Stability of Dynamic Fluid Transport Simulations with Mixing Flows

Mehrnaz Anvari
Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Sankt Augustin, Germany
email: Mehrnaz.Anvari@scai.fraunhofer.de

Anton Baldin
PLEdoc GmbH and Fraunhofer Institute SCAI
Sankt Augustin, Germany
email: Anton.Baldin@scai.fraunhofer.de

Tanja Clees
University of Applied Sciences Bonn-Rhein-Sieg and Fraunhofer Institute SCAI
Sankt Augustin, Germany
email: Tanja.Clees@scai.fraunhofer.de

Bernhard Klaassen
Fraunhofer Research Institution for Energy Infrastructures IEG
Bochum, Germany
email: Bernhard.Klaassen@ieg.fraunhofer.de

Igor Nikitin
Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Sankt Augustin, Germany
email: Igor.Nikitin@scai.fraunhofer.de

Lialia Nikitina
Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Sankt Augustin, Germany
email: Lialia.Nikitina@scai.fraunhofer.de

***Abstract*—This work presents physically based simulation of energy distribution and substance composition for dynamic fluid transport problems. The main addressed problem is the stability of the algorithms for solving the resulting systems of differential-algebraic equations. The challenges encountered include system degeneration, the appearance of stochastic degrees of freedom, jumps in thermodynamic functions during phase transitions, and proper scaling of equations. The proposed solution is the identification and optimal configuration of solver parameters, strongly affecting the stability and speed of the simulation. Such parameters include regularizing and weighting constants, dimensioning of dynamic terms and startup procedure, the size of the integration step and their total number. The main output of the paper is the optimal choice of these parameters that allows to speed up significantly the dynamic simulation of fluid transport for realistically large network scenarios.**

Keywords-simulation and modeling; mathematical and numerical algorithms and methods; mixing flows; pipeline fluid transport; stability.

I. Introduction

This work extends the results of our conference paper [1], which considered modeling of mixing flows in dynamic fluid transport simulation. The extension includes a more precise implementation of heaters and coolers, as well as a detailed stability analysis of dynamic simulation with mixing flows.

The contributions of the study: this paper continues a series of our works on modeling of fluid transport networks. Previous works presented stationary [2] and dynamic [3] modeling of fluid transport networks limited to a single chemical composition and constant temperature. In addition, some aspects of stationary modeling of mixing fluids of different compositions and/or temperatures were considered in [4]. In this paper, flow mixing modeling will be considered in more detail, with special emphasis on the thermodynamic layer of the model. In particular, dynamic mixing equations and algorithms for their solving will be presented. The developed approach is implemented in our Multi-phYsics Network Simulator (MYNTS) [5], which is used to solve actual transport scenarios for natural gas [6], hydrogen [7], carbon dioxide [8], water [9] and other fluids.

State-of-the-art: fluid transport modeling is based on the conservation of mass flows in the form of dynamic Kirchhoff equations; Darcy-Weisbach pipeline pressure drop formula, with empirical friction term by Nikuradse [11] and Hofer [12]; equation of state computation by simplified analytical models by Papay [13], Peng-Robinson [14] and Soave-Redlich-Kwong [15] or more complex ISO-norm models AGA8-DC92 [16] and GERG2008 [17]–[19].

A number of previous studies [20]–[26] considered modeling of pipeline fluid transport, both at the universal mathematical level [20], and in various application scenarios. Such scenarios include transport of natural gas [21] [23], steam transport in oil refineries [22], carbon dioxide transport [24]–[26]. All these works are characterized by the presentation of transport equations as laws of conservation of mass, momentum and energy. In the presence of various substances, conservation of molar flows is added, while the general relations of thermodynamics of open systems [27] regulate the relations of energy and temperature.

The main problem: a common drawback of existing solutions is the closed nature of modeling within blackbox systems. If it is necessary to change the modeling, modify or introduce new equations and variables, the system must be reprogrammed. In addition, existing systems experience difficulties in solving large realistic network problems in the presence of numerical instabilities. The novelty of our approach consists in transparent modeling, where the user can freely change the equations and experiment with different forms of representing physical processes in fluid transport networks. We also pay special attention to the stability and performance of solution algorithms, which is especially important for realistic scenarios with a large number of elements.

The aim of this work: to extend transparent and numerically stable modeling to mixing flows present in realistic fluid transport scenarios. In our early works [2], [4]–[10] an

implementation for a stationary solver was considered. The main strategy for ensuring stability was a gradual sophistication of the modeling, from a pure pipe system with linear equations for control elements, constant temperature and fluid composition, to a full problem with nonlinear control elements and physical distribution of temperature and fluid composition. At each step, the solution was used as a starting point for the next step. The disadvantage of this approach is that simplified modeling does not always yield a physical solution and sometimes gives a bad starting point for the next iterations. Also, theoretically, direct solution of stationary equations does not always yield a limit point of a stable attractive type, it can also yield a repulsive or saddle point. Dynamic modeling automatically finds stationary points of the attracting type, and can also have richer asymptotics, including runaways, limit cycles and random behavior. All this means that the dynamic solver is advantageous, also for solving stationary problems. The key point of our research is to understand how to use the dynamic solver most optimally, in a stable operation mode. Previously [3] we studied only the pressure-massflow subsystem, with constant chemical composition and temperature. Now we study the stability of the dynamic solver for the full system, including mixing flows and temperature modeling.

In this work, Section II presents the modeling of mixing flows incorporating molar and temperature relationships. Section III describes the numerical experiments performed using the developed methods. Section IV considers extended modeling of heaters and coolers. In Section V, extended stability analysis of the full dynamical solver is performed. Section VI summarizes the main results and conclusions of the work.

II. Modeling of mixing flows

This section describes the details of modeling of mixing flows, consisting of modeling fluid molar composition and temperature distribution.

A. Molar fluid composition

A fluid transport network is described by a directed graph consisting of nodes and edges connecting them. The graph is described by an incidence matrix I_{ne}, in which each edge e has nonzero entries for the nodes n that this edge connects; -1 for the node that edge comes from, $+1$ for the node that edge enters. Mixing fluid flows are described by following equations

$$V_n \partial \rho_n / \partial t = \sum_e I_{ne} m_e, \quad (1)$$

$$V_n \partial (\rho_n \mu_n^{-1}) / \partial t = \sum_e I_{ne} m_e \mu_e^{-1}, \quad (2)$$

$$V_n \partial (\rho_n \mu_n^{-1} x_n) / \partial t = \sum_e I_{ne} m_e \mu_e^{-1} x_e, \quad (3)$$

where V_n is the volume assigned to the node; ρ_n represents the mass density at the node; t denotes time; the sum applies to all edges adjacent to the node; m_e is the mass flow in an edge, considered positive if the direction of flow coincides with the direction of the edge, and negative otherwise; $\mu_{n/e}$ is the molar mass assigned to both the node and the edge; $x_{n/e}$ are the mole fractions of the components that make up the fluid.

Physically, the above equations describe various conservation laws. In particular, (1) is the dynamic Kirchhoff equation and describes the conservation of mass. Here, $V_n \rho_n$ on the left side, with V_n representing a time-independent volume, describes the mass of fluid in the node. The sum on the right side accounts for the mass flow into the node, minus the flow out. Equation (2) describes the conservation of the total molar amount of a fluid, where $V_n \rho_n \mu_n^{-1}$ represents the number of moles in a node, and the sum on the right side is the total molar flow in the node. Finally, (3) describes the molar conservation for each component, $V_n \rho_n \mu_n^{-1} x_n$ represents the number of moles of a given component in a node, and the sum is the molar flow of that component. Equations (1) and (2) are valid in the absence of chemical reactions between the components of the fluid.

The x-vector may also include other quantities to which linear molar mixing applies, such as the molar heat value H_m, and linear approximations (T_c, P_c) used in certain equations of state for critical temperature and critical pressure, among others. Alternatively, such quantities can be calculated in post-processing as a linear combination over the molar composition. Explicit inclusion in the mixing equation allows these quantities to be calculated even when the determination of molar composition is disabled.

The conservation equations of type (1)–(3) are standard, can be found in a textbook, e.g., eq. (4.1) in [27]. Now we will rewrite them in a more convenient form, resolved with respect to derivatives:

$$V_n \rho_n \partial \mu_n^{-1} / \partial t = \sum\nolimits'_e I_{ne} m_e (\mu_e^{-1} - \mu_n^{-1}), \quad (4)$$

$$V_n \rho_n \mu_n^{-1} \partial x_n / \partial t = \sum\nolimits'_e I_{ne} m_e \mu_e^{-1} (x_e - x_n), \quad (5)$$

$$\sum\nolimits'_e = \sum\nolimits_{e, I_{ne} m_e > 0}, \quad (6)$$

where the sum is taken over the flows incoming to the node. To prove it, it is necessary to perform the differentiation in (2) and take into account (1), which will result in (4), in which the sums are taken over all flows, incoming and outgoing. Further, if one takes into account that μ_e^{-1} for an outgoing flow is equal to μ_n^{-1} at a node, the sum can be reduced to the incoming flows. The proof for (5) is similar. The condition of equality of mixed quantities in the node and in the outgoing flow can also be used to reduce the total number of variables. Namely, one can completely eliminate the variables in the edge e, replacing them with the values in the upstream node n', $\mu_e^{-1} \to \mu_{n'}^{-1}$, $x_e \to x_{n'}$. When time derivatives are set to zero, these equations are reduced to stationary formula (see eq. (13) in [4]).

