Universal Translation Algorithm for Formulation of Transport Network Problems

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

Fraunhofer
SCAI

Universal Translation Algorithm for Formulation of Transport Network Problems

MYNTS

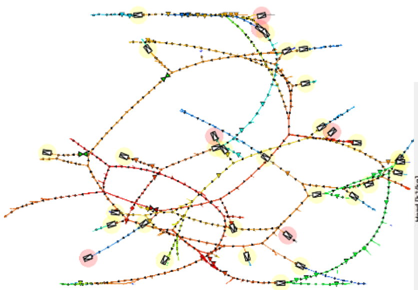
Tanja Clees, Anton Baldin, Kläre Cassirer, Bernhard Klaaßen,

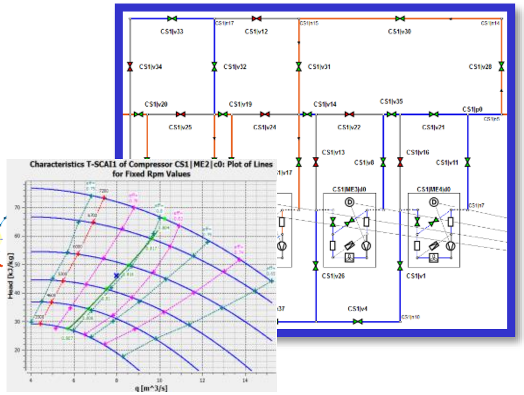
Lialia Nikitina, Igor Nikitin, Sabine Pott, et al.

