_b14glucan_b with default bounds for boundary me
tabolite: b14glucan_b.
Adding exchange reaction EX_pi_b with default bounds for boundary metabolit
e: pi_b.
Adding exchange reaction EX_pi_b with default bounds for boundary metabolit
e: pi_b.
Adding exchange reaction EX_atp_work_b with default bounds for boundary met
abolite: atp_work_b.
Adding exchange reaction EX_atp_work_b with default bounds for boundary met
abolite: atp_work_b.
Adding exchange reaction EX_arg_L_b with default bounds for boundary metabo
lite: arg_L_b.
Adding exchange reaction EX_arg_L_b with default bounds for boundary metabo
lite: arg_L_b.
Adding exchange reaction EX_glyc_b with default bounds for boundary metabol
ite: glyc_b.
Adding exchange reaction EX_sucr_b with default bounds for boundary metabol
ite: sucr_b.
Adding exchange reaction EX_glyc_b with default bounds for boundary metabol
ite: glyc_b.
Adding exchange reaction EX_sucr_b with default bounds for boundary metabol
ite: sucr_b.
Adding exchange reaction EX_so4_b with default bounds for boundary metaboli
te: so4_b.
Adding exchange reaction EX_so4_b with default bounds for boundary metaboli
te: so4_b.
Adding exchange reaction EX_tmp_b with default bounds for boundary metaboli
te: tmp_b.
Adding exchange reaction EX_tmp_b with default bounds for boundary metaboli
te: tmp_b.
Adding exchange reaction EX_fru_b with default bounds for boundary metaboli
te: fru_b.
Adding exchange reaction EX_fru_b with default bounds for boundary metaboli
te: fru_b.
Adding exchange reaction EX_gln_L_b with default bounds for boundary metabo
lite: gln_L_b.
Adding exchange reaction EX_gln_L_b with default bounds for boundary metabo
lite: gln_L_b.
Adding exchange reaction EX_cl_b with default bounds for boundary metabolit
e: cl_b.
Adding exchange reaction EX_cl_b with default bounds for boundary metabolit
e: cl_b.
Adding exchange reaction EX_4abut_b with default bounds for boundary metabo
lite: 4abut_b.
Adding exchange reaction EX_4abut_b with default bounds for boundary metabo
lite: 4abut_b.
Adding exchange reaction EX_amp_b with default bounds for boundary metaboli
te: amp_b.
Adding exchange reaction EX_amp_b with default bounds for boundary metaboli
te: amp_b.
Adding exchange reaction EX_k_b with default bounds for boundary metabolit
e: k_b.
Adding exchange reaction EX_k_b with default bounds for boundary metabolit
e: k_b.
Adding exchange reaction EX_dgmp_b with default bounds for boundary metabol
ite: dgmp_b.
Adding exchange reaction EX_dgmp_b with default bounds for boundary metabol
ite: dgmp_b.
Adding exchange reaction EX_ile_L_b with default bounds for boundary metabo
lite: ile_L_b.
```

```
Adding exchange reaction EX_ile_L_b with default bounds for boundary metabo
lite: ile_L_b.
Adding exchange reaction EX_ca2_b with default bounds for boundary metaboli
te: ca2_b.
Adding exchange reaction EX_ca2_b with default bounds for boundary metaboli
te: ca2_b.
Adding exchange reaction EX_mal_L_b with default bounds for boundary metabo
lite: mal_L_b.
Adding exchange reaction EX_mal_L_b with default bounds for boundary metabo
lite: mal_L_b.
Adding exchange reaction EX_trp_L_b with default bounds for boundary metabo
lite: trp_L_b.
Adding exchange reaction EX_trp_L_b with default bounds for boundary metabo
lite: trp_L_b.
Adding exchange reaction EX_dcmp_b with default bounds for boundary metabol
ite: dcmp_b.
Adding exchange reaction EX_dcmp_b with default bounds for boundary metabol
ite: dcmp_b.
Adding exchange reaction EX_ascb_L_b with default bounds for boundary metab
olite: ascb_L_b.
Adding exchange reaction EX_ascb_L_b with default bounds for boundary metab
olite: ascb_L_b.
Adding exchange reaction EX_glyc3p_b with default bounds for boundary metab
olite: glyc3p_b.
Adding exchange reaction EX_glyc3p_b with default bounds for boundary metab
olite: glyc3p_b.
Adding exchange reaction EX_no3_b with default bounds for boundary metaboli
te: no3_b.
Adding exchange reaction EX_no3_b with default bounds for boundary metaboli
te: no3_b.
Adding exchange reaction EX_glc_D_b with default bounds for boundary metabo
lite: glc_D_b.
Adding exchange reaction EX_glc_D_b with default bounds for boundary metabo
lite: glc_D_b.
Adding exchange reaction EX_icit_b with default bounds for boundary metabol
ite: icit_b.
Adding exchange reaction EX_icit_b with default bounds for boundary metabol
ite: icit_b.
Adding exchange reaction EX_photon_b with default bounds for boundary metab
olite: photon_b.
Adding exchange reaction EX_photon_b with default bounds for boundary metab
olite: photon_b.
Adding exchange reaction EX_damp_b with default bounds for boundary metabol
ite: damp_b.
Adding exchange reaction EX_damp_b with default bounds for boundary metabol
ite: damp_b.
Adding exchange reaction EX_chlb_b with default bounds for boundary metabol
ite: chlb_b.
Adding exchange reaction EX_chlb_b with default bounds for boundary metabol
ite: chlb_b.
Adding exchange reaction EX_ala_L_b with default bounds for boundary metabo
lite: ala_L_b.
Adding exchange reaction EX_ala_L_b with default bounds for boundary metabo
lite: ala_L_b.
Adding exchange reaction EX_lys_L_b with default bounds for boundary metabo
lite: lys_L_b.
Adding exchange reaction EX_lys_L_b with default bounds for boundary metabo
lite: lys_L_b.
Adding exchange reaction EX_akg_b with default bounds for boundary metaboli
te: akg_b.
Adding exchange reaction EX_akg_b with default bounds for boundary metaboli
te: akg_b.
Adding exchange reaction EX_nadph_work_b with default bounds for boundary m
etabolite: nadph_work_b.
```

```
Adding exchange reaction EX_nadph_work_b with default bounds for boundary m
etabolite: nadph_work_b.
Adding exchange reaction EX_leu_L_b with default bounds for boundary metabo
lite: leu_L_b.
Adding exchange reaction EX_leu_L_b with default bounds for boundary metabo
lite: leu_L_b.
Adding exchange reaction EX_cholphya_b with default bounds for boundary met
abolite: cholphya_b.
Adding exchange reaction EX_cholphya_b with default bounds for boundary met
abolite: cholphya_b.
Adding exchange reaction EX_ura_b with default bounds for boundary metaboli
te: ura_b.
Adding exchange reaction EX_ura_b with default bounds for boundary metaboli
te: ura_b.
MNXR98353 does not conform to 'http(s)://identifiers.org/collection/id' o
r'http(s)://identifiers.org/COLLECTION:id
MNXR98353 does not conform to 'http(s)://identifiers.org/collection/id' o
r'http(s)://identifiers.org/COLLECTION:id
```

In [25]:
```
from cobra.util import create_stoichiometric_matrix
S = create_stoichiometric_matrix(veg_model, array_type='DataFrame')
print(S.shape)  # (metabolites × 2183 reactions)
```

```
(2097, 2261)
(2097, 2261)
```

In [26]:
```
print(len(veg_model.reactions))
print(len(veg_model.exchanges))
print(len(veg_model.boundary))
```

```
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
2261
156
78
2261
156
78
```

In [27]:
```
2261-(156-78)
```

Out[27]:
```
2183
```

Out[27]:
```
2183
```

In [28]:
```
print(len(fruit_model.reactions))
print(len(fruit_model.exchanges))
print(len(fruit_model.boundary))
```

```
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
```

```
89
18
18
89
18
18
```

In [ ]:

In [29]:
```
S
```

Out[29]:

	EX_biomass_stem_b	EX_orn_b	EX_mg2_b	EX_5oxpro_b	EX_l
M00957_c	0.0	0.0	0.0	0.0	
q9h2_m	0.0	0.0	0.0	0.0	
ppp9_h	0.0	0.0	0.0	0.0	
m14bq_c	0.0	0.0	0.0	0.0	
3hmp_m	0.0	0.0	0.0	0.0	
...	...	...	...	...	
ifucosterol_c	0.0	0.0	0.0	0.0	
psd5p_c	0.0	0.0	0.0	0.0	
limnen_c	0.0	0.0	0.0	0.0	
pc1829Z12Z1839Z12Z15Z_c	0.0	0.0	0.0	0.0	
indmthhate_c	0.0	0.0	0.0	0.0	

2097 rows × 2261 columns

Out[29]:

	EX_biomass_stem_b	EX_orn_b	EX_mg2_b	EX_5oxpro_b	EX_l
M00957_c	0.0	0.0	0.0	0.0	
q9h2_m	0.0	0.0	0.0	0.0	
ppp9_h	0.0	0.0	0.0	0.0	
m14bq_c	0.0	0.0	0.0	0.0	
3hmp_m	0.0	0.0	0.0	0.0	
...	...	...	...	...	
ifucosterol_c	0.0	0.0	0.0	0.0	
psd5p_c	0.0	0.0	0.0	0.0	
limnen_c	0.0	0.0	0.0	0.0	
pc1829Z12Z1839Z12Z15Z_c	0.0	0.0	0.0	0.0	
indmthhate_c	0.0	0.0	0.0	0.0	

2097 rows × 2261 columns

In [30]:
```
# ================================================================================
# MODEL EXTENSIONS
# Patch wtfruit.xml with reactions missing from the raw SBML, then apply
# boundary direction rules to both models. Call build_fruit_model() once.
# ================================================================================
```

```
VEG_ALLOWED_IMPORTS = {
    'EX_photon_b', 'EX_co2_b', 'EX_h2o_b',
    'EX_no3_b', 'EX_nh4_b', 'EX_pi_b', 'EX_so4_b', 'EX_k_b',
}
VEG_ALLOWED_EXPORTS = {
    'EX_sucr_b', 'EX_gln_L_b',
    'EX_asp_L_b', 'EX_glu_L_b', 'EX_ala_L_b',
    'EX_co2_b', 'EX_o2_b', 'EX_h2o_b',
}

def setup_veg_model(model):
    """Lock non-biological exchanges; enforce export-only on phloem routes."
    for rxn in model.boundary:
        if rxn.id in VEG_ALLOWED_IMPORTS or rxn.id in VEG_ALLOWED_EXPORTS:
            continue
        rxn.lower_bound = 0.0
        rxn.upper_bound = 0.0
    # Without lb=0 on phloem exports, the Gln-maximisation step back-imports
    # Ala at 1000 mmol/g/h, inflating downstream biomass by >1000-fold (Fix
    for rxn_id in ['EX_sucr_b', 'EX_gln_L_b', 'EX_asp_L_b', 'EX_glu_L_b', 'E
        if rxn_id in {r.id for r in model.reactions}:
            model.reactions.get_by_id(rxn_id).lower_bound = 0.0

def _add_rxn(model, rxn_id, name, met_dict, lb=0.0, ub=1000.0):
    """Add a reaction using existing model metabolites, skipping if already
    if rxn_id in {r.id for r in model.reactions}:
        return
    mets = {m.id: m for m in model.metabolites}
    rxn = cobra.Reaction(rxn_id)
    rxn.name = name
    rxn.bounds = (lb, ub)
    rxn.add_metabolites({mets[k]: v for k, v in met_dict.items()})
    model.add_reactions([rxn])

def add_co2_export(model):
    _add_rxn(model, 'EX_CO2', 'CO2 respiratory export', {'CO2': -1.0})

def add_nh4_import(model):
    _add_rxn(model, 'EX_NH4', 'NH4 xylem import', {'NH4': -1.0}, lb=0.0, ub=

def add_phloem_aa_imports(model):
    for met_id, rxn_id in [('Asp','EX_Asp_in'), ('Glu','EX_Glu_in'), ('Ala',
        _add_rxn(model, rxn_id, f'{met_id} phloem import', {met_id: -1.0},

def add_lycopene_pathway(model):
    if 'Vlycop' in {r.id for r in model.reactions}:
        return
    from cobra import Metabolite
    mets   = {m.id: m for m in model.metabolites}
    Lyc    = Metabolite('Lyc',    name='Lycopene',             compartment=
    Lyc_ac = Metabolite('Lyc_ac', name='Lycopene_chromoplast', compartment=
    vlycop = cobra.Reaction('Vlycop')
    vlycop.name = 'Lycopene synthesis (MEP lumped)'
    vlycop.bounds = (0.0, 1000.0)
    vlycop.add_metabolites({
        mets['Pyr']: -8.0, mets['GAP']: -8.0, mets['NADPH']: -16.0,
        mets['ATP']: -8.0, mets['FAD']: -4.0,
        Lyc: 1.0, mets['CO2']: 8.0, mets['NADP']: 16.0,
        mets['ADP']: 8.0, mets['FADH2']: 4.0, mets['PPi']: 8.0,
    })
    vac_lyc = cobra.Reaction('Vac_lyc')
    vac_lyc.bounds = (0.0, 1000.0)
    vac_lyc.add_metabolites({Lyc: -1.0, Lyc_ac: 1.0})
    ex_lyc = cobra.Reaction('EX_Lyc_ac')
```

```
    ex_lyc.bounds = (-1000.0, 1000.0)
    ex_lyc.add_metabolites({Lyc_ac: -1.0})
    model.add_reactions([vlycop, vac_lyc, ex_lyc])

def add_k_transport(model):
    if 'EX_K' in {r.id for r in model.reactions}:
        return
    from cobra import Metabolite
    K     = Metabolite('K',     name='Potassium_cytosol',  compartment='c')
    K_vac = Metabolite('K_vac', name='Potassium_vacuolar', compartment='v')
    ex_k  = cobra.Reaction('EX_K');     ex_k.name  = 'K xylem import'; ex_k
    vac_k = cobra.Reaction('Vac_K');    vac_k.bounds = (0.0, 1000.0)
    ex_kv = cobra.Reaction('EX_K_vac'); ex_kv.bounds = (-1000.0, 1000.0)
    ex_k.add_metabolites({K: -1.0})
    vac_k.add_metabolites({K: -1.0, K_vac: 1.0})
    ex_kv.add_metabolites({K_vac: -1.0})
    model.add_reactions([ex_k, vac_k, ex_kv])

def setup_fruit_model(model):
    """Set exchange directions from dead-end analysis on the modified model.
    input_reactions = {
        "EX_Suc_in", "EX_Gln_in", "EX_NO3", "EX_NH4", "EX_K",
        "EX_Asp_in", "EX_Glu_in", "EX_Ala_in",
    }
    must_export = set()
    for ex_rxn in model.exchanges:
        mets_list = list(ex_rxn.metabolites.keys())
        if not mets_list:
            continue
        met = mets_list[0]
        has_producer = any(r.id != ex_rxn.id and r.metabolites.get(met, 0) >
        has_consumer = any(r.id != ex_rxn.id and r.metabolites.get(met, 0) <
        if has_producer and not has_consumer:
            must_export.add(ex_rxn.id)
    # EX_STARCH and EX_CO2 have internal cycles so dead-end check misses the
    output_reactions = must_export | {"EX_STARCH", "EX_CO2"}
    for rxn in model.exchanges:
        if rxn.id in input_reactions:
            rxn.upper_bound = 0.0
        elif rxn.id in output_reactions:
            rxn.lower_bound = 0.0
        else:
            rxn.lower_bound = 0.0
            rxn.upper_bound = 0.0

def build_fruit_model(model):
    add_co2_export(model)
    add_nh4_import(model)
    add_phloem_aa_imports(model)
    add_lycopene_pathway(model)
    add_k_transport(model)
    setup_fruit_model(model)

setup_veg_model(veg_model)
build_fruit_model(fruit_model)
print("Models configured.")
```

```
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Models configured.
Models configured.
```

In [31]:

```python
# ============================================================================
# PARAMETERS & CONSTANTS
# ============================================================================

# ── VYTOP calibration (Gerlin et al. 2022, Plant Physiology 188:1709–1723)
# Veg model (Supp_File_2.xml) fluxes are in mmol · g[plant DW]⁻¹ · h⁻¹.
# Reference condition: 200 µmol m⁻² s⁻¹, 12 h photoperiod.
#   Photon: 245 mmol/g_plant/day  ÷ 12 light-h = 20.42/light-h
#   Sucrose:  1.0 mmol/g_plant/day ÷ 12         =  0.083/light-h
#   Gln: ~4 % of sucrose by moles (phloem sap: Gln ~12 mM, Suc ~300 mM)
#   *** VYTOP_GLN_PER_LH_AT200 needs verification against VYTOP paper. ***
VYTOP_PHOTON_PER_LH_AT200 = 245.0 / 12.0
VYTOP_SUCR_PER_LH_AT200   =   1.0 / 12.0
VYTOP_GLN_PER_LH_AT200    =  0.04 / 12.0

# Scaling: veg LP output is per g plant DW; multiply by PLANT_DW_PER_FRUIT
# to get per-fruit values. Typical fruiting tomato: ~150–250 g plant DW,
# ~20 fruits → 8 g/fruit. *** Measure from actual experiment before publicat
PLANT_DW_PER_FRUIT = 8.0   # g plant DW per fruit

# ── Environment (replace with sensor data in production) ──────────────────
N_DAYS    = 46
DPA_RANGE = range(8, 54)

tank_no3    = np.full(N_DAYS, 14.0)   # mM in nutrient solution
tank_pi     = np.full(N_DAYS,  1.0)
tank_k      = np.full(N_DAYS,  6.0)
tank_nh4    = np.full(N_DAYS,  0.5)
ppfd_day    = np.full(N_DAYS, 200.0)  # µmol m⁻² s⁻¹ daily average
photoperiod = np.full(N_DAYS,  16.0)  # light hours per day

# Fraction of xylem mineral flow routed to fruit (vs. rest of plant).
xylem_no3_frac = np.full(N_DAYS, 0.15)
nh4_fruit_frac = np.full(N_DAYS, 0.20)
k_fruit_frac   = np.full(N_DAYS, 0.35)

# ── Pool initial conditions (mmol per fruit) ──────────────────────────────
POOL_INIT = {
    "phloem_suc": 0.05,  "phloem_gln": 0.005,
    "phloem_asp": 0.002, "phloem_glu": 0.002, "phloem_ala": 0.002,
    "xylem_no3":  0.05,  "xylem_pi":   0.01,
    "xylem_k":    0.10,  "xylem_nh4":  0.01,
}

