Enhanced Simulation of Pipeline Fluid Transport with Phase Transition Detection and Pipe Subdivision Algorithm

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

Anton Baldin
PLEdoc GmbH and Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Anton.Baldin@scai.fraunhofer.de

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

Bernhard Klaassen
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Bernhard.Klaassen@scai.fraunhofer.de

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

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

Sabine Pott
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Sabine.Pott@scai.fraunhofer.de

***Abstract*—This work considers a stationary simulation of pipeline fluid transport, in the presence of impurities and phase transitions. This simulation finds applications in diverse areas such as energy carrier transportation, including natural gas and hydrogen, as well as the efficient transport of carbon dioxide from emission sources to designated storage sites. Particularly for the transport of carbon dioxide, which is preferably carried out in a liquid or supercritical state, the accurate detection of phase transitions is of utmost importance. Additionally, evaluating the simulation precision based on the selected pipe subdivision is crucial for transporting fluids of any kind. Our implementation includes an algorithm that utilizes the Homogeneous Equilibrium Model and the GERG-2008 thermodynamic equation of state for phase transition detection. We have also developed an optimal pipe subdivision algorithm using empirical formulas derived from extensive numerical experiments. Rigorous testing of the algorithms has been conducted on realistic fluid transport scenarios, confirming their effectiveness in addressing the stated technical challenges.**

Index Terms*—simulation and modeling; mathematical methods and numerical algorithms; advanced applications; fluid transport; carbon dioxide transport; pipe subdivision.

I. Introduction

This paper is an extension of our conference paper [1], which focused on the stationary simulation of carbon dioxide pipeline transport with impurities and phase transition detection. In this current study, we have expanded our simulations to include other fluids such as natural gas and hydrogen. Furthermore, we have developed an algorithm for pipe subdivision to enhance the precision of the simulation as desired.

To reduce greenhouse gas emissions into the atmosphere, Carbon dioxide Capture and Storage (CCS) systems are currently being developed. Typically, such systems consist of 3 parts: (1) capturing carbon dioxide (CO_2) at its source; (2) transporting CO_2 through pipelines to special storage sites; (3) and finally injecting it into wells, when underground storage is used. In this paper, we focus on the second part of the aforementioned process. It is generally required that CO_2 be in the liquid or supercritical phase during transport in order to increase the density and mass flows. It is essential to avoid the transition of fluid phase to gas, which leads to cavitation and destruction of the pipeline during transportation. To ensure reliable operation of the CO_2 pipeline, both an extensive experimental base and stable numerical simulation of the transportation process are required. At the same time, for a long-term planning, it is sufficient to simulate a stationary process of the transportation, with CO_2 in a 1-phase state and an indication of a possible phase transition, in order to prevent it.

The pioneering work [2] has considered in detail the stationary process of transporting pure CO_2 through a pipeline and pumping it into an underground storage, taking into account phase transitions. Subsequent papers, including [3]–[10], have highlighted the significance of considering impurities that can significantly impact transportation parameters even at low

concentrations. These papers have investigated both stationary and dynamic aspects of CO_2 transport. Papers [2]–[9] consider a Homogeneous Equilibrium Model (HEM), in which different phases of a fluid are homogeneously mixed and have the same speed, pressure, temperature and chemical potential. Papers [5]–[7], [9], [10] have also explored the concept of phase split, where the phases are geometrically separated, and phase slip, where the phases have different velocities. Additionally, works such as [5], [7], [9] have examined the formation of a solid phase of CO_2 (dry ice). Other studies [6], [7], [9], [10] have focused on fast transient processes that occur during pipe depressurization, including relevant experimental investigations. Furthermore, the economic aspects of pipeline CO_2 transport have been addressed in papers [11]–[14].

In this paper, we describe a stationary simulation of the CO_2 transport process with the possibility of considering impurities, phase transitions, several sources with different composition, and networks of complex topology. Simulations of this type have extended the capabilities of our software MYNTS [15]–[19]. The system provides an open, freely configurable and user-friendly specification of modeling, defined as a list of variables and equations. An open Python code for workflow procedures is also provided. The main calculations are performed in a fast C++ solver. The system also has a Graphical User Interface (GUI) with the ability to edit networks and scenarios. This architecture allows to formulate and quickly solve very large network problems, as well as the ability to model different energy carriers and couple different energy sectors.

For problems of stationary transportation of fluids, we implement standard pipe transport equations with friction terms by Nikuradse [20], Hofer [21] and spatial discretization of type [22]. The GERG equation of state [23], [24], which is currently the ISO standard [25], is used to accurately model the thermodynamics of fluids, in particular CO_2 with impurities and phase transitions. Additionally, we have developed an algorithm for detecting the proximity to the region of phase transitions. Numerical experiments were conducted to validate the implemented algorithms. These experiments demonstrate that the presence of phase transitions in the system can induce fast and occasionally abrupt behavior, which in turn influences the convergence properties of the numerical algorithms employed for the solution. In the scenarios we have considered, the divergence, if it occurs, is entirely localized in the region of phase transitions. On the other hand, scenarios without phase transitions are converging, which makes it possible to solve them with detection of proximity to the region of phase transitions.

Our implementation is based on standard numerical methods for solving systems of nonlinear equations, described in [26]–[28], applied to piecewise linear resistive systems in [29]–[31]. Questions of discretization of differential equations are considered in detail in general form in [26] and in application to gas networks in [32]–[34]. Our contribution to this area is the formulation of global convergence conditions for solution of nonlinear resistive systems [16] and their application to

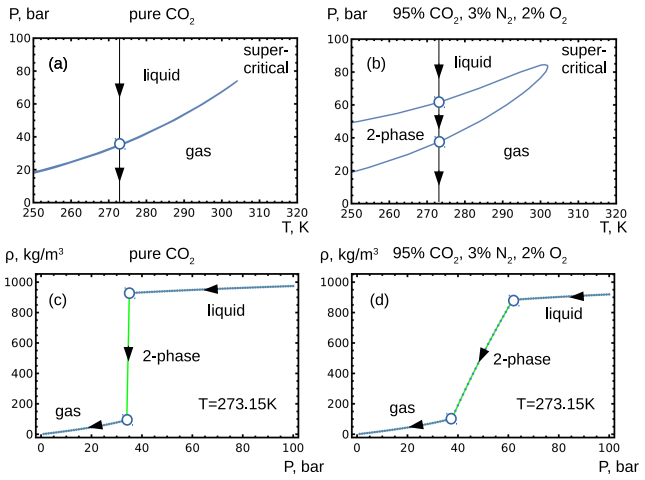

Fig. 1. Phase transitions at fixed temperature: (a),(c) – for pure CO_2; (b),(d) – for CO_2 with impurities. Image from [1].