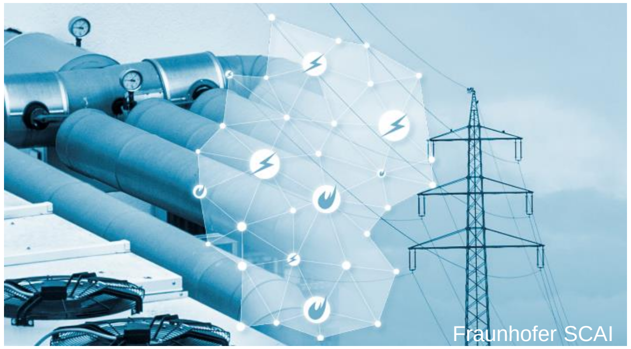
Fraunhofer SCAI

53757 Sankt Augustin, Germany

tanja.clees@scai.fraunhofer.de







Math Energy

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

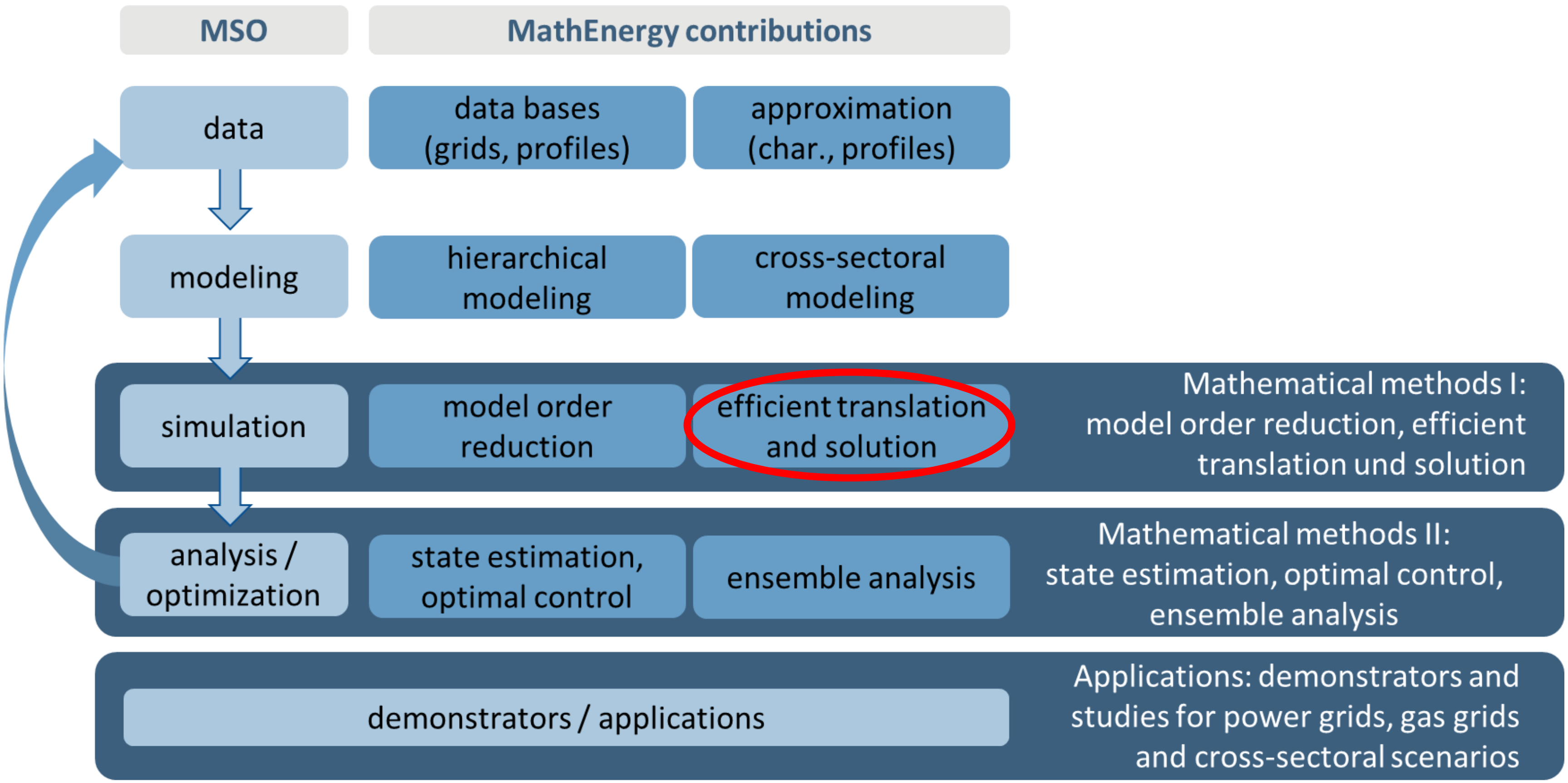
Fraunhofer SCAI

Introduction

- MYNTS (MultiphYsal NeTwork Simulator), a system for physics-based simulation of energy transport networks, is enhanced by our UTrans method.
- UTrans implements a general concept of a “universal translation” (UT) – formulation of a network problem is considered as a translation between two domain specific languages (DSL)
- In our case, UTrans translates from a network description language (DSL1), used by the community, to the problem description language (DSL2), understood by generic non-linear solvers, such as IPOPT, Mathematica, Matlab.
- The translation is performed applying a set of user-configurable translation rules, as contained in the so-called translation matrix (TM).
- Generic syntax of TM allows to formulate network problems for different energy sectors (gas, water, power, etc.) and to model a coupling between them.
- Complexity analysis and benchmarks show efficiency of the overall method.

Gefördert durch:
Bundesministerium für Wirtschaft und Energie

Key Aspects of this Contribution



Gefördert durch:

Bundesministerium für Wirtschaft und Energie

Fraunhofer SCAI

UTrans – Overview

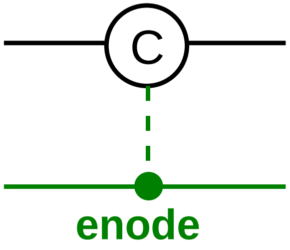
- A novel algorithm UTrans for „universal translation“ was developed and implemented in MYNTS.
- The algorithm translates a network description (NET) in an appropriate (human-readable) format to a problem description (a list of equations for the solver) suitable for a solver. Currently, IPOPT, Mathematica, and Matlab are supported.
- The translation rules (translation matrix, TM) are contained in an ASCII file.
- TM encodes all physical laws (e.g., Kirchhoff, Papay, Hofer, etc.) and control equations (for regulators, compressors, etc.), is human-readable and can be changed freely.
- Coupling of different sectors becomes easy.

