Constraint-based Modeling of Cellular Networks

Exercise 5 – Flux ranges, writing functions

17. 11. 2022

To HW 3: Inital solution – Big M-method

To obtain an initial solution we add artificial variable w to row 4

Row 1: $z - 50*P - 30*C - 0*s - 0*t = 0$
Row 2: $2*P + C + s = 20$
Row 3: $P + C + t = 12$
Row 4: $C - u + w = 5$
$P,C,s,t >= 0$

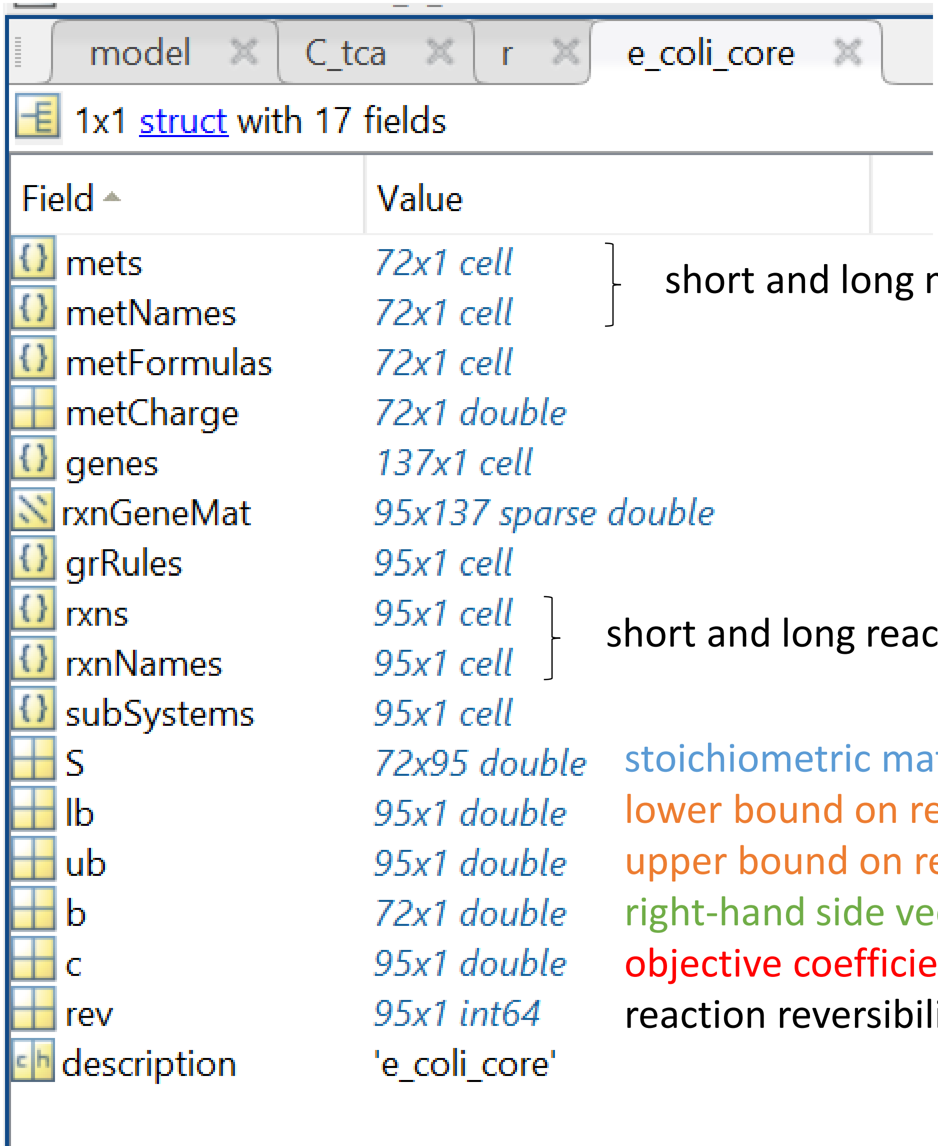
Initial feasible solution is
$P = C = u = 0$, $s = 20$, $t = 12$, $w = 5$

$w = 5 - C + u$

Only solutions with $w=0$ are feasible therefore we penalize w with large constant such that it is driven to be zero. Hence, we change objective to
$z - 50*P - 30*C - 0*s - 0*t + 5000*w = 0$

Substitute w in objective with 5-C+u
$z-50*P-30*C-0*s-0*t+5000*(5-C+u) = z-50*P-5030*C-0*s-0*t+5000*u +25000$

