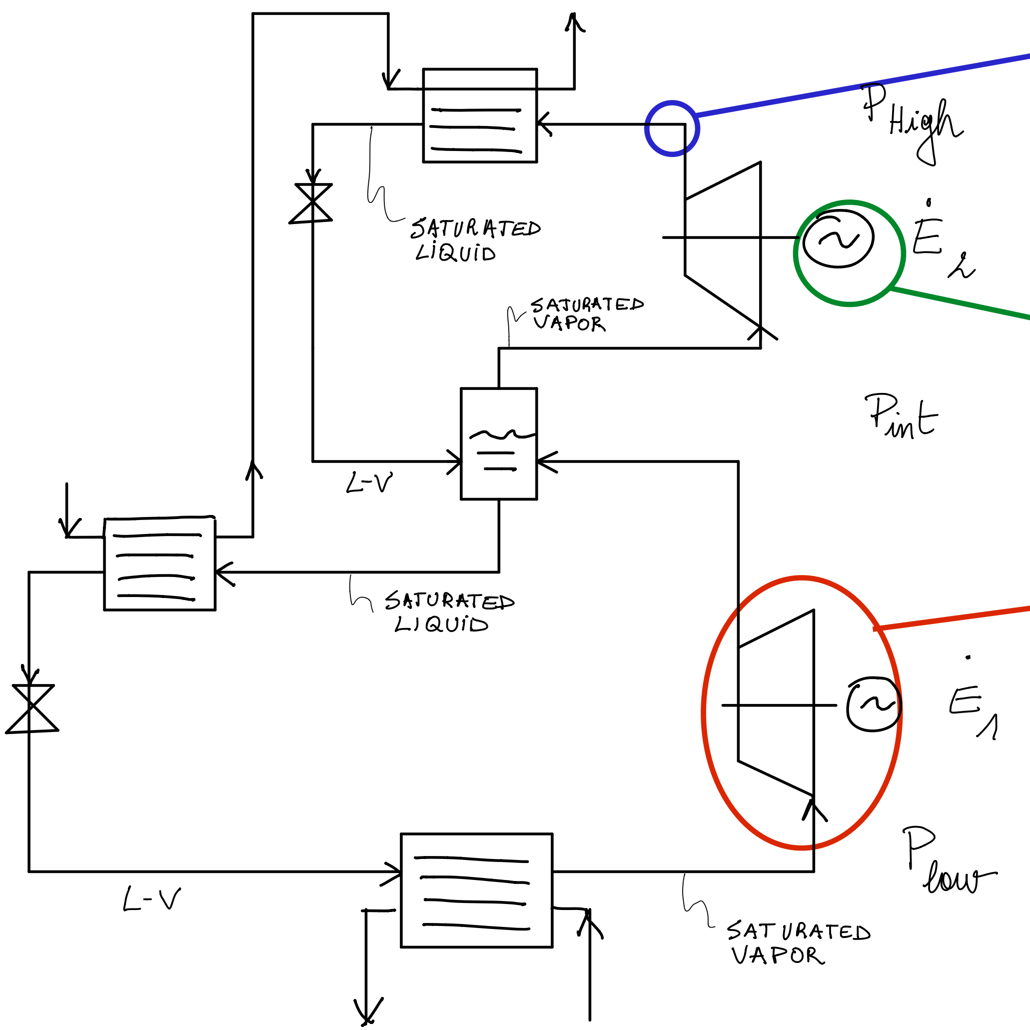
# Solving flowsheet

# Double stage heat pump Flowsheet






**Flows** interconnecting units
- Flowrate
- Thermodynamic State
  T, P, X

Power

**Process unit :** converting flows
- Size
- Type
- Conversion of mass and energy

# Simultaneous resolution

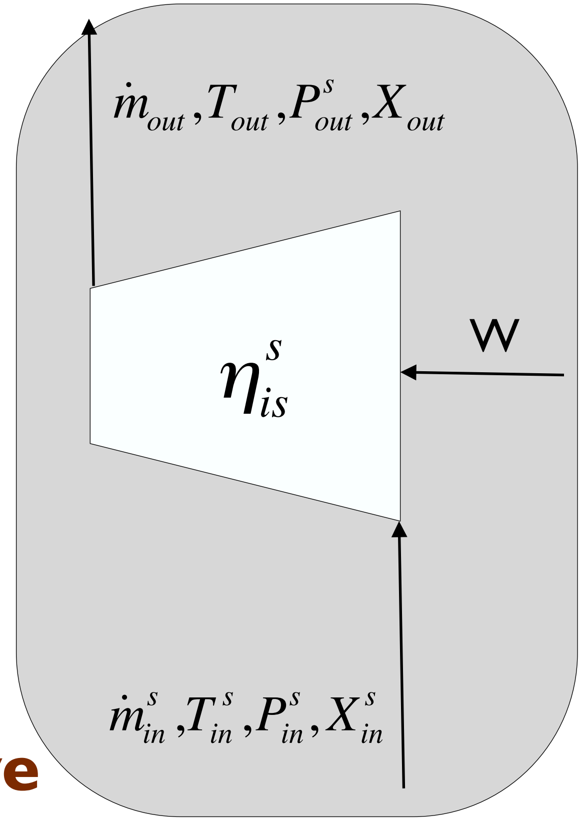
- compressor model

**Model**

$$\dot{m}_{out} - \dot{m}_{in} = 0$$
$$x^{j}_{out} - x^{j}_{in} = 0$$
$$h_{in} - h(T_{in}, P_{in}, X_{in}) = 0$$
$$s_{in} - s(T_{in}, P_{in}, X_{in}) = 0$$
$$h^{is}_{out} - h(s_{in}, P_{out}, X_{out}) = 0$$
$$h_{out} - h_{in} + \eta_{is} * \left(h_{in} - h^{is}_{out}\right) = 0$$
$$T_{out} - T\left(h_{out}, P_{out}, X_{out}\right) = 0$$
$$W - \dot{m}_{in} * (h_{in} - h_{out}) = 0$$

**Specifications**

$$\dot{m}_{in} - \dot{m}^{s}_{in} = 0$$
$$x^{j}_{in} - x^{j,s}_{in} = 0$$
$$T_{in} - T^{s}_{in} = 0$$
$$P_{in} - P^{s}_{in} = 0$$
$$P_{out} - P^{s}_{out} = 0$$
$$\eta_{is} - \eta^{s}_{is} = 0$$




**constitutive equations**

Non linear system resolution using Newton Raphson methods

$$F(X,P) = 0$$
$$S(X,P) = 0 \quad \Rightarrow F(X) = 0 \quad (NxN)$$
$$IN(X) = 0$$

