Selection of variables in initialization of Modelica models

Masoud Najafi

INRIA (French national research institute on computer and control)

Rocquencourt, France

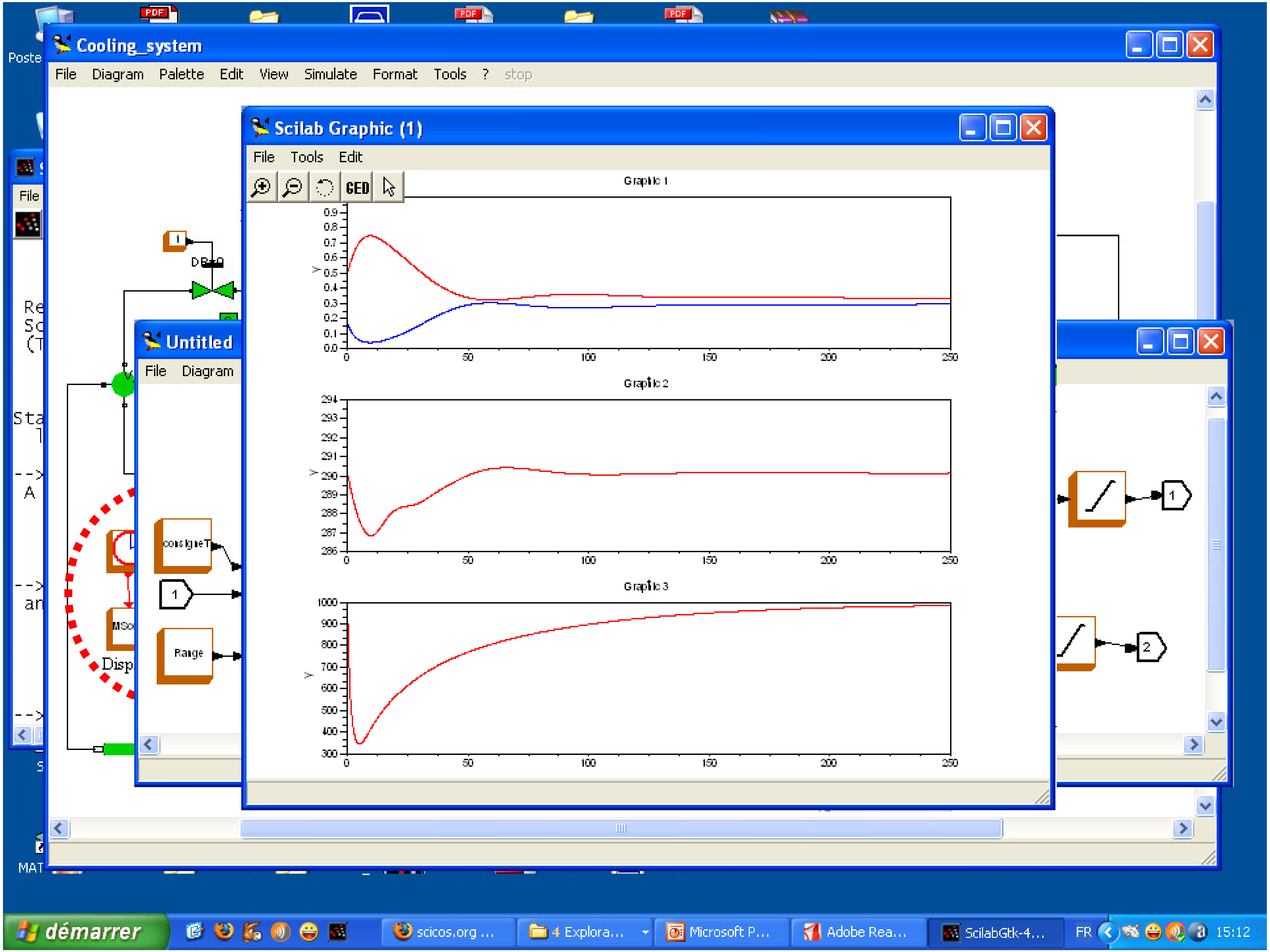
Outline

- The Modelica language in Scicos
- Initialization of Modelica models
- Model diagnostic feature
- Selection of variables