# Pool caps (mmol/fruit): prevent unbounded accumulation during low-demand p
```

```
# Phloem = transit pools; xylem = daily delivery buffer.
# Calibration target: cumulative fruit biomass ~50–200 mmol for a 100 g toma
MAX_PHLOEM_SUC = 1.00;  MAX_PHLOEM_GLN = 0.05
MAX_PHLOEM_ASP = 0.01;  MAX_PHLOEM_GLU = 0.01;  MAX_PHLOEM_ALA = 0.01
MAX_XYLEM_NO3  = 1.00;  MAX_XYLEM_PI   = 0.30
MAX_XYLEM_K    = 2.00;  MAX_XYLEM_NH4  = 0.40

# ── Gompertz fruit growth (sink strength) ──────────────────────────────
# *** w_max, k, t_m should be fit to fresh-weight growth data before publica
GOMPERTZ = {"w_max": 100.0, "k": 0.15, "t_m": 25.0}

# ── Feedback suppression parameters ────────────────────────────────────
# Each FB reduces uptake rate when its pool exceeds the reference level.
FB_SUC_REF, FB_SUC_MAX = 0.5, 0.50   # phloem sucrose → leaf loading
FB_NO3_REF, FB_NO3_MAX = 0.5, 0.40   # xylem NO3     → root uptake
FB_PI_REF,  FB_PI_MAX  = 0.2, 0.50   # xylem Pi      → root uptake

# ── Root respiration coupling ──────────────────────────────────────────
# Carbon cost of active mineral uptake: phloem sucrose is consumed to supply
# *** ATP_PER_* : replace with root FBA model or measured values (Bloom 1992
#     Glass 2003). ATP_PER_SUC is well-constrained biochemistry (low priorit
ATP_PER_NO3 = 2.0   # mol ATP per mol NO3 (H⁺/NO3⁻ symport)
ATP_PER_NH4 = 1.0   # mol ATP per mol NH4 (lower cost, already reduced)
ATP_PER_PI  = 2.0   # mol ATP per mol Pi  (H⁺/Pi symport)
ATP_PER_K   = 0.5   # mol ATP per mol K   (channel-mediated, cheap)
ATP_PER_SUC = 30.0  # mol ATP per mol sucrose oxidised (glycolysis + TCA + (
```

In [32]:

```
# ═══════════════════════════════════════════════════════════════════════
# HELPER FUNCTIONS
# ═══════════════════════════════════════════════════════════════════════

# ── Root kinetic uptake (Michaelis–Menten, mmol h⁻¹ per plant) ──────────
def no3_ceiling(conc_mM):
    Vmax, Km = 50.0 / 24, 0.5    # Siddiqi et al. 1990
    return Vmax * conc_mM / (Km + conc_mM)

def pi_ceiling(conc_mM):
    Vmax, Km = 15.0 / 24, 0.05   # Schachtman et al. 1998
    return Vmax * conc_mM / (Km + conc_mM)

def k_ceiling(conc_mM):
    Vmax, Km = 60.0 / 24, 0.2    # Maathuis & Sanders 1997
    return Vmax * conc_mM / (Km + conc_mM)

def nh4_ceiling(conc_mM):
    Vmax, Km = 10.0 / 24, 0.3
    return Vmax * conc_mM / (Km + conc_mM)

# ── Light ──────────────────────────────────────────────────────────────
def diurnal_ppfd(ppfd_avg, hour, photoperiod_h=16):
    """Sinusoidal diurnal light profile centred on solar noon."""
    sunrise = int((24 - photoperiod_h) / 2)
    sunset  = sunrise + int(photoperiod_h)
    if sunrise <= hour < sunset:
        phase = (hour - sunrise) / photoperiod_h
        return ppfd_avg * (math.pi / 2) * math.sin(math.pi * phase)
    return 0.0

# ── Leaf carbon-priority ratio ─────────────────────────────────────────
def c_priority_from_phloem_cn(phloem_cn_molar=18.0):
    """Fraction of max sucrose export locked after lexicographic step 1."""
    return max(0.5, min(0.95, (2 * phloem_cn_molar - 5) / (2 * phloem_cn_mo

# ── Sink strength (normalised Gompertz derivative, clamped to [0.05, 0.90]
```

```
def sink_strength(dpa):
    w_max, k, t_m = GOMPERTZ["w_max"], GOMPERTZ["k"], GOMPERTZ["t_m"]
    W = w_max * math.exp(-math.exp(-k * (dpa - t_m)))
    demand     = k * W * math.log(w_max / W) if 0 < W < w_max else 0.0
    max_demand = (w_max * k) / math.e
    return min(max(demand / max_demand, 0.05), 0.90)

# ── Inter-organ feedback suppressors ──────────────────────────────
def phloem_loading_fb(S_suc):
    """Suppress leaf phloem loading when sucrose pool is high (Krapp & Stitt
    return max(1.0 - FB_SUC_MAX, 1.0 - FB_SUC_MAX * min(1.0, S_suc / FB_SUC_

def root_no3_fb(S_no3):
    return max(1.0 - FB_NO3_MAX, 1.0 - FB_NO3_MAX * min(1.0, S_no3 / FB_NO3_

def root_pi_fb(S_pi):
    return max(1.0 - FB_PI_MAX, 1.0 - FB_PI_MAX * min(1.0, S_pi / FB_PI_REF)

# ── Phase-dependent fruit objective ───────────────────────────────
def get_fruit_objective(model, dpa):
    if dpa <= 15:   # cell division
        weights = [("EX_DNA_RNA",1.0),("EX_PROTEIN",1.0),
                   ("EX_CW",0.3),("EX_STARCH",0.1),("EX_DAG",0.1)]
    elif dpa <= 35: # cell expansion
        weights = [("EX_STARCH",1.0),("EX_CW",0.8),("EX_PROTEIN",0.5),
                   ("EX_DAG",0.3),("EX_DNA_RNA",0.2),("EX_K_vac",0.4)]
    else:           # ripening
        weights = [("EX_Glc_ac",1.0),("EX_Fru_ac",1.0),("EX_Suc_ac",0.8),
                   ("EX_Cit_ac",0.7),("EX_Mal_ac",0.5),
                   ("EX_Lyc_ac",0.6),("EX_PROTEIN",0.2)]
    return {model.reactions.get_by_id(r): w for r, w in weights}
```

In [33]:

```
# ═══════════════════════════════════════════════════════════════════════
# KALMAN FILTER — POOL STATE MANAGER
# ═══════════════════════════════════════════════════════════════════════
# Replaces bare pool floats with a Gaussian belief (x̂, P).
# Predict step: called every simulated hour with LP-derived pool deltas.
# Update  step: called once per DPA when cv_ml_measurements provides a row.
#
# State vector indices (10-dim)
IDX_PHLOEM_SUC, IDX_PHLOEM_GLN                  = 0, 1
IDX_PHLOEM_ASP, IDX_PHLOEM_GLU, IDX_PHLOEM_ALA = 2, 3, 4
IDX_XYLEM_NO3,  IDX_XYLEM_PI                     = 5, 6
IDX_XYLEM_K,    IDX_XYLEM_NH4                    = 7, 8
IDX_CUM_BIOMASS                                  = 9
N_STATE, N_OBS = 10, 2

H_DIAMETER_PER_MMOL  = 0.50
H_BIOMASS_G_PER_MMOL = 0.18


class PoolKalmanFilter:
    """
    Linear Kalman filter managing 9 metabolic pool variables + cumulative bi

    CV-ML observation vector z (2-dim):
        z[0]  fruit_diameter_mm   → cumulative_biomass
        z[1]  fruit_dry_weight_g  → cumulative_biomass

    innovation_log: {dpa: N_OBS innovation vector} — used for fault detectio
    """

    def __init__(self, x0, Q_diag=None, R_default_diag=None):
        self.x = np.array(x0, dtype=float)
```

```
        self.P = np.diag(np.maximum(np.abs(self.x) * 0.1, 1e-8))
        self.F = np.eye(N_STATE)

        if Q_diag is None:
            Q_diag = np.array([1e-4, 1e-5, 1e-6, 1e-6, 1e-6,
                               1e-3, 1e-4, 1e-3, 1e-4, 1e-3])
        self.Q = np.diag(Q_diag)

        self.H = np.zeros((N_OBS, N_STATE))
        self.H[0, IDX_CUM_BIOMASS] = H_DIAMETER_PER_MMOL
        self.H[1, IDX_CUM_BIOMASS] = H_BIOMASS_G_PER_MMOL

        if R_default_diag is None:
            R_default_diag = np.array([25.0, 0.25])
        self.R_default = np.diag(np.asarray(R_default_diag, dtype=float))
        self._I = np.eye(N_STATE)
        self.innovation_log = {}   # {dpa: full N_OBS innovation vector}

    def predict(self, delta_x):
        """Advance state by LP-derived pool changes."""
        self.x = self.F @ self.x + np.asarray(delta_x, dtype=float)
        self.x = np.maximum(self.x, 0.0)
        self.P = self.F @ self.P @ self.F.T + self.Q
        return self.x.copy()

    def update(self, z, mask=None, R_override=None, dpa=None):
        """
        Fuse a CV-ML measurement. Stores pre-update innovation in innovatio

        z          : [diameter_mm, dry_weight_g]
        mask       : boolean array — False where channel is unavailable
        R_override : per-frame noise matrix
        dpa        : day post-anthesis label for innovation logging
        """
        R = R_override if R_override is not None else self.R_default

        if mask is None:
            mask = np.ones(N_OBS, dtype=bool)
        idx = np.where(mask)[0]
        if not len(idx):
            return self.x.copy()

        H_s   = self.H[idx]
        R_s   = R[np.ix_(idx, idx)]
        S_inn = H_s @ self.P @ H_s.T + R_s
        K     = self.P @ H_s.T @ np.linalg.inv(S_inn)
        innov = np.asarray(z)[idx] - H_s @ self.x

        if dpa is not None:
            full_innov = np.full(N_OBS, np.nan)
            full_innov[idx] = innov
            self.innovation_log[dpa] = full_innov

        self.x = np.maximum(self.x + K @ innov, 0.0)
        self.P = (self._I - K @ H_s) @ self.P
        return self.x.copy()

    @property
    def pools(self):
        labels = ['S_phloem_suc','S_phloem_gln','S_phloem_asp',
                  'S_phloem_glu','S_phloem_ala','S_xylem_no3',
                  'S_xylem_pi',  'S_xylem_k',   'S_xylem_nh4',
                  'cumulative_biomass']
        return dict(zip(labels, self.x))
```

```
    @property
    def sigma(self):
        return np.sqrt(np.diag(self.P))


print("PoolKalmanFilter defined.")
```

```
PoolKalmanFilter defined.
PoolKalmanFilter defined.
```

In [34]:

```
# ============================================================================
# dFBA SIMULATION  (DPA 8–53, hourly steps)
# Restarted from KF posterior after each CV-ML observation.
# Segments: DPA 8–15 → [update] → 16–25 → [update] → 26–36 → ...
# ============================================================================

# ── KF initialisation ──────────────────────────────────────────────────────
x0_kf = [
    POOL_INIT["phloem_suc"], POOL_INIT["phloem_gln"],
    POOL_INIT["phloem_asp"], POOL_INIT["phloem_glu"], POOL_INIT["phloem_ala"
    POOL_INIT["xylem_no3"],  POOL_INIT["xylem_pi"],
    POOL_INIT["xylem_k"],    POOL_INIT["xylem_nh4"],
    0.01,
]
kf = PoolKalmanFilter(x0_kf, R_default_diag=[25.0, 0.25])

# ── CV-ML daily measurement schedule ──────────────────────────────────────
cv_ml_measurements = {
    15: {"z": [20.0, 0.90], "R": [16.0, 0.02]},
    25: {"z": [50.0, 4.00], "R": [25.0, 0.08]},
    36: {"z": [72.0, 7.00], "R": [12.0, 0.05]},
    45: {"z": [82.0, 8.00], "R": [18.0, 0.06]},
    53: {"z": [88.0, 9.00], "R": [10.0, 0.04]},
}

SIGMA_SCALE = 1.0

def _apply_pool_caps(kf_obj):
    caps = [MAX_PHLOEM_SUC, MAX_PHLOEM_GLN, MAX_PHLOEM_ASP,
            MAX_PHLOEM_GLU, MAX_PHLOEM_ALA,
            MAX_XYLEM_NO3,  MAX_XYLEM_PI, MAX_XYLEM_K, MAX_XYLEM_NH4,
            np.inf]
    for j, cap in enumerate(caps):
        kf_obj.x[j] = min(kf_obj.x[j], cap)

def _sigma_slack(kf_obj, idx, cap):
    return min(kf_obj.sigma[idx] * SIGMA_SCALE, cap - kf_obj.x[idx])

fruit_biomass_history = []
hourly_records        = []
kf_posterior_biomass  = {}   # {dpa: cum_biomass after KF update}

# ── Segment boundaries: split at each observation day ─────────────────────
obs_dpas       = sorted(cv_ml_measurements.keys())
seg_end_dpas   = obs_dpas + ([53] if 53 not in obs_dpas else [])
seg_start_dpas = [8] + [d + 1 for d in obs_dpas if d < 53]

for seg_idx, (seg_start, seg_end) in enumerate(zip(seg_start_dpas, seg_end_d

    print(f"\n=== Segment {seg_idx + 1}: DPA {seg_start}–{seg_end} "
          f"| suc={kf.x[IDX_PHLOEM_SUC]:.3f}  no3={kf.x[IDX_XYLEM_NO3]:.3f}
          f"cum_bio={kf.x[IDX_CUM_BIOMASS]:.3f} ===")
```

```
    for current_DPA in range(seg_start, seg_end + 1):
        i = current_DPA - 8
        print(f"DPA {current_DPA}", end=" | ", flush=True)

        ratio      = sink_strength(current_DPA)
        c_priority = c_priority_from_phloem_cn()
        fruit_obj  = get_fruit_objective(fruit_model, current_DPA)
        daily_biomass = 0.0

        for h in range(24):
            # ── Read pool state from KF posterior ──────────────────────
            S_phloem_suc = kf.x[IDX_PHLOEM_SUC]
            S_phloem_gln = kf.x[IDX_PHLOEM_GLN]
            S_phloem_asp = kf.x[IDX_PHLOEM_ASP]
            S_phloem_glu = kf.x[IDX_PHLOEM_GLU]
            S_phloem_ala = kf.x[IDX_PHLOEM_ALA]
            S_xylem_no3  = kf.x[IDX_XYLEM_NO3]
            S_xylem_pi   = kf.x[IDX_XYLEM_PI]
            S_xylem_k    = kf.x[IDX_XYLEM_K]
            S_xylem_nh4  = kf.x[IDX_XYLEM_NH4]

            suc_ceil = S_phloem_suc + _sigma_slack(kf, IDX_PHLOEM_SUC, MAX_P
            gln_ceil = S_phloem_gln + _sigma_slack(kf, IDX_PHLOEM_GLN, MAX_P
            asp_ceil = S_phloem_asp + _sigma_slack(kf, IDX_PHLOEM_ASP, MAX_P
            glu_ceil = S_phloem_glu + _sigma_slack(kf, IDX_PHLOEM_GLU, MAX_P
            ala_ceil = S_phloem_ala + _sigma_slack(kf, IDX_PHLOEM_ALA, MAX_P
            no3_ceil = S_xylem_no3  + _sigma_slack(kf, IDX_XYLEM_NO3,  MAX_X
            nh4_ceil = S_xylem_nh4  + _sigma_slack(kf, IDX_XYLEM_NH4,  MAX_X
            k_ceil   = S_xylem_k    + _sigma_slack(kf, IDX_XYLEM_K,    MAX_X

            ppfd_h   = diurnal_ppfd(ppfd_day[i], h, photoperiod[i])
            is_light = ppfd_h > 0.0

            max_no3_h = no3_ceiling(tank_no3[i])
            max_pi_h  = pi_ceiling(tank_pi[i])
            max_k_h   = k_ceiling(tank_k[i])
            max_nh4_h = nh4_ceiling(tank_nh4[i])

            fb_suc = phloem_loading_fb(S_phloem_suc)
            fb_no3 = root_no3_fb(S_xylem_no3)
            fb_pi  = root_pi_fb(S_xylem_pi)

            eff_no3_h = max_no3_h * fb_no3
            eff_pi_h  = max_pi_h  * fb_pi

            S_xylem_no3 = min(S_xylem_no3 + eff_no3_h  / PLANT_DW_PER_FRUIT,
            S_xylem_pi  = min(S_xylem_pi  + eff_pi_h   / PLANT_DW_PER_FRUIT,
            S_xylem_k   = min(S_xylem_k   + max_k_h    / PLANT_DW_PER_FRUIT,
            S_xylem_nh4 = min(S_xylem_nh4 + max_nh4_h  / PLANT_DW_PER_FRUIT,

            root_atp      = (eff_no3_h * ATP_PER_NO3 + max_nh4_h * ATP_PER_N
                             + eff_pi_h  * ATP_PER_PI  + max_k_h   * ATP_PER_K
            root_suc_cost = root_atp / ATP_PER_SUC * xylem_no3_frac[i]
            S_phloem_suc  = max(0.0, S_phloem_suc - root_suc_cost)

            leaf_suc_h = leaf_gln_h = leaf_asp_h = leaf_glu_h = leaf_ala_h =
            leaf_no3_used = leaf_pi_used = leaf_nh4_used = 0.0

            if is_light:
                max_photon_h = VYTOP_PHOTON_PER_LH_AT200 * (ppfd_h / 200.0)
                sucr_ub      = VYTOP_SUCR_PER_LH_AT200   * (ppfd_h / 200.0)
                gln_ub       = VYTOP_GLN_PER_LH_AT200    * (ppfd_h / 200.0)

                veg_model.reactions.get_by_id('EX_photon_b').lower_bound = -
```

```
            veg_model.reactions.get_by_id('EX_co2_b').lower_bound    = -
            veg_model.reactions.get_by_id('EX_h2o_b').lower_bound    = -
            veg_model.reactions.get_by_id('EX_no3_b').lower_bound = (
                -min(eff_no3_h, S_xylem_no3 * PLANT_DW_PER_FRUIT) / PLAN
            veg_model.reactions.get_by_id('EX_nh4_b').lower_bound = (
                -min(max_nh4_h, S_xylem_nh4 * PLANT_DW_PER_FRUIT) / PLAN
            veg_model.reactions.get_by_id('EX_pi_b').lower_bound = (
                -min(eff_pi_h,  S_xylem_pi  * PLANT_DW_PER_FRUIT) / PLAN
            veg_model.reactions.get_by_id('EX_so4_b').lower_bound = -100
            veg_model.reactions.get_by_id('EX_k_b').lower_bound   = -max

            veg_model.reactions.get_by_id('EX_sucr_b').upper_bound  = su
            veg_model.reactions.get_by_id('EX_gln_L_b').upper_bound = g

            veg_model.objective = 'EX_sucr_b'
            suc_sol = veg_model.optimize()
            if suc_sol.status == 'optimal':
                leaf_suc_h = max(0.0, suc_sol.objective_value) * PLANT_D

            veg_model.reactions.get_by_id('EX_sucr_b').lower_bound = (
                suc_sol.objective_value * c_priority if suc_sol.status =
            veg_model.objective = 'EX_gln_L_b'
            gln_sol = veg_model.optimize()
            if gln_sol.status == 'optimal':
                leaf_gln_h    = max(0.0, gln_sol.objective_value)
                leaf_no3_used = abs(gln_sol.fluxes.get('EX_no3_b', 0.0))
                leaf_pi_used  = abs(gln_sol.fluxes.get('EX_pi_b',  0.0))
                leaf_nh4_used = abs(gln_sol.fluxes.get('EX_nh4_b', 0.0))
                leaf_asp_h    = max(0.0, gln_sol.fluxes.get('EX_asp_L_b
                leaf_glu_h    = max(0.0, gln_sol.fluxes.get('EX_glu_L_b
                leaf_ala_h    = max(0.0, gln_sol.fluxes.get('EX_ala_L_b

            veg_model.reactions.get_by_id('EX_sucr_b').lower_bound = 0.0
            veg_model.reactions.get_by_id('EX_sucr_b').upper_bound = 100