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

Example 1 – Excerpt of a Translation Matrix for Coupling of Multiple Sectors

- multiple sectors as simple concatenation of TM and NET structures with possible cross-references:

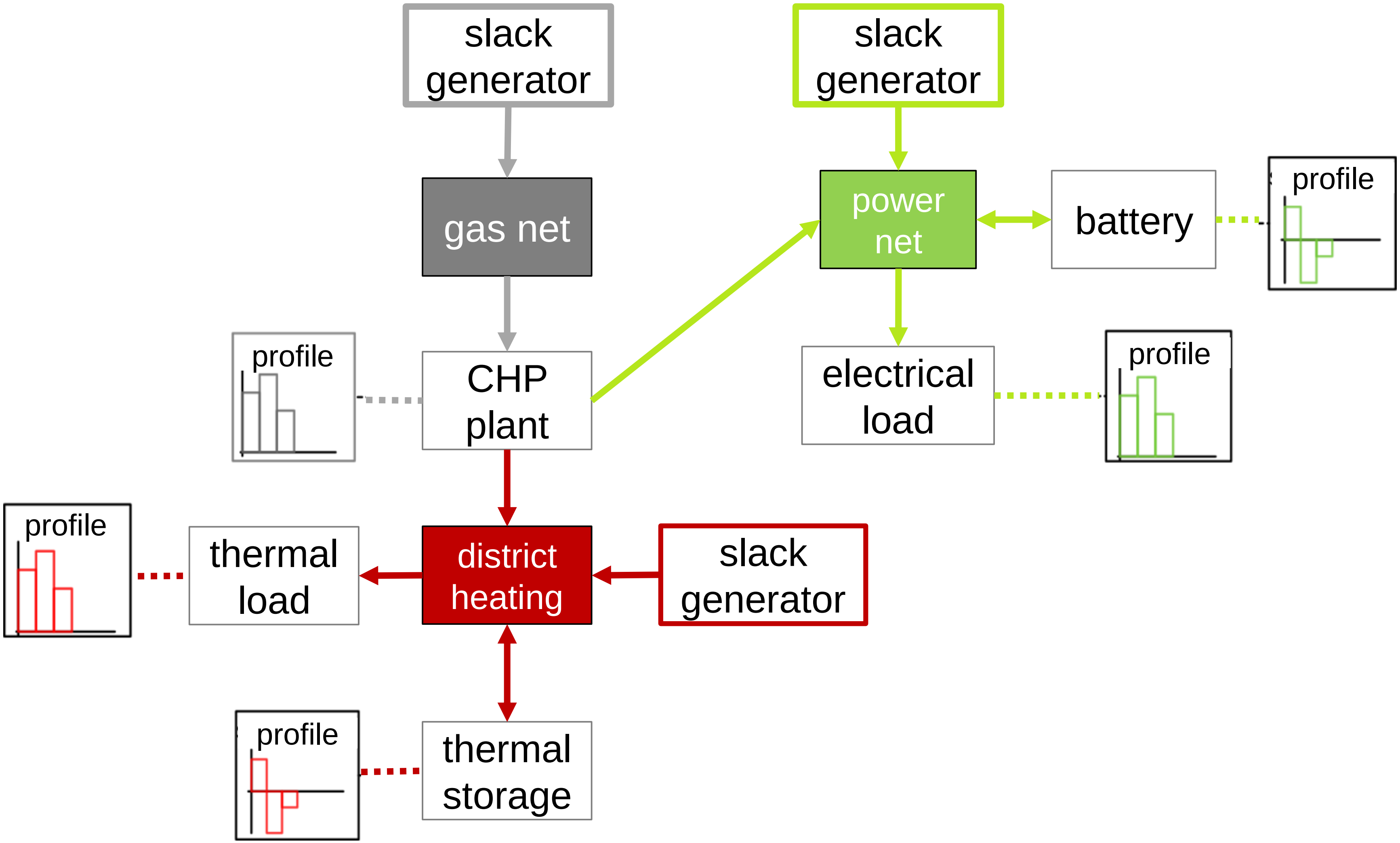



```
class=n, sector=gas, var=P;rho  # list of variables
class=e, sector=gas, var=Q;m
class=n, sector=water, var=P;T
class=e, sector=water, var=Q
class=n, sector=power, var=U
class=e, sector=power, var=I

class=e, sector=gas, eq=[m]-[rho_norm]*[Q]  # list of equations
class=e, sector=gas, type=p, eq=[P@n1]*abs([P@n1])-[P@n2]*abs([P@n2])
                        -[R]*[Q]*abs([Q])+...
class=e, sector=power, type=w, eq=[U@n1]-[U@n2]-[R]*[I]
class=e, sector=water, type=p, eq=...
class=e, sector=gas, type=c, drive=E, eq=[Q]-efun([P@n1],[P@n2],[U@enode])
...
```