Modelica in Scicos

- Scicos is originally a simulator based on causal systems modeling, i.e., blocks with explicit inputs and outputs

- Current version of Scicos supports the Modelica language
  - Allowing acausal modeling (component based modeling)
  - Using a free and open-source Modelica compiler
  - Supporting hierarchical models
  - Supporting most essential continuous-time features (index-1 DAE)
  - A minimum support for discrete-event models


Cooling_system
File Diagram Palette Edit View Simulate Format Tools ? stop
Scilab Graphic (1)
File Tools Edit
GED
Graphic 1
y
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0 50 100 150 200 250
Graphic 2
294
293
292
291
290
289
288
287
286
0 50 100 150 200 250
Graphic 3
1000
900
800
700
600
500
400
300
0 50 100 150 200 250
Untitled
File Diagram
consigneT
1
Range
2
DB
Disp
Poste
MAT
démarrer
scicos.org ...
4 Explora...
Microsoft P...
Adobe Rea...
ScilabGtk-4...
FR
15:12

Initialization in Modelica

- Compilation of a component based model very often ends up to the simulation of a DAE.

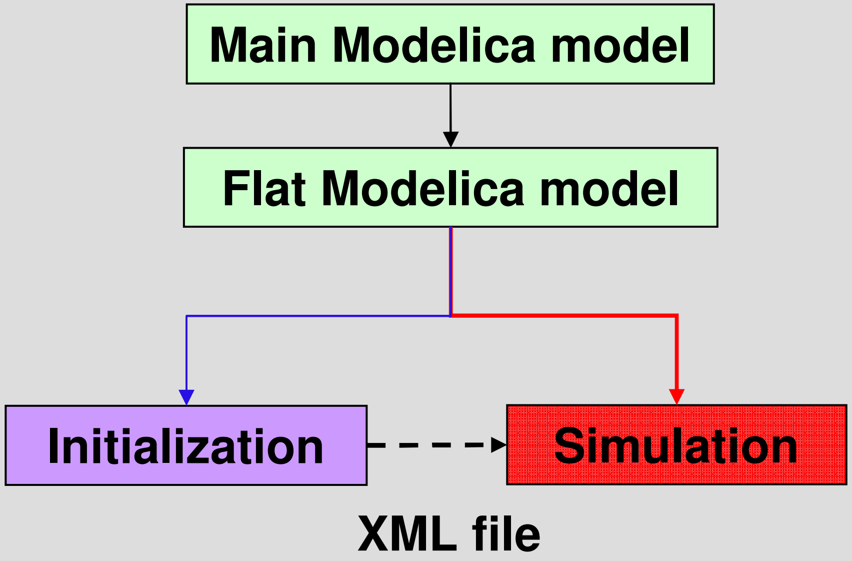
- Simulation of a DAE needs consistent initial conditions.