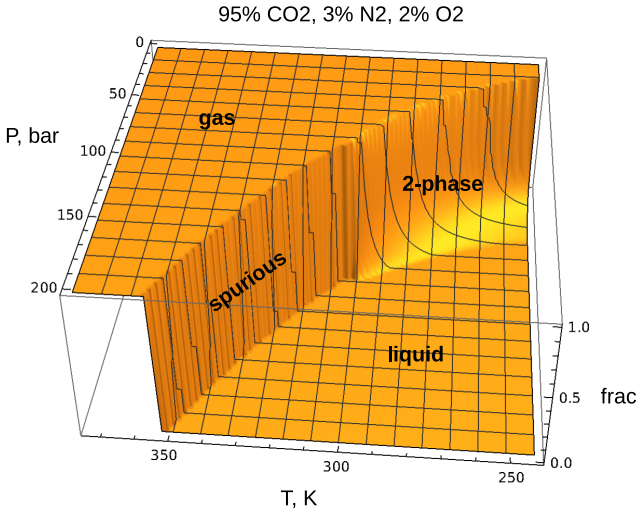

Fig. 2. Fraction of gaseous phase as a function of pressure and temperature. Image from [1].

stationary simulation of gas transport networks. We also constructed a pipe subdivision algorithm to achieve a given precision of stationary simulation of fluid transport networks and present it in this work.

Section II reviews the physics of phase transitions applied to CO_2 with impurities. Section III discusses the transport equations used. Section IV presents the pipe subdivision algorithm. In Section V, we describe numerical experiments, with particular attention paid to the questions of convergence of iterative processes and precision of simulation. Finally, in Section VI, we summarize our results.

II. Physics of Phase Transitions

Phase transitions exhibit slight variations in their occurrence between pure substances and their mixtures. Figure 1a shows the phase transition for pure CO_2. At a constant

temperature, the pressure decreases starting in the region of the liquid state. There is a line of phase transitions on the diagram. When the pressure decreases, the process proceeds until it intersects with this line, after that the pressure decrease stops until all the fluid passes from the liquid state to the gaseous state. At the same time, Figure 1c shows that during this process, the average density changes from large values, typical for the liquid phase, to small values, typical for a gas. In Figure 1b, the behavior of a mixture, i.e., 95% CO_2, 3% N_2, 2% O_2, is depicted. In this case, the two-phase state is not represented by a single line but rather a region on the (T, P)-diagram. The boundary of this region is called the Vapour-Liquid Equilibrium (VLE) diagram, or *phase envelope*. When the pressure decreases, the point enters this region and the fluid also passes from the liquid state to the gaseous state, but here the pressure continues to decrease. Figure 1d shows that in the 2-phase state, the density decreases in the same way as for pure substance, but at a decreasing pressure.

The 3D diagram in Figure 2 shows the behavior of $frac$-value, which varies in the interval $[0, 1]$ and measures the fraction of the gaseous phase in the fluid. In this figure, one can observe the region where the phase transition takes place, which occurs continuously for mixed compositions. Additionally, the diagram exhibits a discontinuity along a line originating from the critical point, although this transition is considered spurious. Above the critical point, there is no significant distinction between gas and liquid phases. However, based on the descriptive framework, a transition from gas to liquid is required at some point. While there is a formal jump in the quantity $frac$, physically measurable quantities do not exhibit such jumps along this line.

Interestingly, this surface resembles the surfaces considered in the theory of functions of a complex variable. Namely, if we take this surface, as well as the $1-frac$ surface and join them together, we get an object that looks like a Riemann surface for a complex square root. The similarity is not accidental, in both cases there is a 2-sheeted surface without the possibility of continuously separating the sheets from each other.

For the thermodynamic description of the fluid, the GERG equation of state and its accompanying implementation [23]–[25] are used. Technically, it is delivered as a software library where one can access a variety of functions describing the fluid state. In addition to the already mentioned phase envelope and $frac$-value, we use the Equation Of State (EOS) and energy functions

$$z = z(T, P, x),\ W = W(T, P, x), \tag{1}$$

where T is absolute temperature, P is pressure, x is a vector describing fluid composition, $W = (H, U, G, A)$ is a vector describing molar energies of different types: enthalpy, internal energy, Gibbs energy, Helmholtz energy, respectively. Compressibility factor z enters in the gas law $P = \rho RTz/\mu$, where R is the universal gas constant, ρ is the mass density, μ is the molar mass.

As an essential parameter for the user, the $frac$-value or a conservative algorithm utilizing $frac$-values in the vicinity of the solution can be employed to identify the proximity of phase transitions:

Algorithm (proximity-alarm):

```
given (T0,P0,x,dT,dP,val)
for T in (T0-dT,T0,T0+dT)
  for P in (P0-dP,P0,P0+dP)
    if frac(T,P,x)!=val return true
return false.
```

The algorithm considers a 3x3 grid created by $(\pm dP, \pm dT)$-variations, and if $frac$ differs from the user-specified val at least at one point, triggers a proximity alarm. This simple algorithm is applied to every node in the network. It has the advantage that it works even in the networks with many fluid compositions, i.e., variable x-values. Alternative algorithms based on the construction of the phase envelope produce many diagrams for different compositions, which complicates the analysis. At the same time, this algorithm has one drawback, it can produce a false alarm when approaching a spurious line. In this case, the user can visually control the solution trajectory on the (T, P)-diagram by constructing a phase envelope for the local network segment with constant x. Our future plans include the development of additional algorithms for automatic detection of phase transitions that can handle variable composition of the fluid within the network.

III. Pipe transport equations

In the stationary case, a pressure drop in the pipe is described by the equation:

$$dP/dL = -\lambda\rho v|v|/(2D) - d(\rho v^2)/dL - \rho g\, dh/dL, \tag{2}$$

where L is the running length along the pipe, v is the speed of the fluid, D is the internal diameter of the pipe, g is the gravitational acceleration, and h is the height. The first term on the right hand side is usually dominant, describing the contribution of the friction force, defined in terms of the dimensionless friction coefficient $\lambda(k/D, Re)$ using the Nikuradse [20] formula or the more accurate Hofer [21] formula. Here, k is the pipe roughness, $Re = 4|Q_m|/(\pi\mu_{visc}D)$ is the Reynolds number, where μ_{visc} is the dynamic viscosity and $Q_m = \rho v\pi D^2/4$ is the mass flow constant along the pipe. In addition, the right-hand side includes the convective and gravitational terms. The flow $Q_m = Q_N\rho_N$ is often expressed in terms of the normal volume flow Q_N and mass density ρ_N at normal conditions $P_N = 1.01325\ bar$, $T_N = 273.15\ K$.