Boundary conditions: $\mu = \mu_{set}$, $x = x_{set}$ are fixed to the specified values in the network entry nodes. The system of (4)–(6) and boundary conditions is closed. Its stationary part on the right side of the equations is non-degenerate if all nodes are connected to at least one entry node in the upstream direction. A complete dynamical system can be non-degenerate even if

this rule is violated, for example, if all flows are zero. In this case, the dynamic term ensures the preservation of the transported quantities, keeping them at the starting values.

Startup algorithm: at entry nodes, the transported values are initialized to set values to satisfy the boundary conditions. In all other nodes, values are initialized to default values, which are either specified by the user or averaged over all set values. As a part of the general procedure [3], the initial pressures are set to a constant, the initial flows are set to zero and all fluid composition-dependent quantities, such as density ρ, are calculated from the appropriate equations of state. This procedure provides a smooth startup, with all equations initially satisfied. Then, fluid starts to propagate from entries to the neighbor nodes with growing massflow, replacing default values with current ones.

V_n-definition: in accordance with the discretization scheme formulated in [3], each pipe contributes half of its volume to the end nodes, and all other elements contribute a nominally specified volume V_0.

Linearity of the system: with known m-flows, the μ^{-1}-subsystem (4) is linear; also, for known m and μ^{-1}, the x-subsystem (5) is linear. This property is convenient for controlling convergence, since each linear subsystem in the non-degenerate case is solvable in one iteration. The following algorithm is used to integrate the equations.

Algorithm (simulation workflow):

```
init;
repeat{ mumix; xmix; Tmix; PM; t+=dt; }
```

Here, `init` represents the initialization of all variables according to the startup algorithm described above. `mumix` is the solution of the μ^{-1}-subsystem, `xmix` is the solution of the x-subsystem, `Tmix` is the solution of the temperature subsystem formulated below, and `PM` is the solution of the pressure-massflow subsystem as formulated in [3]. In this way, it is possible not only to find the dynamic evolution of the system, but also to determine the stationary solution. For the last goal, it is necessary to integrate the system with as large steps as possible until stationarity is achieved. The most stable method suitable for this purpose is time discretization of the implicit Euler type: $\partial v/\partial t \to (v - v_{prev})/dt$, for all dynamic variables v, where v_{prev} is the value from the previous step, dt is the integration step. For a detailed study of dynamic processes, more sophisticated finite-difference schemes [28] [29] can be used.

B. Temperature modeling

The starting point is the law of conservation of energy for open systems (see, for example, eq. (4.14) in [27]):

$$V_n \partial(\rho_n \mu_n^{-1} U_n)/\partial t = \textstyle\sum_e I_{ne} m_e \mu_e^{-1} H_e, \qquad (7)$$

where U is the molar internal energy, $H = U + P\mu/\rho$ is the molar enthalpy, and P is the pressure. The equation is similar to the conditions of molar mixing in (3). The difference is that the derivative of the nodal internal energy is on the left side, and the total enthalpy flow in the node is on the right side. Physically, with each flow, internal energy is introduced into the node, as well as the work of the fluid against the pressure in the node. This work can be combined with internal energy, giving enthalpy on the right side of the equation. On the left side, under the derivative, there is still nodal internal energy. In general case, other terms can be present in the conservation law, vanishing for simple mixing in the node. In particular, no additional work is performed in the node, and due to the assumed absolute thermal insulation of the node, heat transfer becomes zero. Possible processes with additional work and heat transfer are assigned to special edge elements and are described below.

We rewrite equation (7) as follows:

$$V_n \rho_n \mu_n^{-1} \partial H_n/\partial t - V_n \partial P_n/\partial t = \\ = \textstyle\sum_e' I_{ne} m_e \mu_e^{-1} (H_e - H_n), \qquad (8)$$

the derivation is similar to (5), also here the nodal internal energy is re-expressed in terms of enthalpy and pressure in the node.

Boundary conditions: $H = H_{set}$, enthalpy is fixed to the specified value in entry nodes. Alternatively, one can use the condition $T = T_{set}$, which fixes the temperature at the entry nodes.

In addition, according to eq. (4.14) in [27], gravitational and kinetic terms can be added to the internal energy and enthalpy: $H \to H + \mu g h + \mu v^2/2$, where g is the acceleration of free fall, h is the height, and v is the speed of translational motion of the fluid. To calculate the kinetic term, one needs to know the diameter, which is not available for all types of elements. For example, a compressor is a very complex structure to be described by a single diameter. Also, at nodes where many edges join, complex internal motion occurs, which does not coincide with the simple translational motion described by a kinetic term with a single diameter. On the other hand, for the transport of gases, the kinetic term is usually significantly less than the internal energy, for translational velocities significantly lower than the speed of sound. In our simulation, we made it possible to optionally turn off the kinetic term in the temperature equations.

In (8), H_n represents the nodal value, and H_e represents the edge downstream value. The difference from x-mixing is that here the edge downstream value in the general case cannot be replaced by the upstream nodal value, since there are elements that change the enthalpy value. The system cannot be reduced to a purely nodal one; in addition, the system also includes the temperature T of the fluid.

HT-constraint:

$$H = H_{mod}(P, T, x), \qquad (9)$$

$$H = H_{mod}(P, T_{prev}, x) + c_p(T - T_{prev}), \qquad (10)$$

where H_{mod} is the thermodynamic model for enthalpy, $c_p = \partial H_{mod}/\partial T$ is the molar heat capacity calculated at point (P, T_{prev}, x). Equations (9)–(10) and (H, T) variables are introduced per node and edge.

The first equation relates enthalpy and temperature according to the thermodynamic model used. We use GERG2008 [17]–[19] as a concrete implementation of such relation. For software-technical reasons, it cannot be used directly; its call once per internal iteration produces too many total calls of GERG2008 module, resulting in significant slowdown. In addition, the equation is nonlinear, violating the desired linearity property of the Tmix subsystem. The second equation is a linearization of the first, it can be used in internal iterations, with a less frequent update of the coefficients. When using the workflow formulated above, (m, P, ρ) in all mix phases are considered as fixed parameters, updated in PM-phase. For H_{mod} and c_p, updates occur immediately before the start of the Tmix phase.

Default element equation:

$$H_e = m_e > 0?H_{n1} : H_{n2} \tag{11}$$

formulates isenthalpic process [27], where the edge enthalpy is taken from the upstream node, similar to x-mixing. In this and further equations, the edge e goes from node n_1 to node n_2, conditions are written in C-notation: *x?y:z = if(x) then y; else z*. This model is applied to the most of element types, in particular, to valves, regulators, resistors and shortcuts; while the exceptional types are listed below.

Pipe equation:

$$(m_e > 0?(H_{n1} - H_e)\mu_{n1}^{-1} : (H_{n2} - H_e)\mu_{n2}^{-1})|m_e| = \\ = \pi D L c_{ht}(T_e - T_{soil}), \tag{12}$$

the change of enthalpy over the pipe is equal to a heat exchange with the soil, eq. (33.3) in [30]. Here T_{soil} is soil temperature, D is pipe diameter, L is pipe length, and c_{ht} is heat transfer coefficient. The pipe should have sufficiently fine subdivision to model the heat exchange appropriately.

Compressor equation:

$$m_e > 0?(T_e - T_{n1}((|P_{n2}/P_{n1}|^{(\kappa-1)/\kappa} - 1)/\eta + \\ +1)z_{n1}/z_{n2}) : (H_e - H_{n2}) = 0, \tag{13}$$

for positive flow, the change of temperature is described by eq. (38.51) in [30], or a similar formula (eq. (13-31)) without z-correction from [31]; otherwise, isenthalpic process is used. Here κ is isentropic exponent, η is efficiency, z is compressibility factor. This basic model is designed for gas transport, while for liquids, e.g., CO_2 pumps, customer-specific models can be used.

Coolers and heaters:

$$m_e > 0?(A_{set} > 0?(T_e - T_{set}) : (H_e - H_{n1})) \\ : (H_e - H_{n2}) = 0, \tag{14}$$

at the simplest modeling level, we implement these elements by clamp formulas: $T_e = \min(T_{n1}, T_{set})$ for coolers and $T_e = \max(T_{n1}, T_{set})$ for heaters. These formulas are piecewise-linear. Their linearization leads to the common formula above and the active set flag described by the following algorithm.

Algorithm (active set):

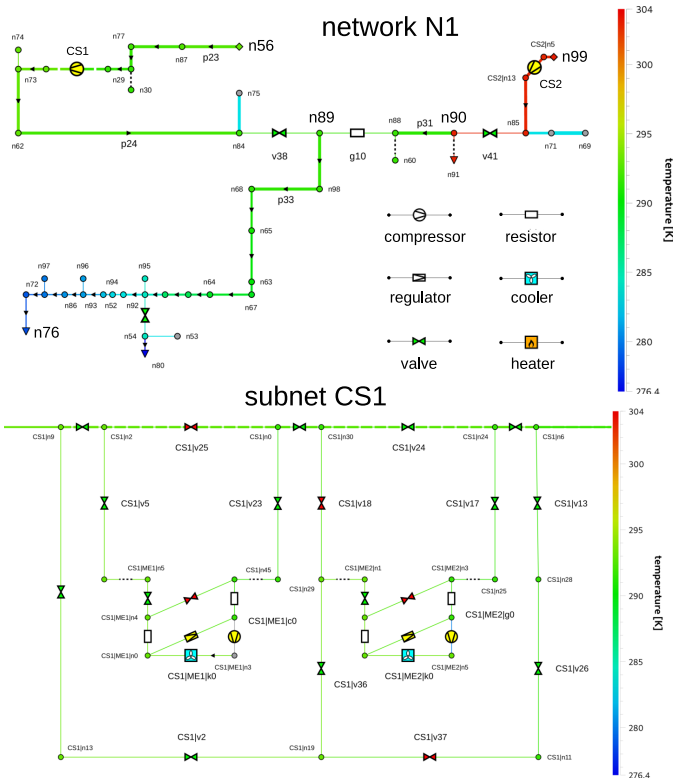

Figure 1. Test network N1.

```
cooler:
  if(Aset==1&&He>Hn1) then Aset=0
  if(Aset==0&&Te>Tset) then Aset=1
heater:
  if(Aset==1&&He<Hn1) then Aset=0
  if(Aset==0&&Te<Tset) then Aset=1
```

Here $A_{set} = 1$ corresponds to an active mode, $A_{set} = 0$ to a standby mode. The algorithm is applied after Tmix-phase, its convergence is tracked.