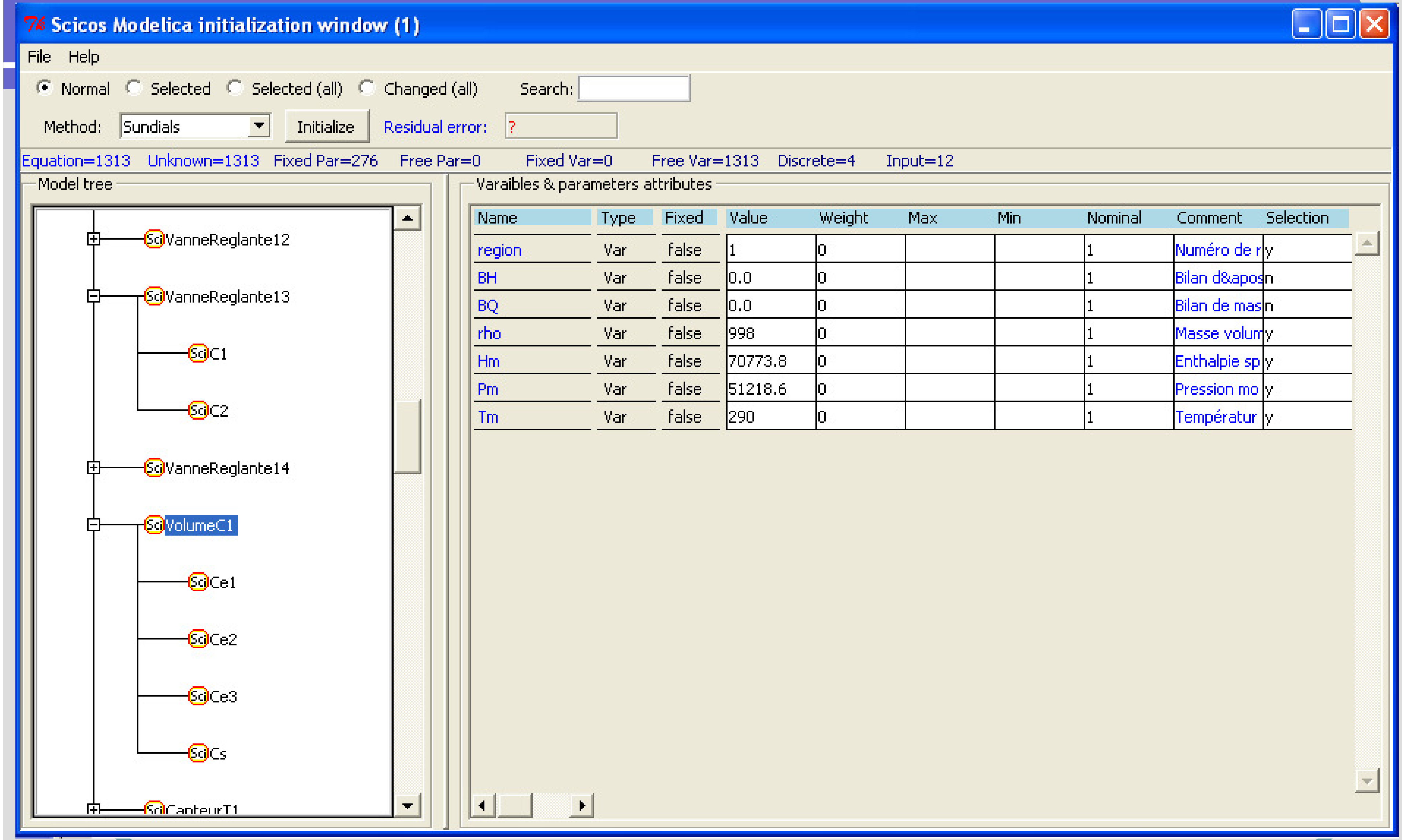
- Model initialization in Modelica:
  - "*start*" value (*initial-value/guess-value)*
  - *"fixed"* attribute (*true/false)*
  - "initial equation" section

- An initialization equation should be obtained:
  - Original DAE: F(dx/dt,x,y,u,t)=0
  - Obtained algebraic equation: G(xd,x,y,u,t0)=0

Simulation of Modelica models in Scicos

- The complete simulation is done in two stages:
  - Initialization, parameter sizing, inverse model
  - Simulation
- Initial values are computed and saved in an XML file
- A GUI as assistance in the model initialization




Scicos Modelica initialization window (1)
File
Help
Initialize
Model tree
Varaibles & parameters attributes

● Normal ○ Selected ○ Selected (all) ○ Changed (all) Search:

Method: Sundials

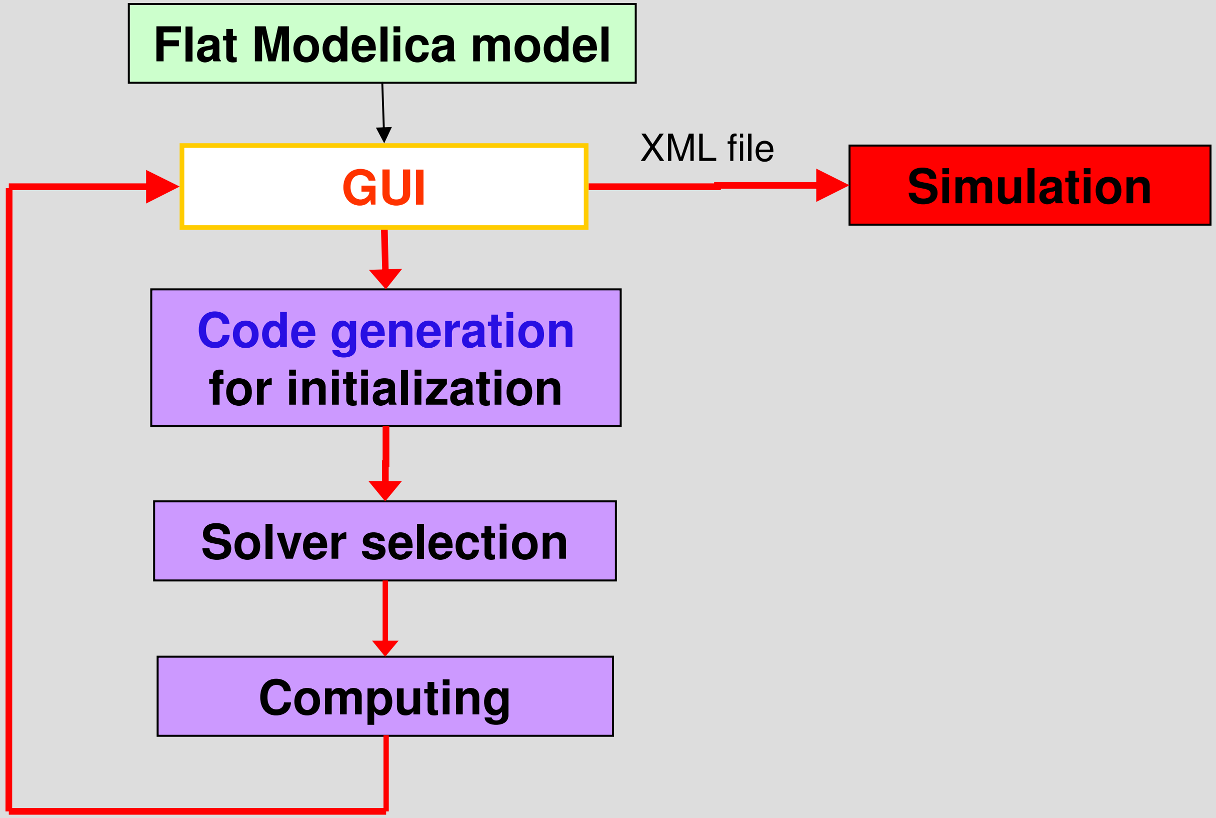
Residual error: ?

Equation=1313 Unknown=1313 Fixed Par=276 Free Par=0 Fixed Var=0 Free Var=1313 Discrete=4 Input=12

- VanneReglante12
- VanneReglante13
  - C1
  - C2
- VanneReglante14
- VolumeC1
  - Ce1
  - Ce2
  - Ce3
  - Cs
- CapteurT1

Name	Type	Fixed	Value	Weight	Max	Min	Nominal	Comment	Selection
region	Var	false	1	0			1	Numéro de r	y
BH	Var	false	0.0	0			1	Bilan d&apos	n
BQ	Var	false	0.0	0			1	Bilan de mas	n
rho	Var	false	998	0			1	Masse volum	y
Hm	Var	false	70773.8	0			1	Enthalpie sp	y
Pm	Var	false	51218.6	0			1	Pression mo	y
Tm	Var	false	290	0			1	Températur	y