        S_phloem_suc = min(S_phloem_suc + leaf_suc_h, MAX_PHLOEM_SUC)
        S_phloem_gln = min(S_phloem_gln + leaf_gln_h, MAX_PHLOEM_GLN)
        S_phloem_asp = min(S_phloem_asp + leaf_asp_h, MAX_PHLOEM_ASP)
        S_phloem_glu = min(S_phloem_glu + leaf_glu_h, MAX_PHLOEM_GLU)
        S_phloem_ala = min(S_phloem_ala + leaf_ala_h, MAX_PHLOEM_ALA)
        S_xylem_no3  = max(0.0, S_xylem_no3 - leaf_no3_used)
        S_xylem_pi   = max(0.0, S_xylem_pi  - leaf_pi_used)
        S_xylem_nh4  = max(0.0, S_xylem_nh4 - leaf_nh4_used)

        suc_for_fruit = min(suc_ceil * ratio / 24, suc_ceil)
        gln_for_fruit = min(gln_ceil * ratio / 24, gln_ceil)
        asp_for_fruit = min(asp_ceil * ratio / 24, asp_ceil)
        glu_for_fruit = min(glu_ceil * ratio / 24, glu_ceil)
        ala_for_fruit = min(ala_ceil * ratio / 24, ala_ceil)
        no3_for_fruit = min(eff_no3_h / PLANT_DW_PER_FRUIT * xylem_no3_
        nh4_for_fruit = min(max_nh4_h  / PLANT_DW_PER_FRUIT * nh4_fruit_
        k_for_fruit   = min(max_k_h    / PLANT_DW_PER_FRUIT * k_fruit_fr

        fruit_model.reactions.get_by_id("EX_Suc_in").lower_bound = -suc_
        fruit_model.reactions.get_by_id("EX_Gln_in").lower_bound = -gln_
        fruit_model.reactions.get_by_id("EX_Asp_in").lower_bound = -asp_
        fruit_model.reactions.get_by_id("EX_Glu_in").lower_bound = -glu_
        fruit_model.reactions.get_by_id("EX_Ala_in").lower_bound = -ala_
        fruit_model.reactions.get_by_id("EX_NO3").lower_bound    = -no3_
        fruit_model.reactions.get_by_id("EX_NH4").lower_bound    = -nh4_
        fruit_model.reactions.get_by_id("EX_K").lower_bound      = -k_fr
        fruit_model.objective = fruit_obj

        hour_biomass = 0.0
```

```
            try:
                fruit_sol = fruit_model.optimize()
                if fruit_sol.status != 'optimal':
                    raise cobra.exceptions.Infeasible

                fruit_suc_used = abs(fruit_sol.fluxes.get('EX_Suc_in', 0.0))
                fruit_gln_used = abs(fruit_sol.fluxes.get('EX_Gln_in', 0.0))
                fruit_asp_used = abs(fruit_sol.fluxes.get('EX_Asp_in', 0.0))
                fruit_glu_used = abs(fruit_sol.fluxes.get('EX_Glu_in', 0.0))
                fruit_ala_used = abs(fruit_sol.fluxes.get('EX_Ala_in', 0.0))
                fruit_no3_used = abs(fruit_sol.fluxes.get('EX_NO3',    0.0))
                fruit_nh4_used = abs(fruit_sol.fluxes.get('EX_NH4',    0.0))
                fruit_k_used   = abs(fruit_sol.fluxes.get('EX_K',      0.0))

                _EPS = 1e-9
                suc_util = fruit_suc_used / suc_for_fruit if suc_for_fruit >
                gln_util = fruit_gln_used / gln_for_fruit if gln_for_fruit >
                no3_util = fruit_no3_used / no3_for_fruit if no3_for_fruit >
                nh4_util = fruit_nh4_used / nh4_for_fruit if nh4_for_fruit >
                k_util   = fruit_k_used   / k_for_fruit   if k_for_fruit   >

                S_phloem_suc = max(0.0, S_phloem_suc - fruit_suc_used)
                S_phloem_gln = max(0.0, S_phloem_gln - fruit_gln_used)
                S_phloem_asp = max(0.0, S_phloem_asp - fruit_asp_used)
                S_phloem_glu = max(0.0, S_phloem_glu - fruit_glu_used)
                S_phloem_ala = max(0.0, S_phloem_ala - fruit_ala_used)
                S_xylem_no3  = max(0.0, S_xylem_no3  - fruit_no3_used)
                S_xylem_nh4  = max(0.0, S_xylem_nh4  - fruit_nh4_used)
                S_xylem_k    = max(0.0, S_xylem_k    - fruit_k_used)

                hour_biomass = (
                    sum(fruit_sol.fluxes.get(r, 0.0)
                        for r in ['EX_STARCH','EX_PROTEIN','EX_CW','EX_DAG',
                    + sum(fruit_sol.fluxes.get(r, 0.0)
                        for r in ['EX_Glc_ac','EX_Fru_ac','EX_Suc_ac','EX_Ma
                                  'EX_Glu_ac','EX_Asp_ac','EX_Ala_ac','EX_S
                )
                daily_biomass += hour_biomass

                hourly_records.append({
                    "dpa": current_DPA, "hour": h,
                    "ppfd": ppfd_h, "is_light": is_light,
                    "leaf_suc": leaf_suc_h, "leaf_gln": leaf_gln_h,
                    "leaf_asp": leaf_asp_h, "leaf_glu": leaf_glu_h, "leaf_a
                    "S_phloem_suc": S_phloem_suc, "S_phloem_gln": S_phloem_g
                    "S_xylem_no3": S_xylem_no3, "S_xylem_pi": S_xylem_pi,
                    "S_xylem_k": S_xylem_k, "S_xylem_nh4": S_xylem_nh4,
                    "fb_suc": fb_suc, "fb_no3": fb_no3, "fb_pi": fb_pi,
                    "fruit_suc_used": fruit_suc_used, "fruit_gln_used": frui
                    "fruit_nh4_used": fruit_nh4_used, "fruit_k_used": fruit_
                    "fruit_lyc_h":   fruit_sol.fluxes.get('EX_Lyc_ac', 0.0),
                    "fruit_k_vac_h": fruit_sol.fluxes.get('EX_K_vac',  0.0),
                    "fruit_biomass_h": hour_biomass,
                    "kf_sigma_suc":   kf.sigma[IDX_PHLOEM_SUC],
                    "kf_sigma_no3":   kf.sigma[IDX_XYLEM_NO3],
                    "kf_sigma_biomass": kf.sigma[IDX_CUM_BIOMASS],
                    "kf_cum_biomass": kf.x[IDX_CUM_BIOMASS],
                    "suc_slack": suc_ceil - S_phloem_suc,
                    "no3_slack": no3_ceil - S_xylem_no3,
                    "suc_util": suc_util, "gln_util": gln_util,
                    "no3_util": no3_util, "nh4_util": nh4_util, "k_util": k_
                })

            except cobra.exceptions.Infeasible:
```

```
                hourly_records.append({
                    "dpa": current_DPA, "hour": h,
                    "ppfd": ppfd_h, "is_light": is_light,
                    "leaf_suc": 0.0, "leaf_gln": 0.0,
                    "leaf_asp": 0.0, "leaf_glu": 0.0, "leaf_ala": 0.0,
                    "S_phloem_suc": S_phloem_suc, "S_phloem_gln": S_phloem_g
                    "S_xylem_no3": S_xylem_no3, "S_xylem_pi": S_xylem_pi,
                    "S_xylem_k": S_xylem_k, "S_xylem_nh4": S_xylem_nh4,
                    "fb_suc": fb_suc, "fb_no3": fb_no3, "fb_pi": fb_pi,
                    "fruit_suc_used": 0.0, "fruit_gln_used": 0.0,
                    "fruit_nh4_used": 0.0, "fruit_k_used": 0.0,
                    "fruit_lyc_h": 0.0, "fruit_k_vac_h": 0.0,
                    "fruit_biomass_h": 0.0,
                    "kf_sigma_suc":   kf.sigma[IDX_PHLOEM_SUC],
                    "kf_sigma_no3":   kf.sigma[IDX_XYLEM_NO3],
                    "kf_sigma_biomass": kf.sigma[IDX_CUM_BIOMASS],
                    "kf_cum_biomass": kf.x[IDX_CUM_BIOMASS],
                    "suc_slack": 0.0, "no3_slack": 0.0,
                    "suc_util": 0.0, "gln_util": 0.0,
                    "no3_util": 0.0, "nh4_util": 0.0, "k_util": 0.0,
                })

            # ── KF predict ──────────────────────────────────────────
            delta_x = np.array([
                S_phloem_suc - kf.x[IDX_PHLOEM_SUC],
                S_phloem_gln - kf.x[IDX_PHLOEM_GLN],
                S_phloem_asp - kf.x[IDX_PHLOEM_ASP],
                S_phloem_glu - kf.x[IDX_PHLOEM_GLU],
                S_phloem_ala - kf.x[IDX_PHLOEM_ALA],
                S_xylem_no3  - kf.x[IDX_XYLEM_NO3],
                S_xylem_pi   - kf.x[IDX_XYLEM_PI],
                S_xylem_k    - kf.x[IDX_XYLEM_K],
                S_xylem_nh4  - kf.x[IDX_XYLEM_NH4],
                hour_biomass,
            ])
            kf.predict(delta_x)
            _apply_pool_caps(kf)

        fruit_biomass_history.append(daily_biomass)

    # ── KF update at segment end → next segment restarts from posterior ──
    if seg_end in cv_ml_measurements:
        entry   = cv_ml_measurements[seg_end]
        z_raw   = np.array(entry["z"], dtype=float)
        R_raw   = np.array(entry["R"], dtype=float)
        mask    = ~np.isnan(z_raw)
        R_raw[np.isnan(R_raw)] = 1e6
        kf.update(z_raw, mask, R_override=np.diag(R_raw), dpa=seg_end)
        _apply_pool_caps(kf)
        kf_posterior_biomass[seg_end] = kf.x[IDX_CUM_BIOMASS]
        print(f"\n  [KF update]  DPA {seg_end}: "
              f"σ(biomass)={kf.sigma[IDX_CUM_BIOMASS]:.4f}  "
              f"cum_biomass={kf.x[IDX_CUM_BIOMASS]:.3f} mmol")
        print(f"  [Restart]    Next segment initialised from KF posterior\n")

print(f"Simulation complete.  "
      f"Total biomass (LP): {sum(fruit_biomass_history):.2f} mmol  |  "
      f"KF cum_biomass: {kf.x[IDX_CUM_BIOMASS]:.2f} mmol")
```

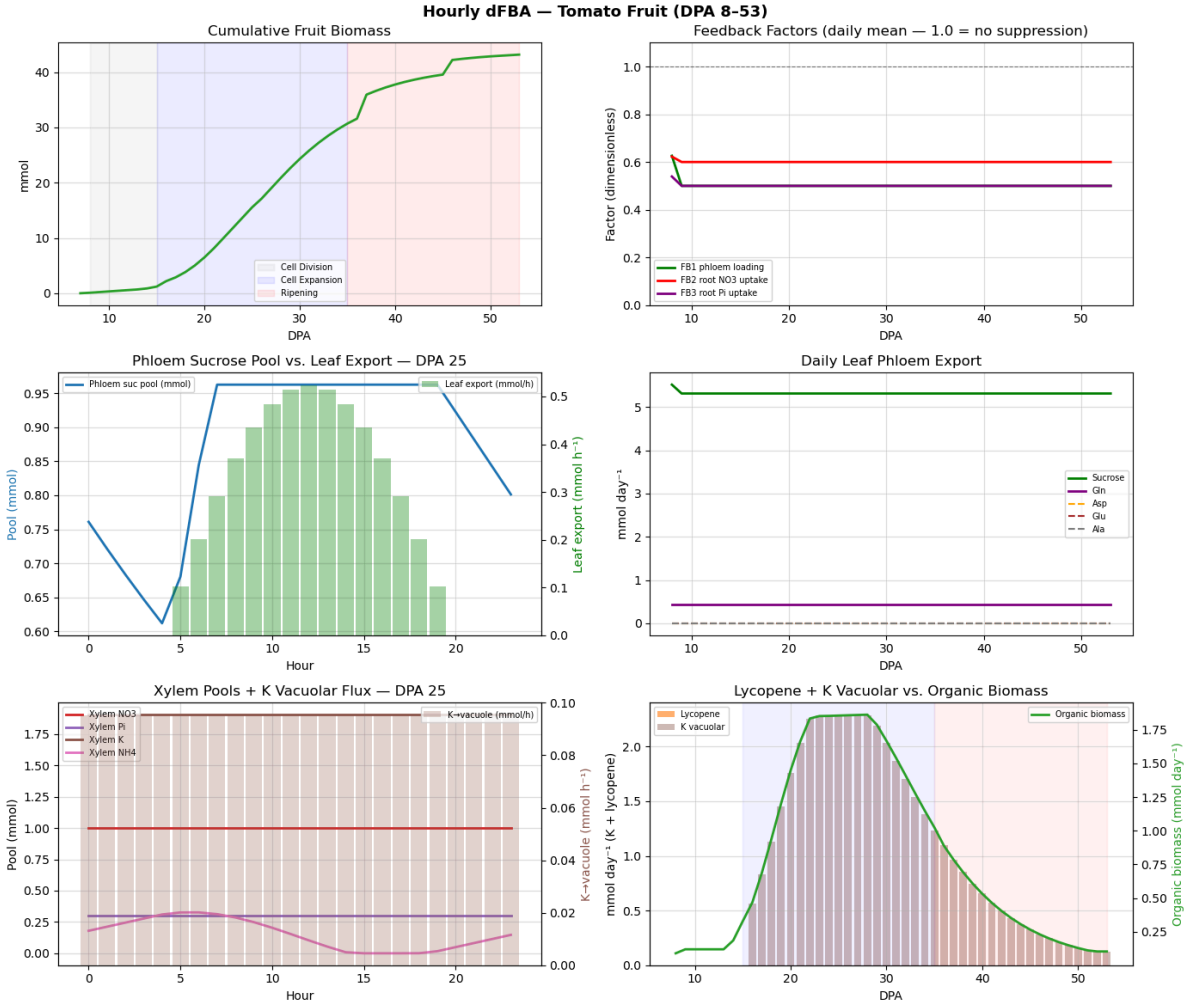
```
=== Segment 1: DPA 8–15 | suc=0.050  no3=0.050  cum_bio=0.010 ===
DPA 8 |
=== Segment 1: DPA 8–15 | suc=0.050  no3=0.050  cum_bio=0.010 ===
DPA 8 | DPA 9 | DPA 9 | DPA 10 | DPA 10 | DPA 11 | DPA 11 | DPA 12 | DPA 12
| DPA 13 | DPA 13 | DPA 14 | DPA 14 | DPA 15 | DPA 15 |
  [KF update]  DPA 15: σ(biomass)=0.3830  cum_biomass=2.198 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 2: DPA 16–25 | suc=0.960  no3=0.997  cum_bio=2.198 ===
DPA 16 |
  [KF update]  DPA 15: σ(biomass)=0.3830  cum_biomass=2.198 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 2: DPA 16–25 | suc=0.960  no3=0.997  cum_bio=2.198 ===
DPA 16 | DPA 17 | DPA 17 | DPA 18 | DPA 18 | DPA 19 | DPA 19 | DPA 20 | DPA
20 | DPA 21 | DPA 21 | DPA 22 | DPA 22 | DPA 23 | DPA 23 | DPA 24 | DPA 24
| DPA 25 | DPA 25 |
  [KF update]  DPA 25: σ(biomass)=0.5773  cum_biomass=17.124 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 3: DPA 26–36 | suc=0.801  no3=1.000  cum_bio=17.124 ===
DPA 26 |
  [KF update]  DPA 25: σ(biomass)=0.5773  cum_biomass=17.124 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 3: DPA 26–36 | suc=0.801  no3=1.000  cum_bio=17.124 ===
DPA 26 | DPA 27 | DPA 27 | DPA 28 | DPA 28 | DPA 29 | DPA 29 | DPA 30 | DPA
30 | DPA 31 | DPA 31 | DPA 32 | DPA 32 | DPA 33 | DPA 33 | DPA 34 | DPA 34
| DPA 35 | DPA 35 | DPA 36 | DPA 36 |
  [KF update]  DPA 36: σ(biomass)=0.6533  cum_biomass=35.935 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 4: DPA 37–45 | suc=0.899  no3=1.000  cum_bio=35.935 ===
DPA 37 |
  [KF update]  DPA 36: σ(biomass)=0.6533  cum_biomass=35.935 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 4: DPA 37–45 | suc=0.899  no3=1.000  cum_bio=35.935 ===
DPA 37 | DPA 38 | DPA 38 | DPA 39 | DPA 39 | DPA 40 | DPA 40 | DPA 41 | DPA
41 | DPA 42 | DPA 42 | DPA 43 | DPA 43 | DPA 44 | DPA 44 | DPA 45 | DPA 45
|
  [KF update]  DPA 45: σ(biomass)=0.6885  cum_biomass=42.207 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 5: DPA 46–53 | suc=0.962  no3=1.000  cum_bio=42.207 ===
DPA 46 |
  [KF update]  DPA 45: σ(biomass)=0.6885  cum_biomass=42.207 mmol
  [Restart]    Next segment initialised from KF posterior


