Shimmer

Safe Hydrogen Injection Modelling and Management for European gas network Resilience

D4.2 Tool for infrastructure optimization: Investments and Capacity

VERSION	DATE	TYPE	DISSEMINATION LEVEL
1.0	20.3.2026	Report	Public

ABSTRACT

To analyse network expansion for hydrogen injection strategies in European natural gas transmission networks, an extension for the open-source EnergyModelsX (EMX) framework was developed in the SHIMMER project: https://github.com/EnergyModelsX/EnergyModelsGasNetworks.jl

The mathematical formulation and the implementation in EMX are described for the new nodes, links to implement quality tracking and flow constraints following from operational pressure levels are described, followed by an installation guide and tutorial on how to use the software.

Results from a validation study against more detailed simulation performed in the project are also presented.

AUTHORSHIP AND APPROVAL INFORMATION

AUTHOR(S)	DATE / SIGN
Raquel Alonso Pedrero / SINTEF Lars Hellemo / SINTEF	27-March-2026 Lars Hellemo (Mar 27, 2026 13:42:41 GMT+1)
REVIEWED BY WP-LEADER	**DATE / SIGN**
Huib Blokland / TNO	Huib Blokland (Mar 30, 2026 11:00:41 GMT+2)
APPROVED BY COORDINATOR	**DATE / SIGN**
Martin Fossen / SINTEF Diana Gonzalez / SINTEF	Diana Gonzalez (Apr 7, 2026 09:41:16 GMT+2)

CO-FUNDED BY THE EUROPEAN UNION. VIEWS AND OPINIONS EXPRESSED ARE HOWEVER THOSE OF THE AUTHOR(S) ONLY AND DO NOT NECESSARILY REFLECT THOSE OF THE EUROPEAN UNION OR THE CLEAN HYDROGEN JOINT UNDERTAKING. NEITHER THE EUROPEAN UNION NOR THE GRANTING AUTHORITY CAN BE HELD RESPONSIBLE FOR THEM.

Release history

VERSION	DATE	VERSION DESCRIPTION
0.1	27-February-2026	First draft version sent for internal review
0.2	20-March-2026	Draft version for project-wide circulation before submission
1.0	27-March-2026	Final version

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Table of contents

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Table of Figures

List of Tables

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

List of Abbreviations

Table 1: List of abbreviations

Term	Explanation
EMX	Energy Models X
H2	Hydrogen
LP	Linear Programming
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Non-Linear Programming
PWA	Piecewise-affine approximation
NG	Natural gas
TSO	Transmission System Operator

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Executive Summary

This document describes the optimisation model extension developed in the SHIMMER project to do network expansion analyses for hydrogen injection strategies in European natural gas transmission networks. To evaluate strategies where existing infrastructure needs to be either repurposed or added, e.g. by adding extra compression capacity, adding or repurposing pipelines or adding storages for buffering.

The optimisation model is implemented as an extension to EnergyModelsX, an open-source energy systems modelling framework. The software along with online documentation is available on GitHub: https://github.com/EnergyModelsX/EnergyModelsGasNetworks.jl

The mathematical formulation of the optimisation model is described for the new functionality provided by the extension, including the tracking of quality through a network with mixing and splitting and quality constraints on the terminals, in the optimisation literature known as the pooling problem. The quality tracking can be combined with a model for pipeline flow constraints linked to operational pressure levels through a linearisation of the Weymouth equation, also accounting for the change in gas properties following from the mixing in the network.

The implementation of the extension in EMX in terms of definition of new resources, nodes, links and necessary data structures is described, followed by an installation guide and tutorial on how to get started with the software extension.

Validation results from comparing with more detailed physical modelling in a simulation model from the project are also presented.

About the project: The European natural gas infrastructure provides the opportunity to accept hydrogen (H_2), as a measure to integrate low-carbon gases while leveraging the existing gas network and contributing to decarbonisation. However, there are technical and regulatory gaps that should be closed, adaptations and investments to be made to ensure that multi-gas networks across Europe will be able to operate in a reliable and safe way while providing a highly controllable gas quality and required energy demand. Aspects such as material integrity of pipelines and components, as well as the lack of harmonisation of gas quality requirements at European level must be addressed in order to facilitate the injection of H_2 in the natural gas network.

In this context, the SHIMMER project (Safe Hydrogen Injection Modelling and Management for European gas network Resilience) was selected for funding as part of the 2023 Clean Hydrogen Partnership programme. SHIMMER aims to enable a higher integration of low-carbon gases and safer H_2 injection management in multi-gas networks by strengthening the knowledge base and improving the understanding of risks and opportunities in H_2 projects.

It will do this by:

- Mapping and assessing European gas T&D infrastructure in relation to materials, components, technology, and their readiness for hydrogen blends.
- Defining methods, tools and technologies for multi-gas network management and quality tracking, including simulation, prediction, and safe management of network operation in view of widespread hydrogen injection in a European-wide context.
- Proposing best practice guidelines for handling the safety of hydrogen in the natural gas infrastructure and managing the risks.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

1 Introduction

This chapter introduces the content of the report, giving the background and motivation for the work performed as part of Task 4.2, subtask 4.2.1 System Design and Capacities. It also serves as a guide to the reader on how to read and make use of the report.

1.1 Purpose of the document

The purpose of this document is to describe the optimisation model extension developed in the SHIMMER project in Task 4.2, subtask 4.2.1 to do network expansion analyses for hydrogen injection strategies in European natural gas transmission networks. To evaluate strategies where existing infrastructure needs to be either repurposed or added, e.g. by adding extra compression capacity, adding or repurposing pipelines or adding storage for buffering.

The software presented in this document is an extension to Energy Models X (EMX), an open-source energy systems modelling framework developed at SINTEF and introduced in more detail in the following chapter.

The model extension adds physical constraints to energy flows to represent the relationship between gas pressures and flows in the network. This allows performing analyses where pipeline capacity in one part of the network can be influenced by the operation in another part of the network through shared pressure dependencies, in contrast with the typical fixed capacities assumed in energy systems models.

Quality tracking in networks is also supported by modelling the mixing of gas with different qualities in the network. Modelling this properly for general network topologies requires the formulation to consider the mathematically challenging class of optimization problems commonly referred to as pooling problems.

Combining these, the model extension allows approximating flow capacity under given pressure levels also when considering changing gas properties throughout the network due to admixing of much lighter hydrogen in the natural gas networks.

The modelling has been performed with the goal of adding more physical detail than is available in energy systems models, balancing the level of detail with computational tractability. While the model offers great capabilities for analyses, including the more accurate formulations as non-linear constraints in addition to using bilinear terms and binary variables can be computationally challenging for networks with large numbers of nodes and pipeline segments and/or time periods.

Hence, the formulations used in `EnergyModelsGasNetworks` still rely on simplifications and approximations compared with more detailed simulation models, but offers a different type of decision support, as it directly addresses the design of the network (e.g. adding pipelines or compressors) and how it should be operated (suggesting pressure level set points and flows).

1.2 Intended readership

The intended audience for this document is analysts working on injection strategies for Transmission System Operators (TSOs) or members of the academic community with an interest in gas transport infrastructure development in general and evaluating hydrogen admixing in natural gas transport networks in particular.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Readers are expected to have prior knowledge of how gas transmission networks and how they are typically operated. Some familiarity with mathematical optimization is an advantage, but not a requirement to read the report, but the software tool is intended for expert users with at least some familiarity with optimization software.

1.3 Structure of this document

This document includes the documentation for the software deliverable, and is structured as follows:

Chapter one gives the background and motivation for the work. In chapter 2 follows an introduction to the EnergyModelsX framework to which EnergyModelsGasNetworks is an extension. Chapter 3 explains how the different parts are modelled mathematically, before giving an overview of how they have been implemented in the software as nodes and links in chapter 4. Chapter 5 is a guide on how to install the software, and Chapter 6 contains a tutorial to help the interested reader to get started with the software. For detailed documentation of the API we refer to the online documentation which will be updated with new releases of the software.