# Sequential resolution : order of resolution

- Compressor Model

$$\dot{m}_{out} = \dot{m}_{in}^{s}$$

$$x_{out}^{j} = x_{in}^{j,s}$$

$$h_{in} = h(T_{in}^{s}, P_{in}^{s}, X_{in}^{s})$$

$$s_{in} = s(T_{in}^{s}, P_{in}^{s}, X_{in}^{s})$$

$$h_{out}^{is} = h(s_{in}, P_{out}^{s}, X_{out})$$

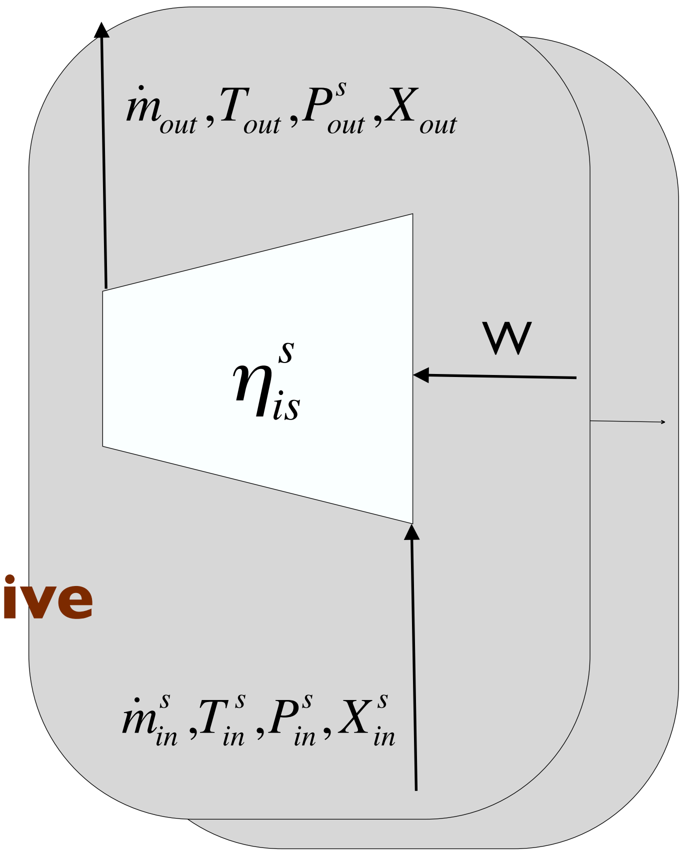
$$h_{out} = h_{in} - \eta_{is}^{s} * \left(h_{in} - h_{out}^{is}\right)$$

$$T_{out} = T\left(h_{out}, P_{out}^{s}, X_{out}\right)$$

$$W = \dot{m}_{in}^{s} * (h_{in} - h_{out})$$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| mout | x | | | | | | | |
| xout | | x | | | | | | |
| Hin | | | x | | | | | |
| Sin | | | | x | | | | |
| Houtis | | x | | x | x | | | |
| Hout | | | x | | x | x | | |
| Tout | | x | | | | x | x | |
| W | | | x | | | x | | x |

**Constitutive equation**




**Incidence matrix is arranged to be diagonal inferior**



EPFL

# Sequence definition : Value i by Equation j

Explicit form $diag_i = f(diag_k, k = 1, \ldots, i-1)$

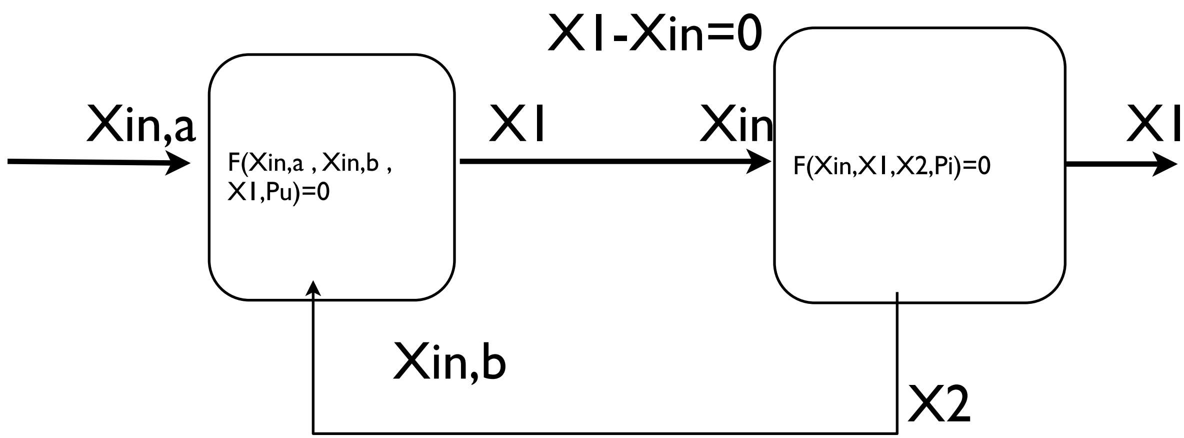
```
Eq1  -> min     x
Eq2  -> xin      x
Eq3  -> Tin       x
Eq4  -> Pin        x
Eq5  -> Pout        x
Eq6  -> Etais        x
Eq7  -> mout    x     x
Eq8  -> xout     x     x
Eq9  -> Hin      xx     x
Eq10-> Sin       xx      x
Eq11-> Houtis       x    xx
Eq12-> Hout         x x    x
Eq13-> Tout         x      xx
Eq14-> W        x   x       xx
```

Solving sequence

# Solving flowsheets with a sequential and modular approach

# Solving Flowsheets : simultaneous approach

- Flowsheet = interconnected modules
  - Simultaneous resolution : needs initial values !




F(X,P)=0 : Models

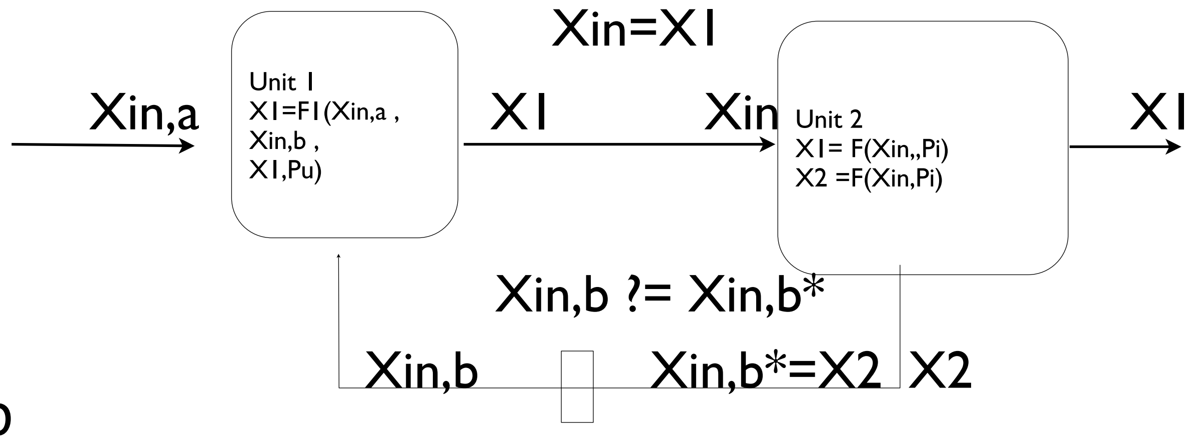
Xs-X*=0 : Specifications (system)