=== Segment 5: DPA 46–53 | suc=0.962  no3=1.000  cum_bio=42.207 ===
DPA 46 | DPA 47 | DPA 47 | DPA 48 | DPA 48 | DPA 49 | DPA 49 | DPA 50 | DPA
50 | DPA 51 | DPA 51 | DPA 52 | DPA 52 | DPA 53 | DPA 53 |
  [KF update]  DPA 53: σ(biomass)=0.6542  cum_biomass=47.109 mmol
  [Restart]    Next segment initialised from KF posterior

Simulation complete.  Total biomass (LP): 36.93 mmol  |  KF cum_biomass: 4
```

```
7.11 mmol

  [KF update]  DPA 53: σ(biomass)=0.6542  cum_biomass=47.109 mmol
  [Restart]    Next segment initialised from KF posterior

Simulation complete.  Total biomass (LP): 36.93 mmol  |  KF cum_biomass: 4
7.11 mmol
```

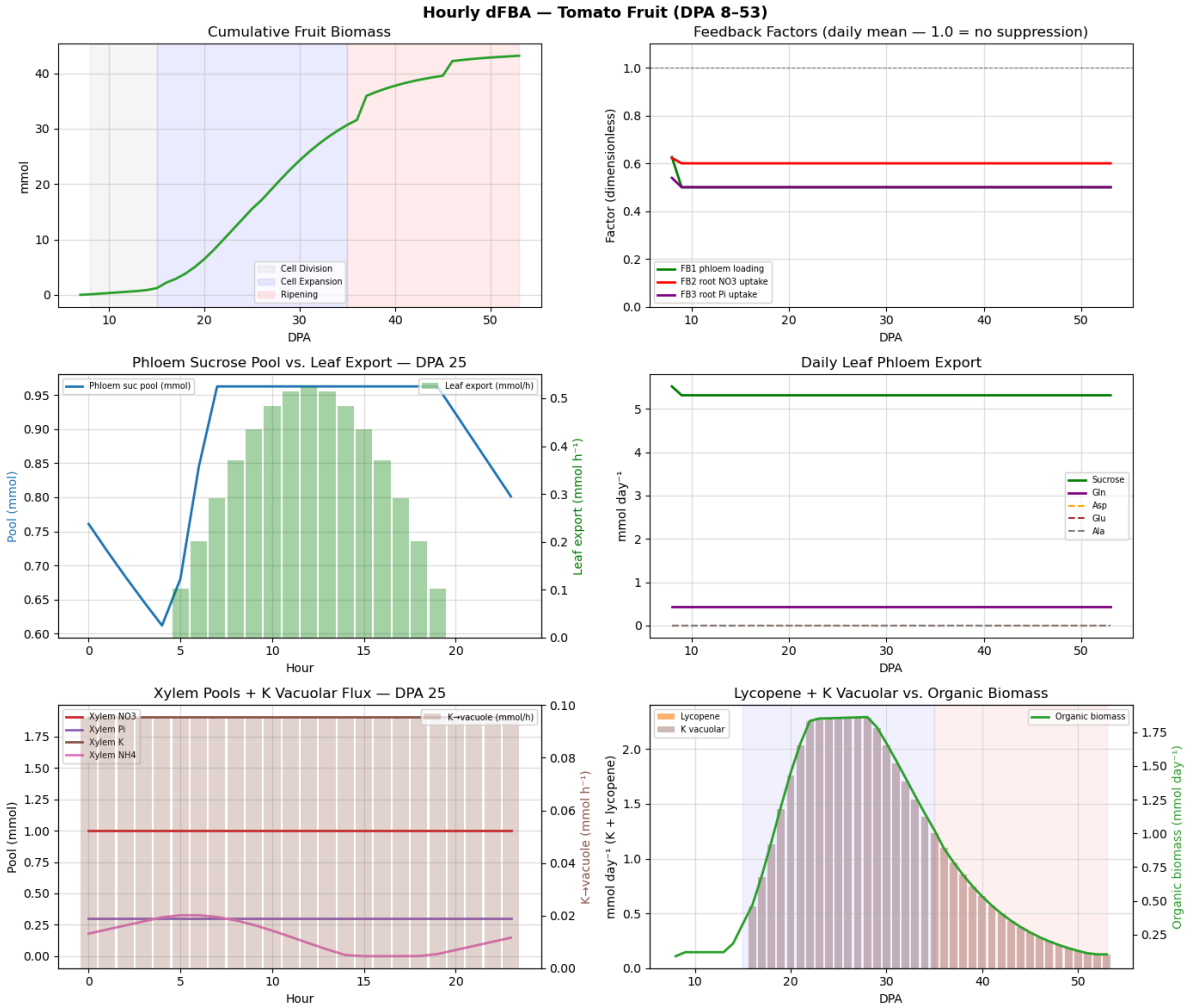
In [35]:

```
hr_df = pd.DataFrame(hourly_records)

PHASE_SHADING = [
    (8,  15, 'gray', 'Cell Division'),
    (15, 35, 'blue', 'Cell Expansion'),
    (35, 53, 'red',  'Ripening'),
]

fig, axes = plt.subplots(3, 2, figsize=(14, 12))
fig.suptitle("Hourly dFBA — Tomato Fruit (DPA 8–53)", fontsize=13, fontweigh

# ── 1. Cumulative fruit biomass ──────────────────────────────────────
cumulative_biomass = [0.01]
running = 0.01
for idx, dg in enumerate(fruit_biomass_history):
    dpa = 8 + idx
    prev_dpa = dpa - 1
    if prev_dpa in kf_posterior_biomass:
        # Snap to green's DPA prev level — no increment added yet
        running = kf_posterior_biomass[prev_dpa]
        cumulative_biomass.append(running)
    else:
        running = running + dg
        cumulative_biomass.append(running)
ax = axes[0, 0]
ax.plot(list(range(7, 54)), cumulative_biomass, color='#2ca02c', linewidth=2
for start, end, col, label in PHASE_SHADING:
    ax.axvspan(start, end, color=col, alpha=0.08, label=label)
ax.set_title("Cumulative Fruit Biomass")
ax.set_xlabel("DPA"); ax.set_ylabel("mmol")
ax.legend(fontsize=7); ax.grid(True, alpha=0.4)

# ── 2. Feedback factor trajectories (daily mean) ─────────────────────
daily_fb = hr_df.groupby('dpa')[['fb_suc','fb_no3','fb_pi']].mean().reset_in
ax = axes[0, 1]
ax.plot(daily_fb.dpa, daily_fb.fb_suc, color='green',  linewidth=2, label='F
ax.plot(daily_fb.dpa, daily_fb.fb_no3, color='red',    linewidth=2, label='F
ax.plot(daily_fb.dpa, daily_fb.fb_pi,  color='purple', linewidth=2, label='F
ax.axhline(1.0, color='k', linestyle='--', linewidth=0.8, alpha=0.5)
ax.set_title("Feedback Factors (daily mean — 1.0 = no suppression)")
ax.set_xlabel("DPA"); ax.set_ylabel("Factor (dimensionless)")
ax.set_ylim(0, 1.1); ax.legend(fontsize=7); ax.grid(True, alpha=0.4)

# ── 3. Phloem sucrose pool + leaf export — DPA 25 ────────────────────
day25 = hr_df[hr_df.dpa == 25]
ax = axes[1, 0]; ax2 = ax.twinx()
ax.plot(day25.hour, day25.S_phloem_suc, color='#1f77b4', linewidth=2, label=
ax2.bar(day25.hour, day25.leaf_suc, color='green', alpha=0.35, width=0.9, la
ax.set_title("Phloem Sucrose Pool vs. Leaf Export — DPA 25")
ax.set_xlabel("Hour"); ax.set_ylabel("Pool (mmol)", color='#1f77b4')
ax2.set_ylabel("Leaf export (mmol h⁻¹)", color='green')
ax.legend(loc='upper left', fontsize=7); ax2.legend(loc='upper right', fonts
ax.grid(True, alpha=0.4)

# ── 4. Daily leaf phloem export — all metabolites ────────────────────
ax = axes[1, 1]
```

```
for col, color, ls, label in [
    ('leaf_suc', 'green',  '-',  'Sucrose'),
    ('leaf_gln', 'purple', '-',  'Gln'),
    ('leaf_asp', 'orange', '--', 'Asp'),
    ('leaf_glu', 'brown',  '--', 'Glu'),
    ('leaf_ala', 'gray',   '--', 'Ala'),
]:
    d = hr_df.groupby('dpa')[col].sum().reset_index()
    ax.plot(d.dpa, d[col], color=color, linewidth=1.5 if ls == '--' else 2,
ax.set_title("Daily Leaf Phloem Export")
ax.set_xlabel("DPA"); ax.set_ylabel("mmol day⁻¹")
ax.legend(fontsize=7); ax.grid(True, alpha=0.4)

# ── 5. Xylem pools + K vacuolar flux — DPA 25 ──────────────────────────
ax = axes[2, 0]; ax2 = ax.twinx()
for col, color, label in [
    ('S_xylem_no3', '#d62728', 'Xylem NO3'),
    ('S_xylem_pi',  '#9467bd', 'Xylem Pi'),
    ('S_xylem_k',   '#8c564b', 'Xylem K'),
    ('S_xylem_nh4', '#e377c2', 'Xylem NH4'),
]:
    ax.plot(day25.hour, day25[col], color=color, linewidth=2, label=label)
ax2.bar(day25.hour, day25.fruit_k_vac_h, color='#8c564b', alpha=0.25, width=
ax.set_title("Xylem Pools + K Vacuolar Flux — DPA 25")
ax.set_xlabel("Hour"); ax.set_ylabel("Pool (mmol)")
ax2.set_ylabel("K→vacuole (mmol h⁻¹)", color='#8c564b')
ax.legend(loc='upper left', fontsize=7); ax2.legend(loc='upper right', fonts
ax.grid(True, alpha=0.4)

# ── 6. Lycopene + K vacuolar vs. organic biomass by phase ──────────────
daily_lyc  = hr_df.groupby('dpa')['fruit_lyc_h'].sum().reset_index()
daily_kvac = hr_df.groupby('dpa')['fruit_k_vac_h'].sum().reset_index()
daily_bm   = hr_df.groupby('dpa')['fruit_biomass_h'].sum().reset_index()
ax = axes[2, 1]; ax2 = ax.twinx()
ax.bar(daily_lyc.dpa,  daily_lyc.fruit_lyc_h,   color='#ff7f0e', alpha=0.6,
ax.bar(daily_kvac.dpa, daily_kvac.fruit_k_vac_h, color='#8c564b', alpha=0.4,
       bottom=daily_lyc.fruit_lyc_h, label='K vacuolar')
ax2.plot(daily_bm.dpa, daily_bm.fruit_biomass_h, color='#2ca02c', linewidth=
for start, end, col, _ in PHASE_SHADING[1:]:
    ax.axvspan(start, end, color=col, alpha=0.05)
ax.set_title("Lycopene + K Vacuolar vs. Organic Biomass")
ax.set_xlabel("DPA"); ax.set_ylabel("mmol day⁻¹ (K + lycopene)")
ax2.set_ylabel("Organic biomass (mmol day⁻¹)", color='#2ca02c')
ax.legend(loc='upper left', fontsize=7); ax2.legend(loc='upper right', fonts
ax.grid(True, alpha=0.4)

plt.tight_layout()
plt.show()
print(f"Final cumulative biomass: {cumulative_biomass[-1]:.2f} mmol")
```




```
Final cumulative biomass: 43.17 mmol
```

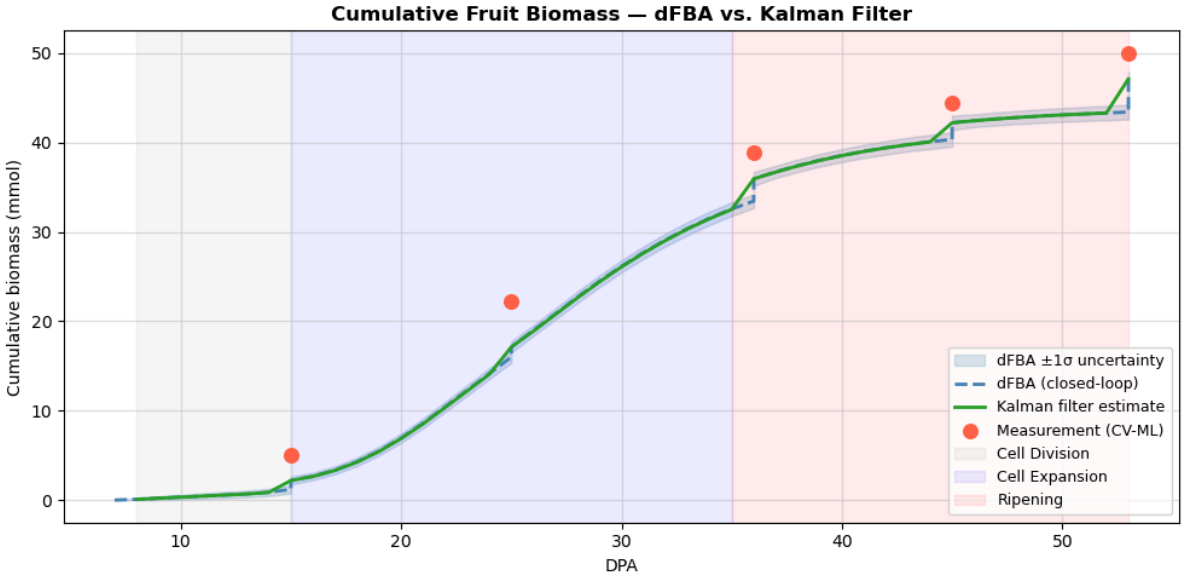


```
Final cumulative biomass: 43.17 mmol
```

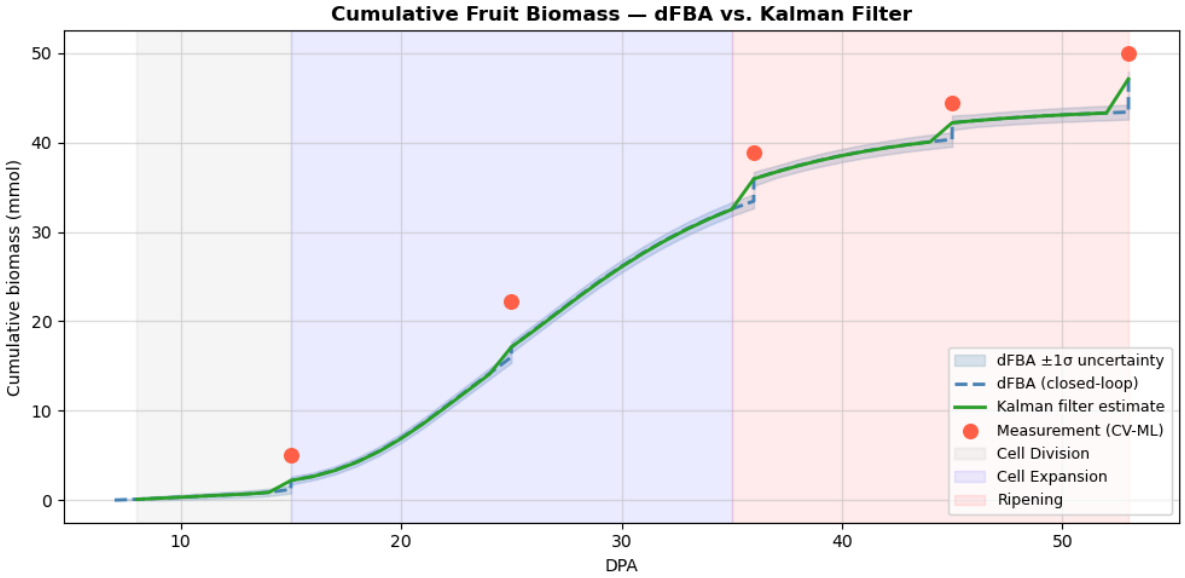
In [36]:

```
# ================================================================================
# RESULT: Cumulative Fruit Biomass — dFBA vs. Kalman Filter vs. Ground Truth
# ================================================================================
# At each observation DPA, the blue line shows TWO points at the same x:
#   1) dFBA prediction (pre-update) — what the model computed
#   2) KF posterior   (post-update) — corrected state, dFBA restarts from he
# This creates a visible vertical segment showing the correction at that DPA
# ================================================================================

fig, ax = plt.subplots(figsize=(10, 5))

# ── Build expanded x/y for blue line with vertical segments at obs days ──
sigma_daily_s = hr_df.groupby("dpa")["kf_sigma_biomass"].last()
x_plot    = [7]
y_plot    = [0.01]
sig_plot  = [0.0]
running   = 0.01

for idx, dg in enumerate(fruit_biomass_history):
    dpa      = 8 + idx
    sig      = float(sigma_daily_s.get(dpa, 0.0))
    running += dg

    if dpa in kf_posterior_biomass:
        # Point 1: pre-update (dFBA prediction for this DPA)
        x_plot.append(dpa);  y_plot.append(running);                        si
        # Point 2: post-update (KF posterior — dFBA restarts from here)
        x_plot.append(dpa);  y_plot.append(kf_posterior_biomass[dpa]);     si
        running = kf_posterior_biomass[dpa]
    else:
        x_plot.append(dpa);  y_plot.append(running);  sig_plot.append(sig)

upper = [y + s for y, s in zip(y_plot, sig_plot)]
lower = [max(0.0, y - s) for y, s in zip(y_plot, sig_plot)]
ax.fill_between(x_plot, lower, upper, color="steelblue", alpha=0.2, label="d
ax.plot(x_plot, y_plot, color="steelblue", linewidth=2, linestyle="--", labe

# ── Kalman filter estimate ──────────────────────────────────────────────
kf_daily = hr_df.groupby("dpa")["kf_cum_biomass"].last().reset_index()
for dpa, val in kf_posterior_biomass.items():
    kf_daily.loc[kf_daily.dpa == dpa, "kf_cum_biomass"] = val
ax.plot(kf_daily.dpa, kf_daily.kf_cum_biomass, color="#2ca02c", linewidth=2,
        label="Kalman filter estimate")