III. Numerical Experiments

We performed a series of simulations on networks of different complexity levels to study in detail the effects of flow mixing, integration stability, and iteration convergence.

N1 network: the network shown in Figure 1 contains 100 nodes, 111 edges and is used for numerical experiments with the transport of natural gas and hydrogen. Detailed settings of supplies in the considered scenario are presented in Table I. Selected time discretization is $dt = 3 \cdot 10^4 s$, $nsteps = 100$. The network has a simple Y-shaped topology, with two supply nodes n99_gm and n56_gm, as well as a mixing node n89, where the flows from the supplies come together, and the rest of the network, ending with the most distant exit node n76.

Figure 2a shows the evolution of inverse molar mass. Figure 2b presents molar heat value, and Figure 2c demonstrates molar fraction of CH_4, representative for chemical composition in the considered test scenario. In all these plots, the

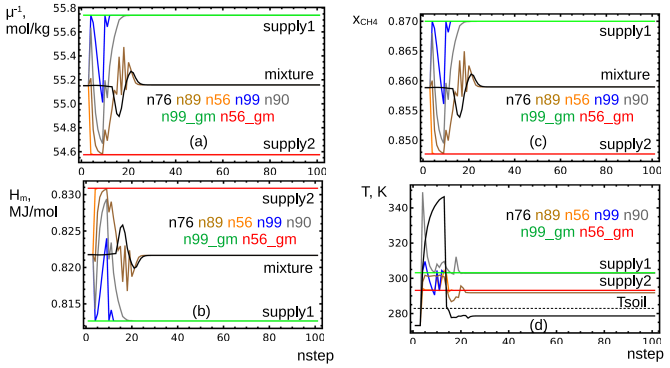

Figure 2. Simulation results (see text for details).

TABLE I
SUPPLY SETTINGS IN VARIOUS SCENARIOS

scenario	entry	composition	temperature
N1 nat.gas	n99_gm	87% CH_4, 1% C_2H_6, 1% C_3H_8, 1% CO_2, 10% N_2	$303.15K$
N1 nat.gas	n56_gm	85% CH_4, 3% C_2H_6, 1% C_3H_8, 1% CO_2, 10% N_2	$293.15K$
dyn-pipe H_2	n0000	95% H_2, 5% N_2	$313.15K$
dyn-pipe CO_2	n0000	95% CO_2, 3% N_2, 2% O_2	$313.15K$

TABLE II
TESTING VARIOUS IMPLEMENTATIONS OF HEATERS ON N85 NETWORKS SET

implementation of heaters	num. of divergent cases
disabled	3
local	0
nonlocal	85
joined	2

values in supply nodes n99_gm and n56_gm are kept constant at set values. In stationary solution, the simple topology of the network leads to a single mixed state, formed in node n89 and propagated downstream to the rest of the network. In the evolution, the values in all nodes tend either to supply values or to this mixed state. Interestingly, in the startup of the evolution, the curves perform several large oscillations between the boundary states, before they relax at the stationary state. This happens due to a complex distribution of flows at the startup phase.

Note that the graphs Figure 2a and Figure 2c have an identical shape, and Figure 2b has the same shape vertically reflected. This happens because there are only two supplies in the network, and the default composition is a linear combination of them. As a result, the trajectory of the system in x-space is limited to a 1-dimensional subspace. Graphs Figure 2a-c are projections of this trajectory to different directions and therefore have the same shape.

Figure 2d shows temperature dependence in selected nodes. During startup evolution, strong heating occurs due to the inverse Joule-Thomson (JT) effect and the influence of the $\partial P/\partial t$-term in (8). With further evolution, the temperature in nodes close to supplies tends to the corresponding constant temperature values of the incoming fluid. In more detail, in the considered scenario, after each supply there is a compressor station, the outlet temperature of which is regulated by a cooler. The outlet temperature of the cooler is set to the same value as that of the corresponding supply. The temperature in network nodes remote from the supply tends to a constant value, slightly below $T_{soil} = 283.15K$, due to the influence of the JT-effect.

N85 networks set: contains 85 realistic natural gas networks, obtained for benchmarking from our industrial partner. The networks are highly resolved, containing up to 4 thousands of nodes each. We used these networks for numerical experiments testing the stability of simulation with a different implementation of heaters. Unlike coolers, which usually control their own output temperature, heaters must control the temperature in an adjacent element, the regulator. In dynamic formulation of the problem, especially at low flows, heaters do not have time to regulate their temperature in order to constantly ensure the set temperature values in the regulator. This leads to divergences. We have tested several options for implementation of heaters, shown in Table II. For disabled heaters, 3 scenarios out of 85 are divergent. For the most stable implementation option, when heaters control their own local temperature, all scenarios are convergent. If the heaters try to control the temperature nonlocally, in the attached regulators, all scenarios diverge, making such implementation impossible. For our selected option, the heaters are joined with regulators, the unified element controls its own output temperature, 2 scenarios out of 85 are divergent, slightly better than the complete disabling of the heaters.

Hydrogen and carbon dioxide pipelines: this is one of our standard test cases, $L = 150km$, $D = 0.5m$ horizontally laid pipeline, transporting gaseous H_2 or CO_2 in liquid or supercritical phase. The case supports variable spatial discretization, for the considered scenario selected to $nsubdiv = 50$. Time discretization is the same as for N1 network. Supply setting is presented in Table I. The considered scenario has a single fluid composition and is used mainly for testing of the temperature modeling. The dynamic simulation starts from $T_{soil} = 283.15K$ and a different $T_{set} = 313.15K$ at the pipeline entry. The simulation converges to stationary solution with nearly exponential fall of temperature from T_{set} to T_{soil}. For CO_2, an observed stronger deviation from the exponent is due to JT-effect and the nonlinear enthalpy model.

Convergence of iterations: in our implementation, we use the globally convergent Newton's solver with Armijo line search rule [32], applied at every time step. For linear problems, it just forwards the solution to the underlying sparse linear solver, that for non-degenerate problems converges in 1 iteration. Due to proper initialization, at the first time step all phases converge in 0 iteration, just keeping the starting values. This provides a good method to test that all variables

are correctly initialized. At the second time step, all mix phases also converge in 0 iteration, while in the last PM phase the network filling begins, and PM phase starts to increase its iteration number. For N1 network and H_2/CO_2 pipe scenarios, all mix phases are solved in 1 iteration on intermediate timesteps, as it should be for non-degenerate linear systems; and in 0 iteration at the last timesteps, due to convergence to stationary solution. For large N85 networks, Tmix phase can have intermediately 2-3 iterations, indicating the remaining degeneracy or the disbalance of scaling factors in Tmix system. This effect will be studied in more details in Section V.

The numerical experiments performed show that the primary purpose of this work has been fully achieved, the modeling has been extended to include mixing flows and is working for scenarios of varying complexity. The modeling in our system is presented in open text form, as a list of variables and equations, which both we and the users can freely modify. This distinguishes us from the existing solutions, in which the modeling is usually hardcoded within the system. We also provide numerical stability of the modeling and the solution algorithms, which allows us to solve large realistic scenarios in fluid transport simulation.

IV. Extended modeling of heaters and coolers

The nonlocal control case is especially difficult to model, when the point with the controlled temperature is not at the heater or cooler output, but in another area of the network, for example, when a temperature sensor is placed there. As shown by the numerical experiments, a direct generalization of the control equations to this case is unstable. At the same time, there is a workaround with the transfer of the temperature control function to the element for which the controlled temperature is local. Although this approach works, it would be desirable to obtain a more realistic modeling, in particular, one that reproduces the correct intermediate temperatures between the heater/cooler and the sensor position. In this section, we consider the extension of modeling necessary for this.