For discretization purposes, we consider a short pipe segment of length L and integrate the equation over it. Expressing the velocity in terms of the mass flow, and keeping only the leading first term for illustration, we get $dP/dL = c_1/\rho$, where c_1 is constant. When performing integration, we substitute the variable density ρ with the average $\bar{\rho} = (\rho_1 + \rho_2)/2$ between the endpoints of the segment. In other words, $P_2 - P_1 = c_1 L/\bar{\rho}$. As an alternative, we multiply the original equation by P, use the gas law $P/\rho = RTz/\mu$, replace the variables T and z with the end averages and, thereby, we get

$(P_2^2 - P_1^2)/2 = c_1 L R \bar{T} \bar{z}/\mu$, in a more familiar quadratic form for gas dynamics [22].

Temperature profiles are described by the equation

$$dH/dL = -\pi D c_h (T - T_s)\mu/Q_m, \quad (3)$$

according to which the enthalpy change in a segment of the pipe is equal to the heat exchange with the soil or other environment. Here, c_h is the heat transfer coefficient, T_s is the soil temperature. Note that when the heat exchange is switched off ($c_h = 0$), the process described by this formula is isenthalpic $dH = 0$, and the temperature change is related to the pressure change by the well-known formula $dT = \mu_{JT} dP$, where $\mu_{JT} = -(\partial H/\partial P)_T/(\partial H/\partial T)_P$ is the Joule-Thomson coefficient. The equation can also be modified by introducing kinetic and gravitational terms.

For discretization purposes, the equation in the form $dH/dL = c_2(T - T_s)$ with a constant c_2, the variable temperature T is substituted with a constant T_x. The value of T_x can be chosen as the average temperature $\bar{T}$ at the endpoint or the outflow temperature T_{out}, depending on the scenario that better represents longer segments. After integration, we get $H_2 - H_1 = c_2 L(T_x - T_s)$. Further, in an iterative solution process in which the pressure profile and fluid composition are kept constant, the enthalpy values can be linearized using the formula $H(T^{i+1}) = H(T^i) + c_p(T^i)(T^{i+1} - T^i)$, where the superscripts indicate the number of iterations and $c_p = (\partial H/\partial T)_P$ is the isobaric molar heat capacity, also calculated by the GERG software library.

Next, we will consider in more detail the process of convergence of the iterations used for the solution. In our previous work [19], the architecture of MYNTS system has been described. Due to software-technical reasons, the solution was divided into 2 parts: (1) *Pressure-Massflow (PM)-iterations*, solved by a sparse non-linear Newtonian solver; and (2) *mix-iterations*, solved by a sparse linear solver. PM iterations determine the pressure, density and mass flow, by solving a relatively small nonlinear system. This system, however, has strong numerical instabilities associated with nearly zero Jacobi matrix eigenvalues and requires special stabilization measures [18]. Mix iterations solve a large linear system defining a multicomponent fluid composition, determine temperature and call external modules, such as GERG that would otherwise be called too often in a fully coupled system. After the temperature linearization described above, all mix equations of the system at each iteration become linear, their solution can be produced by a sparse linear solver such as Pardiso [35]. Further, these two processes are iterated, while using an additional stabilization algorithm called *weighted relaxation* [19], the result of the combined PM-mix-iteration $h(x)$ is replaced by a weighted average $x_{i+1} = wh(x_i) + (1 - w)x_i$.

Among the modeling limitations, it should be mentioned that the GERG module does not consider the solid phase and derives equilibrium conditions for the liquid and gaseous phases under the HEM assumptions. The transport equations considered here treat 2-phase solutions as 1-phase, with the values of thermodynamic parameters calculated by the GERG module in the *total* system, which also means calculations within the HEM framework.

TABLE I
PARAMETERS OF TEST SCENARIOS

parameter	symbol [units]	value
total pipe length	$L_{tot}[km]$	150
pipe internal diameter	$D[m]$	0.5
pipe roughness	$k[mm]$	0.5
heat transfer coefficient	$c_h[W/(m^2K)]$	4
inlet temperature	$T_1[K]$	313.15
soil temperature	$T_s[K]$	283.15
fluid composition	$x(CO_2, N_2, O_2)$	(0.95,0.03,0.02)
inlet pressure	pset [bar]	100
outlet norm.vol.flow, scen1	qset1 $[10^3 m^3/h]$	200
outlet norm.vol.flow, scen2	qset2 $[10^3 m^3/h]$	310
fluid composition	$x(CO_2)$	1
inlet pressure	pset [bar]	96.01325
outlet norm.vol.flow, scen3a	qset $[10^3 m^3/h]$	200
fluid composition	$x(CH_4)$	1
inlet pressure	pset [bar]	50.01325
outlet norm.vol.flow, scen3b	qset $[10^3 m^3/h]$	50
fluid composition	$x(H_2)$	1
inlet pressure	pset [bar]	50.01325
outlet norm.vol.flow, scen3c	qset $[10^3 m^3/h]$	50

At the conclusion of this section, it is important to address a general aspect concerning the simulation. It is common for users to assume the uniqueness of solutions obtained in simulations. However, it should be noted that in general, this assumption may not hold true. Existence and uniqueness theorems for solutions are only formulated in rare cases. So, for example, they are guaranteed for the PM subsystem under the conditions of generalized resistivity [15]. Being combined with the mix system, the uniqueness of the solution is not guaranteed. Theoretically imaginable is the situation when there are two stationary solutions, one 1-phase, the other 2-phase, and it may happen that the stationary solver finds the first one, but in reality the second one will be realized. Consideration of dynamic simulation can decide which solution the trajectory will go to when integrating from a given initial state. But even for a dynamic solver, saddle points, i.e., bifurcations of the solution are possible, where, with a small variation, the solution can go in one direction or the other. Questions about the uniqueness of stationary solutions and the stability of dynamic solutions must be investigated in the practical analysis of simulation results.

IV. PIPE SUBDIVISION ALGORITHM

First, we conduct a series of numerical experiments to analyze the relationship between precision and pipe subdivision. Next, we explore various discretization techniques and evaluate their precision and stability criteria. By carefully analyzing these criteria, we can determine which discretization methods are best suited for our needs. Finally, we implement these derived formulas into our subdivision algorithm. Based on the data gathered from our experiments and evaluations, we derive empirical formulas for pipe subdivision.