# ── Ground truth scatter ────────────────────────────────────────────────
gt_dpa, gt_biomass = [], []
for dpa, entry in cv_ml_measurements.items():
    z = np.array(entry["z"], dtype=float)
    if not np.isnan(z[1]):
        gt_dpa.append(dpa)
        gt_biomass.append(z[1] / H_BIOMASS_G_PER_MMOL)
if gt_dpa:
    ax.scatter(gt_dpa, gt_biomass, color="tomato", zorder=5, s=70,
               marker="o", label="Measurement (CV-ML)")

for start, end, col, label in PHASE_SHADING:
    ax.axvspan(start, end, color=col, alpha=0.08, label=label)

ax.set_title("Cumulative Fruit Biomass — dFBA vs. Kalman Filter",
             fontsize=12, fontweight="bold")
ax.set_xlabel("DPA")
ax.set_ylabel("Cumulative biomass (mmol)")
```

```
ax.legend(fontsize=9)
ax.grid(True, alpha=0.4)
plt.tight_layout()
plt.show()

print(f"Final dFBA (closed-loop): {y_plot[-1]:.2f} mmol")
print(f"Final KF estimate:        {kf_daily.kf_cum_biomass.iloc[-1]:.2f} mmo
```




```
Final dFBA (closed-loop): 47.11 mmol
Final KF estimate:        47.11 mmol
```



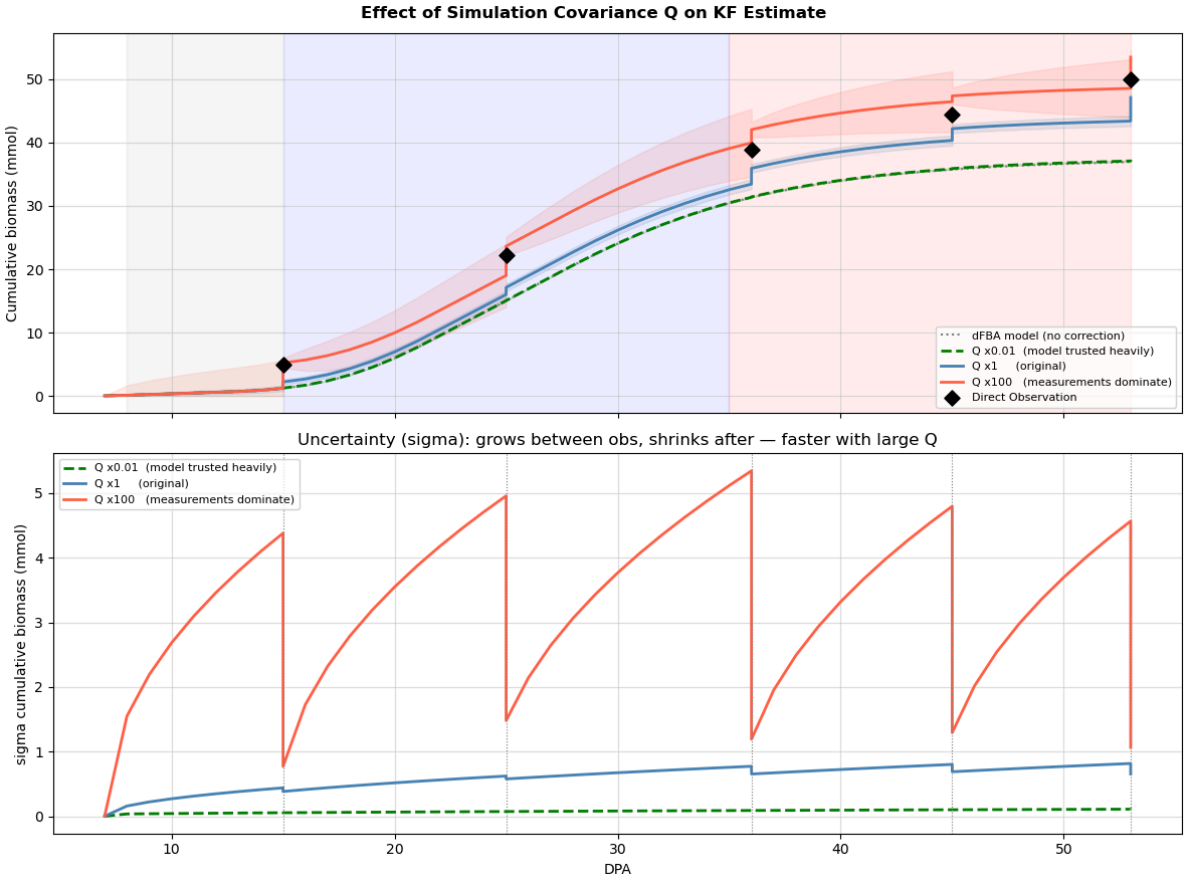

```
Final dFBA (closed-loop): 47.11 mmol
Final KF estimate:        47.11 mmol
```

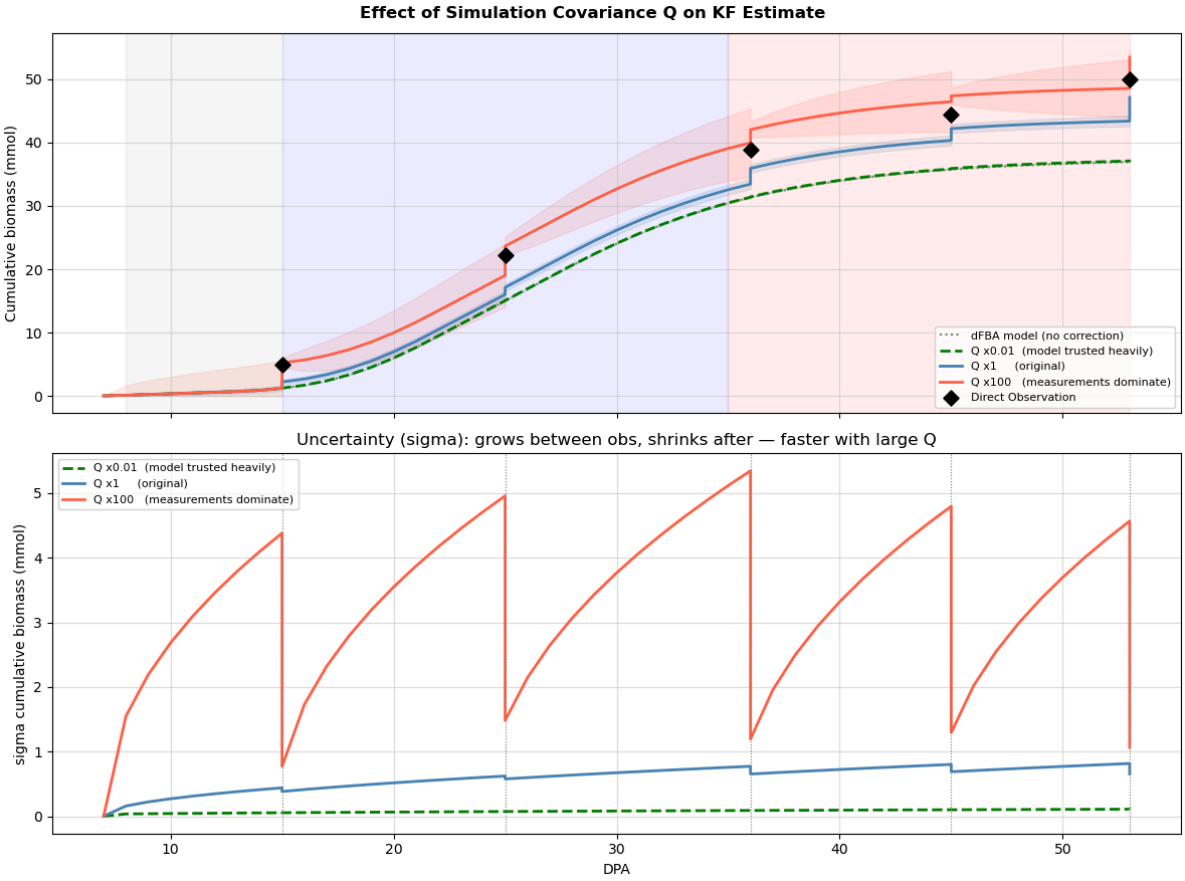
In [37]:

```
# ============================================================================
# Q vs R SENSITIVITY — effect of simulation covariance on KF estimate
# ============================================================================
# Replays the KF using stored hourly biomass increments (dFBA fixed).
# Three Q scalings:
#   Q * 0.01 : model trusted heavily → tiny corrections at obs
#   Q * 1    : current behaviour
#   Q * 100  : Q >> R → measurements dominate → large corrections
#
# R stays fixed. Current Q[9]=1e-3 << R[1]=0.25 → model trusted.
# Q_large[9]=0.1 >> R_biomass → measurements dominate.
# ============================================================================

Q_BASE = np.array([1e-4, 1e-5, 1e-6, 1e-6, 1e-6,
```

```
                    1e-3, 1e-4, 1e-3, 1e-4, 1e-3])

def replay_kf(q_scale):
    kf_r = PoolKalmanFilter(x0_kf, Q_diag=Q_BASE * q_scale, R_default_diag=
    x_r, y_r, sig_r = [7], [0.01], [0.0]
    for current_DPA in DPA_RANGE:
        day_rows = hr_df[hr_df.dpa == current_DPA]
        for _, row in day_rows.iterrows():
            delta_x = np.zeros(N_STATE)
            delta_x[IDX_CUM_BIOMASS] = row['fruit_biomass_h']
            kf_r.predict(delta_x)
        pre_val = kf_r.x[IDX_CUM_BIOMASS]
        pre_sig = kf_r.sigma[IDX_CUM_BIOMASS]
        if current_DPA in cv_ml_measurements:
            entry = cv_ml_measurements[current_DPA]
            z_raw = np.array(entry['z'], dtype=float)
            R_raw = np.array(entry['R'], dtype=float)
            mask  = ~np.isnan(z_raw)
            R_raw[np.isnan(R_raw)] = 1e6
            kf_r.update(z_raw, mask, R_override=np.diag(R_raw))
            post_val = kf_r.x[IDX_CUM_BIOMASS]
            post_sig = kf_r.sigma[IDX_CUM_BIOMASS]
            x_r  += [current_DPA, current_DPA]
            y_r  += [pre_val,     post_val]
            sig_r+= [pre_sig,     post_sig]
        else:
            x_r.append(current_DPA)
            y_r.append(pre_val)
            sig_r.append(pre_sig)
    return x_r, y_r, sig_r

configs = [
    (0.01,  'Q x0.01  (model trusted heavily)', 'green',     '--'),
    (1.0,   'Q x1     (original)',              'steelblue', '-'),
    (100.0, 'Q x100   (measurements dominate)',  'tomato',    '-'),
]

fig, axes = plt.subplots(2, 1, figsize=(12, 9), sharex=True)
fig.suptitle("Effect of Simulation Covariance Q on KF Estimate", fontsize=12

ax = axes[0]
# dFBA model prediction (no KF corrections) — reference line
ol_cum = [0.01]
for dg in fruit_biomass_history:
    ol_cum.append(ol_cum[-1] + dg)
ax.plot(list(range(7, 54)), ol_cum, color="gray", linewidth=1.5,
        linestyle=":", label="dFBA model (no correction)")
for q_scale, label, color, ls in configs:
    xr, yr, sr = replay_kf(q_scale)
    ax.fill_between(xr, [max(0, y-s) for y,s in zip(yr,sr)],
                    [y+s for y,s in zip(yr,sr)], color=color, alpha=0.12)
    ax.plot(xr, yr, color=color, linewidth=2, linestyle=ls, label=label)
for dpa, entry in cv_ml_measurements.items():
    z = np.array(entry['z'], dtype=float)
    if not np.isnan(z[1]):
        ax.scatter(dpa, z[1]/H_BIOMASS_G_PER_MMOL, color='black', zorder=5,
ax.scatter([], [], color='black', marker='D', s=60, label='Direct Observatio
for start, end, col, lbl in PHASE_SHADING:
    ax.axvspan(start, end, color=col, alpha=0.07)
ax.set_ylabel("Cumulative biomass (mmol)")
ax.legend(fontsize=8); ax.grid(True, alpha=0.4)

ax = axes[1]
for q_scale, label, color, ls in configs:
```

```
    xr, yr, sr = replay_kf(q_scale)
    ax.plot(xr, sr, color=color, linewidth=2, linestyle=ls, label=label)
for dpa in cv_ml_measurements:
    ax.axvline(dpa, color='gray', linestyle=':', linewidth=0.8)
ax.set_xlabel("DPA")
ax.set_ylabel("sigma cumulative biomass (mmol)")
ax.set_title("Uncertainty (sigma): grows between obs, shrinks after — faste
ax.legend(fontsize=8); ax.grid(True, alpha=0.4)

plt.tight_layout()
plt.show()
```

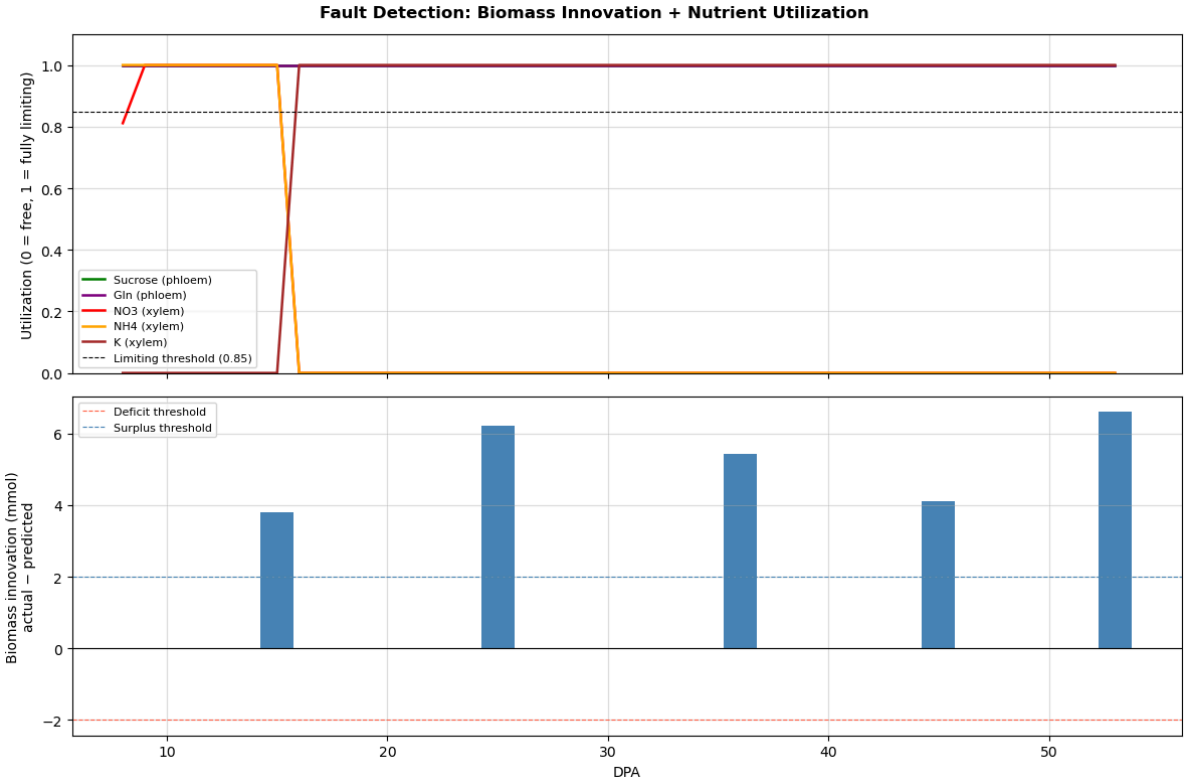


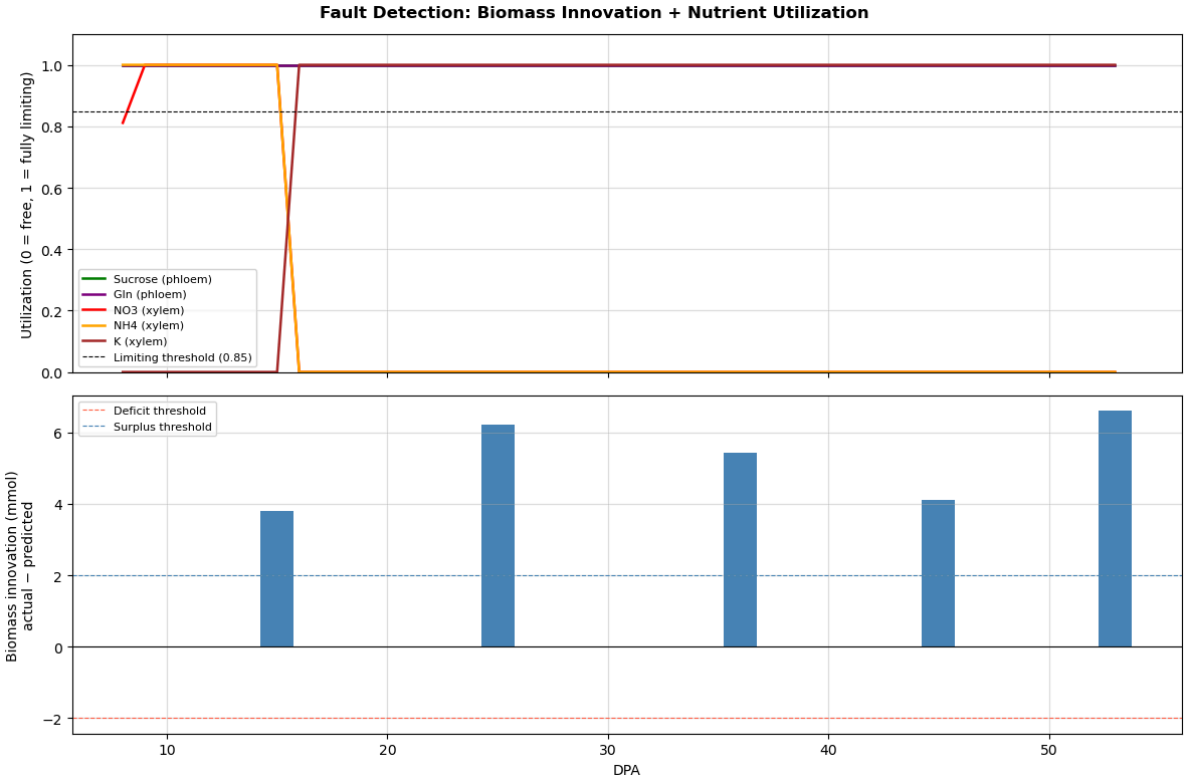


In [38]:
```
# ============================================================================
# FAULT DETECTION — Innovation + Constraint Activity
# ============================================================================
# A negative biomass innovation (actual < predicted) flags underperformance.
# A utilization ratio near 1.0 on the same day identifies the limiting nutri
# Together they answer: "is the plant sick, and which nutrient is the bottle