Initialization flowchart (simplified)



GUI for user assistance

- Solving the initialization equation of medium size engineering models is not easy even for index-1 DAEs.
- GUI can help as a diagnostic tool.
- The structure matrix is a rich information source.
- Modeling diagnostics implemented in GUI:
  1) Fixing and relaxing variables/parameters of the model
  2) Influencing the Modelica compiler for code generation

Closing the degree of the freedom

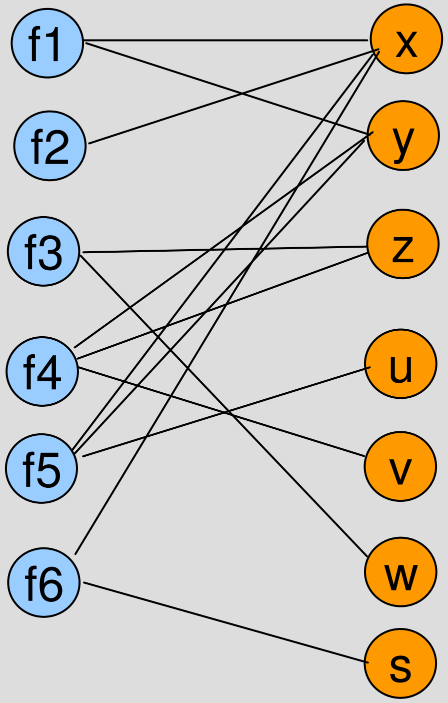
- The initialization equation may be under-determined.
- In the GUI, the user can fix or relax variables/parameters.
- Verifying if a variable/parameter can be fixed or not.
- If the model becomes Over/Under-determined:
  - If O.D., which variables/parameter should be relaxed?
  - If U.D., which variables/parameter should be fixed?

Implementation in Scicos

- Constructing an undirected bipartite graph, e.g.,

f1(x,y)=0
f2(x)=0
f3(z,w)=0
f4(y,z,v)=0
f5(x,y,u)=0
f6(x,s)=0

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



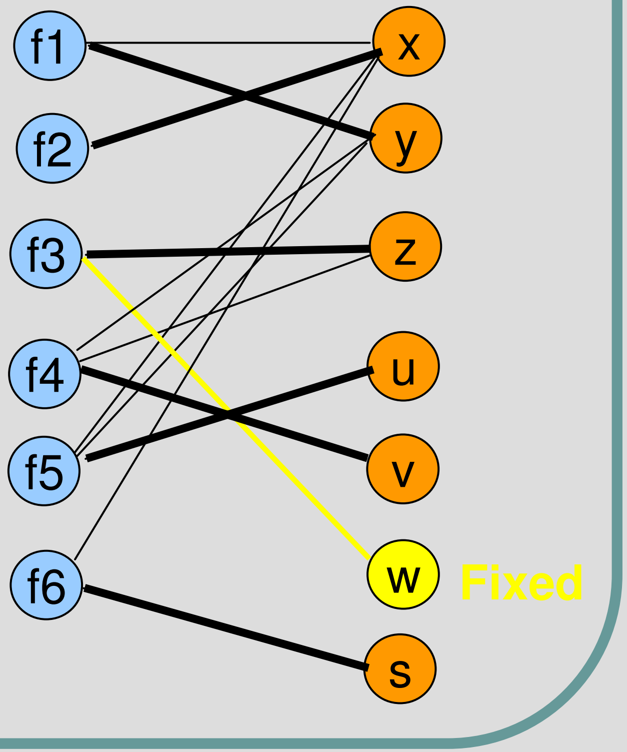
Implementation in Scicos

- Whenever the user tries to fix a variable:
  - Computing the new rank of the structure matrix using the maximum matching method.