TABLE II
DERIVED PARAMETERS OF TEST SCENARIO 3

parameter	symbol [units]	value scen3a	value scen3b	value scen3c
fluid composition		pure CO_2	pure CH_4	pure H_2
inv. molar mass	$\mu^{-1}[mol/kg]$	22.722	62.332	496.06
mass flow	$m[kg/s]$	109.83	17.937	2.2471
heat capacity, molar isobaric, inlet-outlet	$c_p[J/(molK)]$	317.31-109.50	41.675-40.720	29.167-29.008
characteristic length, inlet-outlet	$x_1[km]$	126.03-43.491	7.4158-7.2457	5.1744-5.1462

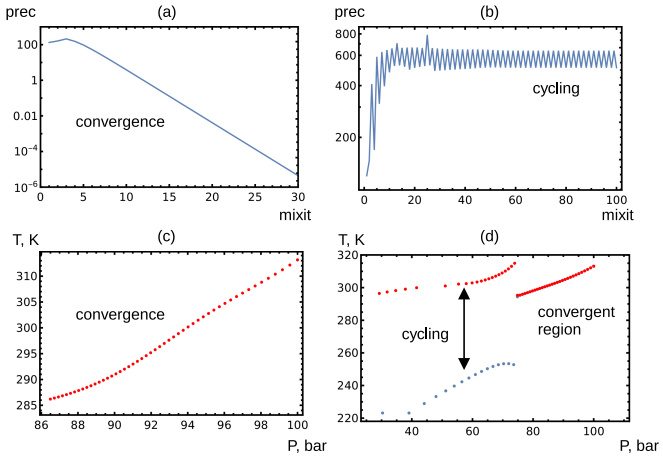

Fig. 3. (a),(c) – convergent iterations for scenario without phase transitions; (b),(d) – cycling iterations for scenario with phase transitions, red color - iteration 100, blue color – iteration 99. Image from [1].

Testing pipe subdivision: a set of simulations with variable pipe subdivision is considered,

$$L = L_{tot}/N_{div},\ N_{div} = 2^n,\ n = 1...n_{max}. \quad (4)$$

In our test case, $L_{tot} = 150km$, $n_{max} = 10$ are selected. In this section, we designate the symbol L to represent the length of a pipe segment. The coordinate along the pipe will be denoted as x, where x ranges from 0 to L_{tot}. Additionally, the symbol m will be used to represent the mass flow.

Precision is defined as maximal deviation of n-th solution from the most precise n_{max}-th solution:

$$\delta P_n = \max_x |P_n(x) - P_{n\_max}(x)|, \quad (5)$$

here for P, and similarly for other variables. The resulting dependencies of the precision on n are shown on Figure 5. The dependencies are mostly following $\sim L$ profile. Three scenarios are considered, with settings described in Table I. For $scen3a/CO_2$, subdivision $N_{div} \sim 16$, $n = 4$, corresponds to $\delta P \sim 0.1bar$. For $scen3b/CH_4$, similar δP is achieved at $N_{div} \sim 6$. For $scen3c/H_2$, such δP is achieved already at original pipe, $N_{div} \sim 1$, no subdivision needed.

Further details: the number of mix iterations is set to $nmixit = 30$, their convergence is controlled. Phase transitions do not happen for all scenarios. A slight bent at $n = 9$ is a methodical issue: since n_{max} subdivision is not exact answer, $n_{max} - 1$ level feels the error of n_{max} level, while the other n-levels are less sensitive to this error.

Various discretizations: the schemes, briefly described in the previous section, will be considered in more details now. In Hofer friction law (2), we track the leading $\sim L/D$ term $dP = -f_R dx$, where $f_R(x) \sim 1/\rho(x)$ with coefficient constant along the pipe segment.

Hofer-quad: multiply both sides by ρ, using that $\rho \sim P$ approximately for gases (gas law $P = \rho RTz/\mu$, where the coefficient RTz/μ is assumed to change slowly along the pipe segment and is represented by its nodal average), integration gives: $lhs = \int PdP = (P_2^2 - P_1^2)/2$ and $rhs \sim \int dx = L$, a known formula for quadratic hydraulic resistance applicable for gases.

Hofer-lin: in f_R take the nodal average $\rho \to (\rho_1 + \rho_2)/2 = const$, then integrate straightforwardly: $lhs = P_2 - P_1$, $rhs \sim \int dx/\rho = L/((\rho_1 + \rho_2)/2)$. A surprising equivalence: the leading terms for Hofer-quad and Hofer-lin coincide.

Proof: in Hofer-quad integrated formula, divide $lhs = (P_2^2 - P_1^2)/2$ to $(P_1 + P_2)/2$, use approximate linearity $\rho \sim P$ above, obtain $rhs \sim L/((\rho_1 + \rho_2)/2)$ coincident with Hofer-lin integrated formula. □

This is what we see on Figure 5, the coincidence of precision for both schemes, perfect for gases (methane at 50bar), slightly deviating for liquids (supercritical CO_2). The deviation is due to the omitted terms in the friction law, which are indeed different for two integration schemes, and due to details of taking nodal average: $Tz \to (T_1 z_1 + T_2 z_2)/2$ vs $(T_1 + T_2)/2 \cdot (z_1 + z_2)/2$, etc. At $n = n_{max}$ level, Hofer-quad and Hofer-lin results coincide at high precision ($\sim 10^{-6}$bar), so that both schemes provide a consistent discretization for the same continuous equation.

Stability considerations: according to [16], for convergence of the simulation, the signature of the whole equation must be $\partial eq/\partial(P_1, P_2, m) \sim (+--)$, while $\partial\rho/\partial P > 0$. In Hofer-quad, this criterion requires to replace $lhs = (P_2|P_2| - P_1|P_1|)/2$, unfolding the expression to $P < 0$ unphysical domain with correct P-signature.

In Hofer-lin, other discretization schemes can be considered, that theoretically can be more stable. Recovering m-dependence: $rhs \sim Lm|m|/\rho$, if ρ is replaced to the nodal average $(\rho_1 + \rho_2)/2$, it will have a wrong signature w.r.t. P_1 or P_2, dependently on the sign of m. A possible alternative is $rhs \sim Lm|m|/(m > 0?\rho_1 : \rho_2)$, C^1-continuous in $m = 0$. Although less precise than the nodal average, it possesses

correct P-signatures and should lead to a more stable solution process. This approach can be extended to other terms in the friction law in its different formulations.

Practically, Hofer-lin scheme is not yet usable in complex scenarios like N85 tests described below. In large natural gas networks, it shows worse stability ($div/tot = 7/85$) in comparison with Hofer-quad ($div/tot = 1/85$). Here div is the number of divergent cases, while tot is the total number of cases. The source of the instability can be in the violation of signature rule, in the leading as well as in sub-leading terms.