UTIL_THRESH = 0.85   # fraction of bound used → nutrient considered limiting
BIOMASS_DEFICIT_THRESH = -2.0   # mmol: innovation below this = significant

# Daily mean utilization per nutrient
hr_df = pd.DataFrame(hourly_records)
daily_util = hr_df.groupby('dpa')[
    ['suc_util','gln_util','no3_util','nh4_util','k_util']
].mean().reset_index()

# Biomass innovation from KF: actual (CV-ML) - model prediction, in mmol
innov_records = []
for dpa, innov_vec in kf.innovation_log.items():
    innov_biomass = innov_vec[1] / H_BIOMASS_G_PER_MMOL if not np.isnan(inno
    innov_records.append({'dpa': dpa, 'innov_biomass': innov_biomass})
innov_df = pd.DataFrame(innov_records)

# ── Text report ──────────────────────────────────────────────────────────
print("=== Fault Detection Report ===")
print(f"{'DPA':>4}  {'Innovation (mmol)':>18}  {'Status':>8}  Limiting nutri
print("-" * 65)
for _, row in innov_df.iterrows():
    dpa   = int(row.dpa)
    innov = row.innov_biomass
    util_row = daily_util[daily_util.dpa == dpa]
    bottleneck = []
    if not util_row.empty:
        u = util_row.iloc[0]
        bottleneck = [col.replace('_util', '') for col in
```

```
                    ['suc_util','gln_util','no3_util','nh4_util','k_util']
                    if u[col] > UTIL_THRESH]
    if np.isnan(innov):
        status = "NO DATA"
    elif innov < BIOMASS_DEFICIT_THRESH:
        status = "DEFICIT"
    elif innov > 2.0:
        status = "SURPLUS"
    else:
        status = "OK"
    print(f"{dpa:>4}  {innov:>+18.2f}  {status:>8}  {bottleneck if bottlenec

# ── Plot ──────────────────────────────────────────────────────────────
fig, axes = plt.subplots(2, 1, figsize=(12, 8), sharex=True)
fig.suptitle("Fault Detection: Biomass Innovation + Nutrient Utilization", f

ax = axes[0]
for col, color, label in [
    ('suc_util', 'green',  'Sucrose (phloem)'),
    ('gln_util', 'purple', 'Gln (phloem)'),
    ('no3_util', 'red',    'NO3 (xylem)'),
    ('nh4_util', 'orange', 'NH4 (xylem)'),
    ('k_util',   'brown',  'K (xylem)'),
]:
    ax.plot(daily_util.dpa, daily_util[col], color=color, linewidth=1.8, lab
ax.axhline(UTIL_THRESH, color='k', linestyle='--', linewidth=0.8, label=f'Li
ax.set_ylabel("Utilization (0 = free, 1 = fully limiting)")
ax.set_ylim(0, 1.1)
ax.legend(fontsize=8); ax.grid(True, alpha=0.4)

ax = axes[1]
if not innov_df.empty:
    colors = ['tomato' if v < BIOMASS_DEFICIT_THRESH else
              'steelblue' if v > 2.0 else 'gray'
              for v in innov_df.innov_biomass]
    ax.bar(innov_df.dpa, innov_df.innov_biomass, color=colors, width=1.5)
    ax.axhline(0, color='k', linewidth=0.8)
    ax.axhline(BIOMASS_DEFICIT_THRESH, color='tomato', linestyle='--', linew
    ax.axhline(2.0, color='steelblue', linestyle='--', linewidth=0.8, label=
ax.set_xlabel("DPA")
ax.set_ylabel("Biomass innovation (mmol)\nactual − predicted")
ax.legend(fontsize=8); ax.grid(True, alpha=0.4)

plt.tight_layout()
plt.show()
```

```
=== Fault Detection Report ===
 DPA   Innovation (mmol)    Status  Limiting nutrients
------------------------------------------------------------------
  15               +3.79   SURPLUS  ['suc', 'gln', 'no3', 'nh4']
  25               +6.22   SURPLUS  ['suc', 'gln', 'k']
  36               +5.44   SURPLUS  ['suc', 'gln', 'k']
  45               +4.10   SURPLUS  ['suc', 'gln', 'k']
  53               +6.60   SURPLUS  ['suc', 'gln', 'k']
=== Fault Detection Report ===
 DPA   Innovation (mmol)    Status  Limiting nutrients
------------------------------------------------------------------
  15               +3.79   SURPLUS  ['suc', 'gln', 'no3', 'nh4']
  25               +6.22   SURPLUS  ['suc', 'gln', 'k']
  36               +5.44   SURPLUS  ['suc', 'gln', 'k']
  45               +4.10   SURPLUS  ['suc', 'gln', 'k']
  53               +6.60   SURPLUS  ['suc', 'gln', 'k']
```





In [39]:
```
# ============================================================================
# MODEL INSPECTION  (optional — run for diagnostics, not needed for simulati
# ============================================================================

def search_reactions(model, term):
    results = model.reactions.query(term)
    if not results:
        print(f"  No matches for '{term}'")
    for rxn in results:
        print(f"  {rxn.id:<20} bounds={rxn.bounds}  {rxn.reaction}")

print("=== Fruit model exchange reactions ===")
for rxn in fruit_model.exchanges:
    print(f"  {rxn.id:<18} {rxn.name:<28} {rxn.reaction}")
```

```
print("\n=== Dead-end metabolite analysis (raw wtfruit.xml) ===")
raw_fruit = cobra.io.read_sbml_model("wtfruit.xml")
print(f"{'Exchange rxn':<22} {'Metabolite':<14} {'Prod':>4} {'Cons':>4}  Cla
print("-" * 75)
must_import = []; must_export = []; bidirectional = []; orphan = []
for ex_rxn in raw_fruit.exchanges:
    mets_list = list(ex_rxn.metabolites.keys())
    if not mets_list:
        continue
    met = mets_list[0]
    producers = [r.id for r in raw_fruit.reactions if r.id != ex_rxn.id and
    consumers = [r.id for r in raw_fruit.reactions if r.id != ex_rxn.id and
    n_p, n_c = len(producers), len(consumers)
    if   n_p == 0 and n_c  > 0: tag = "MUST IMPORT";   must_import.append(ex
    elif n_p  > 0 and n_c == 0: tag = "MUST EXPORT";   must_export.append(ex
    elif n_p  > 0 and n_c  > 0: tag = "BIDIRECTIONAL"; bidirectional.append(
    else:                        tag = "ORPHAN";         orphan.append(ex_rx
    print(f"  {ex_rxn.id:<22} {met.id:<14} {n_p:>4} {n_c:>4}  {tag}")
print(f"\nMust import: {must_import}")
print(f"Must export: {must_export}")
print(f"Bidirectional: {bidirectional}")
print(f"Orphan: {orphan}")

print("\n=== Stoichiometric matrix (fruit model) ===")
from cobra.util import create_stoichiometric_matrix
S = create_stoichiometric_matrix(fruit_model, array_type='DataFrame')
print(f"Shape: {S.shape[0]} metabolites × {S.shape[1]} reactions")
print(f"Non-zero: {int((S != 0).sum().sum())}  ({100*(S!=0).sum().sum()/S.si

fig, ax = plt.subplots(figsize=(20, 14))
vmax = max(abs(S.values.min()), abs(S.values.max()))
img = ax.imshow(S.values, aspect='auto', cmap=plt.cm.RdBu_r, vmin=-vmax, vma
ax.set_xticks(range(len(S.columns))); ax.set_xticklabels(S.columns, rotation
ax.set_yticks(range(len(S.index)));   ax.set_yticklabels(S.index, fontsize=7
ax.set_title("Stoichiometric Matrix S — wtfruit.xml  (red=produced, blue=con
plt.colorbar(img, ax=ax, shrink=0.6, label='Stoichiometric coefficient')
plt.tight_layout(); plt.show()
```

```
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Model does not contain SBML fbc package information.
Model does not contain SBML fbc package information.
```

```
=== Fruit model exchange reactions ===
  EX_Suc_in          EX_Suc_in                    Suc_in <--
  EX_Gln_in          EX_Gln_in                    Gln_in <--
  EX_NO3             EX_NO3                       NO3 <--
  EX_CW              EX_CW                        CW -->
  EX_DAG             EX_DAG                       DAG -->
  EX_PROTEIN         EX_PROTEIN                   PROTEIN -->
  EX_STARCH          EX_STARCH                    STARCH -->
  EX_Glu_ac          EX_Glu_ac                    Glu_ac -->
  EX_Asp_ac          EX_Asp_ac                    Asp_ac -->
  EX_Ala_ac          EX_Ala_ac                    Ala_ac -->
  EX_Ser_ac          EX_Ser_ac                    Ser_ac -->
  EX_Cit_ac          EX_Cit_ac                    Cit_ac -->
  EX_Mal_ac          EX_Mal_ac                    Mal_ac -->
  EX_Glc_ac          EX_Glc_ac                    Glc_ac -->
  EX_Fru_ac          EX_Fru_ac                    Fru_ac -->
  EX_Suc_ac          EX_Suc_ac                    Suc_ac -->
  EX_DNA_RNA         EX_DNA_RNA                   DNA_RNA -->
  EX_Pi              EX_Pi                        Pi -->

=== Dead-end metabolite analysis (raw wtfruit.xml) ===
=== Fruit model exchange reactions ===
  EX_Suc_in          EX_Suc_in                    Suc_in <--
  EX_Gln_in          EX_Gln_in                    Gln_in <--
  EX_NO3             EX_NO3                       NO3 <--
  EX_CW              EX_CW                        CW -->
  EX_DAG             EX_DAG                       DAG -->
  EX_PROTEIN         EX_PROTEIN                   PROTEIN -->
  EX_STARCH          EX_STARCH                    STARCH -->
  EX_Glu_ac          EX_Glu_ac                    Glu_ac -->
  EX_Asp_ac          EX_Asp_ac                    Asp_ac -->
  EX_Ala_ac          EX_Ala_ac                    Ala_ac -->
  EX_Ser_ac          EX_Ser_ac                    Ser_ac -->
  EX_Cit_ac          EX_Cit_ac                    Cit_ac -->
  EX_Mal_ac          EX_Mal_ac                    Mal_ac -->
  EX_Glc_ac          EX_Glc_ac                    Glc_ac -->
  EX_Fru_ac          EX_Fru_ac                    Fru_ac -->
  EX_Suc_ac          EX_Suc_ac                    Suc_ac -->
  EX_DNA_RNA         EX_DNA_RNA                   DNA_RNA -->
  EX_Pi              EX_Pi                        Pi -->

=== Dead-end metabolite analysis (raw wtfruit.xml) ===
```

```
SBML package 'layout' not supported by cobrapy, information is not parsed
SBML package 'layout' not supported by cobrapy, information is not parsed
SBML package 'render' not supported by cobrapy, information is not parsed
SBML package 'render' not supported by cobrapy, information is not parsed
Adding exchange reaction EX_Suc_in with default bounds for boundary metabol
ite: Suc_in.
Adding exchange reaction EX_Suc_in with default bounds for boundary metabol
ite: Suc_in.
Adding exchange reaction EX_Gln_in with default bounds for boundary metabol
ite: Gln_in.
Adding exchange reaction EX_Gln_in with default bounds for boundary metabol
ite: Gln_in.
Adding exchange reaction EX_NO3 with default bounds for boundary metabolit
e: NO3.
Adding exchange reaction EX_NO3 with default bounds for boundary metabolit
e: NO3.
Adding exchange reaction EX_CW with default bounds for boundary metabolite:
CW.
Adding exchange reaction EX_CW with default bounds for boundary metabolite:
CW.
Adding exchange reaction EX_DAG with default bounds for boundary metabolit
e: DAG.
Adding exchange reaction EX_DAG with default bounds for boundary metabolit
e: DAG.
Adding exchange reaction EX_PROTEIN with default bounds for boundary metabo
lite: PROTEIN.
Adding exchange reaction EX_PROTEIN with default bounds for boundary metabo
lite: PROTEIN.
Adding exchange reaction EX_STARCH with default bounds for boundary metabol
ite: STARCH.
Adding exchange reaction EX_STARCH with default bounds for boundary metabol
ite: STARCH.
Adding exchange reaction EX_Glu_ac with default bounds for boundary metabol
ite: Glu_ac.
Adding exchange reaction EX_Glu_ac with default bounds for boundary metabol
ite: Glu_ac.
Adding exchange reaction EX_Asp_ac with default bounds for boundary metabol
ite: Asp_ac.
Adding exchange reaction EX_Asp_ac with default bounds for boundary metabol
ite: Asp_ac.
Adding exchange reaction EX_Ala_ac with default bounds for boundary metabol
ite: Ala_ac.
Adding exchange reaction EX_Ala_ac with default bounds for boundary metabol
ite: Ala_ac.
Adding exchange reaction EX_Ser_ac with default bounds for boundary metabol
ite: Ser_ac.
Adding exchange reaction EX_Ser_ac with default bounds for boundary metabol
ite: Ser_ac.
Adding exchange reaction EX_Cit_ac with default bounds for boundary metabol
ite: Cit_ac.
Adding exchange reaction EX_Cit_ac with default bounds for boundary metabol
ite: Cit_ac.
Adding exchange reaction EX_Mal_ac with default bounds for boundary metabol
ite: Mal_ac.
Adding exchange reaction EX_Mal_ac with default bounds for boundary metabol
ite: Mal_ac.
Adding exchange reaction EX_Glc_ac with default bounds for boundary metabol
ite: Glc_ac.
Adding exchange reaction EX_Glc_ac with default bounds for boundary metabol
ite: Glc_ac.
Adding exchange reaction EX_Fru_ac with default bounds for boundary metabol
ite: Fru_ac.
Adding exchange reaction EX_Fru_ac with default bounds for boundary metabol
ite: Fru_ac.
```

```
Adding exchange reaction EX_Suc_ac with default bounds for boundary metabol
ite: Suc_ac.
Adding exchange reaction EX_Suc_ac with default bounds for boundary metabol
ite: Suc_ac.
Adding exchange reaction EX_DNA_RNA with default bounds for boundary metabo
lite: DNA_RNA.
Adding exchange reaction EX_DNA_RNA with default bounds for boundary metabo
lite: DNA_RNA.
Adding exchange reaction EX_Pi with default bounds for boundary metabolite:
Pi.
Adding exchange reaction EX_Pi with default bounds for boundary metabolite:
Pi.
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtpi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtpi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtpi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtpi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgapdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgapdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgapdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgapdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Veno>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Veno>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Veno>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Veno>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vidh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vidh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vidh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vidh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vkgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vkgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vkgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vkgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VFum>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VFum>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VFum>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VFum>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vmdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vmdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vmdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vmdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vepi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vepi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vepi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vepi>'
```