A list of the chapters in order follows:

1. Introduction
2. Philosophy behind EMX and EnergyModelsGasNetworks
3. Theoretical Background
4. Implementation
5. Installation
6. Tutorial
7. References

1.4 Stakeholder involvement

Stakeholders of this work package have shown great interest in the work and contributed with their experience and expertise to complement each other in reaching the goals of the work package. Specifically, the gas transmission operators have been involved in selecting and prioritising both the use cases and the most relevant analyses, guiding the prioritisation of functionality for the software extension during dedicated workshops in the consortium meetings and dedicated technical workshops. Research partners have contributed to the development of the model extension in providing validation by comparing with results from SHIMMER++ from PoliTO and interoperability of data formats from TNO.

1.5 Relationship with other deliverables

The software extension presented in this document has been tested during development on test cases defined in D4.1 in task T4.1 and presented in the report D4.1 *Report describing the defined simulated test cases, realistic scale testcase(s), and available operational models including required data set and format*. Validation studies have been performed using a selection of test cases and with the tool SHIMMER++ described in D4.4 Open-source fluid-dynamic model with gas quality tracking released with handbook and tutorials. The software presented in this document will be used for the use-case studies that will form the basis for the report D4.3 Report documenting recommendations of mixing strategies and infrastructure needs to be completed later in the project.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

2 Philosophy behind EMX and EnergyModelsGasNetworks

EnergyModelsX (EMX) [1] is a family of Julia packages developed by SINTEF Industry and SINTEF Energy Research for energy systems analyses. EMX is open source and freely available on GitHub[1]. It is designed using the open-source Julia programming language [2] and the JuMP modelling language for optimization [3]. Both the core packages and the published extensions can be easily installed using the Julia package manager.

Energy system models such as EMX can be used to study the operation of systems with multiple energy carriers, generation technologies and demands for different energy carriers or energy services. The system operation is typically optimized to serve the stated demand with the least cost while maintaining the defined operational constraints. Infrastructure development, or what is often referred to as network expansion, considers the maximization of the net present value of the total costs of future developments, accounting for the operational costs and technical requirements required to meet energy demands for different carriers such as power or gas. Many energy systems allow only very simplified technology descriptions assuming linear costs and capacities using a predetermined parametrization (e.g. CAPEX, OPEX, capacities).

EMX is designed to be modular to give users the flexibility to easily add new capabilities such as more detailed description of specific technologies. EMX takes a multi-nodal approach where the energy system is defined as nodes and links between nodes where the types and properties of different energy technologies can be represented through different node types and parameters. New node types or links between nodes can be introduced by the user to extend the techno-economic capabilities of elements in the system without requiring changes in the core structure of the model or indeed changing the source code of the core packages.

Extensions to EMX are typically developed as separate software packages with definitions of new nodes or links that are more specific than the default implementations. The appropriate constraints and variable generation are then overloaded when building the model through Julia's multiple dispatch mechanism.

In SHIMMER, SINTEF Industry has developed an EMX extension called `EnergyModelsGasNetworks` which implements a detailed representation of flow-pressure relationships and tracking of gas composition. In the optimization literature, the tracking of quality in terms of blending gas with different qualities in the network often gives rise to a problem structure known as the "pooling problem", which has a mathematical structure which means it must be modelled using non-linear constraints or binary variables. Both problem classes make the resulting problem instance much harder to solve. In EnergyModelsGasNetworks the constraints are modelled using a non-linear formulation (see Chapter 3 for more details).

As EMX itself, the `EnergyModelsGasNetworks` is open source and freely available on Github, and will be registered in the General registry for Julia packages to be easily available through the Julia

[1] https://github.com/EnergyModelsX/EnergyModelsGasNetworks.jl

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

packet manager when the package has been successfully registered. Julia itself is open source and can be downloaded freely from the Julia website[2].

The energy systems model will need a solver for the appropriate class of problem (e.g. LP, MILP, MINLP). The framework is solver agnostic, and the user can select freely from available solvers. Open-source solvers are available for most problem classes, but the performance of solvers from commercial vendors tends to be much better than the freely available ones.

This report focuses on the development and use of `EnergyModelsGasNetworks` for modelling the admixing problem which is at the operational level. In the analysis for SHIMMER, the module would be used in combination with the modules `EnergyModelsBase`, `EnergyModelsInvestments` and `EnergyModelsRenewableProducers`.

[2] https://julialang.org/downloads/

3 Theoretical Background

To effectively transport natural gas from production sites to demand regions, the gas transmission network requires a complex combination of components that includes compressor stations, valves, pipes, storage tanks, underground storage, among others. The gas transport industry is in charge of providing an effective, reliant and cost-efficient gas system. As such, operational and strategic models for decision making are important for gas operators.

`EnergyModelsGasNetworks` allows defining optimization models for long-term planning of natural gas networks tailored to the needs of transmission operators. The goal would be to maximize the net present value of the total costs of future developments, accounting for the operational costs and technical requirements required to meet energy/gas demands. In alignment with SHIMMER, the module was developed not only to consider single-gas networks but also consider the admixing problem, where the propagation of the quality of different components and its effect on flow and pressure decisions must be modelled.

The following sections explain in detail how the model handles the admixing problem to be studied in SHIMMER. First, the pooling problem is considered to handle the proportions of the different components within the quality bounds set by the network. Secondly, the pressure-flow relationships are modelled.

It should be noted that, differently from simulation models, optimization models require approximating the physics of the gas problems as they might suffer from issues with computational tractability. Therefore, specific considerations for natural gas-hydrogen blends were examined in detail to balance physical correctness and complexity of the model. The methodology adopted was validated with the simulation model SHIMMER++[3], see report D4.4 for more about SHIMMER++.

3.1 Pooling Problem Extension

The pooling problem captures how different gas sources mix as they move through a network. Because sources may inject different gas types (or "commodities"), the resulting model is naturally *multi-commodity*, and it becomes necessary to track and constrain mixture composition at intermediate nodes and at terminals.

Based on the formulation from [4], [5], we define a directed graph $G = (N, A)$ where N is the set of nodes and A the set of arcs. Each arc $(i, j) \in A$ has a known capacity $b_{(i,j)}$. Let us denote the out-neighbours of node i by $N_i^+ = \{j \in N: (i, j) \in A\}$ and the in-neighbours by $N_i^- = \{j \in N: (j, i) \in A\}$. We distinguish:

- **Source Nodes**, S: nodes with no incoming arcs ($N_i^- = \emptyset,\ \forall s \in S$).
- **Terminal Nodes**, T: nodes with no outgoing arcs ($N_i^+ = \emptyset,\ \forall t \in T$).
- **Pooling Nodes**, I: nodes where incoming flows are blended linearly, meaning the proportion (or quality) of each commodity reaching the node is a weighted average of upstream proportions, with weights given by the incoming arc flows ($I = N \backslash (S \cup T)$).

[3] https://github.com/shimmerhydrogen/shimmer.git

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Let K denote the set of commodities. The parameter q_s^k represents the proportion of the commodity k injected from source s. This parameter equals 1 if source s supplies commodity k and 0 otherwise. For each node $n \in N \backslash S$, the parameters $\overline{q}_n^k$ and $\underline{q}_n^k$ define the maximum and minimum proportion bound, respectively.

We define the set of pooling nodes. The composition of the flow arriving at each node depends on the sources from which flows originate. At nodes $n \in (I \cup T)$, incoming flows are blended linearly, meaning that the proportion (or quality) of each commodity at n is computed as a weighted average of the proportions at its upstream neighbour nodes and their corresponding outlet flows towards n. `EnergyModelsGasNetworks` assigns flow values that ensure respecting the quality bounds at terminals and pooling nodes.