Empirical formulas: since numerical experiments show $\sim L$ behavior of precision at large n (small L), here it's only needed to find empirically plausible factors. Employing that the pressure drop is proportional to $dP \sim m^2/\rho \cdot L$, omitting constant factors, evaluating relative error of pressure drop due to ρ-variation, we have $\delta dP/dP = d\rho/\rho$, where δ denotes error estimation, d is a change over the pipe. Absolute values for all changes are taken. This formula assumes intermediate ρ values in pipe changing arbitrarily between their nodal values. An empirical factor 0.5 can be introduced, if the change is taken between the nodal values and the nodal average. The resulting empirical formula for relative error of pressure drop is

$$errP = \delta dP/dP = 0.5 d\rho/\rho. \quad (6)$$

For T-dependence, a simplified exponential model can be used: $T = T_s + (T_1 - T_s)\exp(-x/x_1)$, with the characteristic length $x_1 = c_p m/(\mu D c_h \pi)$. Here the main factor is proportional to L/x_1, since subdivisions with $L \sim x_1$ give roughly acceptable quality, $L \ll x_1$ fine quality, $L \gg x_1$ inacceptable. What "roughly acceptable" means can be found in comparison of discrete and continuous integration of the simplified model: $\delta T = |T_1 - T_s||(1 + L/x_1)^{(-x_1/L)} - 1/e|$, we obtain $errT = 0.132121$ at $L = x_1$, conservatively giving a factor 0.2. Thus, the following expression can serve as an estimator of relative T-precision:

$$errT = \delta dT/dT = 0.2 L/x_1. \quad (7)$$

The values of the characteristic length for 150km pipe scenario with different fluid composition are shown in Table II. Figure 5 confirms that the derived formulas provide tight conservative estimators for the pipe subdivision error.

Implementation of pipe subdivision: taking the desired level of relative error, we construct an estimator for subdivision number, with maximum taken over P- and T-estimators. This estimator should be evaluated for every pipe. If further accepted by the user, it provides the following subdivision algorithm with a minimal user assistance.

Algorithm (pipe-subdivision):

```
given (x1min,mmin,nmax,err_desired)
for every pipe
  compute x1,errT,errP
  if x1<x1min or |m|<mmin
    errT=0
  err=max(errT,errP)
```

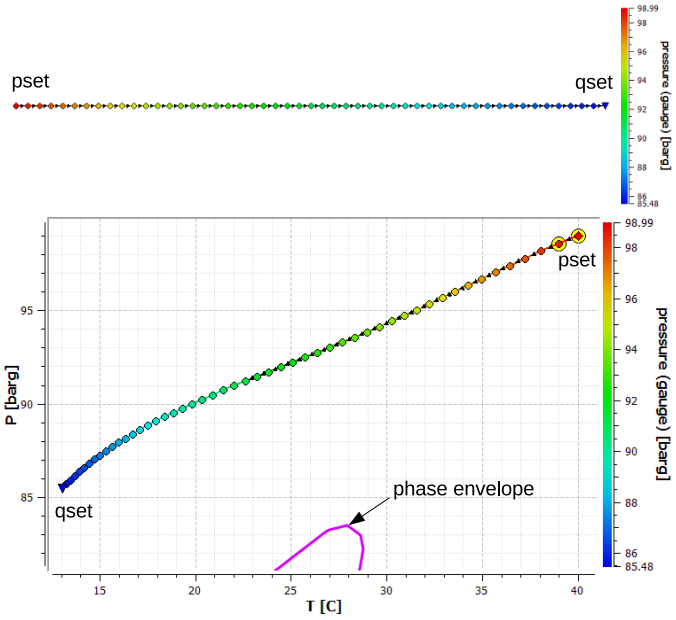

Fig. 4. Screenshot of MYNTS GUI for scenario without phase transitions. Image from [1].

```
n_suggested=[err/err_desired]+1
if n_suggested>nmax
  n_suggested=nmax.
```

Details: for better efficiency, some cutoffs are necessary. At small x_1 values, in particular, at small m, the value T rapidly jumps to T_s. Under these conditions, uniform subdivision algorithm would provide too large $n\_suggested$. To prevent this, for $x_1 < x_{1,min}$ or $|m| < m_{min}$, $errT = 0$ is set, subdivision is defined by $errP$ only. In addition, if $n\_suggested > n_{max}$, it is set to n_{max}.

The discretizations used here possess $\sim L$ precision dependence. A study of the other schemes with a higher order $\sim L^n$ dependence is in our further plans. The tradeoff between precision and stability should be also considered. For higher order schemes, the empirical formulas should be upgraded, while the subdivision algorithm remains the same.

The approach requires at least one preliminary simulation to find all necessary parameters. The pipe subdivision is sensitive to such details as mass flow, temperature, density, gas composition, and is performed for a given scenario. If scenario is changed, subdivision should be repeated starting from the raw level.

The usage of the algorithm proceeds via specification of input parameters, listed in Table III. The resulting output values are stored per pipe and can be visualized. The application of the algorithm to realistic networks is presented in the next section.

V. Numerical Experiments

To test the implemented algorithms, we apply them to a number of realistic network problems. At first, we test phase transition detection, then pipe subdivision algorithm.