Ps-P=0 : Specifications (system)

Xi-Xj=0 : Links (unit interconnections)

# Solving Flowsheets : sequential

- Flowsheet = interconnected modules
  - Sequential



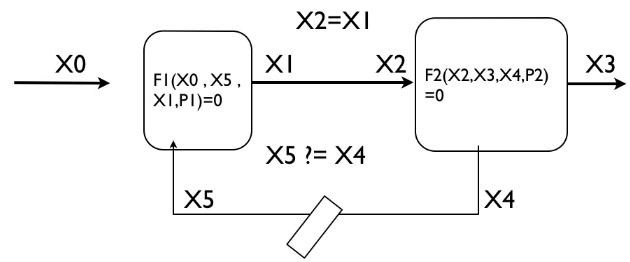

0. Locate Xin,b
1. Guess Xin,b
   2. Solve unit 1 => X1
   3. Solve unit 2 => X2
   4. Test Xin,b ?= X2
      yes => out
   5. Propose a new value to Xin,b
   & Goto 2.

# Defining a resolution sequence

Solving flowsheets
Sequential Modular Approach

# Conclusions

- Flow sheet can be solved
  - simultaneously
  - in sequence

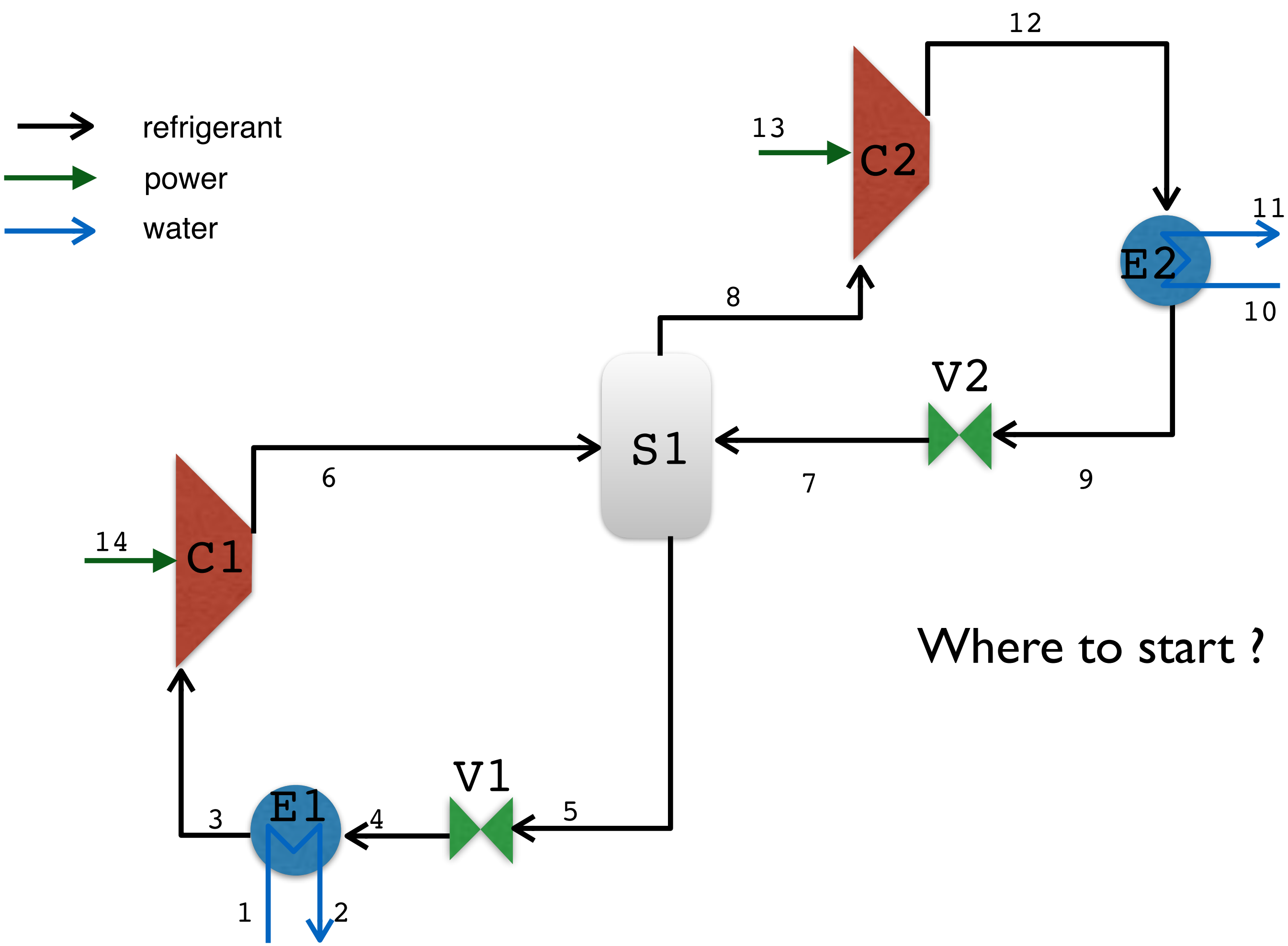



- Open Questions
  - Defining a resolution sequence
    - define an ordered list of equation solving
    - Identify loops (i.e. calculated values that are needed to solve the sequence)
      - » Tear the loops and guess an initial value
      - » Iterate (do the sequence of internal loop) until convergence of the loop by solving x=f(x)
  - Find $X^0$ to solve F(X)=0
    - this is typically done by using resolution sequence :
      - » $X^0$ is built up progressively
        more $X^0$ are calculated while adding $F_i$ (X)=0 in the list of equations