The required diagram for the heater is shown in Figure 3a. It consists of three branches: on – the temperature at the controlled point is maintained at the required value: $T_c = T_{set}$, the heater is on: $T_e > T_{n1}$; standby – the temperature at the controlled point exceeds the required value: $T_c > T_{set}$, the heater is off: $T_e = T_{n1}$; an additional max branch is introduced – the temperature at the controlled point is less than the required value: $T_c < T_{set}$, the heater operates at maximum: $T_e = T_{max}$. The reason for introducing an additional branch is that in some cases the set control goal is unachievable. An example is a vanishingly small flow, when the contribution of the heated fluid from the heater has virtually no effect on the temperature at the controlled point. Also, due to a network configuration error, the controlled point may be outside the influence zone of the heater, for example, behind a closed valve. If there is no max branch in the control equation, then, in the case of a decrease in the controlled temperature below

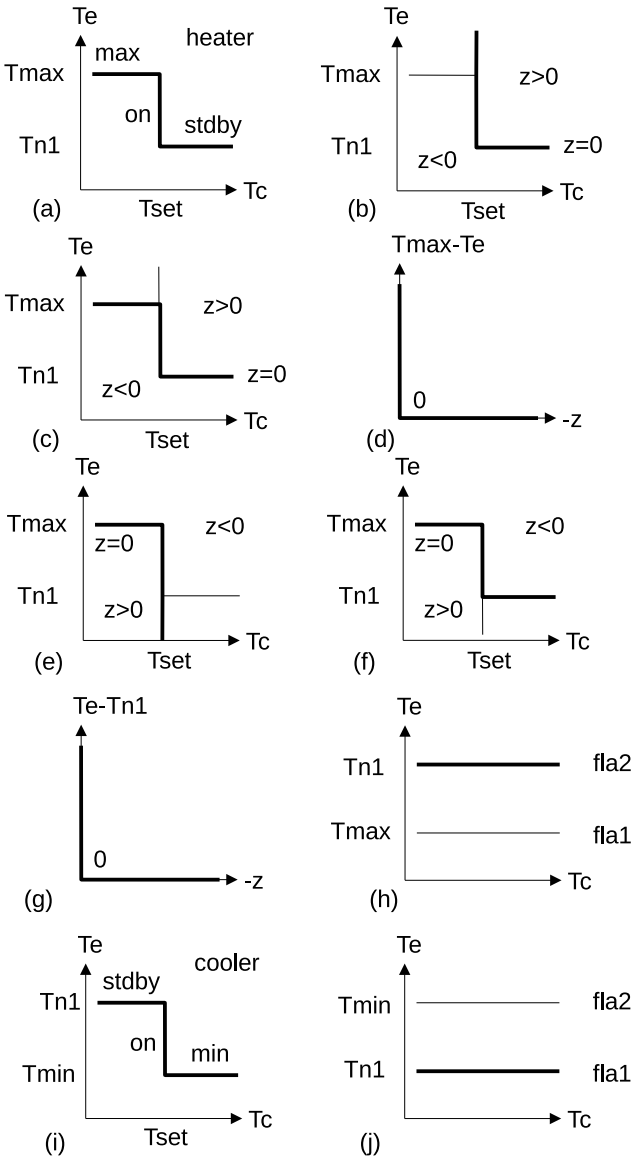

Figure 3. Construction of minmax formulas for heaters and coolers (see text for details).

the required value, the heater will try to heat the fluid more and more, eventually leading to simulation divergence. Introducing the max branch in this case gives a physically reasonable alternative scenario with a limited temperature. For cooler the modification is similar, here the min branch is introduced, as shown in Figure 3i. Extended control equations are as follows:

heater:

$$\max(\min(T_{set} - T_c, T_{max} - T_e), T_{n1} - T_e) = 0, \quad (15)$$

cooler:

$$\max(\min(T_c - T_{set}, T_e - T_{min}), T_e - T_{n1}) = 0, \quad (16)$$

where T_e is the local temperature in the heater/cooler; T_{n1} is the temperature at the heater/cooler inlet; T_c is the temperature at the controlled node/edge, at the sensor location; T_{set} is the

set temperature at that location; $T_{max/min}$ are the temperature limits at the heater/cooler, by default set to $T_{min} = 223.15K$, $T_{max} = 423.15K$. Note that the formulas are now piecewise linear rather than linear. It is not possible to preserve the overall linearity of the simulation, but the new simulation is more stable and does not require convergence of the active set iterations.

We will now provide a detailed derivation of the minmax formulas. Similar formulas are used in other parts of our simulation, and their derivation uses a similar procedure. First, let us consider the heater simulation, represented by the diagram in Figure 3a. Next, in Figure 3b the zero level of the function $z = \min(T_c - T_{set}, T_e - T_{n1})$ is marked with a bold line, dividing the plane into regions of positive and negative values of this function. In Figure 3c we break this line to the desired shape of the diagram, consisting of two pieces:

$$(z \le 0 \& T_e = T_{max}) | (z = 0 \& T_e \le T_{max}). \quad (17)$$

In Figure 3d we use the coordinates $(-z, T_{max} - T_e)$, transform it to the standard representation

$$(-z \ge 0 \& T_{max} - T_e = 0) | (-z = 0 \& T_{max} - T_e \ge 0), \quad (18)$$

equivalent to the equation $\min(-z, T_{max} - T_e) = 0$. After substitutions and algebraic transformations we obtain

$$\min(-\min(T_c - T_{set}, T_e - T_{n1}), T_{max} - T_e) = 0, \quad (19)$$
$$\max(\min(T_c - T_{set}, T_e - T_{n1}), T_e - T_{max}) = 0, \quad (20)$$

hereinafter denoted as formula1.

Alternatively, in Figure 3e, we start constructing the diagram from the other end, considering the zero level of the function $z = \min(T_{set} - T_c, T_{max} - T_e)$; in Figure 3f we obtain the form

$$(z \le 0 \& T_e = T_{n1}) | (z = 0 \& T_e \ge T_{n1}); \quad (21)$$

in Figure 3g in coordinates $(-z, T_e - T_{n1})$ reduced to the standard form

$$(-z \ge 0 \& T_e - T_{n1} = 0) | (-z = 0 \& T_e - T_{n1} \ge 0); \quad (22)$$

or $\min(-z, T_e - T_{n1}) = 0$. Now, after trivial algebra we obtain another formula:

$$\min(-\min(T_{set} - T_c, T_{max} - T_e), T_e - T_{n1}) = 0, \quad (23)$$
$$\max(\min(T_{set} - T_c, T_{max} - T_e), T_{n1} - T_e) = 0, \quad (24)$$

hereinafter denoted as formula2.

It is interesting that these two formulas give an equivalent representation of the diagram shape in Figure 3a, but are not absolutely identical. In the special, physically important case $T_{n1} > T_{max}$, when the input temperature exceeds the maximum limit, these formulas give different results, shown in Figure 3h. Indeed, in formula1:

$$\max(\min(T_c - T_{set}, T_e - T_{n1}), T_e - T_{max}) = 0, \quad (25)$$
$$T_{n1} > T_{max} \Rightarrow T_e - T_{n1} < T_e - T_{max}, \quad (26)$$
$$\text{case1: } T_c - T_{set} \ge T_e - T_{n1}, \quad (27)$$
$$\max(T_e - T_{n1}, T_e - T_{max}) = T_e - T_{max} = 0; \quad (28)$$
$$\text{case2: } T_c - T_{set} < T_e - T_{n1}, \quad (29)$$
$$\max(T_c - T_{set}, T_e - T_{max}) = T_e - T_{max} = 0, \quad (30)$$

case1 and case2 produce the same answer. In formula2:

$$\max(\min(T_{set} - T_c, T_{max} - T_e), T_{n1} - T_e) = 0, \quad (31)$$
$$T_{n1} > T_{max} \Rightarrow T_{n1} - T_e > T_{max} - T_e; \quad (32)$$
$$\text{case1: } T_{set} - T_c \ge T_{max} - T_e, \quad (33)$$
$$\max(T_{max} - T_e, T_{n1} - T_e) = T_{n1} - T_e = 0; \quad (34)$$
$$\text{case2: } T_{set} - T_c < T_{max} - T_e, \quad (35)$$
$$\max(T_{set} - T_c, T_{n1} - T_e) = T_{n1} - T_e = 0, \quad (36)$$

here we also get a horizontal line on Figure 3h, but a different one. Physically, in the special case under consideration, formula1: $T_e = T_{max} < T_{n1}$ leads to the fact that the heater cools the fluid, so here we should choose the answer $T_e = T_{n1}$, described by formula2.

Let's move on to considering cooler, with the diagram shape shown in Figure 3i. Interestingly, it coincides with the diagram for heater, up to the redesignations $T_{n1} \to T_{min}$, $T_{max} \to T_{n1}$. Thus, instead of repeating the derivation, we can make such a redesignation in the answer for heater and obtain two formulas:

formula1:
$$\max(\min(T_c - T_{set}, T_e - T_{min}), T_e - T_{n1}) = 0; \quad (37)$$
formula2:
$$\max(\min(T_{set} - T_c, T_{n1} - T_e), T_{min} - T_e) = 0. \quad (38)$$

For the special case $T_{n1} < T_{min}$, formula2: $T_e = T_{min} > T_{n1}$ would mean that the cooler heats the fluid, so for physical reasons the answer $T_e = T_{n1}$ described by formula1 should be chosen here.

V. Extended stability analysis

Stability analysis of fluid transport simulations was performed in our previous works, for the stationary case in [2], [10], for the dynamic case in [3]. In these works only the PM phase of the simulation was analyzed. Stability analysis for mixing flows modeling will be performed in this section. The main challenge is the configuration of the dynamic solver for solving stationary problems by integrating to a stationary state, while ensuring the stability of the simulation for realistic large-size networks. First, we present the main results for the PM phase, then we move on to the analysis of mixing phases.

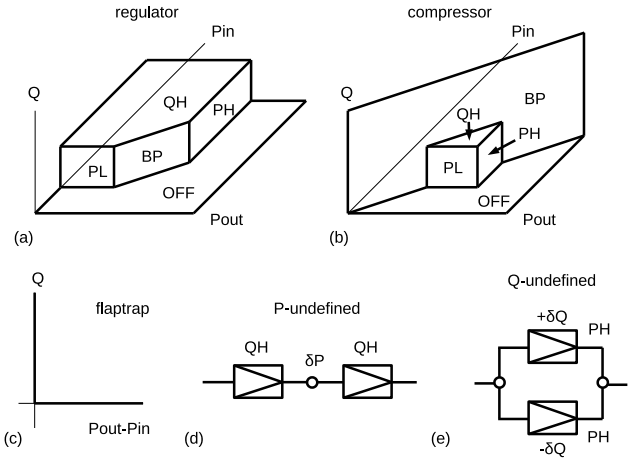

Figure 4. (a)-(c): working diagrams for control elements; (d),(e): possible degenerations of the system; reprinted from [3] by permission (copyright IOP).