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

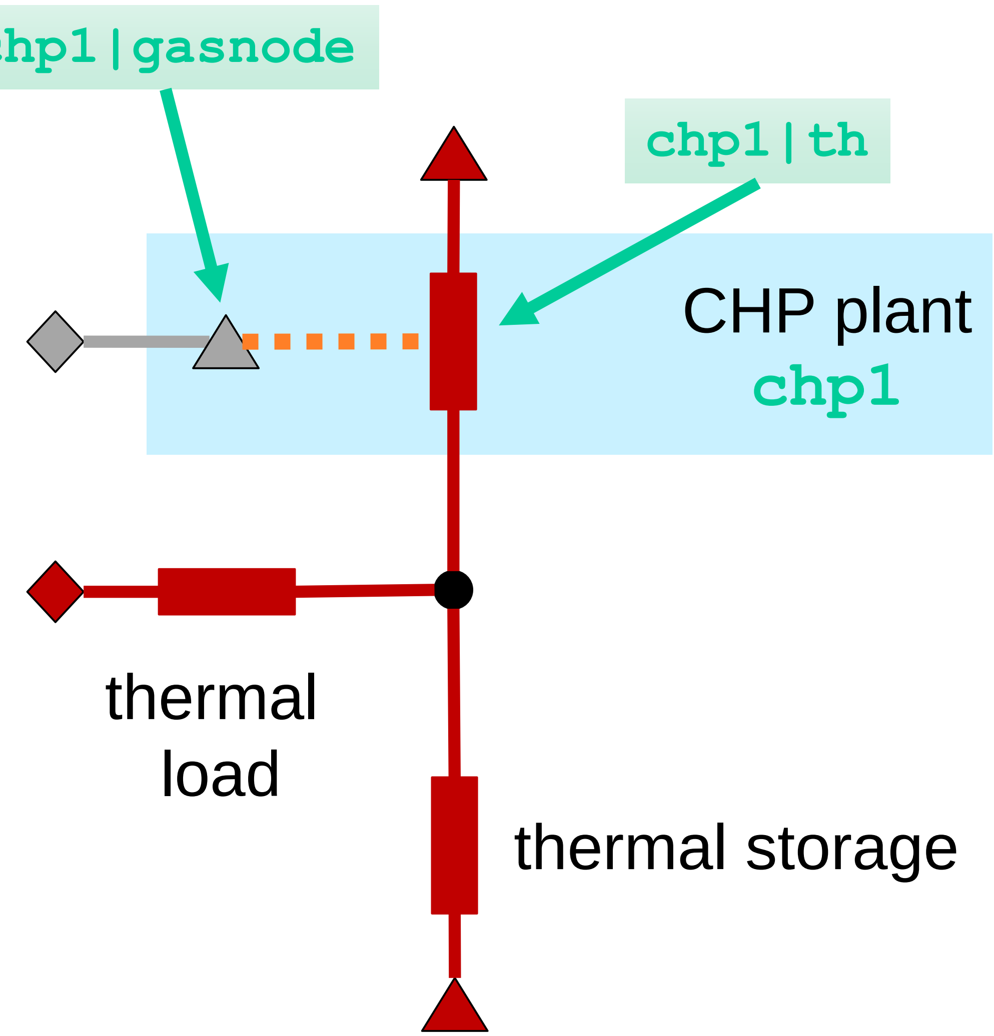
Example 2 – Sketch



Gefördert durch:
Bundesministerium für Wirtschaft und Energie

Example 2 – Basic MYNTS Network (for Demonstration)

- For demonstration purposes, we restrict ourselves to a simplified version here:
- The basic network for MYNTS contains two sectors,
  - gas (gray)
  - and water (district heating, red)
- and the following elements:
  - CHP plant (power delivery not shown here)
  - thermal load
  - thermal storage
- coupling: orange



Gefördert durch:
Bundesministerium für Wirtschaft und Energie