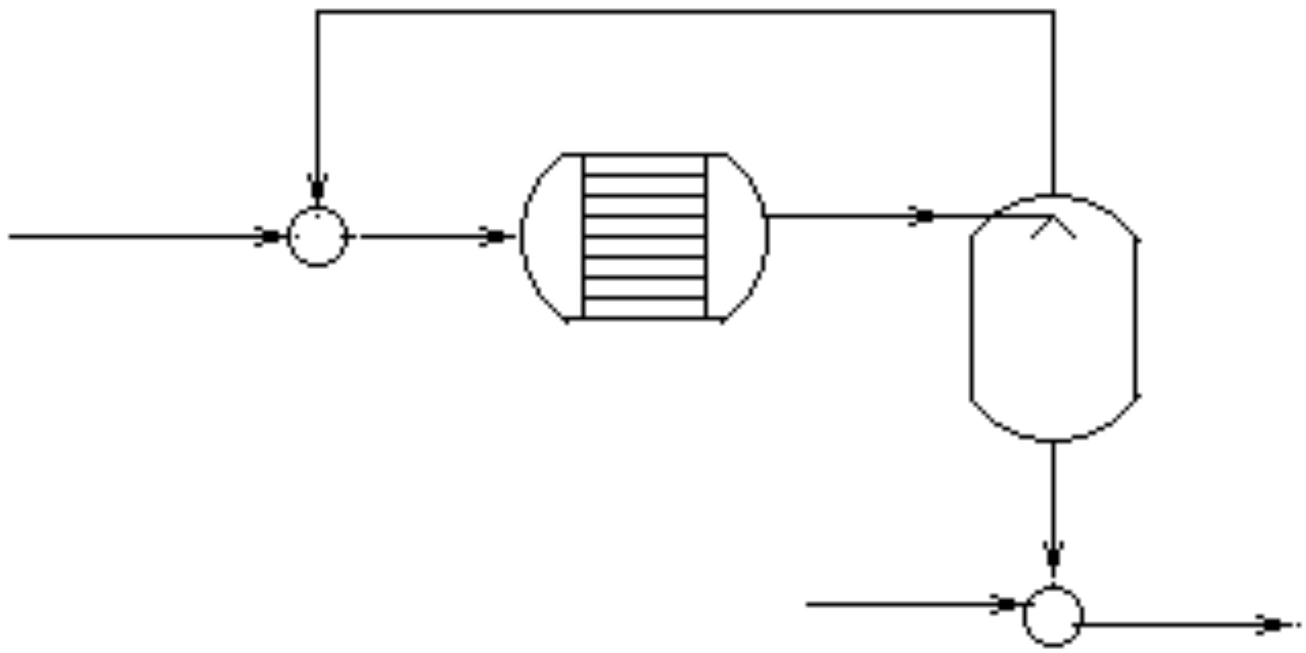
EPFL

# Two stages heat pump : initialisation




Where to start ?

# Example : Flowsheet

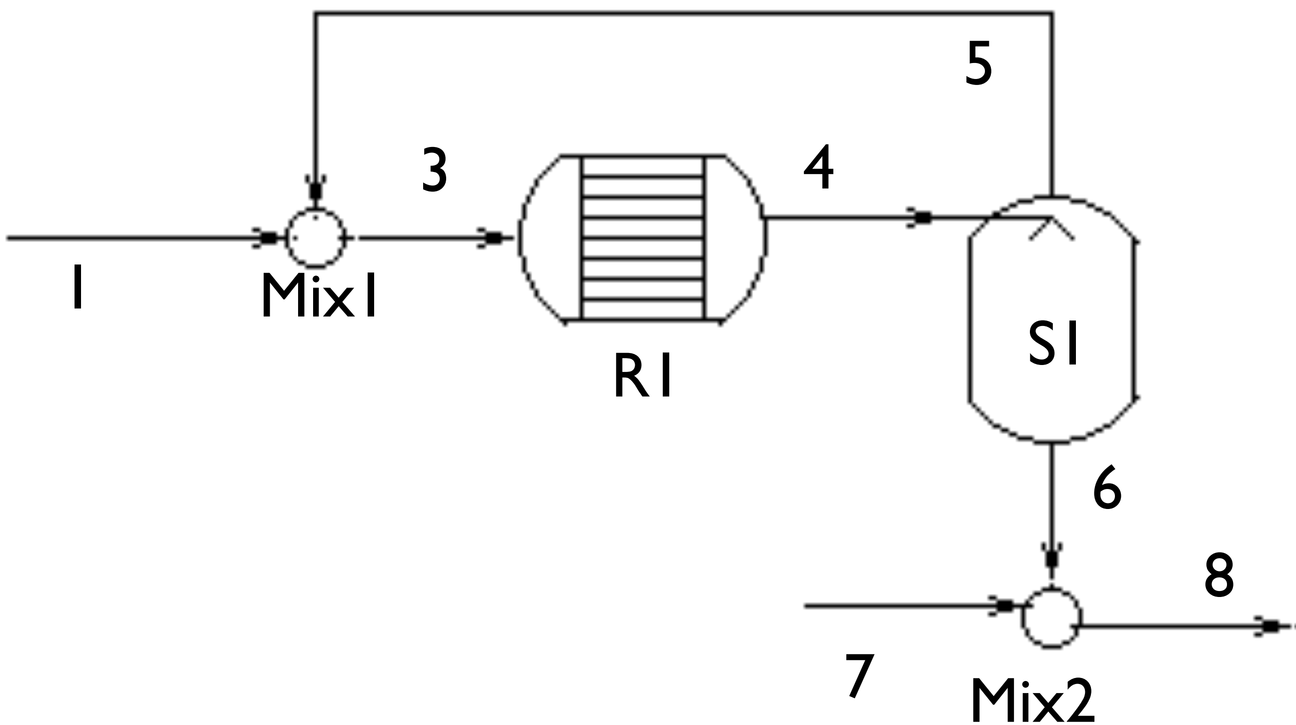




EPFL

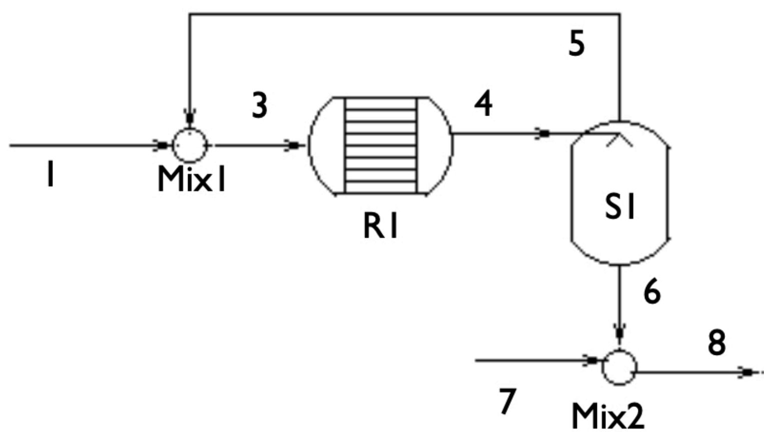
# Step 1 : describe what you want to know

- Name the streams
  - what you want to know
- Name the units - define the model of the unit operation
  - what is modeling the thermodynamic transformations in the system



# Step 11 : define a sequence

- A unit model u calculates the output knowing the input :
  $U_{out} = f_u(U_{in})$
- Order of resolution sequence
  1. set all units to unsolved and set $X_{in}$ as known
  2. 0 Identify units u that is not solved for which all the $U_{in}$ are known : For each u identified
     - 2.1. Calculate $U_{out} = f_u(U_{in})$
     - 2.2. Mark $U_{out}$ as known and u as solved
     - 2.3. Back to 2.0.