PM phase: the main problem for the stability of the simulations is represented by regulators, compressors and flaptraps. The behavior of these elements is given by the diagrams shown in Figure 4a-c. The polyhedral surfaces for regulators and compressors are represented by complex minmax formulas given in [2], the specific form of which is not important for us now. What is important is that combinations of several such elements can be located on certain faces of the surfaces that conflict with each other. For example, in a stationary simulation, two regulators in series on QH-face actually impose the equation $Q = QH$ twice, with one equation wasted, and one unconstrained degree of freedom appears in the system. This degree of freedom corresponds to the undefined pressure at the intermediate point, P-undefined conflict, see Figure 4d. Similarly, two regulators in parallel on PH-face impose the equation $P_2 = PH$ twice, with one equation wasted, and the balance of flows through the regulators is undefined, Q-undefined conflict, see Figure 4e. The described conflicts are not limited to series and parallel connections. The problem is also represented by a long pipe, at the beginning and end of which there are QH-regulators; Y-connection of 3 PH-regulators; conflict between the regulator and Pset/Qset boundary condition at entry or exit; etc. In addition, during the solution process, the working point can change the face on the control diagram, so during the simulation, the described conflicts can spontaneously arise in any part of the network.

For numerical simulations, these conflicts lead to degeneration of the system, the appearance of zero eigenvalues in the Jacobian matrix [2], [10], which leads to divergence of the solver. The general approach to solving this problem is to regularize the equations, to reformulate them as follows:

$$f_{reg} = (1 - \epsilon_s) f + \epsilon_s (P_1 - P_2 - R_s Q) - \epsilon_d \partial m / \partial t, \quad (39)$$

where in the first term f are the original control equations. The second term represents the linear resistor equation, the coefficient $0 \leq \epsilon_s \leq 1$ is chosen so that the regularization can be completely removed at $\epsilon_s = 0$ or, conversely, the control equation can be deformed to a linear resistor at $\epsilon_s = 1$. This type of regularization is static, independent of time derivatives. The third term contains the time derivative and represents dynamic regularization. When choosing the implicit Euler finite difference scheme, this term takes the form $-\epsilon_d (m - m_{prev})/dt$, with $\epsilon_d > 0$. Here Q and m represent the flow in different normalizations and are proportional to each other with a positive coefficient. The common signs in this formula are chosen so that the derivatives of the result with respect to the variables (P_1, P_2, m) have the signature $(+, -, -)$, which, according to [2], is necessary for the convergence of the PM phase of the simulation.

The dynamic term in (39) contains only the m-variable and effectively regularizes only the Q-undefined conflict. The regularization of the P-undefined conflict is performed by the dynamic term $V_n \partial \rho_n / \partial t$ in the Kirchhoff equation (1). This term is able to describe the evolution of the density and the associated pressure even in situations where the control equations do not capture them. The regularizing parameter here is the nodal volume $V_n > 0$, which can also be replaced by one freely adjustable value $V_1 > 0$, without changing the stationary result.

In practice, the use of static regularization leads to the undesirable effect of shifting the solution from the faces of the control equation, violating the control conditions $Q = QH$, $P = PH$. These violations are controlled by the regularizing parameter ϵ_s; for small values, the equation is too singular to solve, and for large values, the physically desirable conditions will be violated. As a tradeoff value, we chose $\epsilon_s = 10^{-3}$, corresponding to 0.1% violation of the control equations and an acceptable level of convergence of the simulations. In the case of divergences, if the cause can be traced back to the control equations via residuals, the user is advised to increase the parameter to $\epsilon_s = 10^{-2}$.

For dynamic regularization, the time derivatives vanish as the stationary solution is reached. Therefore, the dynamic regularizers are switched off in the stationary limit, and no violations of the control equations occur. The limiting factor here is the too slow convergence of the solution for large values of the regularizer. Also, the equations include combinations of ϵ_d/dt, V_1/dt, so for integration with a large step, it is also necessary to artificially increase the regularizing parameters. In our numerical experiments, we varied the described parameters in wide limits and investigated their influence on the simulation stability.

The choice of regularizing parameters was carried out on large natural gas simulations of the N85 type described above and is illustrated by the graphs in Figure 5. At first, we included only the PM phase and investigated its stability separately. Figure 5a shows the idealized case of $\epsilon_s = 1$, when all control equations are replaced by linear resistors. In this experiment, all nodal volumes were also replaced by a single value of V_1. As a result, very fast collective convergence of all simulations below the nominal value $res = 1\%$ is obtained. This numerical experiment shows that in the PM phase we

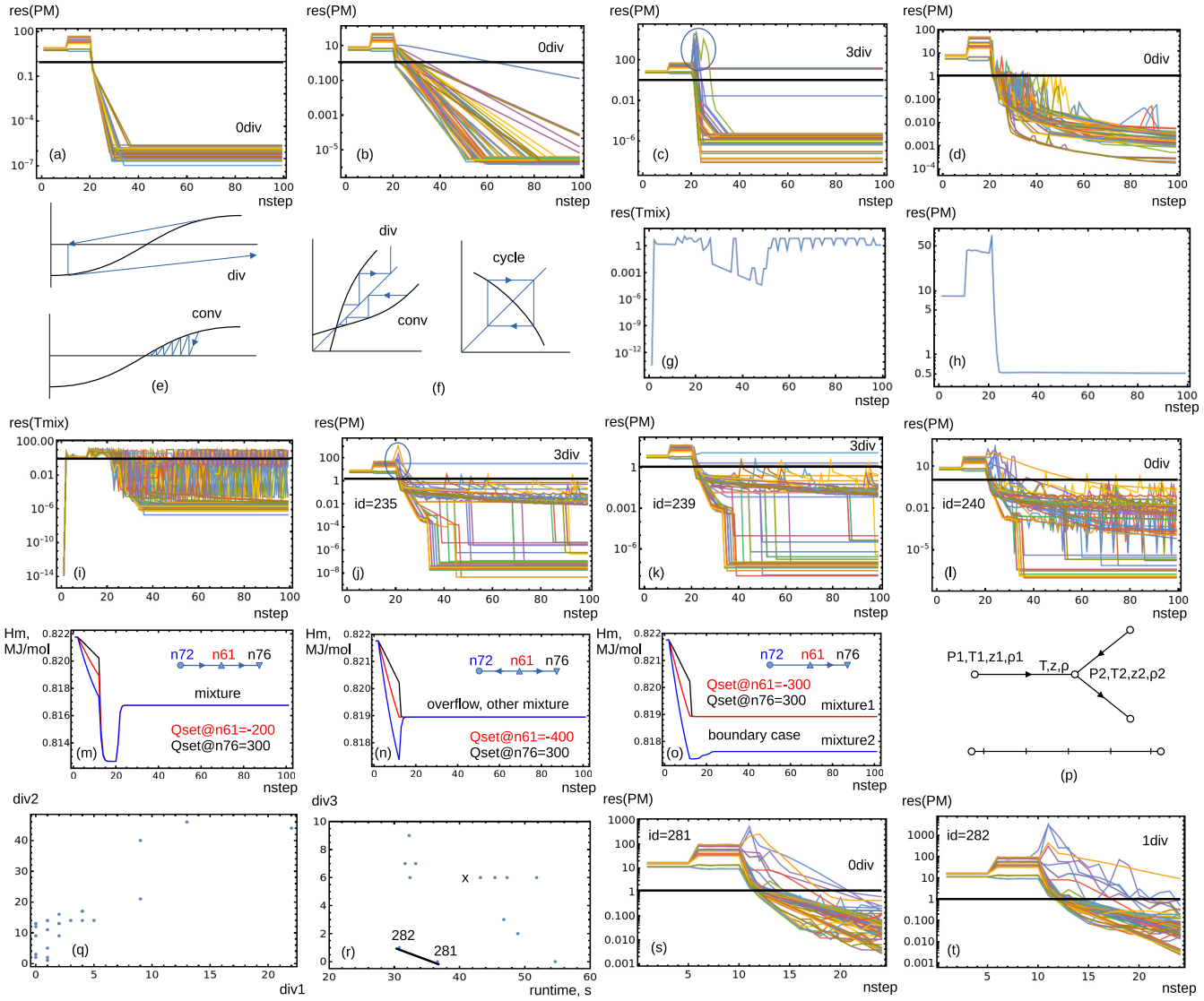

Figure 5. Extended stability analysis (see text for details).

have taken all causes of divergence under control. In Figure 5b we set the nodal volumes to their actual values. The result is still acceptable, but the convergence rate varies from one test case to another. This indicates the advantage of choosing one value for all nodal volumes. In Figure 5c we chose the values $\epsilon_s = 10^{-3}$, $\epsilon_d = 30bar/(kg/s^2)$, $V_1 = 300m^3$, $dt = 3 \cdot 10^5 s$. The configuration is still acceptable, with only 3 out of 85 cases diverged. In this figure, the two initial plateaus correspond to the starting procedure [3] of changing the boundary conditions, first raising all Psets from the starting one to the desired values, then all Qsets. In the second part, we also made a continuous deformation of the regularizing parameter from $\epsilon_s = 1$ to $\epsilon_s = 10^{-3}$. At the end of this interval, the system approaches a singularity, so the characteristic residual peaks are visible in the figure. In Figure 5c, we changed $\epsilon_d = 3 \cdot 10^3 bar/(kg/s^2)$, $V_1 = 30m^3$, and as a result, all simulations went below the nominal threshold. This time, the convergence is slower, but the peak after the starting procedure that generated divergences has disappeared.