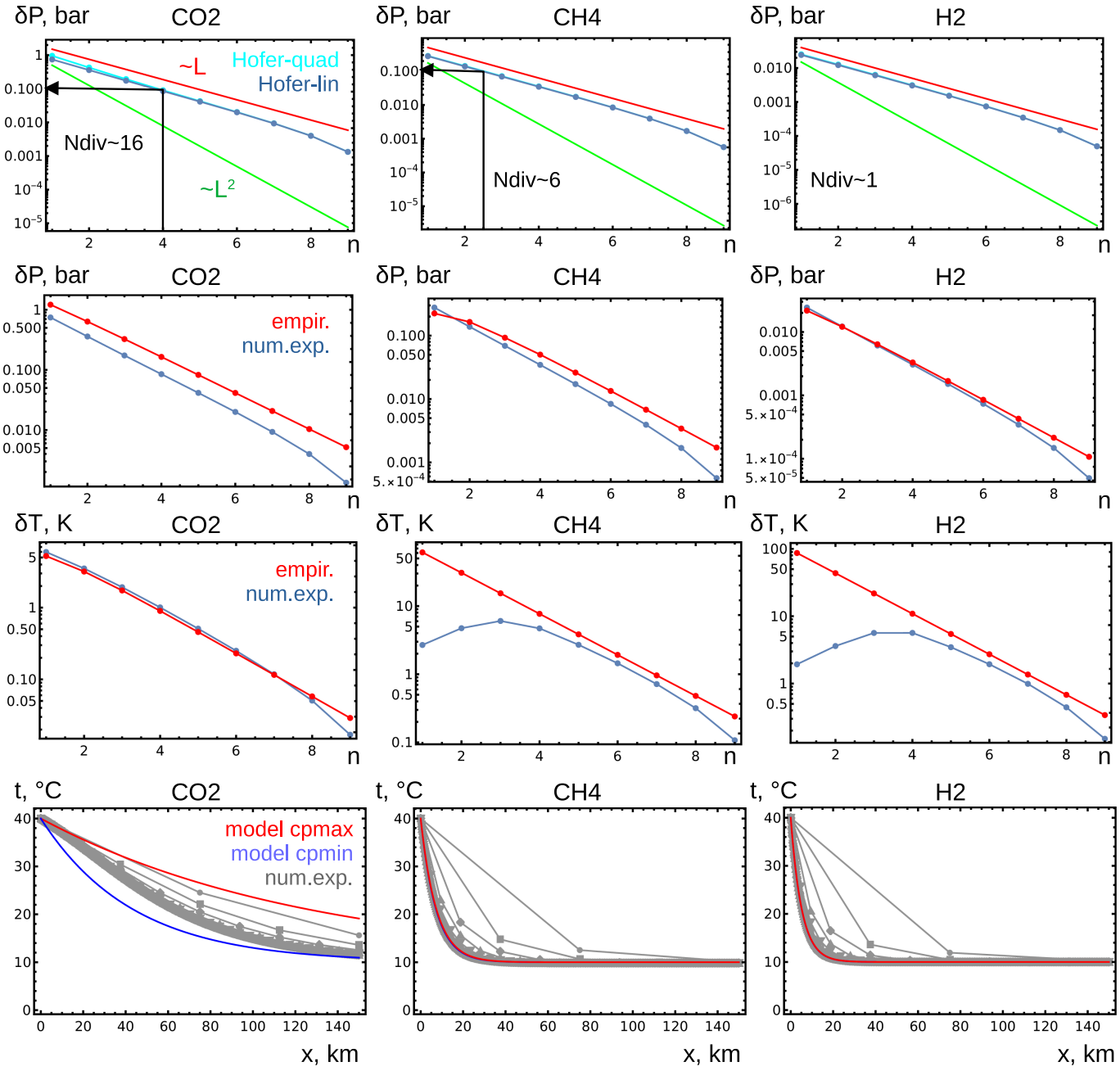

Fig. 5. Dependence of the simulation precision of pressure δP and absolute temperature δT on subdivision level n; dependence of relative temperature t on coordinate along the pipe x.

Testing phase transition detection (scen1-2): here we use a pipe segment with parameters taken from [2]. In our experiments, two scenarios are considered, see Table I. In the first scenario, a small flow is set, at which phase transitions do not occur. The entire pipe is filled with liquid or supercritical fluid. In the second scenario, a larger flow is set, the pressure drops more strongly, and a phase transition occurs in the system. Both scenarios use a mixture of 95% CO_2, 3% N_2, 2% O_2, see Figure 9 in [2]. The pipe is laid horizontally with $h = 0$.

Figure 3 shows the convergence characteristics for our test scenarios, left column for scen1, right column for scen2. The dimensionless precision parameter $prec = \max(res_i/norm_i)$ is defined as the maximum of the residuals of the equations divided by the normalizing value, for each equation its own. For the Kirchhoff equation of conservation of flow, the friction law in quadratic form, and the gas law expressed with respect to density, the normalization factors $norm = (1kg/s, 100bar^2, 1kg/m^3)$ are chosen, respectively. In our system, the equations and their normalizing factors can be

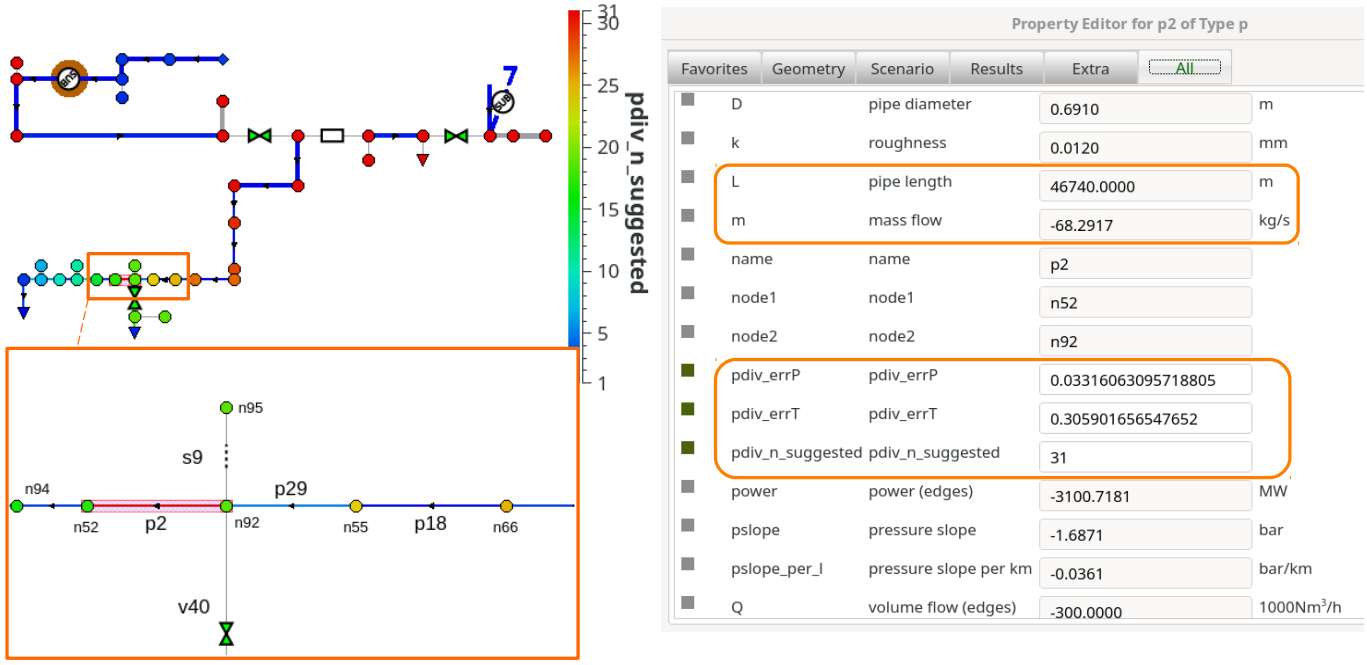

Fig. 6. Top-left: distribution of pipe subdivision estimator $n\_suggested$ over the test network N1. Bottom-left: closeup to the pipe with the largest subdivision. Right: parameters of the selected pipe.