The extension was defined to allow the definition of *generalised pooling problem*, allowing pool-to-pool and source-to-terminal connections. As such, qualities depend on recursive mixing and can lead to much harder nonconvex problems. The formulation used in the `EnergyModelsGasNetworks` is based on the multi-commodity flow formulation proposed in [4]. In this formulation, at each pooling and sink node, we define the variable y_i^s denoting the proportion of flow coming from s.

The proportion of flow at node $i \in I$ from source s is calculated as follows:

$$y_i^s = \frac{\sum_{i \in N_i^-} f_{(j,i)}\, y_j^s}{\sum_{i \in N_i^-} f_{(j,i)}},\ \forall i \in I, \forall s \in S_i$$

where S_i are the source nodes who are connected to arcs that reach i.

Furthermore, the quality bounds are guaranteed by:

$$\sum_{j \in N_t^-} \sum_{s \in S_j} (q_s^k - \overline{q}_t^k) y_j^s f_{(j,t)} \leq 0,\ \forall t \in T$$
$$\sum_{j \in N_t^-} \sum_{s \in S_j} (q_s^k - \underline{q}_t^k) y_j^s f_{(j,t)} \geq 0,\ \forall t \in T$$

Other constraints considered are the nodal flow balance, capacity limits at nodes and links.

3.2 Pressure-Flow Problem Extension

One main consideration when modelling gas network models is to capture the relationship between pipeline flows and pressure drop. The three fundamental flow equations (i.e., mass conservation, momentum conservation and equation of state) form a system of nonlinear partial differential equations which are too complex to be included in optimization models.

To tackle this limitation, `EnergyModelsGasNetworks` captures flow-pressure relationship using the Weymouth equation, which simplifies the laws and assumes a steady-state system. The Weymouth equation is expressed as:

$$Q^2 = W(p_{in}^2 - p_{out}^2)$$

where

$$W = \frac{D^5}{KTfLzg}$$

Q is the flow through the pipeline, p_{in} and p_{out} are the pressures at the beginning and end of the pipe, D and L are the inside diameter and length of the pipe, K is a global constant, T is the gas temperature, f is the Darcy-Weisbach friction factor in the pipeline, while the z and g are the compressibility and specific gravity of the gas. These last two parameters in the Weymouth constant depend on the gas composition, while the rest of the parameters are pipe specific.

In pipes transporting single gases, like pure methane, the Weymouth equation is linearised using the first-order Taylor approximation [6]. The outer approximation of Equation is as follows:

$$Q_{ij} \leq W_{ij} \frac{PIN_{ijl}}{\sqrt{PIN_{ijl}^2 - POUT_{ijl}^2}} p_{ij}^{in} - W \frac{POUT_{ijl}}{\sqrt{PIN_{ijl}^2 - POUT_{ijl}^2}} p_{ij}^{out}$$

This approximation is suitable for transmission networks given that pressures are high and variation in pressure are small relative to absolute pressure, which makes the Weymouth already close to linear and there are no low-flow regimes narrowing the upper end of feasible flows. This basically implies that the Taylor is an accurate approximation at transmission networks' regimes.

In the SHIMMER use cases, the goal is to examine the admixing problem. Given that the first-order Taylor approximation considers a fixed Weymouth constant W, we cannot represent the effect of changes in gas composition through the compressibility and specific gravity.

To tackle this, we developed a new methodology that allows to reformulate the flow-pressure relationship based on the Weymouth equation. The method was developed by examining the case of hydrogen and natural gas blends.

As mentioned, the Weymouth constant depends on the compressibility factor of the gas, z and the specific gravity g. At constant pressure drop, the addition of hydrogen in the mixture increases the flow rate of the gas (non-linear increase) [7]. For instance, additions of 10 and 20 mol% induces a flow increase of approximately 4 and 9%, respectively, while transporting pure hydrogen implies a 165% [7].

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

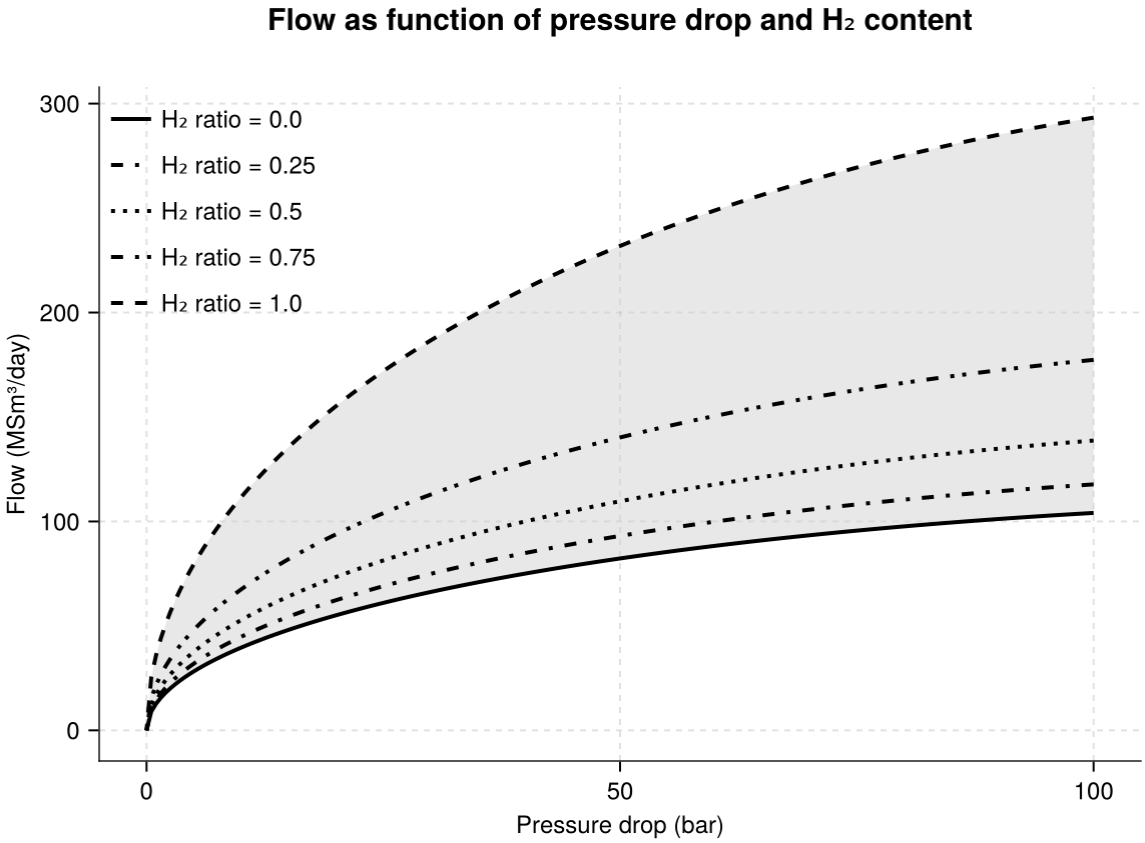

Figure 1. Representation on how hydrogen injection changes flow capacity at given pressure drops. Different hydrogen content ratios mixed with natural gas are presented.