Example 2 – Excerpt from Translation Matrix

- Excerpt from the respective translation matrix,
  showing one exemplary coupling of the gas and water sectors:

- Properties **`eta_th_set`** and **`powset`**
  of the **`n_gas`** node from the gas sector
  are referenced by the **`kwk_th`** element from the water sector:

```
sector=water, class=e, type=kwk_th, mtype=1,
    eq="[c]*fabs([m])*([T]-[T@in])-
            [eta_th_set@n_gas]/100*[powset@n_gas]*1e6"
```

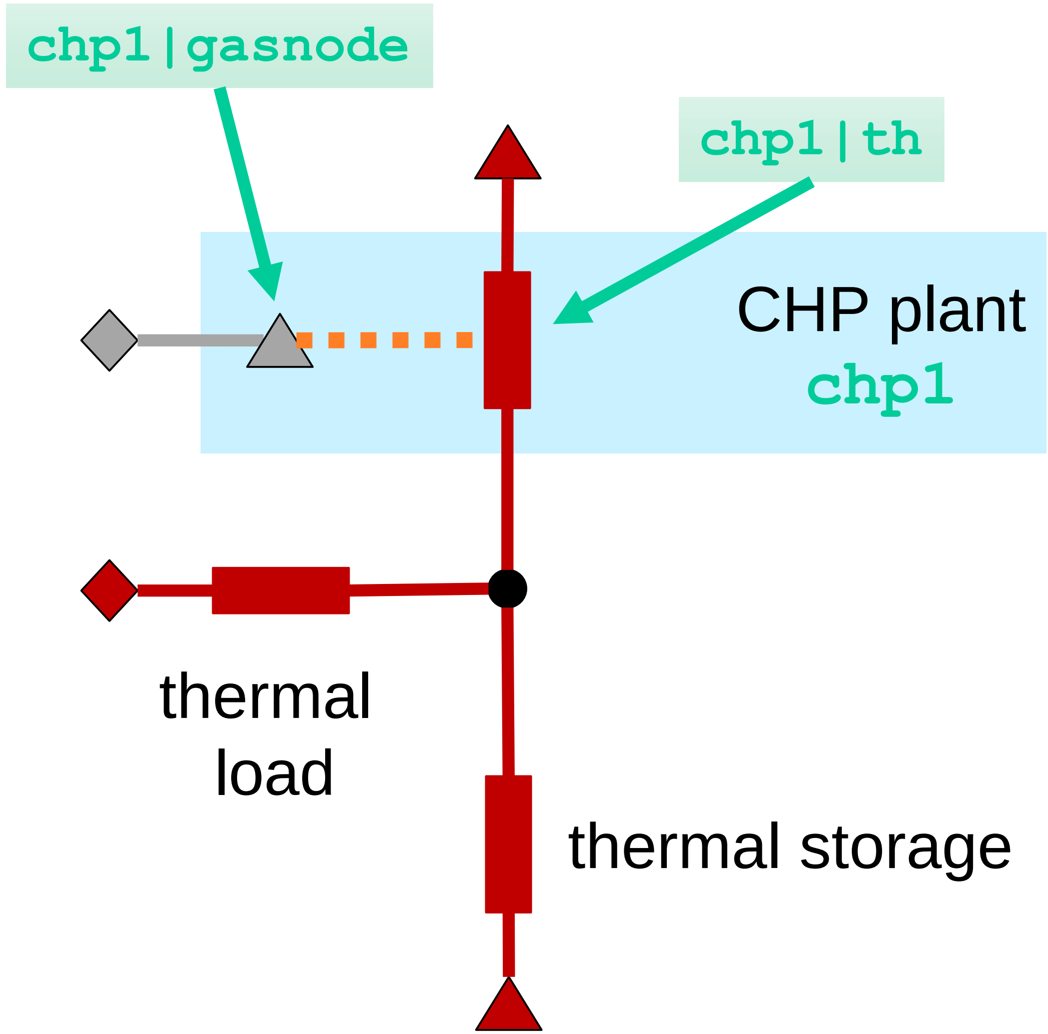
- In the net, an element of type **`kwk_th`** needs, in particular, a property **`n_gas`** containing the name of the respective gas node.