```
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vriso>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vriso>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vriso>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vriso>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtk2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtk2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vtald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtald>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vtald>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgm>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpgm>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgm>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpgm>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsusy>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsusy>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsusy>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsusy>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Valaat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Valaat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Valaat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Valaat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaspat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vaspat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaspat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vaspat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vserat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vserat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vserat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vserat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vadk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vadk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vadk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vadk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VSuc_up
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VSuc_up
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VSuc_up>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VSuc_up>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VGln_up
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction VGln_up
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VGln_up>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction VGln_up>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgk>'
```

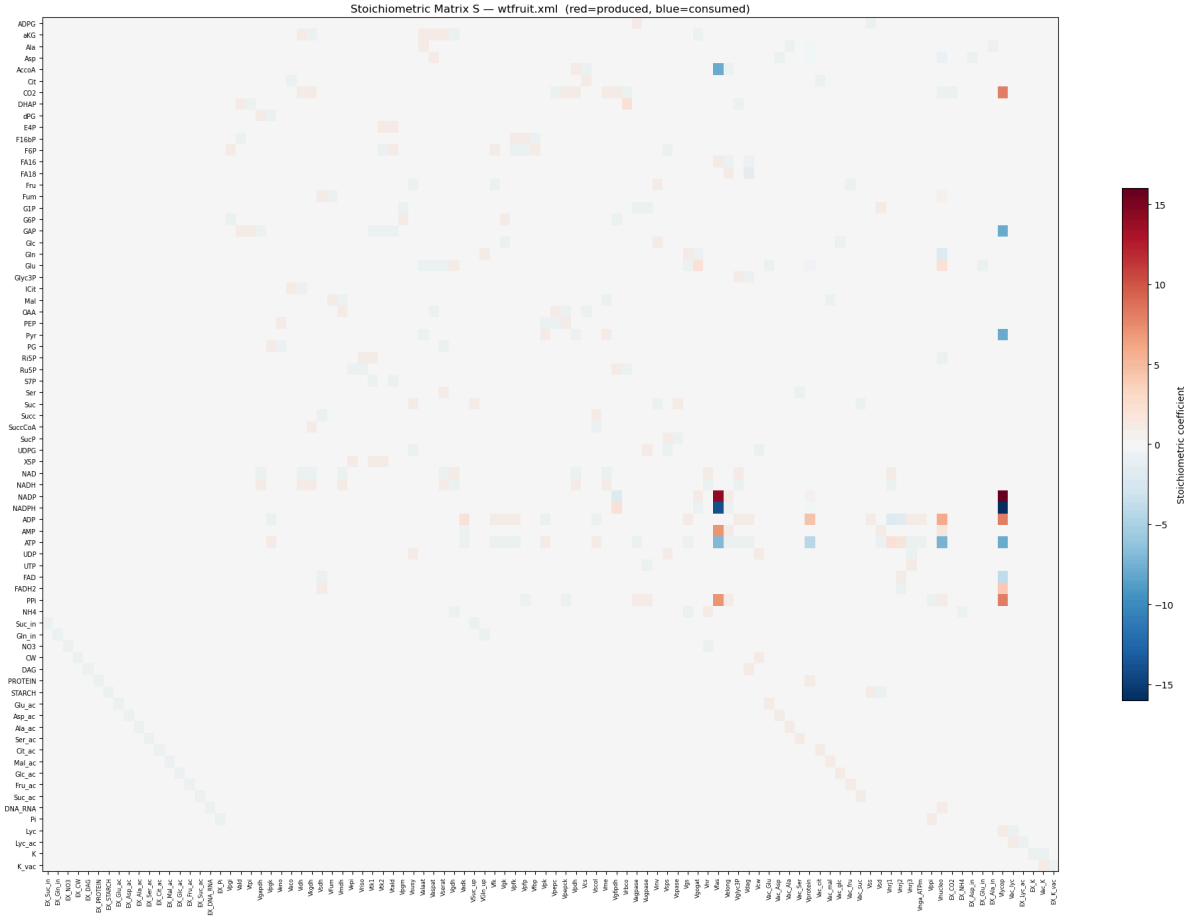
```
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpfp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpfp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfbp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfbp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfbp>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfbp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpk>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpk>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepc>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepc>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepck>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpepck>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepck>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpepck>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vpdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vpdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vscol>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vscol>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vme>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vme>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vscol>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vscol>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vme>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vme>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vg6pdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vg6pdh>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vg6pdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vrbco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vrbco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vagpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vg6pdh>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vrbco>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vrbco>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vagpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vagpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vugpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vugpase>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vagpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vugpase
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vugpase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vinv>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vinv>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsps>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsps>'
```

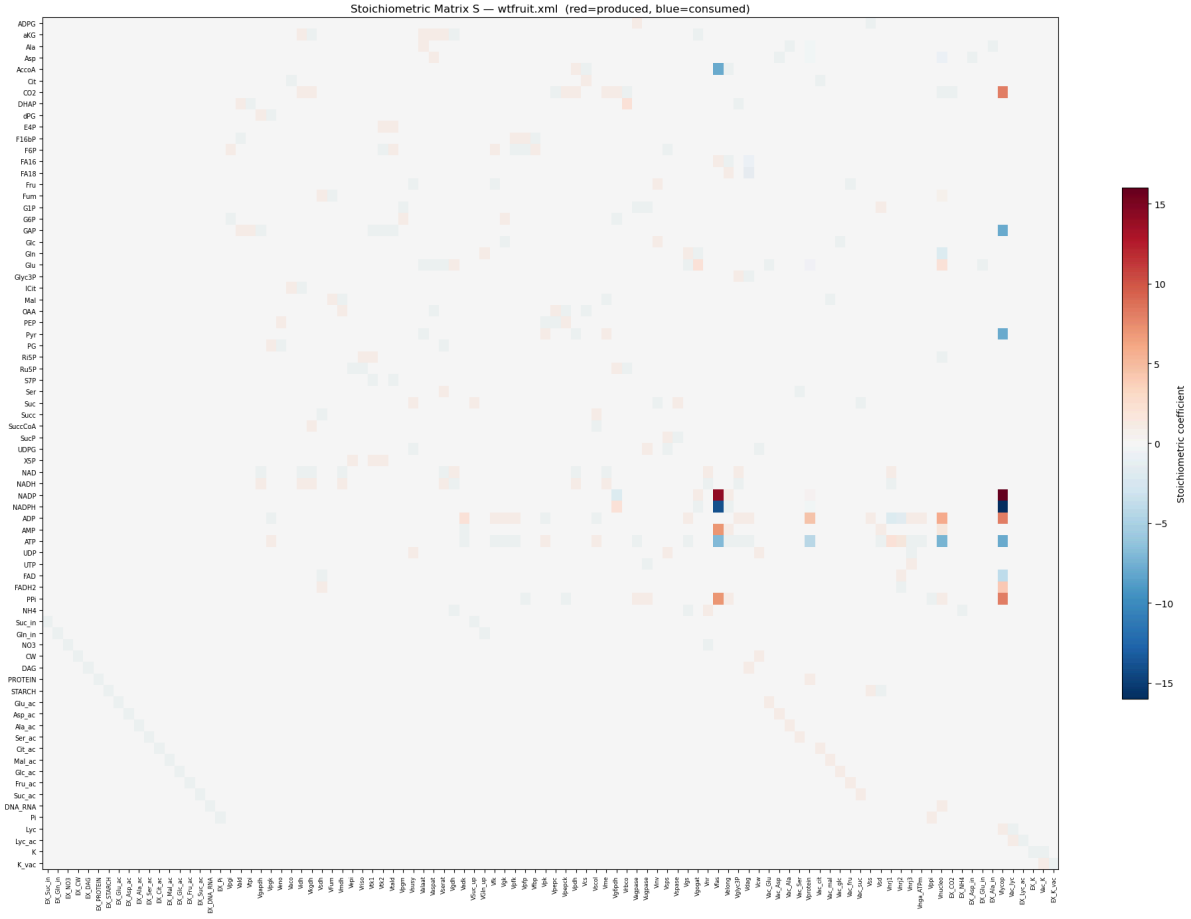
```
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vinv>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vinv>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsps>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsps>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vspase>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vspase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgogat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vspase>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vspase>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgs>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgs>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vgogat>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgogat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnr>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnr>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfas>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfas>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Velong>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Velong>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vgogat>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnr>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnr>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vfas>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vfas>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Velong>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Velong>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vglyc3P
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vglyc3P>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vdag>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vdag>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcw>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vglyc3P
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vglyc3P>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vdag>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vdag>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vcw>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcw>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Glu
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Glu>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vcw>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Glu
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Glu>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Asp
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Asp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ala
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ala>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Asp
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Asp>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ala
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ala>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ser
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ser>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vprotein
```

```
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_Ser
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_Ser>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vprotein
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vprotein
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_cit
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vprotein
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_cit
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_cit>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_mal
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_mal>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_cit>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_mal
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_mal>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_glc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_glc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_fru
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_fru>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_suc
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_glc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_glc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_fru
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_fru>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vac_suc
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_suc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vss>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vss>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsd>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vac_suc>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vss>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vss>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vsd>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsd>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vsd>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj1>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj1>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj2>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj2>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj3>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj3>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnga_ATP
m>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnga_ATPm
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vppi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vppi>'
```

```
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnucleo
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnucleo>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnrj3>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnrj3>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnga_ATP
m>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnga_ATPm
>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vppi>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vppi>'
Missing lower flux bound set to '-1000.0' for reaction: '<Reaction Vnucleo
>'
Missing upper flux bound set to '1000.0' for reaction: '<Reaction Vnucleo>'
No objective coefficients in model. Unclear what should be optimized
Missing flux bounds on reactions set to default bounds.As best practise and
to avoid confusion flux bounds should be set explicitly on all reactions.
No objective coefficients in model. Unclear what should be optimized
Missing flux bounds on reactions set to default bounds.As best practise and
to avoid confusion flux bounds should be set explicitly on all reactions.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
Could not identify an external compartment by name and choosing one with th
e most boundary reactions. That might be complete nonsense or change sudden
ly. Consider renaming your compartments using `Model.compartments` to fix t
his.
```

```
Exchange rxn           Metabolite     Prod Cons  Classification
------------------------------------------------------------------------------
  EX_Suc_in              Suc_in            0    1  MUST IMPORT
  EX_Gln_in              Gln_in            0    1  MUST IMPORT
  EX_NO3                 NO3               0    1  MUST IMPORT
  EX_CW                  CW                1    0  MUST EXPORT
  EX_DAG                 DAG               1    0  MUST EXPORT
  EX_PROTEIN             PROTEIN           1    0  MUST EXPORT
  EX_STARCH              STARCH            1    1  BIDIRECTIONAL
  EX_Glu_ac              Glu_ac            1    0  MUST EXPORT
  EX_Asp_ac              Asp_ac            1    0  MUST EXPORT
  EX_Ala_ac              Ala_ac            1    0  MUST EXPORT
  EX_Ser_ac              Ser_ac            1    0  MUST EXPORT
  EX_Cit_ac              Cit_ac            1    0  MUST EXPORT
  EX_Mal_ac              Mal_ac            1    0  MUST EXPORT
  EX_Glc_ac              Glc_ac            1    0  MUST EXPORT
  EX_Fru_ac              Fru_ac            1    0  MUST EXPORT
  EX_Suc_ac              Suc_ac            1    0  MUST EXPORT
  EX_DNA_RNA             DNA_RNA           1    0  MUST EXPORT
  EX_Pi                  Pi                1    0  MUST EXPORT

Must import: ['EX_Suc_in', 'EX_Gln_in', 'EX_NO3']
Must export: ['EX_CW', 'EX_DAG', 'EX_PROTEIN', 'EX_Glu_ac', 'EX_Asp_ac', 'E
X_Ala_ac', 'EX_Ser_ac', 'EX_Cit_ac', 'EX_Mal_ac', 'EX_Glc_ac', 'EX_Fru_ac',
'EX_Suc_ac', 'EX_DNA_RNA', 'EX_Pi']
Bidirectional: ['EX_STARCH']
Orphan: []

=== Stoichiometric matrix (fruit model) ===
Shape: 74 metabolites × 100 reactions
Non-zero: 303  (4.1% filled)
Exchange rxn           Metabolite     Prod Cons  Classification
------------------------------------------------------------------------------
  EX_Suc_in              Suc_in            0    1  MUST IMPORT
  EX_Gln_in              Gln_in            0    1  MUST IMPORT
  EX_NO3                 NO3               0    1  MUST IMPORT
  EX_CW                  CW                1    0  MUST EXPORT
  EX_DAG                 DAG               1    0  MUST EXPORT
  EX_PROTEIN             PROTEIN           1    0  MUST EXPORT
  EX_STARCH              STARCH            1    1  BIDIRECTIONAL
  EX_Glu_ac              Glu_ac            1    0  MUST EXPORT
  EX_Asp_ac              Asp_ac            1    0  MUST EXPORT
  EX_Ala_ac              Ala_ac            1    0  MUST EXPORT
  EX_Ser_ac              Ser_ac            1    0  MUST EXPORT
  EX_Cit_ac              Cit_ac            1    0  MUST EXPORT
  EX_Mal_ac              Mal_ac            1    0  MUST EXPORT
  EX_Glc_ac              Glc_ac            1    0  MUST EXPORT
  EX_Fru_ac              Fru_ac            1    0  MUST EXPORT
  EX_Suc_ac              Suc_ac            1    0  MUST EXPORT
  EX_DNA_RNA             DNA_RNA           1    0  MUST EXPORT
  EX_Pi                  Pi                1    0  MUST EXPORT

Must import: ['EX_Suc_in', 'EX_Gln_in', 'EX_NO3']
Must export: ['EX_CW', 'EX_DAG', 'EX_PROTEIN', 'EX_Glu_ac', 'EX_Asp_ac', 'E
X_Ala_ac', 'EX_Ser_ac', 'EX_Cit_ac', 'EX_Mal_ac', 'EX_Glc_ac', 'EX_Fru_ac',
'EX_Suc_ac', 'EX_DNA_RNA', 'EX_Pi']
Bidirectional: ['EX_STARCH']
Orphan: []

=== Stoichiometric matrix (fruit model) ===
Shape: 74 metabolites × 100 reactions
Non-zero: 303  (4.1% filled)
```





Greenhouse Extension — N_ROWS × N_COLS grid

Design choices:

- One `PoolKalmanFilter` per plant; pools carry per-plant metabolic uncertainty.
- **Veg LP** runs once per *column* per hour using the column-average pool state. Columns differ only in PPFD (wall-shadow gradient); rows in the same column share the same leaf-export fluxes.
- **Fruit LP + KF predict** run once per *plant* per hour.
- **CV-ML observations** are per-plant with camera noise (σ_d = 2 mm, σ_w = 0.15 g).
- `FAST_MODE = True` samples 4 h/DPA (h = 0, 6, 12, 18) and scales ×6 — quick sanity check.
- `FAST_MODE = False` runs all 24 h/DPA — full diurnal dynamics, takes ~30 min.

In [40]:
```
# ============================================================================
# GREENHOUSE LAYOUT + PER-PLANT INITIALISATION
# ============================================================================

FAST_MODE = True    # set False for full 24-hour diurnal simulation
N_ROWS, N_COLS = 4, 6

def col_ppfd_scale(col, n_cols=N_COLS):
    """Wall-shadow gradient: edge columns get 15 % less PPFD than centre."""
    centre = (n_cols - 1) / 2.0
    dist   = abs(col - centre) / max(centre, 1e-9)
    return 1.0 - 0.15 * dist

COL_PPFD_SCALE = np.array([col_ppfd_scale(c) for c in range(N_COLS)])
print("Column PPFD fractions:", np.round(COL_PPFD_SCALE, 3))

# ── Per-plant KF: ±5 % pool noise at planting ──────────────────────────
rng_gh = np.random.default_rng(seed=7)
kf_grid = {}
for _r in range(N_ROWS):
    for _c in range(N_COLS):
        _x0 = np.array(x0_kf, dtype=float) * (1.0 + rng_gh.normal(0, 0.05, 
        kf_grid[(_r, _c)] = PoolKalmanFilter(
            np.maximum(_x0, 0.0).tolist(),
            R_default_diag=[25.0, 0.25]
        )

# ── Per-plant CV-ML observations (same DPA schedule, per-plant camera noise
CV_REF_GH = {
    15: {"z": [20.0, 0.90], "R": [16.0, 0.02]},
    25: {"z": [50.0, 4.00], "R": [25.0, 0.08]},
    36: {"z": [72.0, 7.00], "R": [12.0, 0.05]},
    45: {"z": [82.0, 8.00], "R": [18.0, 0.06]},
    53: {"z": [88.0, 9.00], "R": [10.0, 0.04]},
}
cv_plant_obs = {}
for _r in range(N_ROWS):
    for _c in range(N_COLS):
        for _dpa, _entry in CV_REF_GH.items():
            _z = [
                float(np.clip(_entry["z"][0] + rng_gh.normal(0, 2.0),  1, 20
                float(np.clip(_entry["z"][1] + rng_gh.normal(0, 0.15), 0,  5
            ]
            cv_plant_obs[(_r, _c, _dpa)] = {"z": _z, "R": _entry["R"]}

print(f"Initialised {len(kf_grid)} plant KFs  |  {len(cv_plant_obs)} CV obse
```

```
Column PPFD fractions: [0.85 0.91 0.97 0.97 0.91 0.85]
Initialised 24 plant KFs  |  120 CV observations.
Column PPFD fractions: [0.85 0.91 0.97 0.97 0.91 0.85]
Initialised 24 plant KFs  |  120 CV observations.
```

In [41]:

```
# ============================================================================
# GREENHOUSE SIMULATION  (DPA 8–53)
# Veg LP: once per column per hour  (N_COLS × 24 × 46 calls)
# Fruit LP: once per plant per hour (N_ROWS*N_COLS × 24 × 46 calls)
# ============================================================================

gh_records   = []
gh_daily_bio = {(_r, _c): [] for _r in range(N_ROWS) for _c in range(N_COLS)

hours_sample = list(range(0, 24, 6)) if FAST_MODE else list(range(24))
hour_scale   = 24 // len(hours_sample)   # multiply deltas to represent full

# Reuse segment boundaries from the single-fruit simulation cell
for seg_idx, (seg_start, seg_end) in enumerate(zip(seg_start_dpas, seg_end_d
    print(f"\n=== GH Segment {seg_idx+1}: DPA {seg_start}\u2013{seg_end} ===

    for current_DPA in range(seg_start, seg_end + 1):
        i          = current_DPA - 8
        ratio      = sink_strength(current_DPA)
        c_priority = c_priority_from_phloem_cn()
        fruit_obj  = get_fruit_objective(fruit_model, current_DPA)
        print(f"  DPA {current_DPA}", end=" ", flush=True)

        for h in hours_sample:

            # ── VEG LP: one call per column (column-average pool state) ──
            veg_col_cache = {}
            for col in range(N_COLS):
                _kfs   = [kf_grid[(_r, col)] for _r in range(N_ROWS)]
                avg_suc = float(np.mean([_k.x[IDX_PHLOEM_SUC] for _k in _kfs
                avg_no3 = float(np.mean([_k.x[IDX_XYLEM_NO3]  for _k in _kfs
                avg_pi  = float(np.mean([_k.x[IDX_XYLEM_PI]   for _k in _kfs
                avg_nh4 = float(np.mean([_k.x[IDX_XYLEM_NH4]  for _k in _kfs

                scale_c    = COL_PPFD_SCALE[col]
                ppfd_h_c   = diurnal_ppfd(ppfd_day[i] * scale_c, h, photoper
                is_light_c = ppfd_h_c > 0.0

                fb_suc_c  = phloem_loading_fb(avg_suc)
                fb_no3_c  = root_no3_fb(avg_no3)
                fb_pi_c   = root_pi_fb(avg_pi)
                eff_no3_c = no3_ceiling(tank_no3[i]) * fb_no3_c
                eff_pi_c  = pi_ceiling(tank_pi[i])   * fb_pi_c

                leaf = dict(suc=0.0, gln=0.0, asp=0.0, glu=0.0, ala=0.0,
                            no3=0.0, pi=0.0, nh4=0.0,
                            eff_no3=eff_no3_c, eff_pi=eff_pi_c,
                            ppfd=ppfd_h_c, is_light=is_light_c)

                if is_light_c:
                    max_ph_c = VYTOP_PHOTON_PER_LH_AT200 * (ppfd_h_c / 200.0
                    sucr_ub  = VYTOP_SUCR_PER_LH_AT200   * (ppfd_h_c / 200.0
                    gln_ub   = VYTOP_GLN_PER_LH_AT200    * (ppfd_h_c / 200.0

                    veg_model.reactions.get_by_id('EX_photon_b').lower_bound
                    veg_model.reactions.get_by_id('EX_co2_b').lower_bound
                    veg_model.reactions.get_by_id('EX_h2o_b').lower_bound
                    veg_model.reactions.get_by_id('EX_no3_b').lower_bound
                        -min(eff_no3_c, avg_no3 * PLANT_DW_PER_FRUIT) / PLAN
```