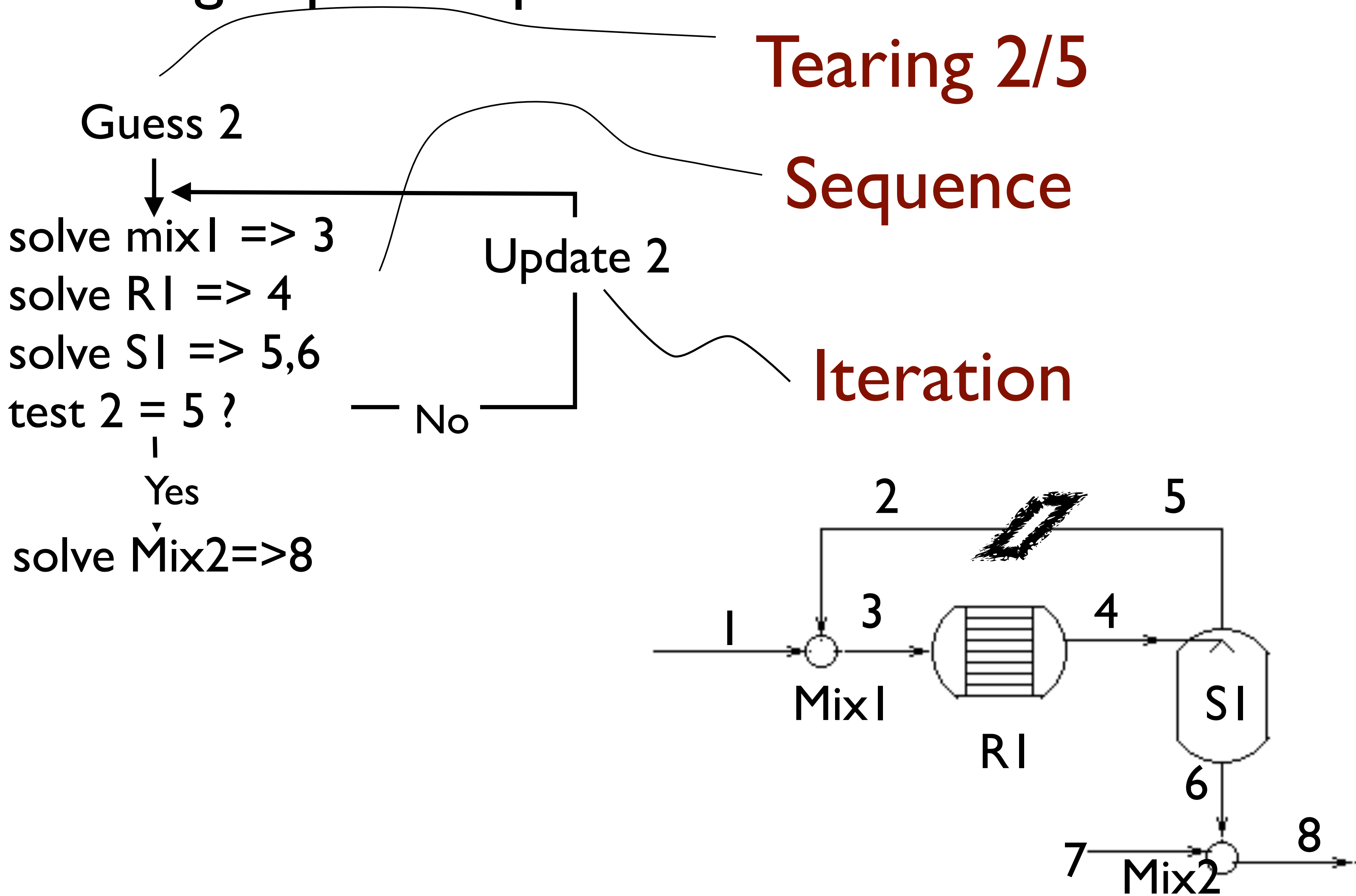





EPFL

# Solving flowsheets :sequence definition

- Tearing : open loops when units are not solved



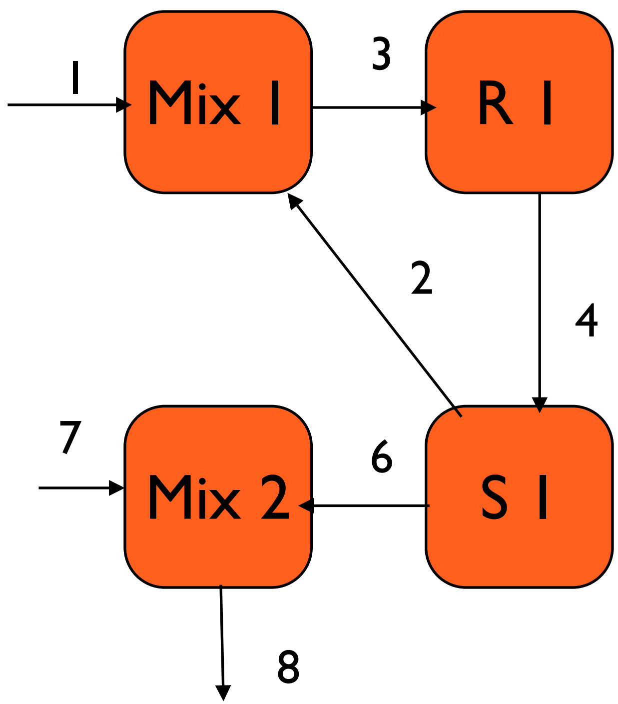
# Resolution sequence

- Defining a resolution sequence
  - Identify the teared streams (where to open the loops)
  - Define an **ordered** list of **units** to solve
  - Define the **streams** calculated by the **units**
  - Define the tears to be solved
    $X^{guess} - F(X^{guess}) = 0$