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

Example 2 – Relationship of Translation Matrix and Network

- In the basic MYNTS net:
  - `chp1|th` is an edge of type `kwk_th`
  - It needs, in particular, a property `n_gas` containing the name of the respective gas node, here: `chp1|gasnode`
- Via TM, the connection between the sectors is made.

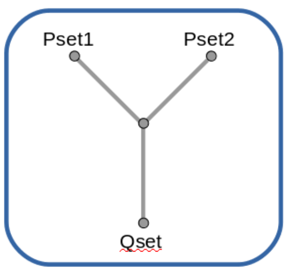


Gefördert durch:
Bundesministerium
für Wirtschaft
und Energie

Fraunhofer
SCAI

UTrans – Basic Workflow for NL-Format (IPOPT Solver)

- `net-w`: writes MYNTS network
- `preNL`: creates NL-ready format TM-NL of translation matrix TM
- `utrans`: creates core system to be solved based on net and (here) TM-NL
- `postNL`: writes system (incl. headers etc.) in suitable format, here: NL
- `solver`: here, IPOPT
- `x2net`: converts solver output back to MYNTS



x

Translation matrix for water networks with laminar flow (TM)

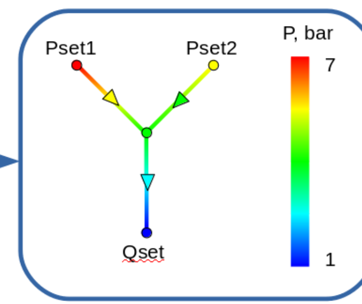
```
# variables with start values
class=n, var=P, var0=1
class=e, type=p, var=Q, var0=0

# equations of elements (linear resistors)
class=e, type=p,
      eq="[P@node1]-[P@node2]-[R]*[Q]"
```

=

NL: eqs encoded in Polish prefix notation (PPN)

```
7 7                        #nvars,neqs
o1 v0 n7                   #eq0
o1 v1 n5                   #...
o1 o1 v0 v2 o2 n0.3 v4
o1 o1 v1 v2 o2 n0.05 v5    #...
o1 o1 v2 v3 o2 n0.1 v6     #eq6
```

solver



net-w preNL utrans postNL x2net

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

Fraunhofer SCAI

Complexity of the Algorithm

- Complexity estimation:

$$O(N_{elem}\, N_{prop}\, N_{eq})$$

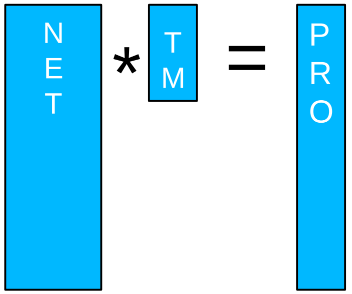
N_{elem} – number of elements,

N_{prop} – (average) number of properties per element,

N_{eq} – (average) number of equations per element.