The increase in flow is primarily due to the decrease in specific gravity resulting from the addition of hydrogen. This is calculated as the fraction of the gas molecular weight and the molecular air of air (see Equation below). Since the molecular weight of methane (16.04) is eight times greater than that of hydrogen (2.02), adding hydrogen significantly lowers the specific gravity of the blend. In contrast, there is a minor decrease in the compressibility factor, which as a negligible effect on the final flow of the gas [7].

$$g^{gas} = \frac{M^{gas}}{M^{air}}$$

We then reformulate the Weymouth constant, W, such that the compressibility factor remains to that of the natural gas, despite hydrogen injections, and the specific gravity is written as a function of the molecular weight of the mixture (weighted sum of the molecular weight of its components). This reformulation is as follows

$$W = K\frac{1}{g^{mix}} = K\frac{\sum_{i=1}^{n} w_i M^i}{M^{air}}$$

where K is a composition-independent Weymouth constant that aggregates pipe and operating parameters, while the effect of gas composition is accounted through the specific gravity g^{mix}.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

$$K = \frac{D^5}{KTfLz}$$

Thus, the Weymouth equation for pipes carrying blended gases becomes:

$$f = \sqrt{K \frac{\sum_{i=1}^{n} w_i M^i}{M^{air}} (p_{in}^2 - p_{out}^2)}$$

As such, additionally to the flow and pressures, the weight of each component becomes a variable.

3.3 Linearisation of the adapted Weymouth equation

The blended Weymouth equation introduces an additional variable through the mixture-dependent coefficient. This coefficient is affine in the mixture weights and remains nonnegative under physical bounds. On the physical operating conditions (inlet pressure ≥ outlet pressure, nonnegative pressures and flows), the resulting pressure–flow relation is smooth and convex (see Figure 2).

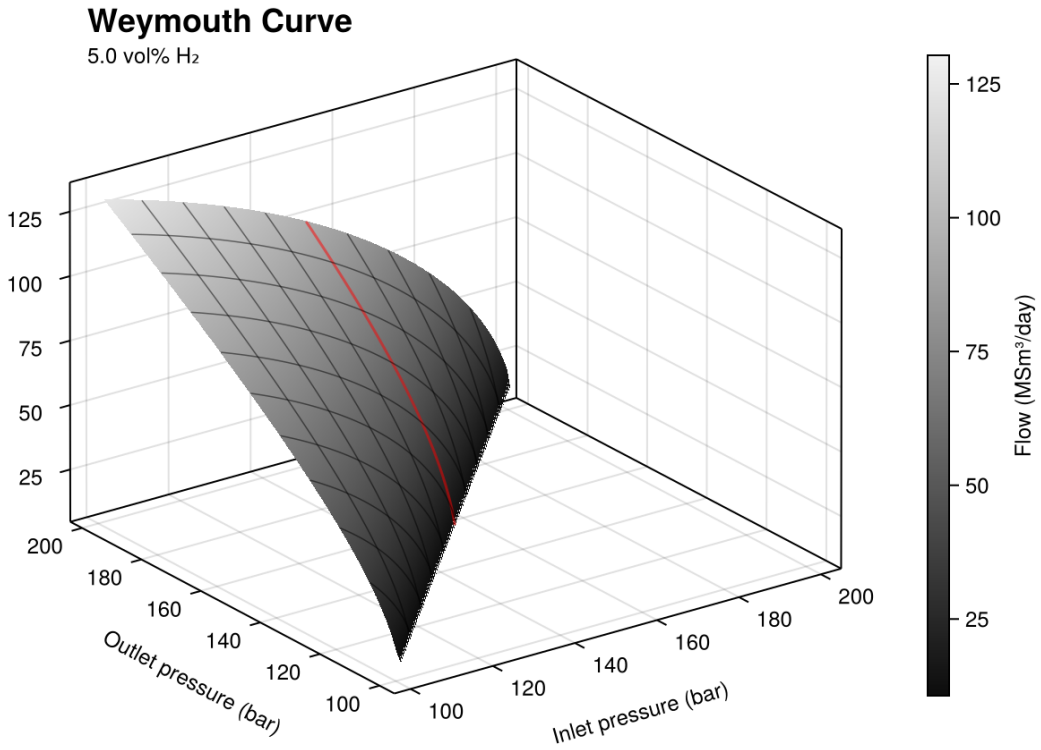

Figure 2. Surface of flow capacity as a function of inlet and outlet pressures for a fixed hydrogen ratio of 0.05 and W=0.3

To obtain a pure MILP formulation, we approximate the nonlinear surface using a convex piecewise-affine (PWA) outer approximations based on tangent planes. The planes were defined using the Julia package PiecewiseAffineApprox.jl[4] developed by SINTEF Industry. The adequate number of planes should be defined to ensure a proper approximation tightness to the curve.

[4] https://github.com/sintefore/PiecewiseAffineApprox.jl

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Figure 3 exemplifies how the PWA with tangent planes is used in `EnergyModelsGasNetworks` to approximate the non-linear flow-pressures surfaces.

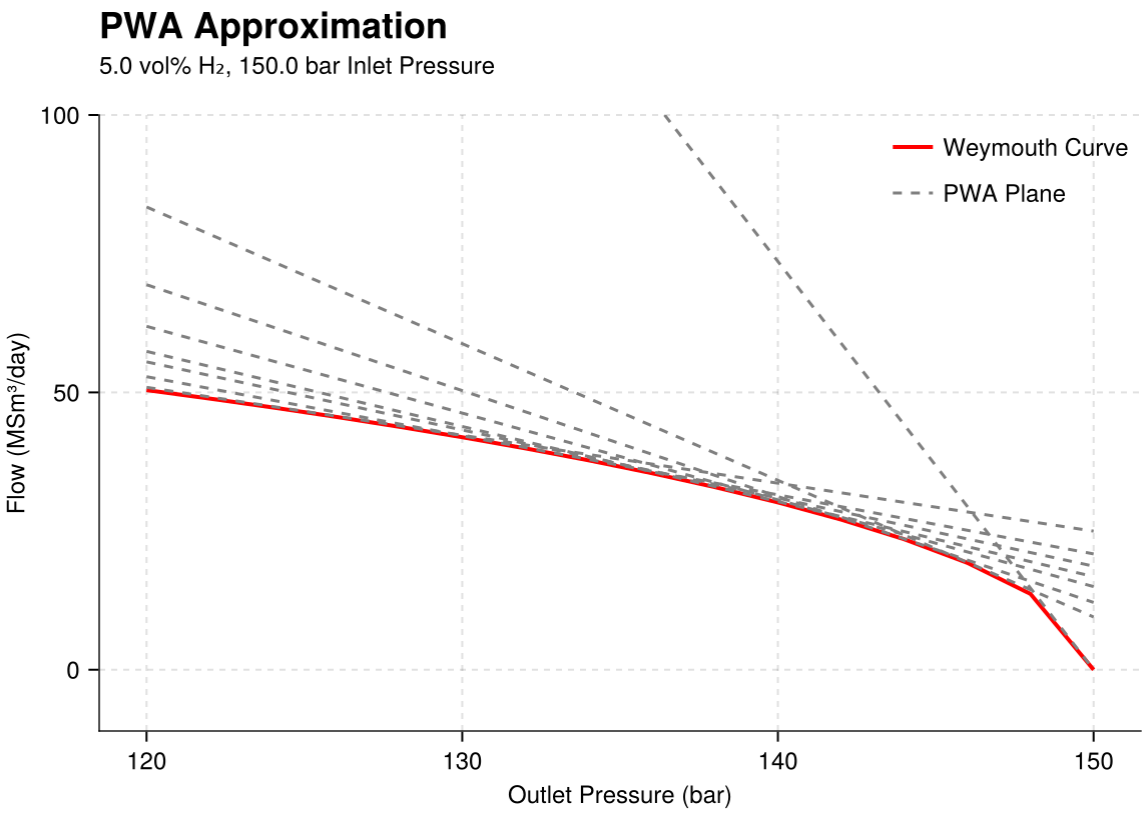

Figure 3. Example of piecewise-affine approximation of the Weymouth surface. The figure shows the flow with respect to the outlet pressure assuming 0.05 hydrogen ratio, W=0.3 and a fixed inlet pressure of 150 bars

To summarise, `EnergyModelsGasNetworks` is able to determine whether to apply the Taylor approximation or the PWA depending on the type of gas (i.e., blend or single gas) flowing through a pipeline.