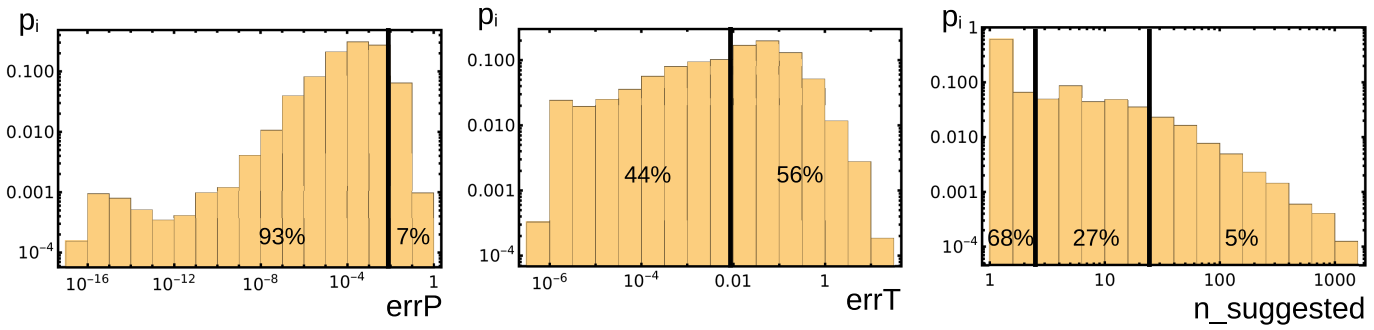

Fig. 7. Distribution of pipe subdivision estimators for the test networks N85.

TABLE III
PARAMETERS OF PIPE SUBDIVISION ALGORITHM

parameter	meaning
activation: pdiv	0/1 (default 0, inactive)
input parameters:	
pdiv_x1min	x1-cutoff (default 1 [m])
pdiv_mmin	m-cutoff (default 1 [kg/s])
pdiv_nmax	n-clamp (default 1000)
pdiv_err_desired	relative error desired (default 0.01)
output values, per pipe:	
pdiv_errP	estimated relative error of pressure change
pdiv_errT	estimated relative error of temperature change
pdiv_n_suggested	number of subdivisions suggested

freely configured by the user. For a purely 1-phase solution scen1 shown in Figure 3(a) and (c), the value of $prec$ decreases exponentially with the number of iterations and the solution procedure converges. For scen2, as seen in Figure 3(b) and (d), the procedure has cycling. In more detail, we see that there is a converging region for the 1-phase and a part of the 2-phase state, after which a temperature jump occurs, and oscillations are observed in the remaining pipe segment.

Along with the two main scenarios, we ran a number of additional simulations with small qset variations around the specified values. Simulations show stability of the effects, convergence in the 1-phase solution, and divergence in the 2-phase solution. The reason for this divergence is that EOS and the enthalpy function receive large derivatives in the phase transition region. These functions are actually jump-like for a pure substance and formally continuous for a mixture, but at a low concentration of impurities, the derivatives are still large.

A prototype example of such instability is the logistic map: $x_{i+1} = rx_i(1-x_i)$, which characterizes the behavior of simple iterations near the root $x = 1-1/r$. When r rises from 1, and

passes the value 3, the absolute value of the r.h.s. derivative of the logistic map equation exceeds 1, which is a critical value for the convergence of simple iterations. Below this value, the iterations converge. Above it, limit cycles appear, first with a multiplicity of 2, then they double, and finally the system goes to chaos.

Qualitatively, the same effects happen in our case. In principle, the stabilization algorithm helps to overcome such divergences, but for an ever higher derivative it becomes less and less effective. We are going to explore this problem in more detail in our future work. In order to overcome the divergence, we can try to adjust the weight parameter in the stabilizing algorithm. The dynamic solver behaves in much the same way as weighted relaxation with a low weight; with a decrease in the integration step, the stability of the integration also increases. As shown in Figure 1, high derivatives only occur for EOS in the form $\rho(T, P)$, changing variables to $P(T, \rho)$ could also be a solution of the problem.

At the same time, only scenarios in which phase transitions and associated divergences do not occur are to be considered within the scope of the technical task set. For such solutions, it is required to determine the proximity of the solution to the region of phase transitions. That can be done using the proximity-alarm algorithm described above.

Figure 4 shows the screenshots for scen1 solution in MYNTS GUI. At the top, there is the pipe geometry with the pressure profile shown in color. At the bottom, there is the solution on the (T, P)-plane, where a part of the phase envelope is also shown. The yellow disks show the proximity-alarm triggered in the given node for the values $dT = 1K$, $dP = 1bar$. The first 2 nodes near pset appear to be close to the spurious line on the phase diagram. The alarm in them can be canceled, because they are located top-right to the phase envelope, in the supercritical region. In general, this visual criterion is difficult to automate, since phase envelopes can have a more complex appearance than in the figures of this paper. Further, the figure shows how the solution trajectory passes at a safe distance from the phase envelope, providing the required CO_2 transport without phase transitions.

Testing pipe subdivision algorithm (scen3): here we consider the same pipe, filled with pure $\{CO_2, CH_4, H_2\}$ fluids, in various subdivisions (4). Other settings are given in Table I. The results are displayed in Figure 5.

The first row shows δP precision dependence on subdivision number n. Numerical experiments are shown by blue line with dots. This line closely follows $\sim L$ dependence on the length of pipe segment, for all fluids. The values of $N_{div} = 2^n$ for $\delta P = 0.1bar$ are shown. Two discretizations, Hofer-quad and Hofer-lin, shown by shades of blue, are almost coincident for CO_2 and coincident for other fluids.

The second and the third rows show the same numerical experiments in comparison with δP and δT empirical estimators. The estimators restrict the experiments from above almost everywhere, they are almost coincident for $\delta P(H_2)$ and $\delta T(CO_2)$, and closely approaching the experimental points at large n for other cases.

The last row shows the temperature distribution in these numerical experiments in comparison with the simplified model, used in the derivation of δT-estimator. Red and blue lines show the model results for maximal and minimal value of the heat capacity c_p, see Table II. For CO_2, these lines are different, they restrict the experimental subdivision, shown by gray lines with dots, from above and from below. For CH_4/H_2, the upper and lower c_p-values are very close and produce visually coincident model curves. Also typical for supercritical/liquid CO_2 are much larger values of c_p, in comparison with gaseous CH_4/H_2. This leads to a larger characteristic length x_1 and a slower temperature drop over the length.