- For linear problems with only one type of elements, complexity is identical to that of a matrix multiplication

$$NET * TM = PRO$$

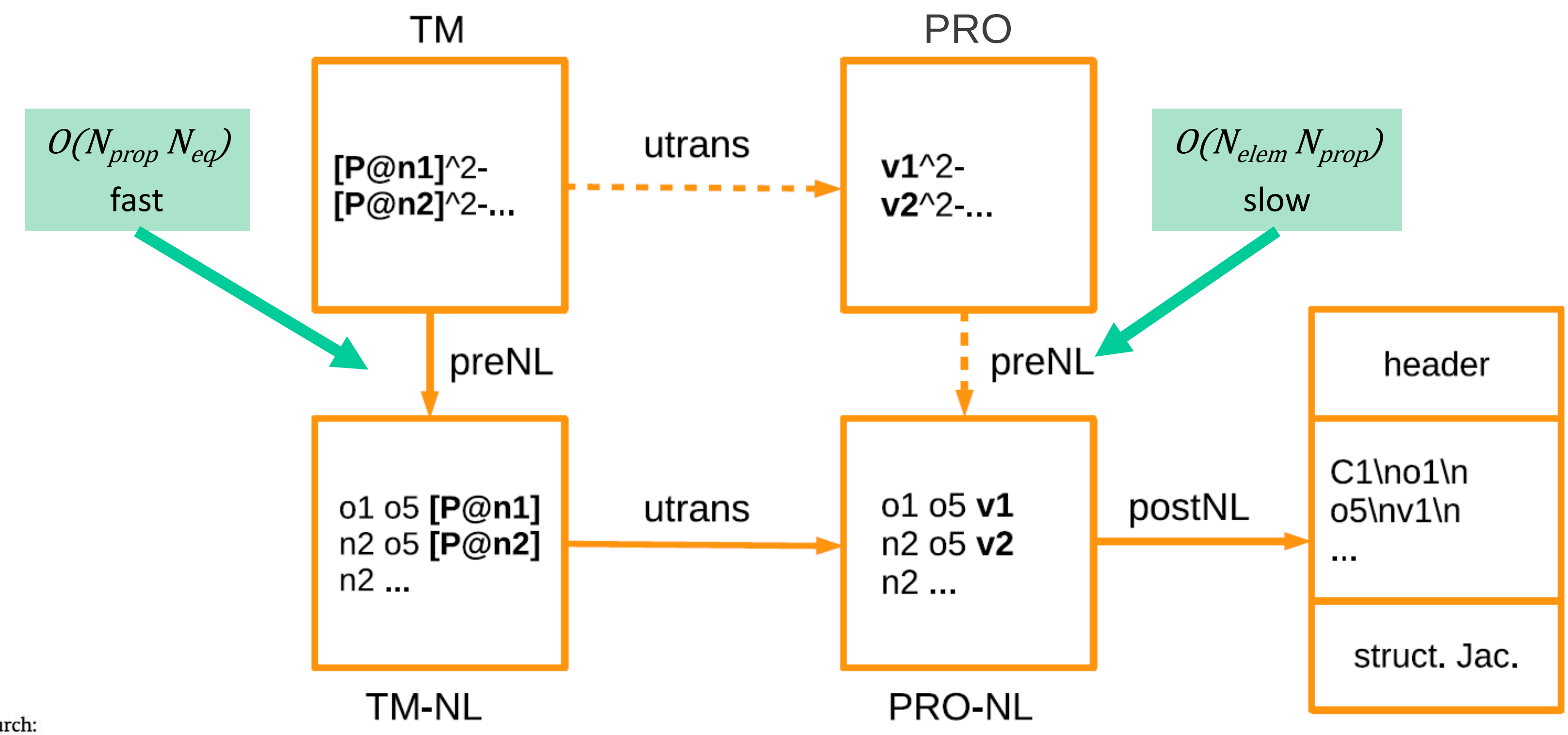


where NET is an $N_{elem}\ x\ N_{prop}$, TM an $N_{prop}\ x\ N_{eq}$, and PRO an $N_{elem}\ x\ N_{eq}$ matrix.