Mixing phases: instabilities are present only in the Tmix phase, the others work without problems. Stabilization can be done using dynamic regularization

$$f_{reg} = f + \epsilon_H \partial H/\partial t \tag{40}$$

with the coefficient $\epsilon_H > 0$, when choosing the sign for the original equation $\partial f/\partial H > 0$. Due to the identity $\partial H/\partial T = c_p > 0$, which relates this derivative to the heat capacity, the regularizing term can be reexpressed in via temperature: $\epsilon_H \partial H/\partial t \rightarrow \epsilon_T \partial T/\partial t$, with a new regularizing parameter $\epsilon_T > 0$. A static regularizing term can also be added to this expression, for example, $\epsilon_s(T - T_{soil})$. This term can lead to physically undesirable effects, for example, a temperature of T_{soil} can be established at the output of a low-flow

regulator, despite the existing thermal insulation. Therefore, if dynamic regularization works, we try to refrain from using static regularization. Note that for moderate-sized systems, the described simulations very rarely lead to divergences and can be used directly. For large systems, such as the N85 set used in our tests, the problems are potentiated, and the simulations require special stabilizing measures. Below, we will analyze in detail the available equations and the instabilities associated with them.

Enthalpy mix equation: it already has time derivatives and does not require additional regularization. When switching off the dynamic terms in (8), the remaining stationary system can be degenerate. The problem occurs for $m = 0$, in particular, at the starting conditions. Also, since in this equation only flows entering a node are included in the sum, the problem occurs for all subgraphs not connected to Tset-nodes in the upstream direction. Physically, this singularity means a T-undefined state in the stationary limit for such subgraphs. The dynamic terms resolve this ambiguity, however, for small V_n/dt the regularization is weak, the system matrix is close to singular. An important tuning factor is the nodal volume. It is also possible to equip both dynamic terms in (8) with free coefficients. This allows one to further strengthen the contribution of the $\partial H_n/\partial t$ term, as well as to weaken or disable the $\partial P_n/\partial t$ term. According to our numerical experiments, removal of $\partial P_n/\partial t$ term improves overall stability.

Temperature equation: when using the linearized version of the simulation, equation (10) has a new type of problem. In fact, it describes a Newton iteration in the temperature variable. Although Newton's method is used in the inner iteration, at each integration step, it has been specially stabilized there [32], while the outer iteration described by (10) is unstabilized. It is widely known that the unstabilized Newton's method produces divergences. As shown in Figure 5e above, the sequence of tangents to the curve may go to infinity if the starting point is chosen poorly. The simplest way to overcome this problem is to increase the slope of the lines above the tangent position, which is equivalent to introducing a coefficient $H = H_{mod} + c_1 c_p (T - T_{prev})$, $c_1 > 1$. As shown in Figure 5e below, this can enforce convergence. Although the convergence rate of such an iteration may be slower than Newton's, it turns out to be more stable. Another way to stabilize is to use the original nonlinear equation (9). In a case when there is a closed analytical formula for this equation, it can be used directly. Of course, this will lead to nonlinearity of the Tmix phase and an increase in the number of internal iterations for its solution, the advantage of this approach is better stability of the simulation.

Compressors: in equation (13) a problem similar to the temperature equation arises. In this equation, there is a strong coupling with the PM-phase, in particular, through the z-coefficients present in it. An increase in T_e at a given iteration leads to an increase in z_{n2} at the next iteration, which through the formula (13) triggers a decrease in T_e at the next iteration. Under strong coupling, this iteration sequence can loop or diverge. Figure 5f illustrates the possible behavior of a one-dimensional iteration, showing prototypical examples of instability. The simple solution proposed in [4] consists in introducing a weighting procedure: $T = T_{eq} w + T_{prev}(1 - w)$, with a constant $0 \leq w \leq 1$. In this case, the new value of the variable is not taken directly from the equation, but is weighted with the previous iteration. In practical applications, this approach allows stabilizing looped or diverging iterations that arise due to strong coupling. After rewriting the weighting procedure as the equation $(T - T_{eq})w + (T - T_{prev})(1 - w) = 0$ and comparing the stabilizing terms $(T - T_{prev})/dt \sim \partial T/\partial t$, it becomes clear that the weighting method is completely equivalent to both dynamic regularization and the stabilization of the Newton iteration presented above, up to a redefinition of the coefficients. Another method for stabilizing the compressor equation is to substitute analytical expressions for z-coefficients, if any, into (13).

Coolers and heaters: equations (15)-(16) have problems similar to compressors. For example, if the heater was in standby at the previous iteration and at the controlled point T_c becomes slightly less than T_{set}, then the heater goes into max mode. If the flow through the heater is small, this may lead to a small increase in T_c over T_{set}, and the heater is forced to return to standby. This may lead to iteration loops. The solution here is also dynamic regularization or the equivalent weighting procedure.

Pipes: equation (12) already contains a regularizing term of the static type $\sim (T - T_{soil})$, so the temperature modeling of pipes is stable. A necessary condition is the presence of a physically reasonable heat exchange coefficient.

Default element equation: in the simple-looking equation (11) the strongest instability is located. When passing through the value $m_e = 0$, the edge enthalpy H_e jumps between the nodal values H_{n1} and H_{n2}. Changes in the sign of the flow can occur both at intermediate steps and at the end of integration. A specific example is small numerical fluctuation of the flow in network sections with zero stationary flow. In this case, a unique situation arises when in the final, physically stationary state there are randomly fluctuating variables of undamped amplitude. The jumps are experienced by both the variables themselves and by the residuals of equations defining them, see Figure 5g, which shows the residual of Tmix phase for one scenario N85.1. The residuals use the maximum norm over the equations, as a result of this definition, the jumps can be separate or merging into a plateau. At the same time, the residual of the PM phase shown in Figure 5h does not have such jumps, and repeats the shape of the residual of the pure PM phase shown in Figure 5c.

A detailed analysis shows that the stochastic edge degrees of freedom (H_e, T_e) formed in the system decouple from the nodal (H_n, T_n). Indeed, coupling is carried out by means of equation (8), in which H_e are multiplied by m_e. Thus, the jumps of H_e at $m_e = 0$ are suppressed. The PM phase includes only the nodal values of T_n, so the stochastic degrees of freedom are decoupled from the PM phase as well. In practice, the Tmix phase residual shown in Figure 5i for the entire N85 set is so noisy that it becomes unusable. The PM

phase residual, Figure 5j, can be used as successfully as for the pure PM phase. Indirectly, the PM residual also controls the nodal values of the Tmix phase, via the strong coupling T_n/P_n in the equations of state. The edge values of the Tmix phase undergo jumps around $m_e = 0$, which arise due to their definition as edge downstream values and are not physically important. Thus, our current recommendation is to ignore the Tmix residual and use only the PM residual to control the convergence of the simulation.

Looking at this issue in even more detail, jumps occur in all edge equations where the separation into $m_e > 0$ and $m_e < 0$ branches is used, and they are also suppressed by the m_e-factor in the nodal coupling. The introduction of branches is necessary, otherwise degeneracies arise in the system. As an example, consider the compressor equation $T_e = T_{n1}a$, $a > 1$, in the stationary limit. For $m_e > 0$, the outlet temperature is further transferred to $T_{n2} = T_e$ via the nodal coupling (8). Negative flow through the compressor is possible due to ϵ_s-regularization for infeasible solutions, both at the intermediate and final integration steps. If the compressor equation remains the same for $m_e < 0$, then nodal coupling will lead to $T_{n1} = T_e$, an overdetermined equation on T_e, and no condition on T_{n2}. As a result, the stationary system will become degenerate, and the stationary solver will diverge. There are additional regularizers for the dynamic solver, but their efficiency will be reduced if they have to suppress a more degenerate stationary system. Introducing the isenthalpic branch into the equations ensures non-degeneracy of the system, and it also generates jumps in the solution. Note that suppressing jumps in the edge equations by introducing dynamic damping or weighting procedures does not work here, it only reduces the amplitude of the jumps by a factor of w. The value $w = 0.5$ is practically acceptable for stabilization in our numerical experiments; for smaller values, the convergence of integration becomes too slow.

Figure 5j shows the PM residual for simulations with values $\epsilon_s = 10^{-3}$, $\epsilon_d = 30bar/(kg/s^2)$, $V_1 = 0.3m^3$. Characteristic is the loss of the collective convergence property, which was present for pure PM simulations. This property is a consequence of single-phase modeling, in which the convergence of the solution at the previous iteration leads to convergence at the next one, with small variations due to small dynamic terms. In the full simulation, mix phases are involved in the iterative process, and the convergence of the outer iterative loop is decisive for the convergence of the simulation. In Figure 5k, we increased $\epsilon_s = 10^{-2}$, which resulted in the absence of the residual peak at the end of the startup procedure, which also led to better stability of the inner iterations and a decrease in runtime. In Figure 5l, with $\epsilon_s = 10^{-3}$, $\epsilon_d = 3 \cdot 10^3 bar/(kg/s^2)$, $V_1 = 30m^3$ were increased. Here, as for the pure PM simulation, convergence became slower, but the stability of the simulation has been improved.