f1(x,y)=0
f2(x)=0
f3(z,w)=0
f4(y,z,v)=0
f5(x,y,u)=0
f6(x,s)=0

Perfect matching

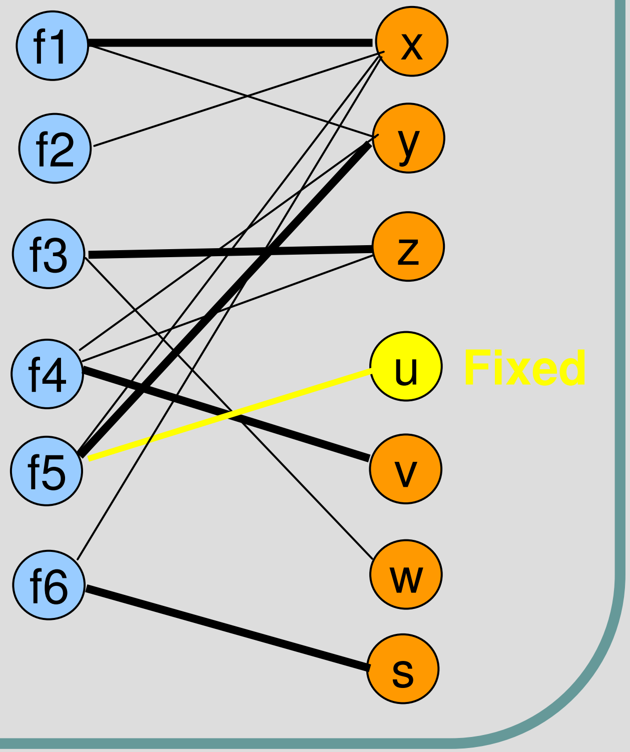
$\Rightarrow$ ***operation allowed***



Implementation in Scicos

- Whenever the user tries to fix a variable:
  - Computing the new rank of the structure matrix using the maximum matching method.

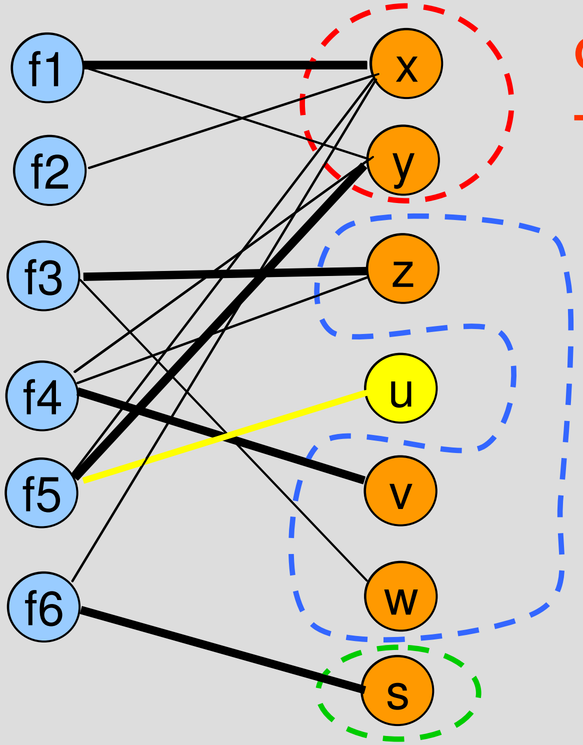
f1(x,y)=0
f2(x)=0
f3(z,w)=0
f4(y,z,v)=0
f5(x,y,u)=0
f6(x,s)=0



Implementation in Scicos

- Whenever the user tries to fix a variable:
  - If rank deficient, using the Dulmage-Mendelsohn algorithm partition the matrix into under/over-constrained parts.