Repetition Cobra Models



1x1 struct with 17 fields

Field	Value	
mets	72x1 cell	short and long metabolite names
metNames	72x1 cell	
metFormulas	72x1 cell	
metCharge	72x1 double	
genes	137x1 cell	
rxnGeneMat	95x137 sparse double	
grRules	95x1 cell	
rxns	95x1 cell	short and long reaction names
rxnNames	95x1 cell	
subSystems	95x1 cell	
S	72x95 double	stoichiometric matrix
lb	95x1 double	lower bound on reaction flux
ub	95x1 double	upper bound on reaction flux
b	72x1 double	right-hand side vector
c	95x1 double	objective coefficients
rev	95x1 int64	reaction reversibility
description	'e_coli_core'	

Flux balance analysis

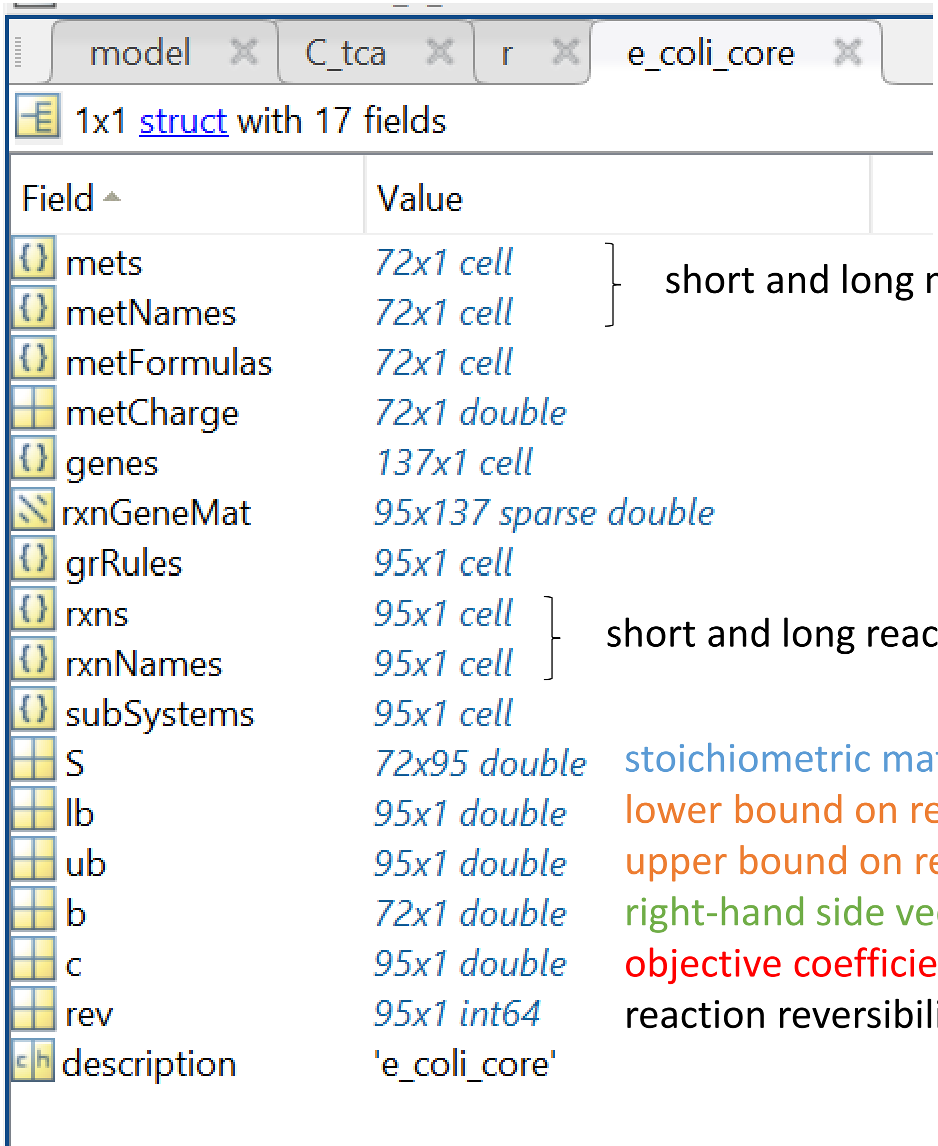
$$\max c^T v$$

s.t.

$$Sv = b$$

$$lb \leq v \leq ub$$

Repetition Cobra Models



1x1 struct with 17 fields

Field	Value	
mets	72x1 cell	short and long metabolite names
metNames	72x1 cell	
metFormulas	72x1 cell	
metCharge	72x1 double	
genes	137x1 cell	
rxnGeneMat	95x137 sparse double	
grRules	95x1 cell	
rxns	95x1 cell	short and long reaction names
rxnNames	95x1 cell	
subSystems	95x1 cell	
S	72x95 double	stoichiometric matrix
lb	95x1 double	lower bound on reaction flux
ub	95x1 double	upper bound on reaction flux
b	72x1 double	right-hand side vector
c	95x1 double	objective coefficients
rev	95x1 int64	reaction reversibility
description	'e_coli_core'	

Flux balance analysis

$$\max c^T v$$

s.t.