Phase transitions: should be considered, in particular, for CO2 transport [8]. The problem is the presence of a jump in the function $W(T)$ for pure substances or a rapid change in this function in the presence of small impurities. This leads to the failure of the Newtonian method, both in internal and external iterations. Dynamic regularization or weighting do not help here. In this case, jumps also occur in nodal variables, propagate to the PM phase and break the convergence of the simulation altogether. Usually, scenarios without phase transitions are considered in applications, CO2 is transported in a liquid/supercritical dense phase or in a gaseous phase. In the absence of phase transitions, the simulation does not have problems of the described type. In order to prevent phase transitions also for all intermediate states on the integration path, the simulation should be started with (P_{start}, T_{start}) values in the region of the expected solution.

FE-nodes: Figures 5m-o show the behavior of FE-nodes, Qset-supplies without specified mix quantities. For such supplies, the mix quantities are assumed to be taken from the incoming flow. The experiments are done on N1 test network. In Figure 5m, a normal operation is shown, where the added flow is less than for a downstream exit, and the mix quantities are taken from the incoming flow. Figure 5n shows an overflow scenario, where the added flow prevails, and there are no incoming, but only outgoing flows. In this case, in stationary problem, the mix value is undefined. The dynamic modeling has the time-derivative term, making the problem non-degenerate even in this case. The resulting mix values are defined by the history of integration. Typically they remain at the starting default values, different from the mixed state of the normal operation mode. Figure 5o shows a boundary case, when the added flow exactly equals the exit flow. In this case, two different mixed states are formed.

Downstream mismatch: in the PM phase there is a problem of a different type, see Figure 5p. The upper part of the figure shows a pipe, with nodal values of pressure, temperature, compressibility and density (P_1, T_1, z_1, ρ_1) and (P_2, T_2, z_2, ρ_2). Consider the section of the pipe immediately adjacent to the downstream node. By continuity, the pressure at this point coincides with the nodal P_2. Otherwise, the pressure jump would create a non-zero force that would act on a vanishingly small mass of the section and lead to infinite acceleration. The temperature in the section, however, may differ from the nodal one, due to a possible inflow of fluid of a different temperature into the node. Compressibility and density depend on temperature and may also not coincide with the nodal values. As already mentioned, PM modeling uses only nodal values for the mentioned quantities. In particular, the PM equation for pipes includes their nodal average. The described mismatch can lead to a local variation of the result near the downstream node. One possible solution would be to introduce edge quantities (T, z, ρ) and a state equation relating them. In fact, this is not a very good idea, since these quantities have random jumps around $m = 0$, and the stochastic behavior would penetrate into the PM phase. Another, simpler solution is shown in the lower part of the figure. To improve the accuracy of the simulation, long pipes should be split into smaller ones. This procedure can include two short segments, say $\Delta L = 1m$, at the beginning and end of the pipe. As a

result, the downstream mismatch problem will be concentrated in these segments. At the same time, since the pressure drop on short pipe segments is negligible, the influence of the described problem on the result will be excluded. In addition to pipes, the problem can occur in compressors (13), if there is z-correction in their equation. On the other hand, in real scenarios, a cooler is usually installed immediately after the compressor, and the described problem does not arise. When using stand-alone compressors or pumps, it is recommended to insert a short section of pipe immediately after them to avoid possible downstream mismatch.

Scaling: Newton's method, in particular the stabilization algorithms [32] used in it, are sensitive to the scaling of equations. For optimal operation of these algorithms, all our equations were scaled so that their variation in the working region of the variable change was of the same value, nominally chosen as 100 units. As a result of such normalization, the residuals of the equations become dimensionless quantities measuring the current absolute value of the equation as a percentage of its variation in the working region. Further, the residuals are maximized over the equations and characterize the convergence of the solution phases. For a detailed characterization of the convergence, two residuals are introduced, for the inner and outer iterations. The residual at the end of the inner Newton iteration measures the convergence of each integration step. The residual at the beginning of the inner Newton iteration measures the convergence of the integration steps to a stationary solution. When the stationary solution is reached, the variables begin to converge to constant values, and the equations also stop changing. In this case, the residual at the beginning of the inner iteration becomes small, ideally less than the stop-criterion $tol = 10^{-5}\%$, so that the inner iterations should not even start, or less than the acceptable threshold $tol_2 = 1\%$.

Clamping: is another technically necessary procedure. In all equations, such quantities as $v = (P, T, z, \rho)$ must be clamped into the physical domain of change: $v \to \min(\max(v, v_{min}), v_{max})$. In the process of solving, such quantities may go beyond the physical domain, for example, become negative. This may happen for infeasible problems that have no solution in the physical domain, as well as for stationary feasible problems at intermediate iterations. According to the general strategy [2], we maintain convergence of the solver in these domains as well, to ensure stability and localize possible infeasibility. As an example, consider the mixing equation (4) with a dynamic term $\sim \rho_n \partial \mu_n^{-1}/\partial t$. If ρ_n becomes negative during the solution, this term will effectively undergo a time reversal, which will immediately lead to divergence of the integrator. Clamping ρ_n into the positive domain solves the problem. Clamping should be carefully introduced into all equations, however, one should not overdo it. Consider the compressor equation $T_e = T_{n1}a$, $a > 1$. Here one can enter clamping to the T_{n1}, a, or a combined $T_{n1}a$ term. One cannot enter clamping to the T_e term, since this equation is the definition of T_e. In the case of T_e clamping, when it is triggered, the T_e dependence drops out of the equation, which will lead to degeneration. We also experimented with introducing clamping to all T variables not in the equations, but between the integration steps. At first glance, this eliminates the need to introduce T-clamping in numerous equations. However, this leads to a deeper problem. If at the current integration step the solution of the equations is located outside the T-clamps, and a clamp is applied before the next step, then the starting point of the next step will no longer satisfy the equations. This can increase the residual and unnecessarily trigger additional iterations. Therefore, we recommend avoiding the use of clamps and any solution-modifying algorithms between the integration steps.

TABLE III
FINE-TUNING PROCEDURE

id	div1	div2	div3	runtime, s
235	3	14	3	73
239	1	5	3	67
240	0	3	0	78
258	4	14	3	47
259	5	14	7	32
260	1	14	6	43
261	9	21	15	34
262	4	17	2	73
263	2	13	6	72
266	1	14	6	32
267	4	17	6	52
268	4	17	9	32
269	2	13	6	47
270	2	13	7	33
271	2	16	6	45
275	0	9	0	55
276	1	2	2	49
277	13	46	16	109
278	2	9	11	38
279	1	1	11	31
280	22	44	58	44
281	**0**	**13**	**0**	**37**
282	**0**	**12**	**1**	**31**
283	9	40	17	56

TABLE IV
FINE-TUNING RESULTS

par	id=281	id=282
n	25	25
dt, s	$6 \cdot 10^4$	$6 \cdot 10^3$
t_1, s	$3 \cdot 10^5$	$3 \cdot 10^4$
t_2, s	$6 \cdot 10^5$	$6 \cdot 10^4$
t_{end}, s	$1.5 \cdot 10^6$	$1.5 \cdot 10^5$
ϵ_s	10^{-3}	10^{-3}
$\epsilon_d, bar/(kg/s^2)$	30	30
V_1, m^3	0.3	0.3
w	0.5	0.5

Fine-tuning: after we have found parameter values with satisfactory convergence characteristics, see Figure 5j-l, we fine-tune the parameters to achieve optimal runtime while maintaining acceptable stability. To do this, we introduce the following characteristics: div1 – number of cases with divergent Newton iteration at the last integration step; div2 – number of cases with divergent Newton iteration at any integration step; div3 – number of cases that do not reach

stationarity after integration, at nominal level $res = 1\%$. The runtime value is averaged over all cases from the N85 set, convergent or not. Simulations were performed on i7-14700K CPU computer. The values of div1-3 and runtime should be minimized. As the analysis shows, the values div1-2 are correlated with each other, see Figure 5q, they are also weakly correlated with div3. Also, the value div3 is weakly anticorrelated with runtime. For the analysis, one graph Figure 5r is sufficient, representing numerical experiments in coordinates (runtime,div3). The best solution marked with a line in the figure marks the tradeoff boundary, the Pareto front, on which these criteria cannot be simultaneously reduced. The characteristics of the stationary simulator are marked with a cross in the figure. It is evident from the graph that the dynamic solver clearly overperforms the stationary one.

In greater detail, the characteristics of fine-tuning runs are given in Table III. The first three lines (235-240) correspond to the configuration of Figure 5j-l. Next, we chose the point (235) and optimized the dynamic schedule described by three parameters (n, dt, t_1), the number of integration steps, the step size, and the time of the first startup phase. The dependent parameters are $(t_2 = 2t_1, t_{end} = n\,dt)$, the time of the second startup phase, and the total integration time. At the beginning (258-259), we decreased n from the starting value $n = 100$ to $n = 50, 25$, which corresponds to a shortening of the integration interval t_{end} with fixed (dt, t_1). Then (260-261), we increased $dt \to dt\,a$ and decreased $n \to n/a$, $a = 2, 4$, which corresponds to more sparse integration with constant (t_1, t_{end}). If we consider weighting as equivalent to dynamic damping, then changing dt above corresponds to changing the weight from the initial $w = 0.5$ to $w = 0.67, 0.8$. Next (262–263) we increased $dt \to dt\,a$ simultaneously with $(t_1, t_{end}) \to (t_1, t_{end})a$, $a = 2, 4$, with n remaining constant, which is equivalent to decreasing the dynamic damping in all equations; in this case, w was varied as described above. The system has an exact symmetry: scaling the step dt and the coefficients of dynamic terms such as (V_1, ϵ_d) simultaneously does not change the equations. The three transformations described exhaust the space of variables (n, dt, t_1). In subsequent experiments (266-271) we considered combinations of these transformations corresponding to their cross-effects. Next, we selected 3 points (258,259,266) on the Pareto front (runtime,div3) as the most promising candidates. For them, we decreased $(dt, t_1, t_{end}) \to (dt, t_1, t_{end})/a$, $a = 10, 10^2, 10^3$, with n remaining constant. This corresponds to an increase in dynamic damping in the equations and was done to catch a solution with strong damping like Figure 5l. In this case, $w = 0.5$ was not changed, since it is already strong enough.