# Defining the sequence : Graph representation

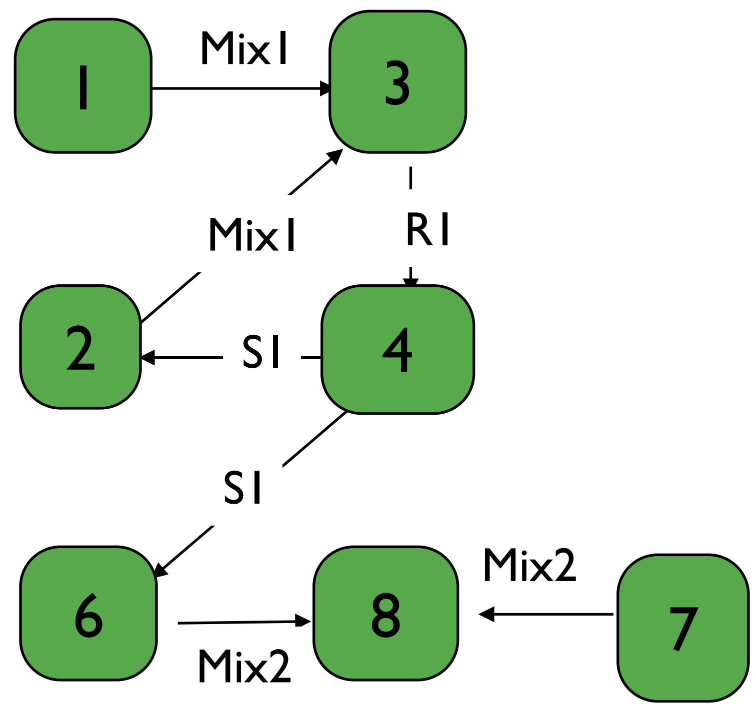
## Rheogram

### primal representation




Streams to compute units
Variables to compute equations

### dual representation

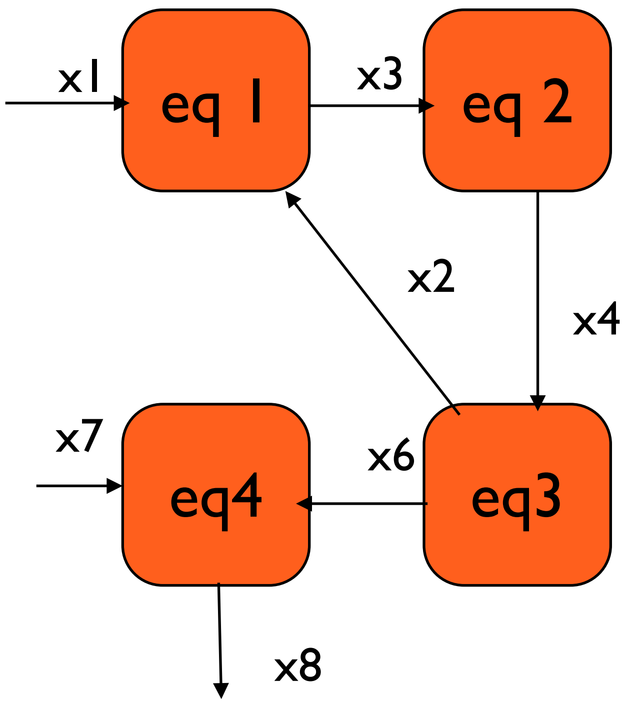



Units to compute streams
Equations to compute variables

# Systematic definition : Graph representation

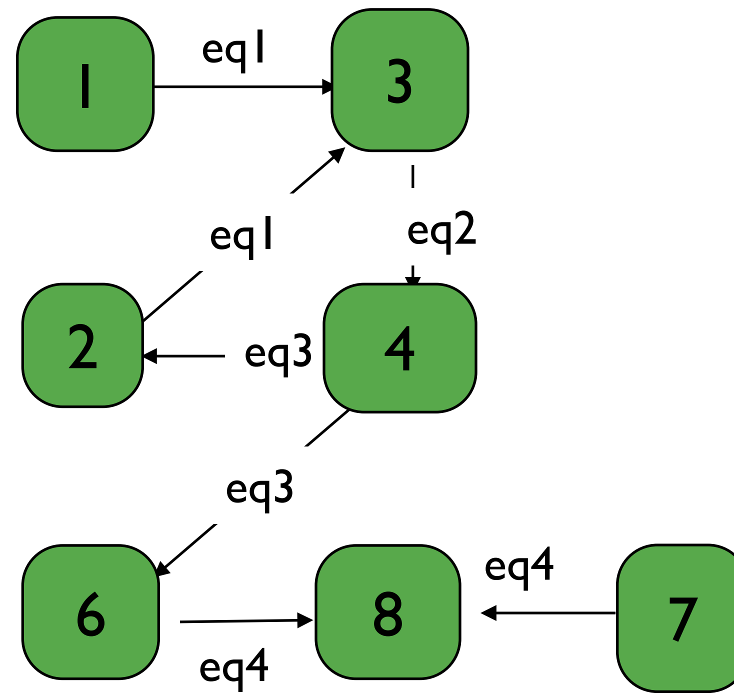
Analogy with equations set y= f(x)

primal representation




Streams to compute units
Variables to compute equations

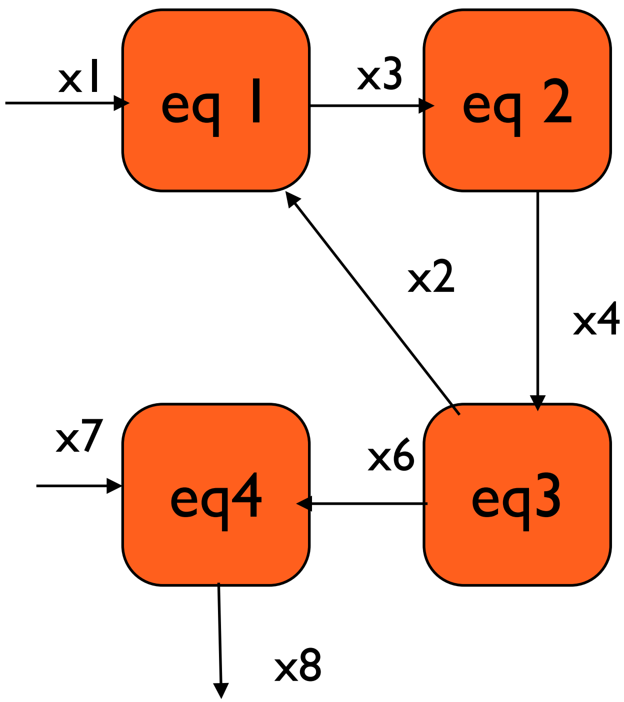
dual representation



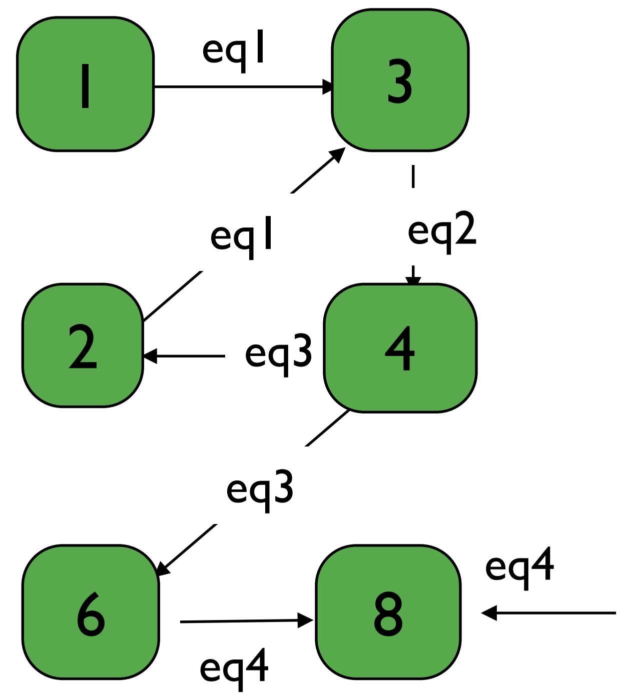

Units to compute streams
Equations to compute variables

# Defining anteriors

- x3 : needs x1 & x2 and is obtained by solving eq 1
  - x1 and x2 are anteriors of x3 : each time I see x3 I can replace it by x1,x2
- x4 : needs x3 and is obtained by solving eq 2
  - x3 is an anterior of x4 : each time I see x4 I can replace it by x3