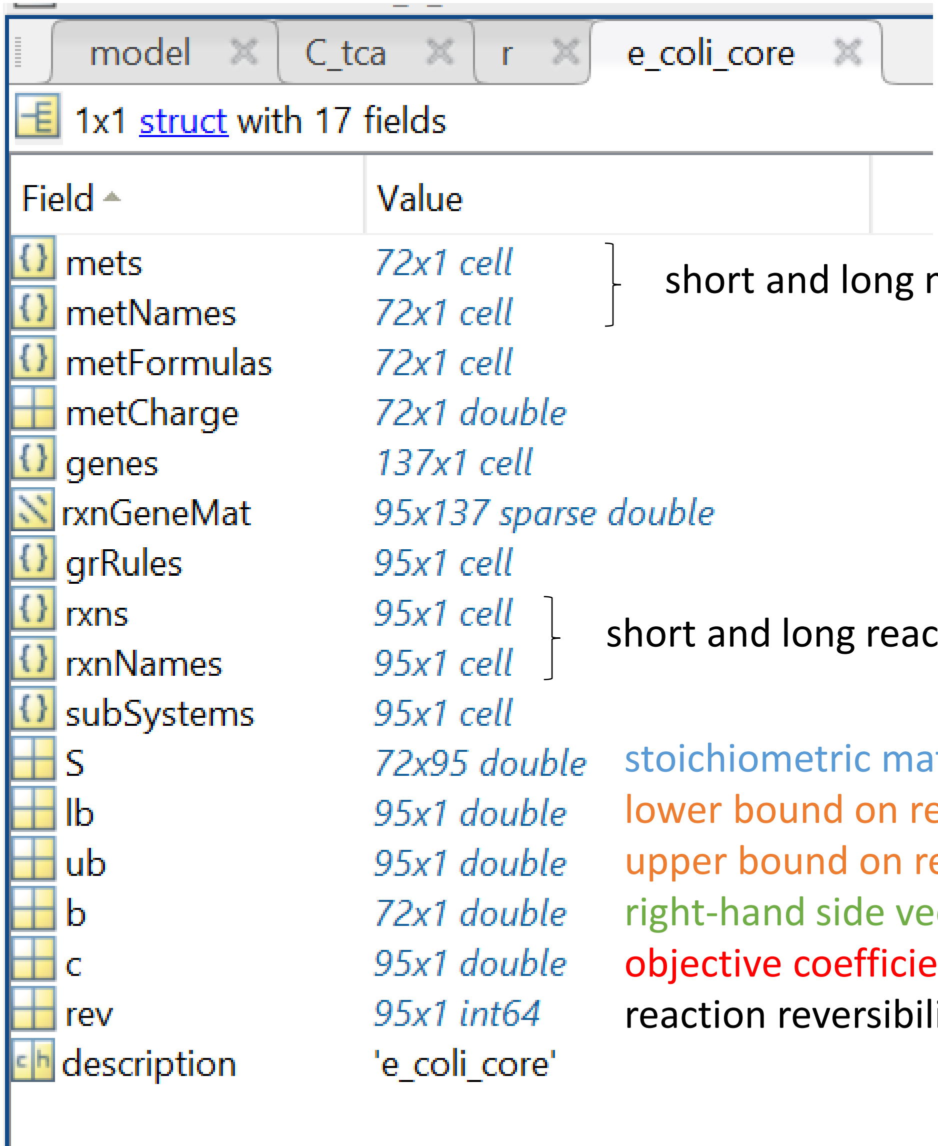
$$Sv = b$$
$$lb \le v \le ub$$

- Which reaction flux is optimized?

```
e_coli_core.rxnNames(e_coli_core.c~=0)
```

```
ans = 1x1 cell array
    {'Biomass Objective Function with GAM'}
```