```
                    veg_model.reactions.get_by_id('EX_nh4_b').lower_bound
                        -min(nh4_ceiling(tank_nh4[i]), avg_nh4 * PLANT_DW_PE
                    veg_model.reactions.get_by_id('EX_pi_b').lower_bound
                        -min(eff_pi_c,  avg_pi  * PLANT_DW_PER_FRUIT) / PLAN
                    veg_model.reactions.get_by_id('EX_so4_b').lower_bound
                    veg_model.reactions.get_by_id('EX_k_b').lower_bound
                        -k_ceiling(tank_k[i]) / PLANT_DW_PER_FRUIT)
                    veg_model.reactions.get_by_id('EX_sucr_b').upper_bound
                    veg_model.reactions.get_by_id('EX_gln_L_b').upper_bound

                    veg_model.objective = 'EX_sucr_b'
                    _ss = veg_model.optimize()
                    if _ss.status == 'optimal':
                        leaf['suc'] = max(0.0, _ss.objective_value) * PLANT_
                        veg_model.reactions.get_by_id('EX_sucr_b').lower_bou
                            _ss.objective_value * c_priority)
                        veg_model.objective = 'EX_gln_L_b'
                        _gs = veg_model.optimize()
                        if _gs.status == 'optimal':
                            leaf['gln'] = max(0.0, _gs.objective_value) * PL
                            leaf['no3'] = abs(_gs.fluxes.get('EX_no3_b', 0.0
                            leaf['pi']  = abs(_gs.fluxes.get('EX_pi_b',  0.0
                            leaf['nh4'] = abs(_gs.fluxes.get('EX_nh4_b', 0.0
                            leaf['asp'] = max(0.0, _gs.fluxes.get('EX_asp_L_
                            leaf['glu'] = max(0.0, _gs.fluxes.get('EX_glu_L_
                            leaf['ala'] = max(0.0, _gs.fluxes.get('EX_ala_L_
                        veg_model.reactions.get_by_id('EX_sucr_b').lower_bou
                        veg_model.reactions.get_by_id('EX_sucr_b').upper_bou

                veg_col_cache[col] = leaf

            # ── FRUIT LP + KF PREDICT: per plant ──────────────────────────
            for row in range(N_ROWS):
                for col in range(N_COLS):
                    kf_p = kf_grid[(row, col)]
                    vc   = veg_col_cache[col]

                    Sp_suc = kf_p.x[IDX_PHLOEM_SUC]; Sp_gln = kf_p.x[IDX_PHL
                    Sp_asp = kf_p.x[IDX_PHLOEM_ASP]; Sp_glu = kf_p.x[IDX_PHL
                    Sp_ala = kf_p.x[IDX_PHLOEM_ALA]
                    Sx_no3 = kf_p.x[IDX_XYLEM_NO3];  Sx_pi  = kf_p.x[IDX_XYL
                    Sx_k   = kf_p.x[IDX_XYLEM_K];    Sx_nh4 = kf_p.x[IDX_XYL

                    suc_ceil = Sp_suc + _sigma_slack(kf_p, IDX_PHLOEM_SUC, N
                    gln_ceil = Sp_gln + _sigma_slack(kf_p, IDX_PHLOEM_GLN, N
                    asp_ceil = Sp_asp + _sigma_slack(kf_p, IDX_PHLOEM_ASP, N
                    glu_ceil = Sp_glu + _sigma_slack(kf_p, IDX_PHLOEM_GLU, N
                    ala_ceil = Sp_ala + _sigma_slack(kf_p, IDX_PHLOEM_ALA, N
                    no3_ceil = Sx_no3 + _sigma_slack(kf_p, IDX_XYLEM_NO3,  N
                    nh4_ceil = Sx_nh4 + _sigma_slack(kf_p, IDX_XYLEM_NH4,  N
                    k_ceil   = Sx_k   + _sigma_slack(kf_p, IDX_XYLEM_K,    N

                    max_no3_h = no3_ceiling(tank_no3[i])
                    max_pi_h  = pi_ceiling(tank_pi[i])
                    max_k_h   = k_ceiling(tank_k[i])
                    max_nh4_h = nh4_ceiling(tank_nh4[i])

                    eff_no3_h = max_no3_h * root_no3_fb(Sx_no3)
                    eff_pi_h  = max_pi_h  * root_pi_fb(Sx_pi)

                    Sx_no3 = min(Sx_no3 + eff_no3_h / PLANT_DW_PER_FRUIT, MA
                    Sx_pi  = min(Sx_pi  + eff_pi_h  / PLANT_DW_PER_FRUIT, MA
                    Sx_k   = min(Sx_k   + max_k_h   / PLANT_DW_PER_FRUIT, MA
                    Sx_nh4 = min(Sx_nh4 + max_nh4_h / PLANT_DW_PER_FRUIT, MA
```

```
                root_atp   = (eff_no3_h * ATP_PER_NO3 + max_nh4_h * ATP_
                           + eff_pi_h  * ATP_PER_PI  + max_k_h   * ATP_
                Sp_suc     = max(0.0, Sp_suc - root_atp / ATP_PER_SUC *

                Sp_suc = min(Sp_suc + vc['suc'], MAX_PHLOEM_SUC)
                Sp_gln = min(Sp_gln + vc['gln'], MAX_PHLOEM_GLN)
                Sp_asp = min(Sp_asp + vc['asp'], MAX_PHLOEM_ASP)
                Sp_glu = min(Sp_glu + vc['glu'], MAX_PHLOEM_GLU)
                Sp_ala = min(Sp_ala + vc['ala'], MAX_PHLOEM_ALA)
                Sx_no3 = max(0.0, Sx_no3 - vc['no3'])
                Sx_pi  = max(0.0, Sx_pi  - vc['pi'])
                Sx_nh4 = max(0.0, Sx_nh4 - vc['nh4'])

                suc_f = min(suc_ceil * ratio / 24, suc_ceil)
                gln_f = min(gln_ceil * ratio / 24, gln_ceil)
                asp_f = min(asp_ceil * ratio / 24, asp_ceil)
                glu_f = min(glu_ceil * ratio / 24, glu_ceil)
                ala_f = min(ala_ceil * ratio / 24, ala_ceil)
                no3_f = min(eff_no3_h / PLANT_DW_PER_FRUIT * xylem_no3_f
                nh4_f = min(max_nh4_h / PLANT_DW_PER_FRUIT * nh4_fruit_f
                k_f   = min(max_k_h   / PLANT_DW_PER_FRUIT * k_fruit_fra

                fruit_model.reactions.get_by_id("EX_Suc_in").lower_bound
                fruit_model.reactions.get_by_id("EX_Gln_in").lower_bound
                fruit_model.reactions.get_by_id("EX_Asp_in").lower_bound
                fruit_model.reactions.get_by_id("EX_Glu_in").lower_bound
                fruit_model.reactions.get_by_id("EX_Ala_in").lower_bound
                fruit_model.reactions.get_by_id("EX_NO3").lower_bound
                fruit_model.reactions.get_by_id("EX_NH4").lower_bound
                fruit_model.reactions.get_by_id("EX_K").lower_bound
                fruit_model.objective = fruit_obj

                hour_bio_p = 0.0
                try:
                    f_sol = fruit_model.optimize()
                    if f_sol.status == 'optimal':
                        suc_u = abs(f_sol.fluxes.get('EX_Suc_in', 0.0))
                        gln_u = abs(f_sol.fluxes.get('EX_Gln_in', 0.0))
                        asp_u = abs(f_sol.fluxes.get('EX_Asp_in', 0.0))
                        glu_u = abs(f_sol.fluxes.get('EX_Glu_in', 0.0))
                        ala_u = abs(f_sol.fluxes.get('EX_Ala_in', 0.0))
                        no3_u = abs(f_sol.fluxes.get('EX_NO3',    0.0))
                        nh4_u = abs(f_sol.fluxes.get('EX_NH4',    0.0))
                        k_u   = abs(f_sol.fluxes.get('EX_K',      0.0))
                        Sp_suc = max(0.0, Sp_suc - suc_u)
                        Sp_gln = max(0.0, Sp_gln - gln_u)
                        Sp_asp = max(0.0, Sp_asp - asp_u)
                        Sp_glu = max(0.0, Sp_glu - glu_u)
                        Sp_ala = max(0.0, Sp_ala - ala_u)
                        Sx_no3 = max(0.0, Sx_no3 - no3_u)
                        Sx_nh4 = max(0.0, Sx_nh4 - nh4_u)
                        Sx_k   = max(0.0, Sx_k   - k_u)
                        hour_bio_p = sum(f_sol.fluxes.get(r, 0.0)
                            for r in ['EX_STARCH','EX_PROTEIN','EX_CW',
                                      'EX_Glc_ac','EX_Fru_ac','EX_Suc_ac
                                      'EX_Glu_ac','EX_Asp_ac','EX_Ala_ac
                except Exception:
                    pass

                delta_x = np.array([
                    Sp_suc - kf_p.x[IDX_PHLOEM_SUC],
                    Sp_gln - kf_p.x[IDX_PHLOEM_GLN],
                    Sp_asp - kf_p.x[IDX_PHLOEM_ASP],
```

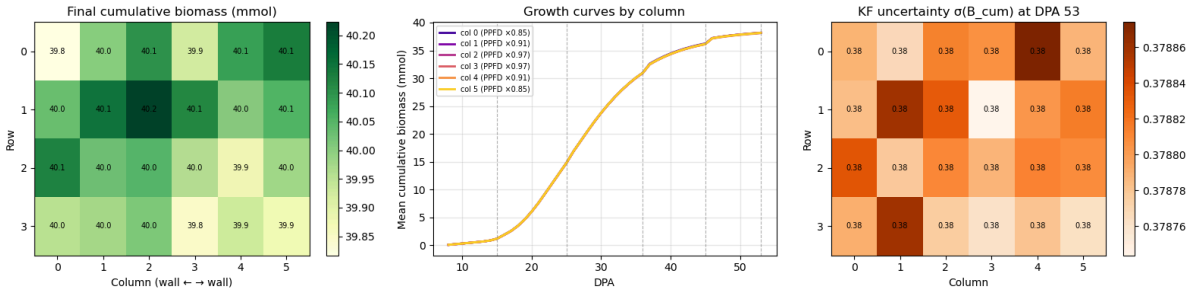
```
                        Sp_glu = kf_p.x[IDX_PHLOEM_GLU],
                        Sp_ala = kf_p.x[IDX_PHLOEM_ALA],
                        Sx_no3 = kf_p.x[IDX_XYLEM_NO3],
                        Sx_pi  = kf_p.x[IDX_XYLEM_PI],
                        Sx_k   = kf_p.x[IDX_XYLEM_K],
                        Sx_nh4 = kf_p.x[IDX_XYLEM_NH4],
                        hour_bio_p,
                    ]) * hour_scale
                    kf_p.predict(delta_x)
                    _apply_pool_caps(kf_p)
            # end per-plant loop
        # end hours loop

        # ── record daily state per plant ──────────────────────────────
        for row in range(N_ROWS):
            for col in range(N_COLS):
                _kp = kf_grid[(row, col)]
                gh_daily_bio[(row, col)].append(_kp.x[IDX_CUM_BIOMASS])
                gh_records.append({
                    'row': row, 'col': col, 'dpa': current_DPA,
                    'cum_biomass':   _kp.x[IDX_CUM_BIOMASS],
                    'sigma_biomass': _kp.sigma[IDX_CUM_BIOMASS],
                    'S_phloem_suc':  _kp.x[IDX_PHLOEM_SUC],
                    'S_xylem_no3':   _kp.x[IDX_XYLEM_NO3],
                })

    # ── KF update at segment end (per plant) ──────────────────────────
    if seg_end in CV_REF_GH:
        for row in range(N_ROWS):
            for col in range(N_COLS):
                _e   = cv_plant_obs[(row, col, seg_end)]
                _z   = np.array(_e["z"], dtype=float)
                _R   = np.diag(np.array(_e["R"], dtype=float))
                kf_grid[(row, col)].update(_z, R_override=_R, dpa=seg_end)
                _apply_pool_caps(kf_grid[(row, col)])
        print(f"\n  [KF update] DPA {seg_end}")

print("\nGreenhouse simulation complete.")
```

```
=== GH Segment 1: DPA 8–15 ===
  DPA 8
=== GH Segment 1: DPA 8–15 ===
  DPA 8   DPA 9   DPA 9   DPA 10   DPA 10   DPA 11   DPA 11   DPA 12   DPA
12   DPA 13   DPA 13   DPA 14   DPA 14   DPA 15   DPA 15
  [KF update] DPA 15

=== GH Segment 2: DPA 16–25 ===
  DPA 16
  [KF update] DPA 15

=== GH Segment 2: DPA 16–25 ===
  DPA 16   DPA 17   DPA 17   DPA 18   DPA 18   DPA 19   DPA 19   DPA 20   D
PA 20   DPA 21   DPA 21   DPA 22   DPA 22   DPA 23   DPA 23   DPA 24   DPA
24   DPA 25   DPA 25
  [KF update] DPA 25

=== GH Segment 3: DPA 26–36 ===
  DPA 26
  [KF update] DPA 25

=== GH Segment 3: DPA 26–36 ===
  DPA 26   DPA 27   DPA 27   DPA 28   DPA 28   DPA 29   DPA 29   DPA 30   D
PA 30   DPA 31   DPA 31   DPA 32   DPA 32   DPA 33   DPA 33   DPA 34   DPA
34   DPA 35   DPA 35   DPA 36   DPA 36
  [KF update] DPA 36

=== GH Segment 4: DPA 37–45 ===
  DPA 37
  [KF update] DPA 36

=== GH Segment 4: DPA 37–45 ===
  DPA 37   DPA 38   DPA 38   DPA 39   DPA 39   DPA 40   DPA 40   DPA 41   D
PA 41   DPA 42   DPA 42   DPA 43   DPA 43   DPA 44   DPA 44   DPA 45   DPA
45
  [KF update] DPA 45

=== GH Segment 5: DPA 46–53 ===
  DPA 46
  [KF update] DPA 45

=== GH Segment 5: DPA 46–53 ===
  DPA 46   DPA 47   DPA 47   DPA 48   DPA 48   DPA 49   DPA 49   DPA 50   D
PA 50   DPA 51   DPA 51   DPA 52   DPA 52   DPA 53   DPA 53
  [KF update] DPA 53

Greenhouse simulation complete.

  [KF update] DPA 53

Greenhouse simulation complete.
```

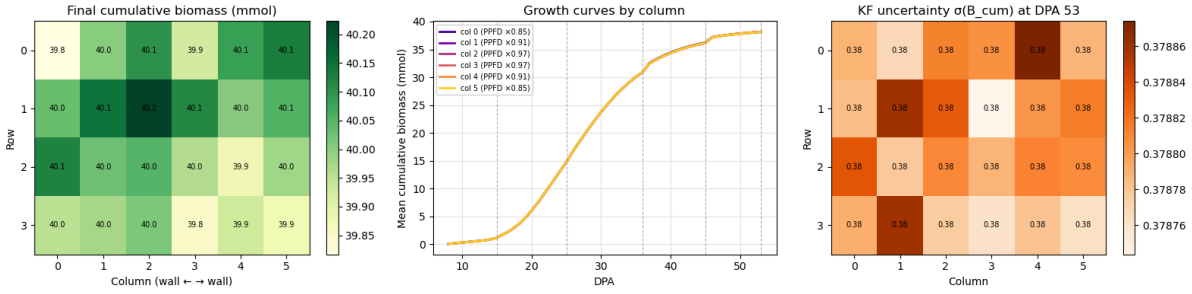
In [42]:

```
# ============================================================================
# GREENHOUSE VISUALISATION
# 1) Heatmap — final cumulative biomass per plant (N_ROWS × N_COLS)
# 2) Growth curves grouped by column (shows PPFD gradient effect)
# 3) Heatmap — final KF uncertainty σ(B_cum)
# ============================================================================

gh_df = pd.DataFrame(gh_records)

fig, axes = plt.subplots(1, 3, figsize=(16, 4))

# ── 1. Biomass heatmap ──────────────────────────────────────────────
```

```
bio_grid = np.array([[kf_grid[(r, c)].x[IDX_CUM_BIOMASS]
                     for c in range(N_COLS)] for r in range(N_ROWS)])
im0 = axes[0].imshow(bio_grid, cmap='YlGn', aspect='auto')
axes[0].set_title('Final cumulative biomass (mmol)')
axes[0].set_xlabel('Column (wall ← → wall)'); axes[0].set_ylabel('Row')
axes[0].set_xticks(range(N_COLS)); axes[0].set_yticks(range(N_ROWS))
plt.colorbar(im0, ax=axes[0])
for r in range(N_ROWS):
    for c in range(N_COLS):
        axes[0].text(c, r, f"{bio_grid[r,c]:.1f}", ha='center', va='center',

# ── 2. Growth curves per column ──────────────────────────────────────
col_palette = plt.cm.plasma(np.linspace(0.1, 0.9, N_COLS))
for col in range(N_COLS):
    col_data = gh_df[gh_df['col'] == col].groupby('dpa')['cum_biomass'].mean
    axes[1].plot(col_data.index, col_data.values,
                 label=f'col {col} (PPFD ×{COL_PPFD_SCALE[col]:.2f})',
                 color=col_palette[col], linewidth=2)
for _dpa in CV_REF_GH:
    axes[1].axvline(_dpa, color='gray', linewidth=0.8, linestyle='--', alpha
axes[1].set_xlabel('DPA'); axes[1].set_ylabel('Mean cumulative biomass (mmo
axes[1].set_title('Growth curves by column')
axes[1].legend(fontsize=7); axes[1].grid(True, alpha=0.3)

# ── 3. Uncertainty heatmap ──────────────────────────────────────
sig_grid = np.array([[kf_grid[(r, c)].sigma[IDX_CUM_BIOMASS]
                     for c in range(N_COLS)] for r in range(N_ROWS)])
im2 = axes[2].imshow(sig_grid, cmap='Oranges', aspect='auto')
axes[2].set_title('KF uncertainty σ(B_cum) at DPA 53')
axes[2].set_xlabel('Column'); axes[2].set_ylabel('Row')
axes[2].set_xticks(range(N_COLS)); axes[2].set_yticks(range(N_ROWS))
plt.colorbar(im2, ax=axes[2])
for r in range(N_ROWS):
    for c in range(N_COLS):
        axes[2].text(c, r, f"{sig_grid[r,c]:.2f}", ha='center', va='center',

plt.tight_layout()
plt.savefig('greenhouse_results.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"\nBiomass range: {bio_grid.min():.2f} – {bio_grid.max():.2f} mmol")
print(f"PPFD effect:   centre cols avg {bio_grid[:, 2:4].mean():.2f} mmol  "
      f"vs  edge cols avg {bio_grid[:, [0,-1]].mean():.2f} mmol")
```




```
Biomass range: 39.82 – 40.22 mmol
PPFD effect:   centre cols avg 40.03 mmol  vs  edge cols avg 39.99 mmol
```



```
Biomass range: 39.82 – 40.22 mmol
PPFD effect:   centre cols avg 40.03 mmol  vs  edge cols avg 39.99 mmol
```