3.4 Validation

The approximation methodology was validated against detailed simulation models to verify that the resulting operating points satisfy the physical pressure-flow relation. First, the model was used compared to standard/basic testing from academic relevant cases like the Haverly case [8]. Then it was validated using simulation results from SHIMMER++[5], a simulation model for admixing developed by Politecnico di Torino in SHIMMER.

Three cases for validation were used: only methane, uniform blend in the network (same proportion hydrogen-natural gas across the system) and admixing. In all the cases, the relative errors in flows and pressures are below 1% compared to the simulation results.

[5] https://github.com/shimmerhydrogen/shimmer.git

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

Examples of results from the validation study are shown below. Note that for the validation the pressure differences have been fixed to force similar results, as the optimization model in many cases will suggest different setpoints for pressures and flows, making comparison difficult. In all cases the results from EMX are shown with lines in the following plots, while results from SHIMMER++ from PoliTO are shown with dots.

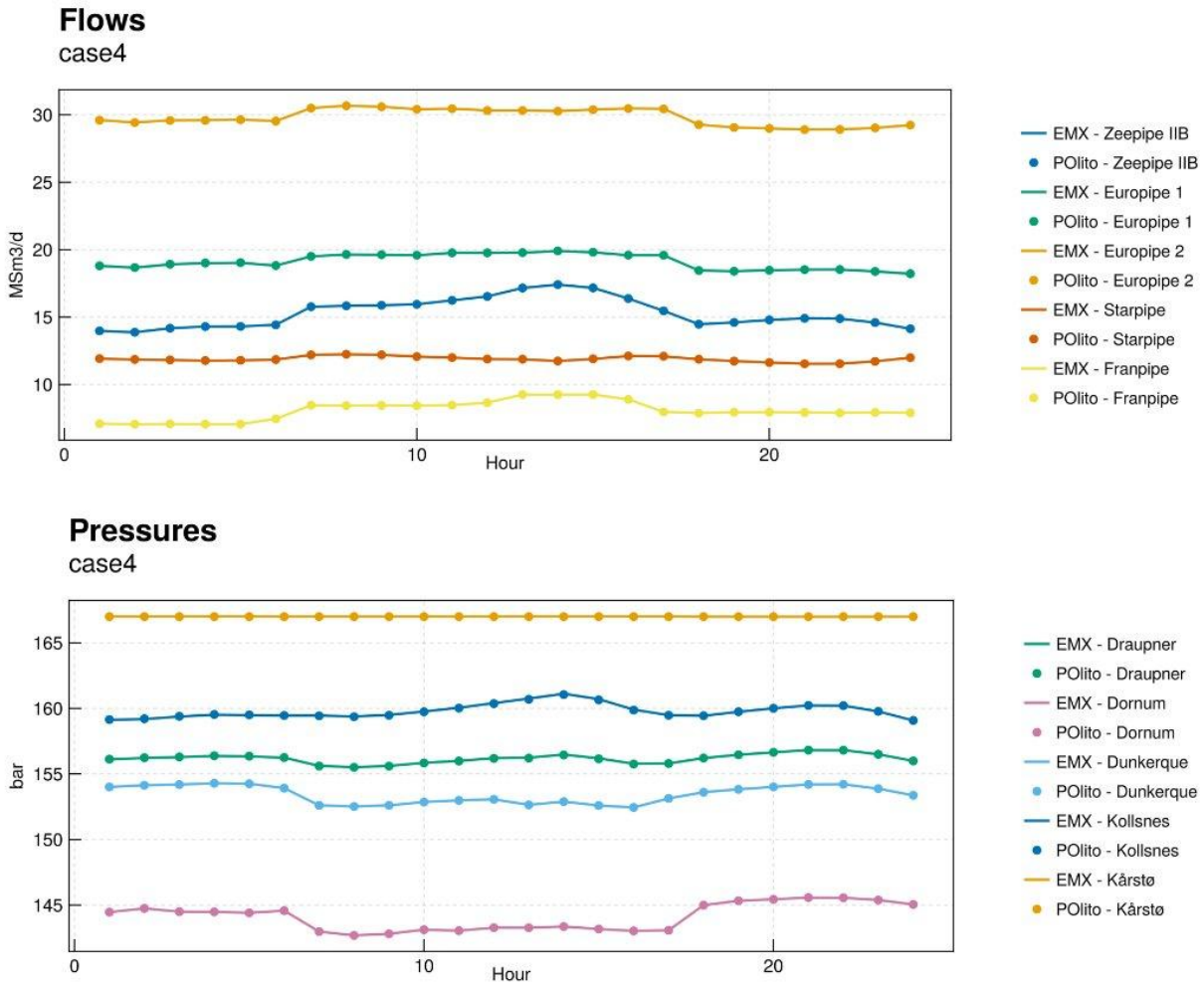

Figure Comparison results between EMX and SHIMMER++ for pure hydrogen.

For the validation of pure hydrogen flows, we can see that with fixed pressures the results in hours 1-24 are very similar across the pipelines, each shown with distinct colours:

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

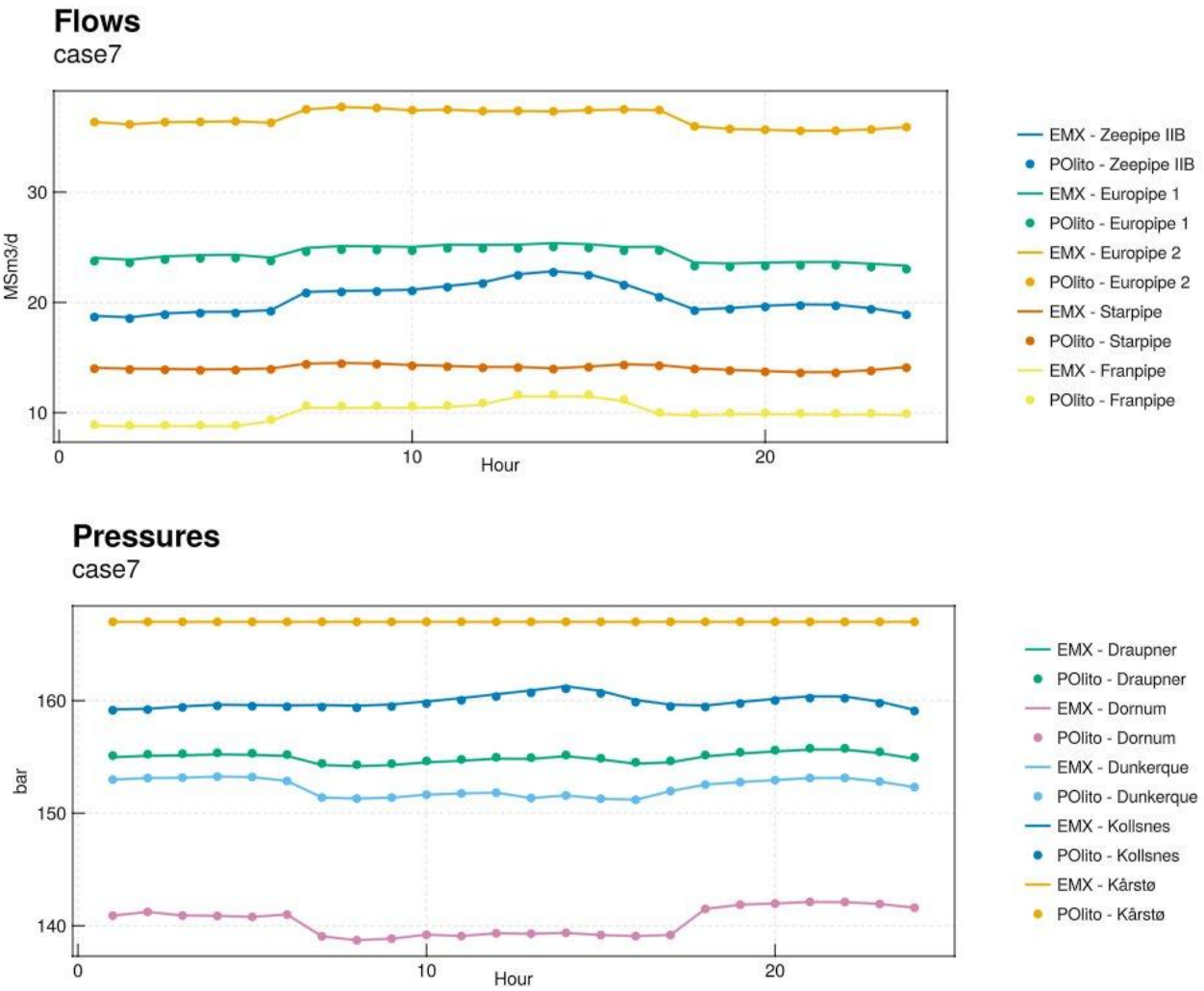

Figure 4 Comparison results between EMX and SHIMMER++ for fixed hydrogen percentage (5% at Kollsnes).