f1(x,y)=0
f2(x)=0
f3(z,w)=0
f4(y,z,v)=0
f5(x,y,u)=0
f6(x,s)=0



Over-Constrained

The variable to be relaxed

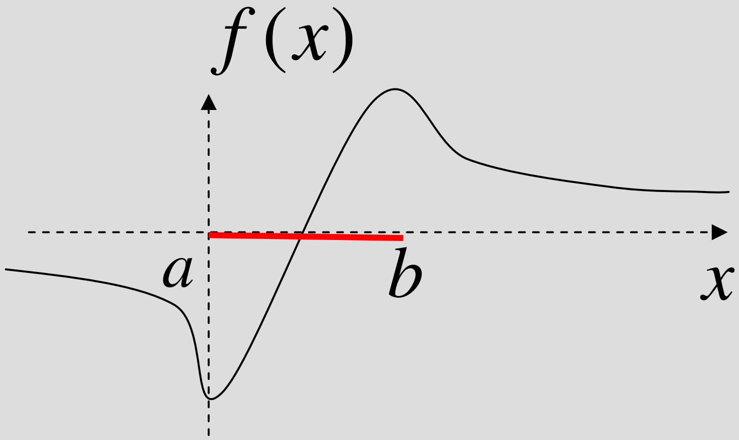
Under-Constrained

A variable to be fixed

Well-Constrained

Helping the solver to converge

- Once a well-constrained algebraic equation obtained, it should be solved.
- The algebraic equation is very often nonlinear and huge, even for medium size models.
- Newton-based solvers need good guess values of the variables to converge to a solution, e.g.,



Guess values

- Guess values are provided via the “start” attribute in Modelica.
- If not specified, they are set to zero automatically.
- It is unreasonable to ask the user to provide all guess values.
- Many variables are tightly related, e.g.,

$$\begin{cases} 0 = f(x) & \longrightarrow \quad 0 = \bar{f}(x) \quad \text{KO} \\ ay + bx = c & \longrightarrow \quad 0 = \bar{f}(y) \quad \text{OK} \end{cases}$$

$$y(start = y_0)$$

- In order to facilitate the solver’s task, the model is simplified.
- The choice of variables to be kept in the final model is important.
- GUI helps to get rid of “*variables without guess-values*”
  - “Marking” the variables with known guess-values in GUI
  - The Modelica compiler tries to keep marked variables.

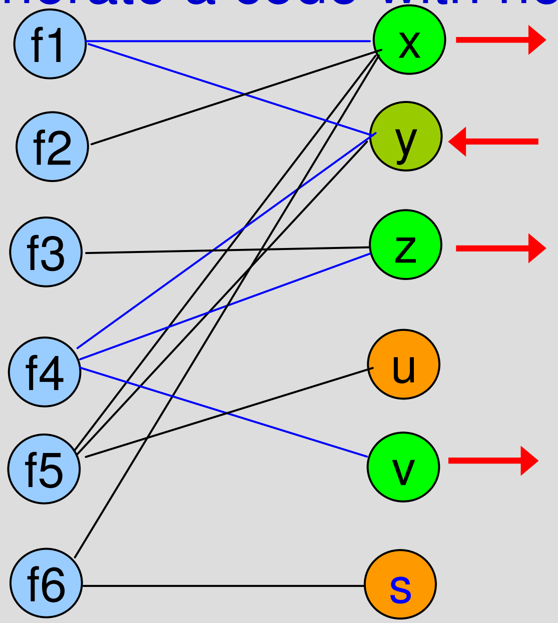
Guess values

- Problem:
  - Some unmarked variables may remain in the simplified code.
  - Remaining unmarked variables may not be familiar to the user.
- Solution:
  - The GUI proposes the alternatives to remaining unmarked variables.
  - Using the same bipartite graph and the Ford-Fulkerson algorithm.

Guess values

- *E.g.*, **x**, **z**, **v** are alternative variables to **y**.
- The user can give the guess value of a familiar alternative and marks it.
- The compiler tries to generate a code with new marked variables.

f1(x,y)=0
f2(x)=0
f3(z)=0
f4(y,z,v)=0
f5(x,y,u)=0
f6(x,s)=0



Conclusion

- A specialized GUI for the initialization of Modelica Models.
- GUI as a user assistance for model diagnostics.
- Finding under/over-determined part of the model.
- Influencing the Modelica compiler to select “marked” variables
- Proposing alternatives variables for unmarked remaining variables after the model simplification

Questions…

Thank you