Repetition Cobra Models



1x1 struct with 17 fields

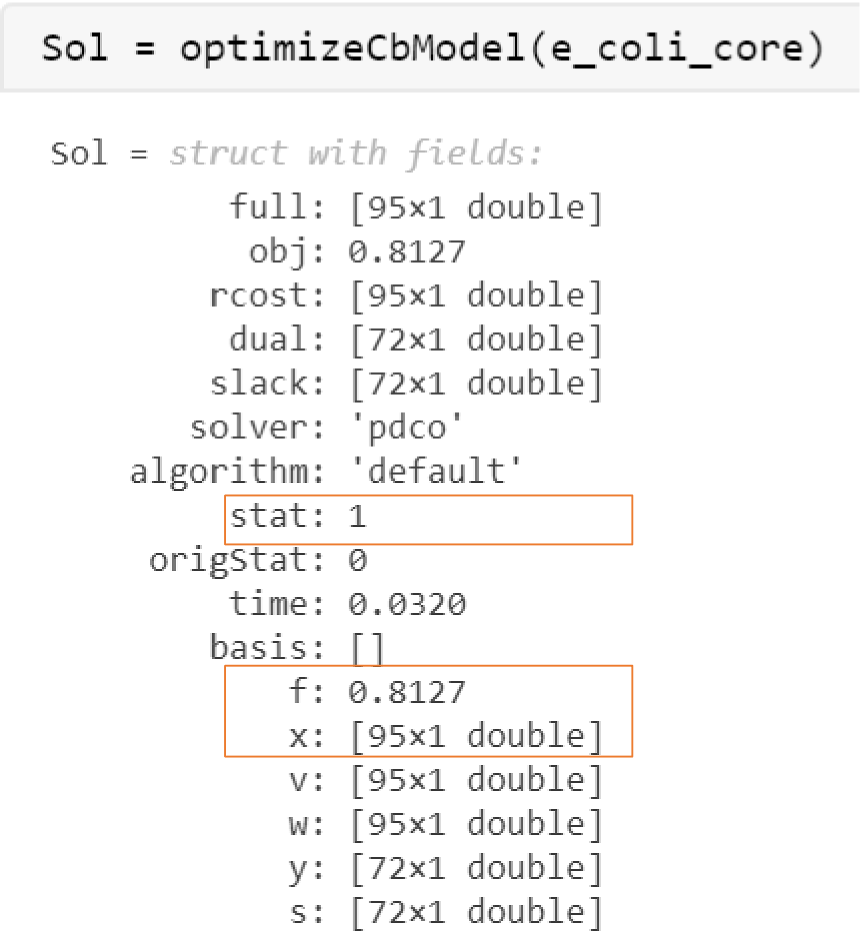
Field	Value	
mets	72x1 cell	short and long metabolite names
metNames	72x1 cell	
metFormulas	72x1 cell	
metCharge	72x1 double	
genes	137x1 cell	
rxnGeneMat	95x137 sparse double	
grRules	95x1 cell	
rxns	95x1 cell	short and long reaction names
rxnNames	95x1 cell	
subSystems	95x1 cell	
S	72x95 double	stoichiometric matrix
lb	95x1 double	lower bound on reaction flux
ub	95x1 double	upper bound on reaction flux
b	72x1 double	right-hand side vector
c	95x1 double	objective coefficients
rev	95x1 int64	reaction reversibility
description	'e_coli_core'	

Flux balance analysis

$$\max c^T v$$

s.t.