Gefördert durch:
Bundesministerium für Wirtschaft und Energie

Behind the Scenes

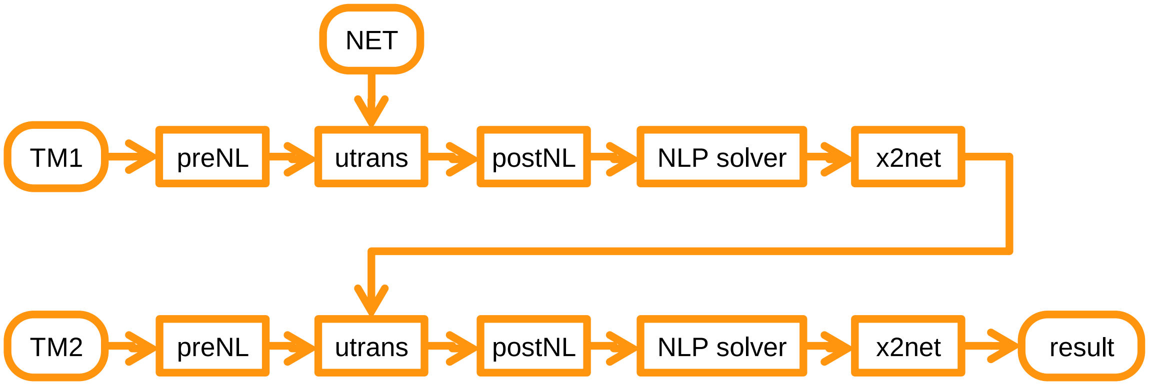
- Retranslation of the problem to NL form: The translation path, shown by solid lines, is much faster than the one shown by dashed lines.



Usage of UTrans – Exemplary Solver Workflow

- Call to the new kernel invokes a Python workflow
- It describes a multiphase solution procedure
  - e.g., start with forced goals of compressors and regulators
  - proceed with free compressors
  - then with advanced
  - iterate mixing of temperature and gas properties
  - …

Example:
2-phase solution process



Gefördert durch:
Bundesministerium für Wirtschaft und Energie

Exemplary Results

- Performance benchmarks for selected realistic gas transport networks:

network	nodes	edges	translation [sec]	solution* [sec]
N1	100	111	0.04	0.02
N2	931	1047	0.15	0.24
N3	4466	5362	0.39	0.42

- Timings on a workstation with 3 GHz Intel i7 CPUs and 8 GB RAM
- *) one-phase solution process

Gefördert durch:
Bundesministerium für Wirtschaft und Energie

More Remarks on Features and Performance

- UTrans comes with several filters, e.g. for topological reduction*.
- Particular representation of NET and TM structures plays no role – different implementations for the same abstract structure (sets of maps)
- Implementation based on set and map containers of C++ STL, allowing usage of efficient search and set-theoretical algorithms
- Syntax of TM flexible and completely in user's hand:

  One can define other names for the properties in NET and adopt TM accordingly. One can also change the syntax of equations in TM according to the solver language – main algorithm independent of format!
- Translation algorithm is highly parallelizable (future work):

  Processing loops can be separated element-wise and proportionally accelerated by using multiprocessor architectures.

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

**) see separate set of slides*

Fraunhofer SCAI

Thanks for Your Attention

www.mathenergy.de

mynts@scai.fraunhofer.de

Gefördert durch:

Bundesministerium für Wirtschaft und Energie