Similar results can be seen for a test case where the proportion of hydrogen admixing is fixed to 5% hydrogen at the injection point at Kollsnes:

It should be noted that simulation models tend to have a lower number of degree freedom than optimisation models, meaning that the amount of input parameters for these models is typically larger than for optimisation models that should decide the optimal decision. For example, a simulation of a single pipe might require determining the inlet and outlet pressures at all times in order to observe the resulting flows and hydrogen percentage flowing throughout the pipe. Nonetheless, the optimisation model, with a specific objective function that could be, for instance, maximise revenues costs, will only require defining minimum and maximum pressures. The model will then determine which pressure drop is the most optimal to achieve the objective.

4 Implementation

`EnergyModelsGasNetworks` expanded EMX by including the following elements:

4.1 Resources

- `ResourcePressure` – Resource that enables the activation of pressure/potential variables and constraints

- `ResourcePooling` – Resource that represents a blend of subresources of type `ResourcePressure` or EnergyModelsBase.`ResourceCarrier`

4.2 Nodes

- `SimpleCompressor <: Compressor` – Compressor that adds a pressure increase and has variable cost a variable cost associated with the inflow capacity
- `PoolingNode` – NetworkNode that guarantees the inflow of resources lead to an outflow of blended resources (`ResourcePooling`)
- `RefConversion` – Auxiliary node defined to convert units (e.g., from MSm3/d to MWh/d)

4.3 Links

- `CapDirect` – A direct link between nodes that includes maximum capacity and allows activating pressure-flow constraints.

4.4 Data structures

- `PressureLinkData` – Data to include in links to model the flow-pressure behaviours. Fields include the Weymouth constant and the maximum and minimum potential to consider when performing the approximations
- `FixPressureData`, `MaxPressureData` and `MinPressureData` – Data for nodes or links to set a fixed pressure or set minimum and maximum bounds. The value is set to the outlet of the node, except in Sink nodes which applies to the inlet.
- `RefBlendData` – Data for controlling the quality of the nodes.
- `BlendLinkData` – Data for including necessary information for correctly setting the approximations.

5 Installation

To start using `EnergyModelGasNetworks` one can follow these steps:

1. Install the most recent version of Julia, freely available from the Julia website[6]
2. Install the package EnergyModelsBase[7] and the time package TimeStruct[8] and JuMP[9], by running:

   ```
   ] add TimeStruct EnergyModelsBase JuMP
   ```

   These packages are required as we do not only use them internally, but also for building the model. It is hence necessary to load it in each model run explicitly.
3. Install the package EnergyModelsGasNetworks10

[6] https://Julialang.org/downloads/

[7] https://energymodelsx.github.io/EnergyModelsBase.jl/

[8] https://sintefore.github.io/TimeStruct.jl/

[9] https://github.com/jump-dev/JuMP.jl/

[10] https://energymodelsx.github.io/EnergyModelsGasNetworks.jl/

```
] add EnergyModelsGasNetworks
```

4. Install a suitable solver 'EnergyModelsX' models are in general agnostic towards which solver is used. They are not automatically included. Therefore, they require you to explicitly load the corresponding solver. Some constraints defined in EnergyModelsGasNetworks are nonlinear and nonconvex

    a. **Pooling constraints.** The pooling formulation introduces bilinear terms (nonconvex quadratic), defining a nonconvex QCQP (and, if integer variables are present, a nonconvex MINLP).

    b. **Pressure-flow constraints for blended gases.** The pressure-flow relation for blended gases is also nonlinear and nonconvex. However, within the physically operating region (i.e., inlet pressures higher or equal to outlet pressure, nonnegative flows and bounded pressures), the relations is smooth and monotonic. To obtain a MILP, we approximate the nonlinear surface using a piecewise-affine outer approximation built from tangent planes via PiecewiseAffineApprox[11].

    Consequently, when pooling and/or pressure-flow constraints are enabled, you need a solver capable of handling nonconvex MINLP problems, such as Alpine[12], which needs a a MIP solver and a continous NLP solver internally. For instance:

```
] add Alpine, Ipopt, HiGHS
using JuMP, Alpine, Ipopt, HiGHS
ipopt = optimizer_with_attributes(Ipopt.Optimizer, "print_level" => 0)
        highs = optimizer_with_attributes(HiGHS.Optimizer, "output_flag"
=> false)
model = Model(
            optimizer_with_attributes(
                Alpine.Optimizer,
                "nlp_solver" => ipopt,
                "mip_solver" => highs,
            ),
        )
```

6 Tutorial

The following tutorial provides a step-by-step walkthrough of the case `"examples/haverly.jl"` to show how to set up `EnergyModelsGasNetworks` for solving the admixing problem.

The network in this example, showed in Figure 5.1 consists of:

- Three gas sources with fixed outlet pressures (130 bars), two for methane (nodes #1 and #2) and the other for hydrogen (node #3). Each gas source has an associated cost per unit of gas injected into the network.

[11] https://sintefore.github.io/PiecewiseAffineApprox.jl/stable/

[12] https://github.com/lanl-ansi/Alpine.jl

- Three compressors (nodes #7 to #9) at the injection point to increase the gas pressure. Each compressor requires power for functioning and has a maximum capability of pressure increase of 50 bars.
- Power source for compressors (node #10). This node imposes a penalty/cost to ensure the pressures are not raised more than needed.
- Pooling node (node #4)
- Two gas sinks (nodes #5 and #6) with minimum inlet pressure (130 bars) and hydrogen quality constraints.
- Pipeline connections. For the sake of simplicity, all the pipelines are assumed to share the same Weymouth constant, $W = 0.24$. The pipelines in slash lines are simplified pipes where flow-pressure relationships are not modelled, they just allow the transmission of pressures between nodes. As such, the outlet pressure of the sources equals the inlet pressure of the compressors.

The main objective is to determine the optimal gas flow distribution while respecting pressure and pooling constraints and minimizing operational costs.