Table IV presents two optimal configurations (281,282), the first column corresponds to enhanced stability div1-3=(0,13,0) and runtime=37s, the second – to acceptable stability div1-3=(0,12,1) and the shortest runtime=31s. These solutions are also shown in Figure 5s-t. Based on the results of the analysis, the user can independently select the required mode and has a sufficient number of handles for detailed adjustment of the convergence.

VI. Conclusion

This paper considered the modeling of mixing flows in dynamic simulation of pipeline fluid transport. Mixed characteristics include molar mass, heat value, chemical composition and temperature of the transported fluids. In the absence of chemical reactions, the modeling is based on the universal conservation laws for molar flows and total energy. The modeling leads to a system of differential algebraic equations, including linear molar mixing formulas, nonlinear temperature-energy relationships, and piecewise-linear element equations for coolers and heaters. In one approach, for nonlinear relations, linearization is carried out in the vicinity of the previous integration step, piecewise-linear relations are reduced to linear ones using the active set method. The resulting sequence of linear systems is solved by a sparse linear solver, typically in one iteration per integration step. In alternative implementation, exact minmax formulas for coolers and heaters are used, solution is performed by a stabilized Newtonian solver. The functionality and stability of the developed approach have been tested in a number of realistic network scenarios.

Numerical experiments on the moderate size N1 network allow us to follow the mixing processes in detail, including the evolution of molar mass, heat value, chemical composition, and temperature. Experiments on the N85 set of large-scale natural gas networks demonstrate the stability of the developed methods and its sensitivity to such details as nonlocality of equations used in the implementation of heaters. Hydrogen and carbon dioxide pipeline scenarios are also used for testing the temperature modeling and the convergence of simulation.

Based on numerous simulations, the stability of the dynamic solver was studied in detail. The factors affecting stability and runtime were identified, and their optimal configuration was selected. Each equation was analyzed separately, as well as their full set. The challenges encountered during the analysis include

- system degeneration,
- the appearance of stochastic degrees of freedom,
- jumps in thermodynamic functions on phase transitions,
- proper scaling of equations.

Parameters that greatly affect the stability and speed of simulation were identified. These include

- regularizing and weighting constants,
- dimensioning of dynamic terms and startup procedure,
- the size of the integration step,
- the total number of the integration steps.

The optimal choice of these parameters allowed us to accelerate significantly the dynamic simulation of fluid transport for realistically large network scenarios.

Our further research includes fine-tuning the underlying sparse linear solvers, adaptive choice of the number of integration steps, and hardware acceleration.

Acknowledgments

The work has been partly supported by Fraunhofer research cluster CINES and partly by BMBF project TransHyDE-Sys,

grant no. 03HY201M. We also acknowledge support from Open Grid Europe GmbH in the development and testing of the software.

REFERENCES

[1] M. Anvari, A. Baldin, T. Clees, B. Klaassen, I. Nikitin, and L. Nikitina, "Mixing flows in dynamic fluid transport simulations", in Proc. of ADVCOMP 2024, International Conference on Advanced Engineering Computing and Applications in Sciences, pp. 34-39, IARIA, 2024.

[2] T. Clees, I. Nikitin, and L. Nikitina, "Making network solvers globally convergent", Advances in Intelligent Systems and Computing, vol. 676, 2018, pp. 140-153.

[3] M. Anvari et al., "Stability of dynamic fluid transport simulations", J. Phys.: Conf. Ser., vol. 2701, 2024, 012009.

[4] A. Baldin et al., "On advanced modeling of compressors and weighted mix iteration for simulation of gas transport networks", Lecture Notes in Networks and Systems, vol. 601, 2023, pp. 138-152.

[5] T. Clees et al., "MYNTS: Multi-physics network simulator", in Proc. of SIMULTECH 2016, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 179-186, SciTePress, 2016.

[6] A. Baldin, T. Clees, B. Klaassen, I. Nikitin, and L. Nikitina, "Topological reduction of stationary network problems: example of gas transport", International Journal On Advances in Systems and Measurements, vol. 13, 2020, pp. 83-93.

[7] T. Clees et al., "Efficient method for simulation of long-distance gas transport networks with large amounts of hydrogen injection", Energy Conversion and Management, vol. 234, 2021, 113984.

[8] M. Anvari et al., "Simulation of pipeline transport of carbon dioxide with impurities", in Proc. of INFOCOMP 2023, the 13th International Conference on Advanced Communications and Computation, pp. 1-6, IARIA, 2023.

[9] A. Baldin et al., "Universal translation algorithm for formulation of transport network problems", in Proc. SIMULTECH 2018, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, vol. 1, pp. 315-322.

[10] A. Baldin et al., "Principal component analysis in gas transport simulation", in Proc. of SIMULTECH 2022, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 178-185, SciTePress, 2022.

[11] J. Nikuradse, "Laws of flow in rough pipes", NACA Technical Memorandum 1292, Washington, 1950.

[12] P. Hofer, "Error evaluation in calculation of pipelines", GWF-Gas/Erdgas, vol. 114, no. 3, 1973, pp. 113-119 (in German).

[13] J. Saleh, ed., Fluid Flow Handbook, McGraw-Hill 2002.

[14] D.-Y. Peng and D.P. Robinson, "A new two-constant equation of state", Ind. Eng. Chem. Fundam, vol. 15, 1976, pp. 59-64.

[15] G. Soave, "Equilibrium constants from a modified Redlich-Kwong equation of state", Chemical Engineering Science, vol. 27, 1972, pp. 1197-1203.

[16] ISO 12213-2: Natural gas – calculation of compression factor, International Organization for Standardization, 2020.

[17] ISO 20765-2: Natural gas – Calculation of thermodynamic properties – Part 2: Single-phase properties (gas, liquid, and dense fluid) for extended ranges of application, International Organization for Standardization, 2020.

[18] O. Kunz and W. Wagner, "The GERG-2008 wide-range equation of state for natural gases and other mixtures: An expansion of GERG-2004", J. Chem. Eng. Data, vol. 57, 2012, pp. 3032-3091.

[19] W. Wagner, Description of the Software Package for the Calculation of Thermodynamic Properties from the GERG-2008 Wide-Range Equation of State for Natural Gases and Similar Mixtures, Ruhr-Universität Bochum, 2022.

[20] E. Egger and J. Giesselmann, "Stability and asymptotic analysis for instationary gas transport via relative energy estimates", Numerische Mathematik, Springer 2023.

[21] J. K. van Deen and S. R. Reintsema, "Modelling of high-pressure gas transmission lines", Appl. Math. Modelling, vol. 7, 1983, pp. 268-273.

[22] C.-Y. Chang, S.-H. Wang, Y.-C. Huang, and C.-L. Chen, "Transient response analysis of high pressure steam distribution networks in a refinery", in Proceedings of the 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), May 28-31, 2017, Taipei, Taiwan, pp. 418-423, IEEE 2017.

[23] T.-P. Azevedo-Perdicoúlis, F. Perestrelo, and R. Almeida, "A note on convergence of finite differences schemata for gas network simulation", in Proceedings of the 22nd International Conference on Process Control, June 11-14, 2019, Štrbské Pleso, Slovakia, pp. 274-279, IEEE 2019.

[24] M. Chaczykowski and A. J. Osiadacz, "Dynamic simulation of pipelines containing dense phase/supercritical CO2-rich mixtures for carbon capture and storage", International Journal of Greenhouse Gas Control, vol. 9, 2012, pp. 446-456.

[25] P. Aursand, M. Hammer, S. T. Munkejord, and Ø. Wilhelmsen, "Pipeline transport of CO2 mixtures: models for transient simulation", International Journal of Greenhouse Gas Control, vol. 15, 2013, pp. 174-185.

[26] S. T. McCoy and E. S. Rubin, "An engineering-economic model of pipeline transport of CO2 with application to carbon capture and storage", International Journal of Greenhouse Gas Control, vol. 2, 2008, pp. 219-229.

[27] M. J. Moran and H. N. Shapiro, Fundamentals of Engineering Thermodynamics, John Wiley and Sons, 2006.

[28] C. Himpe, S. Grundel, and P. Benner, "Next-gen gas network simulation", in: Progress in Industrial Mathematics at ECMI 2021, pp. 107-113, Springer 2022.

[29] C. Himpe, S. Grundel, and P. Benner, "Model order reduction for gas and energy networks", J. Math. Industry, vol. 11:13, 2021, pp. 1-46.

[30] J. Mischner, H.-G. Fasold, and K. Kadner, gas2energy.net, Systemplanung in der Gasversorgung, Gaswirtschaftliche Grundlagen, Oldenbourg Industrieverlag GmbH, 2011 (in German).

[31] GPSA Engineering Data Book, 14th Edition, Gas Processors Suppliers Association, 2016.

[32] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.