$$Sv = b$$
$$lb \leq v \leq ub$$

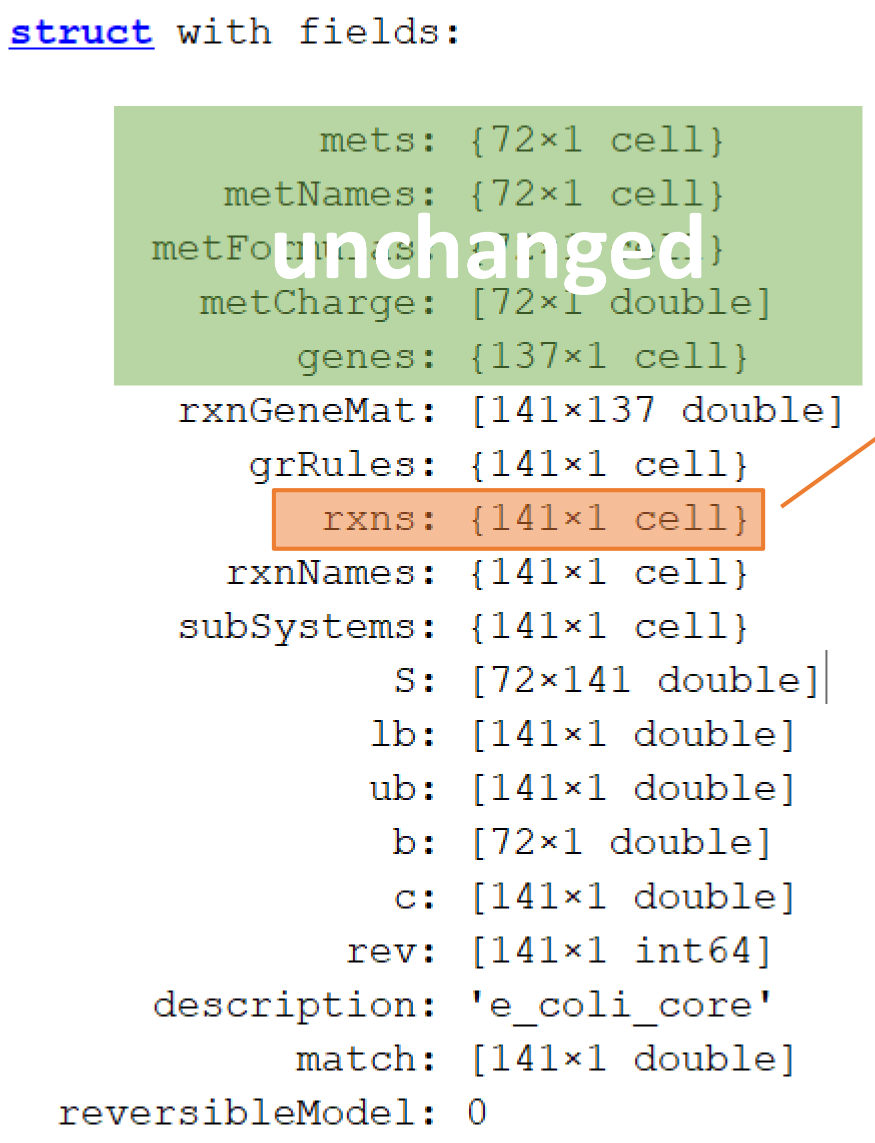


```
Sol = optimizeCbModel(e_coli_core)
```

```
Sol = struct with fields:

       full: [95×1 double]
        obj: 0.8127
      rcost: [95×1 double]
       dual: [72×1 double]
      slack: [72×1 double]
     solver: 'pdco'
  algorithm: 'default'
       stat: 1
   origStat: 0
       time: 0.0320
      basis: []
          f: 0.8127
          x: [95×1 double]
          v: [95×1 double]
          w: [95×1 double]
          y: [72×1 double]
          s: [72×1 double]
```

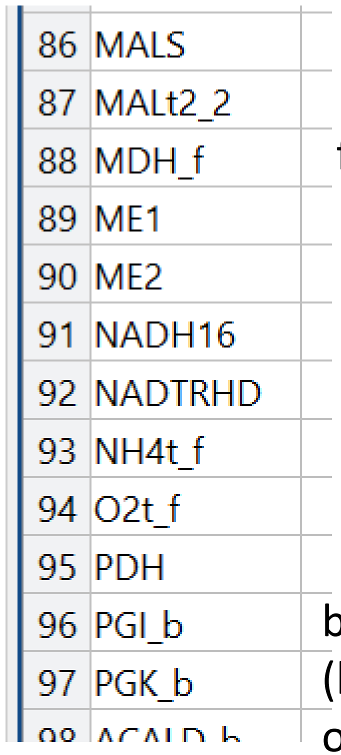
ConvertToIrreversible(e_coli_core)



```
struct with fields:

            mets: {72×1 cell}
        metNames: {72×1 cell}
     metFormulas: {72×1 cell}
       metCharge: [72×1 double]
           genes: {137×1 cell}
      rxnGeneMat: [141×137 double]
         grRules: {141×1 cell}
            rxns: {141×1 cell}
        rxnNames: {141×1 cell}
      subSystems: {141×1 cell}
               S: [72×141 double]
              lb: [141×1 double]
              ub: [141×1 double]
               b: [72×1 double]
               c: [141×1 double]
             rev: [141×1 int64]
     description: 'e_coli_core'
           match: [141×1 double]
 reversibleModel: 0
```