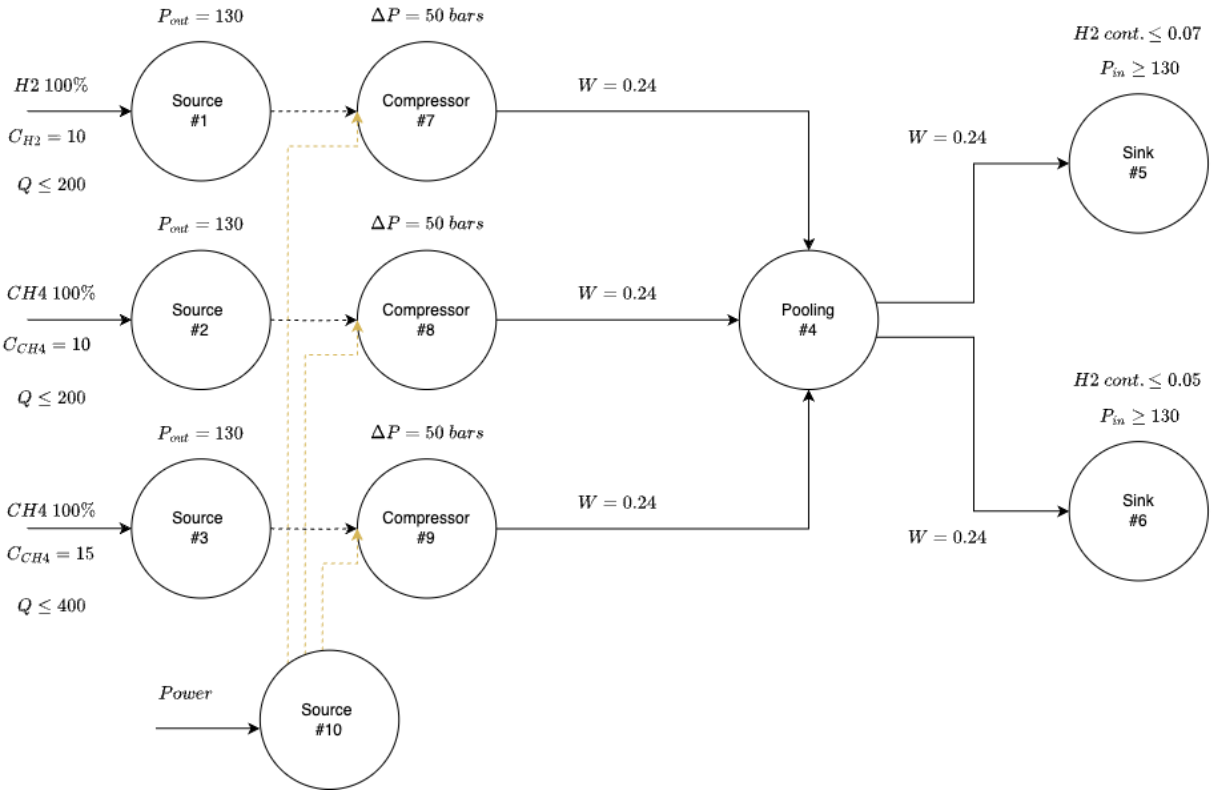

Figure 5. Network considered in the tutorial

6.1 Step 1: Environment Setup

First, we activate the local environment and load the required packages:

```
using Pkg
    Pkg.activate(joinpath(@__DIR__))
```

```
using HiGHS
using JuMP
using EnergyModelsBase
using EnergyModelsGasNetworks
using TimeStruct
using PrettyTables
```

We define convenient aliases for frequently used packages:

```
const EMB = EnergyModelsBase
const EMGN = EnergyModelsGasNetworks
const TS = TimeStruct
```

6.2 Step 2: Resource, Node and Link Definition

We define the function `generate_haverly_case()` to generate the necessary data for the case.

First, we define four main resources within the system to instantiate the model.

```
H2 = ResourcePressure("H2", 1.0)
CH4 = ResourcePressure("CH4", 1.0)
Gas = ResourcePooling("Gas", [H2, CH4])
Power = ResourceCarrier("Power", 0.0)
CO2 = ResourceEmit("CO2", 1.0)
products = [CO2, Gas, H2, CH4, Power]
```

The resources `H2` and `CH4` are subresources of type `ResourcePressure` as this type ensure the activation of flow-pressure constraints in the model. In networks where these constraints are not necessary (e.g., `examples/pressure.jl` the resources can be defined as the standard EMB type `ResourceCarrier`.

The resource `Gas` of type `ResourcePooling` is composed by H2 and CH4 and will ensure the activation of the pooling constraints. The CO2 and the list products are standard requirements for the correct generation of the model in EMX.

In order to control the time dimension, EMX utilises the library TimeStruct.jl[13] developed by SINTEF Industry. For this tutorial, we model a single operational timestep. For more operational times or long-time horizons (e.g., when modelling investments) other time structures can be used. Information about the usage of the library is extensively documented in its GitHub repository.

```
op_duration = 1
    op_number = 1
    operational_periods = TS.SimpleTimes(op_number, op_duration)
    op_per_strat = op_duration * op_number

    T = TS.TwoLevel(1, 1, operational_periods; op_per_strat)
```

The next step is to define the individual nodes as showed in Figure 5.

The three gas sources with different input resource and operational costs can be defined as:

[13] https://sintefore.github.io/TimeStruct.jl/stable/

```
RefSource(
    "source_h2_1",                          # Node identifier
    FixedProfile(200),                      # Maximum flow rate (MSm3/d)
    FixedProfile(10),                       # Variable OPEX (EUR/(MSm3/d))
    FixedProfile(0),                        # Fixed OPEX (EUR/(MSm3/d)/a)
    Dict(H2 => 1),                          # Output resources
    [FixPressureData(FixedProfile(130))],   # Fixed outlet pressure (bar)
)
```

The three sources have the following characteristics:

- **source_h2_1:** Variable OPEX = 10 EUR/(MSm^3/d), fixed pressure = 130 bar, capacity = 200 MSm^3/d
- **source_ch4_2:** Variable OPEX = 10 EUR/(MSm^3/d), fixed pressure = 130 bar, capacity = 200 MSm^3/d
- **source_ch4_3:** Variable OPEX = 15 EUR/(MSm^3/d), fixed pressure = 130 bar, capacity = 400 MSm^3/d

Three identical compressors are defined to increase gas pressure in the inlets:

```
SimpleCompressor(
    "compressor_1",                         # Node identifier
    FixedProfile(1e6),                      # Maximum flow rate (MSm3/d)
    FixedProfile(0),                        # Variable OPEX (EUR/(MSm3/d))
    FixedProfile(0),                        # Fixed OPEX (EUR/(MSm3/d)/a)
    Dict(CH4 => 1, Power => 1),           # Input resources
    Dict(CH4 => 1),                         # Output resources
    FixedProfile(50),                       # Max pressure difference (bar)
    [MaxPressureData(FixedProfile(190))],   # Max outlet pressure (bar)
)
```

All compressors are assumed to have a maximum pressure increase of 50 bars with a maximum outlet pressure of 190 bars. There is no OPEX nor flow restrictions. Note that in this version of `EnergyModelsGasNetworks` the compressors are used to define the minimum pressure increase that allow meeting our objective. Further extensions can focus on more detailed power consumption of compressors given they are a particular expensive asset for TSOs.

A power source supplies energy to the compressors:

```
RefSource(
    "power_compressors",
    FixedProfile(1e6),      # Maximum energy input to the compressors in MWh/d
    FixedProfile(0.3),      # Variable OPEX in €/MWh
    FixedProfile(0),        # Fixed €/(MWh)/a
    Dict(Power => 1),       # Output resource
)
```

The penalty to ensure the correct behaviour of the compressor is defined with the variable OPEX of the power source, such that increasing pressure will consume power and therefore has a cost in the objective function.

Finally, the terminals or sinks represents the gas demand:

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

```
RefSink(
    "sink_1",                           # Node 5 - Sink node
    FixedProfile(500),                  # Required demand in MSm3/d
    Dict(:surplus => FixedProfile(0), :deficit => FixedProfile(300)), # Surplus
and deficit penalty for the node in €/(MSm3/d)
    Dict(Gas => 1),                     # Demanded resource and its correspondi
ng coefficient for the node
    [
        RefBlendData(
            Gas,                        # `ResourcePooling`
            Dict(H2 => 0.07, CH4 => 1.0),   # Maximum allowed pooling shares of
the subresources of the `ResourcePooling` (volumetric %)
            Dict(H2 => 0.0, CH4 => 0.0)),   # Minimum required pooling shares o
f the subresources of the `ResourcePooling` (volumetric %)
        MinPressureData(FixedProfile(130))            # Minimum required pressu
re in bars
    ]),
```

This sink has a demand of 500 MSm3/d for the represented timestep for the blended resource Gas. The penalties are set such that deficits are heavily penalised while surplus are not rewarded. In this network, we also define a minimum inlet pressure of 130 bars and a maximum quality bound for the hydrogen of 7 vol-%.