Streams to compute units
Variables to compute equations




Units to compute streams
Equations to compute variables

=> Sequence of solving equations if anteriors are known

# Mottard Algorithm

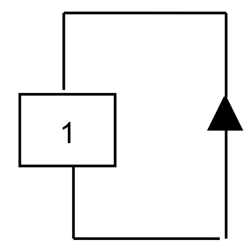
## From streams anteriority table

For each stream : what are the streams needed to compute its value by solving 1 unit

**1.- Suppress streams that have no anteriors** (you know them)

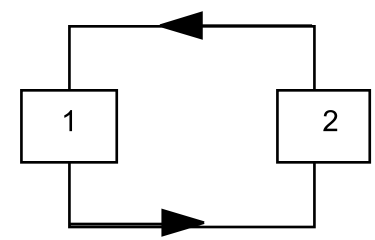
**2.- Replace the streams that have only one anterior by their anterior**

(if X3 depends only of X2 and X2 depends only of X1, X3 depends only of X1)



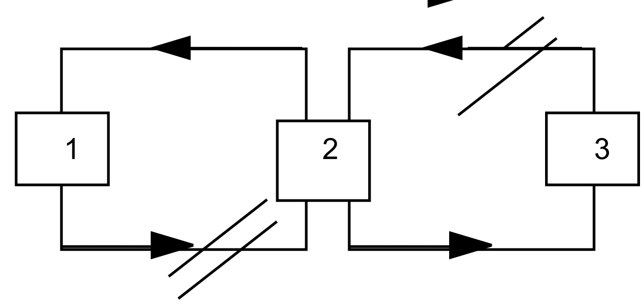

**3.- when a stream depends of himself**
**Open loop by tearing**

mark the teared stream as known =>
consider that the teared stream has no anterior back to 1.




**4.- when an anterior stream depends of his anterior**
**Open loop by tearing**

**5.- Tear streams with the highest number of anterior**

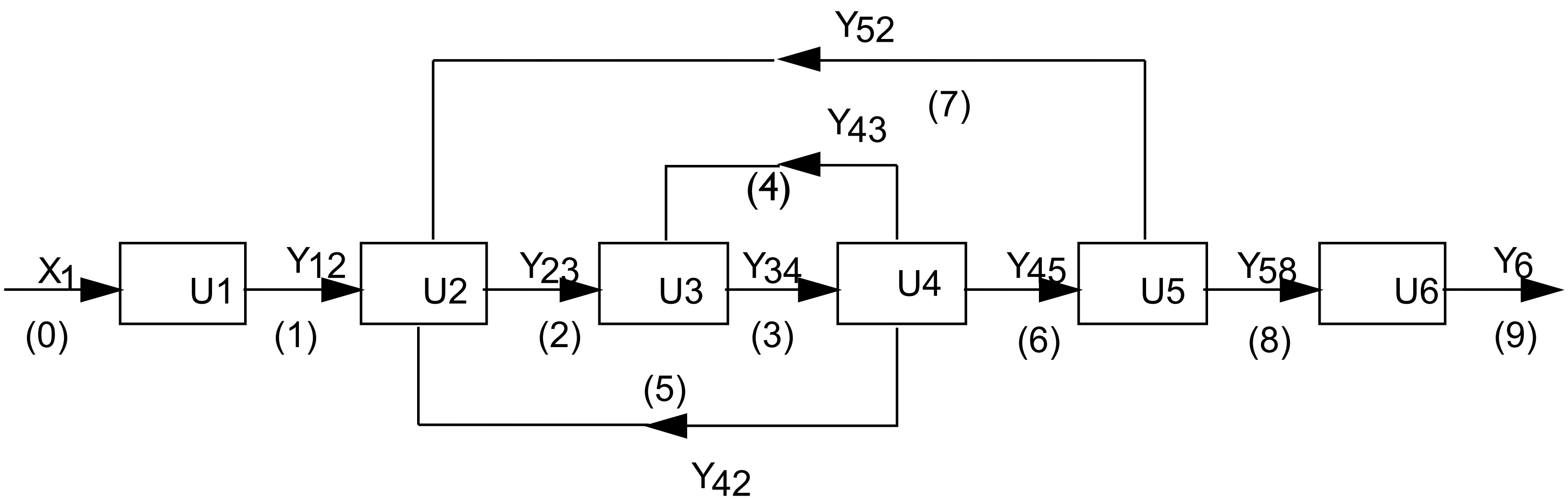



***A teared stream has no anterior***
***Guess the value and restart in 1***



EPFL

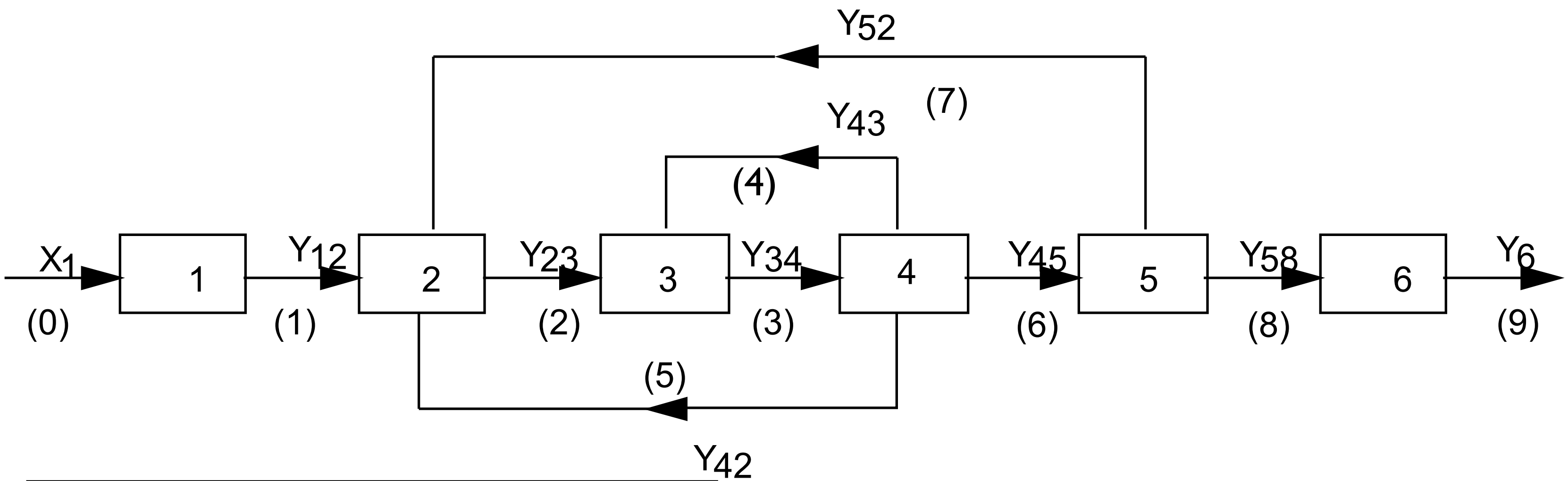
# APPLICATION OF THE MOTTARD ALGORITHM




| STREAMS | ANTERIOR STREAMS | |
|---|---|---|
| 0 | - | - |
| 1 | 0 | - |
| 2 | 1,5,7 | 1,5,7 |
| 3 | 2,4 | 2,4 |
| 4 | 3 | 3 |
| 5 | 3 | 3 |
| 6 | 3 | 3 |
| 7 | 6 | 6 |
| 8 | 6 | 6 |