unchanged

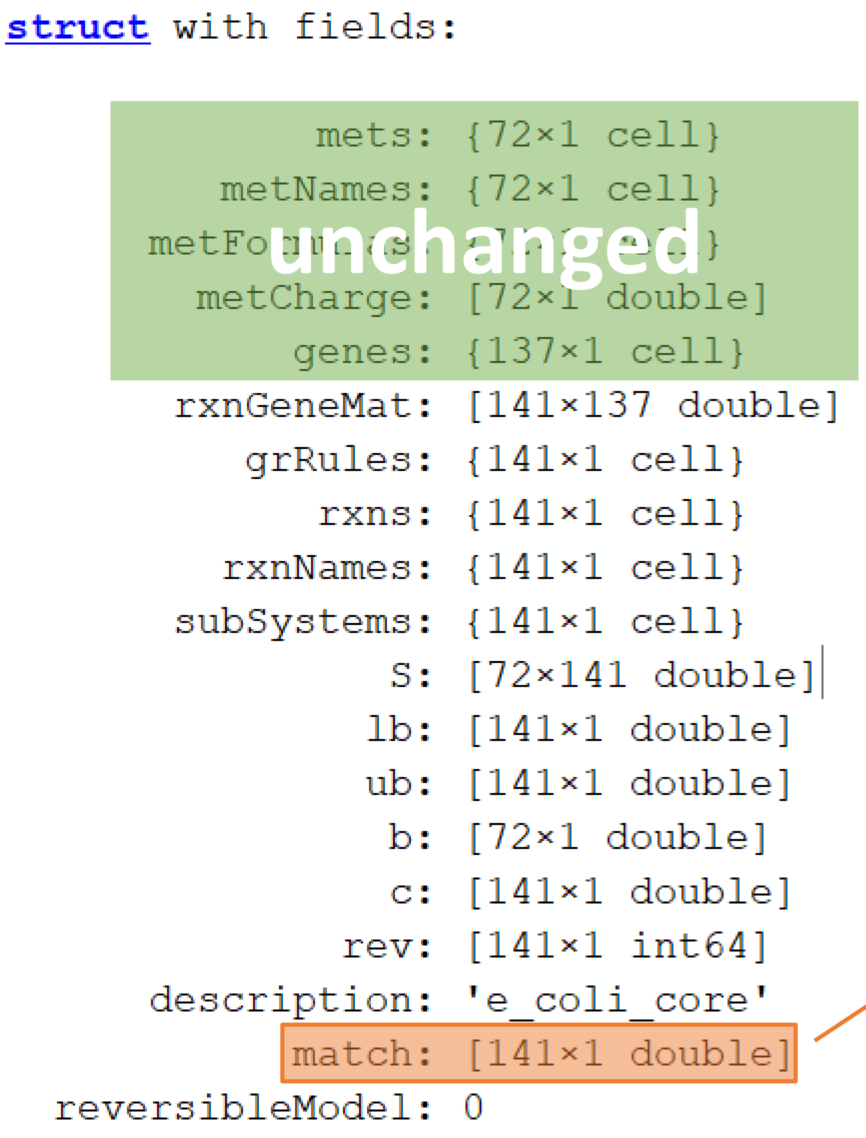


86	MALS
87	MALt2_2
88	MDH_f
89	ME1
90	ME2
91	NADH16
92	NADTRHD
93	NH4t_f
94	O2t_f
95	PDH
96	PGI_b
97	PGK_b
98	ACALD_b

foreward direction indicated by _f

backward direction indicated by _b (backward reactions appended to the end of original stoichiometric matrix)

ConvertToIrreversible(e_coli_core)



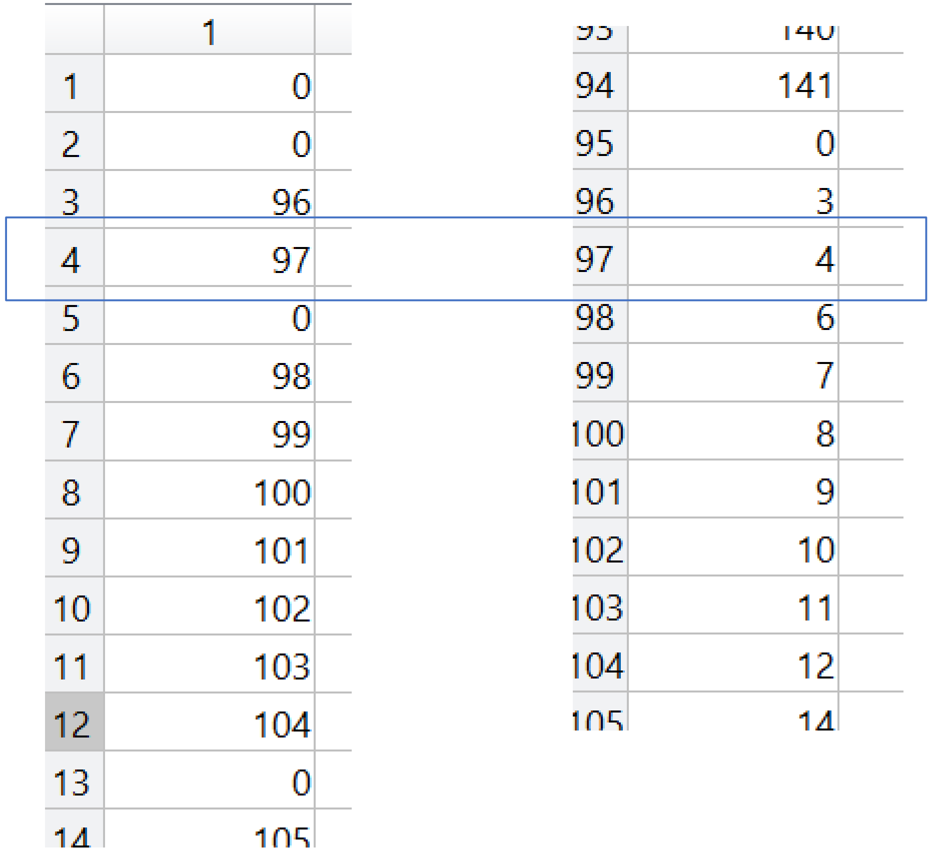
```
struct with fields:

            mets: {72×1 cell}
        metNames: {72×1 cell}
     metFormulas: {72×1 cell}
       metCharge: [72×1 double]
           genes: {137×1 cell}
      rxnGeneMat: [141×137 double]
         grRules: {141×1 cell}
            rxns: {141×1 cell}
        rxnNames: {141×1 cell}
      subSystems: {141×1 cell}
               S: [72×141 double]
              lb: [141×1 double]
              ub: [141×1 double]
               b: [72×1 double]
               c: [141×1 double]
             rev: [141×1 int64]
     description: 'e_coli_core'
           match: [141×1 double]
 reversibleModel: 0
```