The other sink in the network demands 300 MSm3/d of Gas at a hydrogen quality maximum to 5 vol-%

Finally, we can define the links connected between nodes:

```
Direct("source_1_compressor_1", nodes[1], nodes[7], Linear()),
```

The links source-compressor and power source-compressors (see dashed lines in Figure 5.1) are defined with the standard EMB link `Direct`. This is a type of link that allows resource transmission and ensures that the outlet pressure of the origin node equals the inlet pressure of the destination node.

The rest of the links are defined similarly to:

```
CapDirect(
    "compressor_1_pool_1",
    nodes[7],
    nodes[4],
    Linear(),
    FixedProfile(200),
    [PressureLinkData(0.24, 180, 130)]), # (weymouth constant, maximum pressure
and minimum pressure for approximation)

 CapDirect(
    "pool_1_sink_1",
    nodes[4],
    nodes[5],
    Linear(),
    FixedProfile(200),
    [
        PressureLinkData(0.24, 180, 130), # (weymouth constant, maximum pressur
```

```
e and minimum pressure for approximation)
        BlendLinkData(
            Gas,
            Dict{ResourcePressure{Float64},Float64}(H2=>2.016), # Molar mass of
tracking resource
            0.1, # Maximum proportion of the tracking resource
            0.0, # Minimum proportion of the tracking resource
            Dict{ResourcePressure{Float64},Float64}(CH4=>16.04), # Other resour
ces in the blend and their molar mass
        )
    ]),
```

The `CapDirect` link type was developed as part of `EnergyModelsGasNetworks` to model the flow-pressure relationships. In the first type of the example, the resource `H2` is transported, so the necessary information for the Taylor approximation (e.g., Weymouth constant, maximum pressure and minimum pressure) are required. The second link, however, transfer `Gas` implying that additional information like molar masses are necessary for a correct determination of the piecewise-affine approximation of the Weymouth equation.

6.3 Step 3: Case and Model Creation

We assemble all components into a case structure:

```
case = Case(T, products, [nodes, links], [[get_nodes, get_links]])
model = OperationalModel(
    Dict(CO2 => FixedProfile(0)), # emission limits
    Dict(CO2 => FixedProfile(0)), # emission costs
    CO2,
)
```

The `OperationalModel` specifies emission limits and costs (in this case, zero limits and costs as emissions are not the focus).

6.4 Step 4: Optimize

Once the `case` and `model` elements are generated, we can proceed to optimize our network. First, we will need to define an optimizer. In this case, as we require pooling and flow-pressure constraints we set the optimizer using `Alpine`, `Ipopt`, and `Juniper` as well as a MIP solvers like `HiGHS` or `Xpress`.

```
    nl_solver = optimizer_with_attributes(Ipopt.Optimizer, MOI.Silent() => true
, "sb" => "yes")
    minlp_optimizer = optimizer_with_attributes(
        Juniper.Optimizer,
        MOI.Silent() => true,
        "mip_solver" => mip_solver,
        "nl_solver" => nl_solver,
    )
    optimizer = optimizer_with_attributes(
        Alpine.Optimizer,
        "nlp_solver" => nl_solver,
        "mip_solver" => mip_solver,
        "minlp_solver" => minlp_optimizer,
```

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

```
        "rel_gap" => 0.01,
        "presolve_bt" => false,
        "time_limit" => 300,
    )
```

We need to define the optimizer for calculating the piecewise-affine approximation using the function `set_optimizer_pwa!(mip_solver)`. Then, the model can be solved:

```
set_optimizer_pwa!(optimizer_with_attributes(Xpress.Optimizer, MOI.Silent() =>
true))
m = create_model(case, model; check_timeprofiles = true)
set_optimizer(m, optimizer)
optimize!(m)
```

Note that MIP solvers like `Xpress` might require a license. An open source solver is for instance `HiGHS`. The selection of the solver will have a large impact on the solving times of the model.

6.5 Step 5: Results Processing

Once the model is solved, we can proceed to extract the results.

First we can analyse the gas delivered to each of the terminals. In this case, the same flows reach of the terminals (44.7 MSm3/d), with the same hydrogen percentage of 5 vol-%. This percentage is restricted by sink #2 and the fact that both sinks receive their flow from the same pooling node. The two pipelines reaching the sinks from the pooling node have an inlet pressure of 157.7 and outlet of 130 (set as the minimum pressure allowed in the sinks). For the given pressures and the hydrogen percentage the tangent planes from the PWA results in 44.73 MSm3/d while applying the Weymouth equation will result in 44.72 MSm3/d (0.02% relative error).

The hydrogen source delivers 4.47 MSm3/d while the two other sources 42.49 MSm3/d. Both sources deliver the same amount despite the cost difference as they are exporting as much as possible within the pressure restrictions of the compressors. The compressors #8 and #9 work at its maximum pressure increase of 50 bars (from 130 to 180), while compressor #7 increases the pressure from 130 to 158 bars for delivering a total flow of hydrogen of 4.47 MSm3/d.

The repository EnergyModelsGasNetworks contains several examples (‘examples‘ folder) that illustrate the steps to follow to for different cases.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

7 References

[1] L. Hellemo, E. F. Bødal, S. E. Holm, D. Pinel, and J. Straus, 'EnergyModelsX: Flexible Energy Systems Modelling withMultiple Dispatch', *J. Open Source Softw.*, vol. 9, no. 97, p. 6619, May 2024, doi: 10.21105/joss.06619.

[2] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, 'Julia: A Fresh Approach to Numerical Computing', *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, Jan. 2017, doi: 10.1137/141000671.

[3] M. Lubin, O. Dowson, J. D. Garcia, J. Huchette, B. Legat, and J. P. Vielma, 'JuMP 1.0: recent improvements to a modeling language for mathematical optimization', *Math. Program. Comput.*, vol. 15, no. 3, pp. 581–589, Sep. 2023, doi: 10.1007/s12532-023-00239-3.

[4] M. Alfaki and D. Haugland, 'A multi-commodity flow formulation for the generalized pooling problem', *J. Glob. Optim.*, vol. 56, no. 3, pp. 917–937, Jul. 2013, doi: 10.1007/s10898-012-9890-7.

[5] Alfaki, Mohammed, 'Models and Solution Methods for the Pooling Problem', PhD, University of Bergen, 2012. [Online]. Available: https://hdl.handle.net/1956/5847

[6] F. Rømo, A. Tomasgard, L. Hellemo, M. Fodstad, B. H. Eidesen, and B. Pedersen, 'Optimizing the Norwegian Natural Gas Production and Transport', *Interfaces*, vol. 39, no. 1, pp. 46–56, Feb. 2009, doi: 10.1287/inte.1080.0414.

[7] A. B. Galyas, L. Kis, L. Tihanyi, I. Szunyog, M. Vadaszi, and A. Koncz, 'Effect of hydrogen blending on the energy capacity of natural gas transmission networks', *Int. J. Hydrog. Energy*, vol. 48, no. 39, pp. 14795–14807, May 2023, doi: 10.1016/j.ijhydene.2022.12.198.

[8] C. A. Haverly, 'Studies of the behavior of recursion for the pooling problem', *ACM SIGMAP Bull.*, no. 25, pp. 19–28, Dec. 1978, doi: 10.1145/1111237.1111238.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.

The research leading to these results has received funding from Horizon Europe, the European Union's Framework Programme for Research and Innovation under grant agreement n° 101111888.