| STREAMS | ANTERIOR STREAMS |
|---|---|
| 2 | 5, 7 |
| 3 | 2,4 |
| 4 | 3 |
| 5 | 3 |
| 6 | 3 |
| 7 | 6 |
| 8 | 6 |

# APPLICATION OF THE MOTTARD'S ALGORITHM



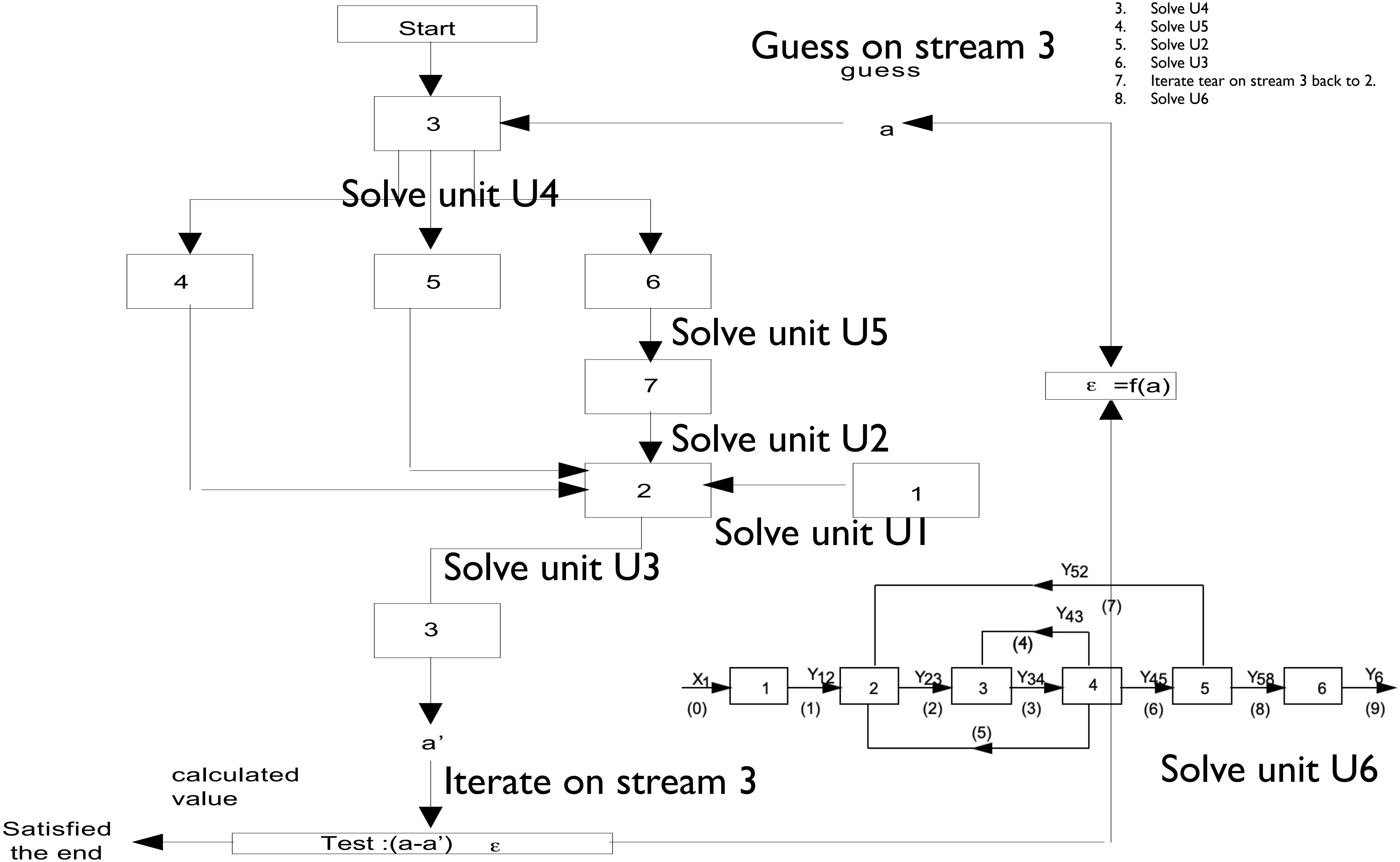

| STREAMS | ANTERIOR STREAMS | |
|---|---|---|
| 2 | 5,7 | 3,3 |
| 3 | 2,4 | 2,3 |
| 4 | 3 | 3 |
| 5 | 3 | 3 |
| 6 | 3 | 3 |
| 7 | 6 | 3 |
| 8 | 6 | 3 |

| STREAMS | ANTERIOR STREAMS |
|---|---|
| 2 | 3 |
| 3 | 2,3 |

| STREAMS | ANTERIOR STREAMS |
|---|---|
| 3 | 3 |

# Calculation sequence

1. Solve U1
2. Guess tear on stream 3
3. Solve U4
4. Solve U5
5. Solve U2
6. Solve U3
7. Iterate tear on stream 3 back to 2.
8. Solve U6



# Types of tearing streams

- **Depending on the process unit calculation model**
  - Mass flow ($N_c$ Variables)
    - e.g. if the temperature is specified
  - Temperature, Pressure (2 Variables)
    - e.g. if the flow is specified
  - Total ($N_c$+2 variables)
    - Typical case
- **Tearing Equations : (X is an array)**
  - Substitution form

$$X_{tear}^{k+1} = X_{tear}(X_{tear}^{k})$$

  - Equation form

$$X_{tear}^{k+1} - X_{tear}(X_{tear}^{k}) = 0$$

# Conclusions

- **Defining a resolution sequence**
  - Define the anterior (information needed to compute an information)
  - Eliminate what is known
  - define an ordered list of equation/unit solving
    - Identify loops (i.e. calculated values that are need to solve the sequence)
      - Guess a value
      - Iterate (do the sequence of internal loop) until convergence of the loop ($x=f(x)$)
        - » Simple substitution
        - » Newton Raphson : $x-f(x)=0$

# Some remarks

- **The units are not always out=f(in)**
  - calculation mode or specifications can sometimes calculate in=f(out)
- **Use to find the initial values for a simultaneous resolution**
  - tear convergence not needed if good initial guess
- **Understanding the problem helps to improve the method**
  - principles remain the same