unchanged

You can see that...

- backward flux of rxn 4 is rxn 97
- equally we see that rxn 4 is foreward flux of rxn 97



	1
1	0
2	0
3	96
4	97
5	0
6	98
7	99
8	100
9	101
10	102
11	103
12	104
13	0
14	105

93	140
94	141
95	0
96	3
97	4
98	6
99	7
100	8
101	9
102	10
103	11
104	12
105	14

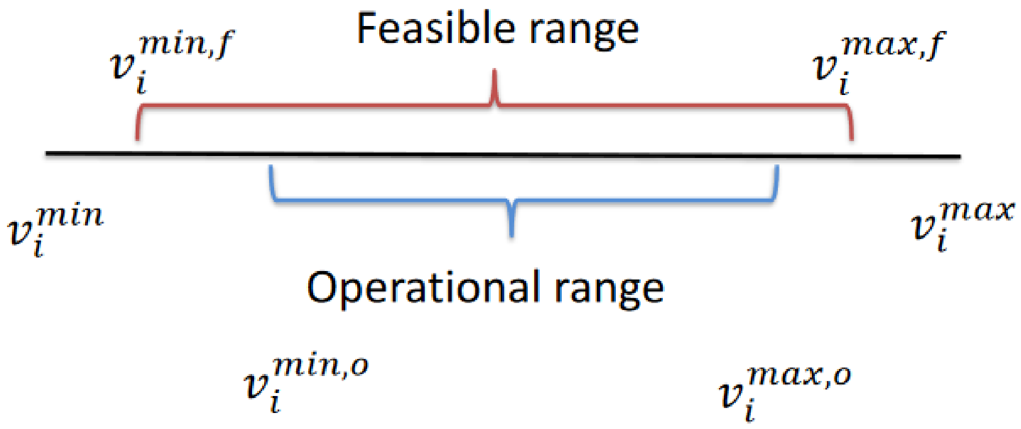
Flux ranges

Q: What is the difference between

- Generic flux bounds
- Feasible range
- Operational range

- Q: What is a blocked reaction?

Illustration

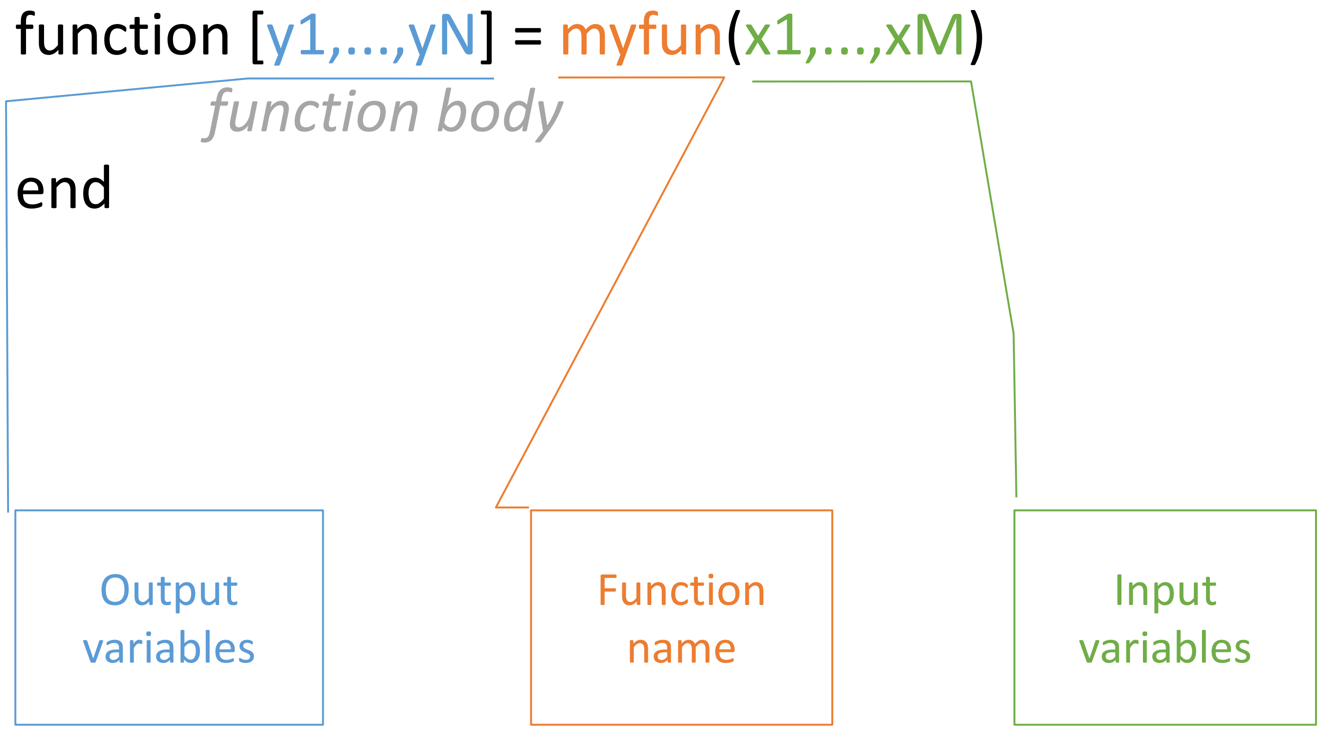


Task for today

Write matlab function to calculate feasible flux range.

How do we write functions in MATLAB?

Matlab functions



```
function [y1,...,yN] = myfun(x1,...,xM)
        function body
end
```

Output variables

Function name

Input variables

How to save functions

function file (global)

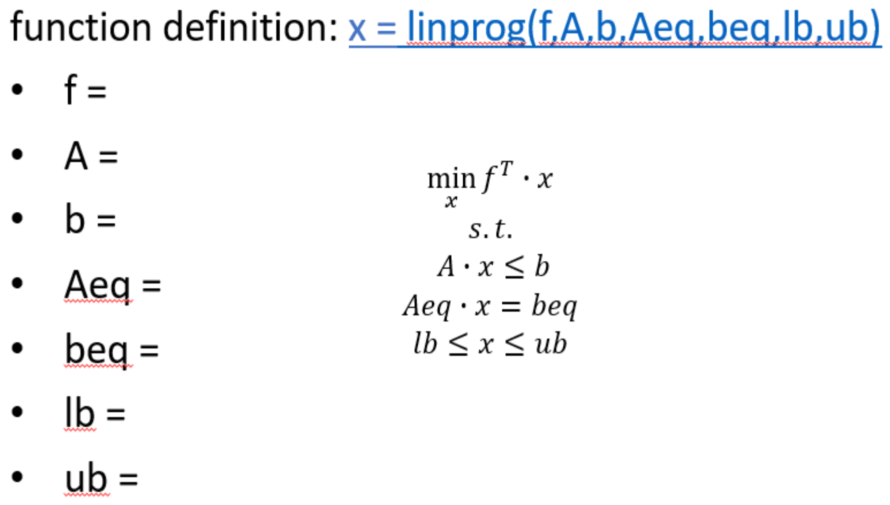
- contains only function definition, name of the file must match the name of the function in the file

in a script file with commands (local)

- functions must be at the end of the file
- Script files cannot have the same name as a function in the file

[minimum_flux,maximum_flux] = FluxRange(M,w)

```
for each j in w
    do solve min/max FBA:
    minimum_flux(j)    =    min $v_j$
                            s.t.
                            Nv=0
                            lb<=v<=ub
    maximum_flux(j)    =    max $v_j$
                            s.t.
                            Nv=0
                            lb<=v<=ub
end
```



function definition: x = linprog(f,A,b,Aeq,beq,lb,ub)

- f =
- A =
- b =
- Aeq =
- beq =
- lb =
- ub =

$$\min_{x} f^T \cdot x$$
$$s.t.$$
$$A \cdot x \le b$$
$$Aeq \cdot x = beq$$
$$lb \le x \le ub$$