Details: the simplified model should not coincide with simulation exactly, it contains only the simplest $c_p dT$ term in the temperature equation, while the simulation is more precise, contains additional terms, such as Joule-Thomson effect, gravity term, etc. In the cases considered, the pressure drop is small and the pipe is laid horizontally, so such effects are negligible. However, they can be activated in other scenarios. Also, the simplified model is valid only for constant c_p and x_1, while their variation over the pipe makes the model solution more approximate. In our simulation, the c_p dependence on pressure and temperature is computed by the GERG module.

Applying pipe subdivision algorithm (scen4-6): in the next scenario scen4, we consider a natural gas network N1 of moderate size, shown in Figure 6 top left. It contains 100 nodes, connected by 111 edges, 34 of them are pipes. The pipe subdivision algorithm with default settings produced $n\_suggested$ value, visualized in the figure. The value is peaked at $n\_suggested = 31$ on a 47km long pipe, shown on a closeup (Figure 6 bottom left), with parameters shown in Figure 6 right. The pipe possesses a moderate flow and large $errT$, that defines the subdivision.

In the following scenario scen5, we consider the same 150km pipe as in scen1-3 experiments, filled by pure CO_2, other parameters selected as in scen3a. The first iteration of pipe subdivision algorithm produces: $errP = 0.173$, $errT = 2.04$, $n\_suggested = 205$. Again, T-estimator defines the subdivision. Taking the suggested value, the pipe is subdivided to 205 equal pieces, and the second iteration of the algorithm is applied. It produces $errP$ varied in the range $0.00389 - 7.39 \cdot 10^{-5}$ from inlet to outlet, and $errT$ inbetween $0.00973 - 0.01005$, the value $n\_suggested$ is now $1 - 2$. We see that $errT$ is very close to the desired value 0.01, while $n\_suggested$ is balanced on the border of this value, so that no further subdivision is necessary. The final result satisfies all criteria and can be accepted.

Details: $errP$ changes over segments, due to the coupling to a non-linear density profile. As a result, a uniform subdivision dx gives non-uniform subdivision $d\rho$, initially large, then smaller, while our estimation supposed uniform $d\rho$. In the given scenario, $errP < err\_desired$, this effect is not important. In other scenarios, if $errP > err\_desired$, one more iteration of the algorithm can be needed. The alternative is to construct an adaptive subdivision, following $d\rho/\rho$ profile.

In the next scenario scen6, we consider a set of 85 natural

gas networks of large size [19], provided for benchmarking by our industrial partner. Each has 3000 to 4000 edges, mostly pipes. One iteration of pipe subdivision algorithm has been applied. The output is shown as a histogram in Figure 7, here p_i is probability of location in the i-th bin. The results for $err\_desired = 0.01$ show that 68% of the pipes do not require subdivision ($n\_suggested \sim 1 - 2$), 27% require moderate subdivision ($n\_suggested \sim 3 - 20$), and only the remaining 5% require large subdivision ($n\_suggested > 20$).

VI. Conclusion

In this paper, we have considered a numerical simulation of the stationary process of CO_2 transport with impurities and phase transitions. We have developed the algorithms that allow to solve scenarios of CO_2 transport in the liquid or supercritical phase and to detect proximity to the phase transition region. We have analyzed a convergence of the solution algorithms in connection with fast and abrupt changes of the equation of state and the enthalpy function in the region of phase transitions.

The performed numerical experiments show that the scenarios with a single CO_2 phase converge. For the obtained temperature and pressure profiles, a conservative algorithm for detecting the proximity of phase transitions can be applied, giving the solution to the technical problem posed. At the same time, divergences can occur in scenarios with phase transitions due to the abrupt change of thermodynamic parameters. Questions about the possible suppression of these divergences as well as improved detection of phase transitions are the subject of our further work.

Also, in this paper, an algorithm for subdivision of pipes for achieving a required precision of simulation is constructed. The algorithm uses empirical formulas for conservative error estimation, derived on the basis of numerical experiments. The application of the algorithm to realistic $CO_2/CH_4/H_2$ transport scenarios shows a good correspondence of predicted and measured precision. An additional study is planned on implementation of higher order finite difference schemes and an adaptive non-uniform subdivision for further improvement of the efficiency of the algorithm.

Acknowledgments

The work has been supported by Fraunhofer research cluster CINES. We acknowledge support from Open Grid Europe GmbH in the development and testing of the software. We also thank the organizers and participants of the conference INFOCOMP 2023 for fruitful discussions.

References

[1] 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.

[2] M. Nimtz, M. Klatt, B. Wiese, M. Kühn, and H.-J. Kraautz, “Modelling of the CO2 process- and transport chain in CCS systems – Examination of transport and storage processes”, Chemie der Erde – Geochemistry, vol. 70, suppl. 3, 2010, pp. 185-192.

[3] S. Liljemark, K. Arvidsson, M. T. P. Mc Cann, H. Tummescheit, and S. Velut, “Dynamic simulation of a carbon dioxide transfer pipeline for analysis of normal operation and failure modes”, Energy Procedia, vol. 4, 2011, pp. 3040-3047.

[4] 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.

[5] 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.

[6] L. Raimondi, “CO2 Transportation with Pipelines - Model Analysis for Steady, Dynamic and Relief Simulation”, Chemical Engineering Transactions, vol. 36, 2014, pp. 619-624.

[7] M. Drescher et al., “Towards a Thorough Validation of Simulation Tools for CO2 Pipeline Transport”, Energy Procedia, vol. 114, 2017, pp. 6730-6740.

[8] B. Chen, H. Guo, S. Bai, and S. Cao, “Optimization of process parameters for pipeline CO2 transportation with impurities”, IOP Conf. Series: Earth and Environmental Science, vol. 300, 2019, 022002.

[9] M. Vitali et al., “Risks and Safety of CO2 Transport via Pipeline: A Review of Risk Analysis and Modeling Approaches for Accidental Releases”, Energies, vol. 14, 2021, 4601.

[10] L. Raimondi, “CCS Technology - CO2 Transportation and Relief Simulation in the Critical Region for HSE Assessment”, Chemical Engineering Transactions, vol. 91, 2022, pp. 43-48.

[11] 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.

[12] X. Luo, M. Wang, E. Oko, and C. Okezue, “Simulation-based Techno-economic Evaluation for Optimal Design of CO2 Transport Pipeline Network”, Applied Energy, vol. 132, 2014, pp. 610-620.

[13] V. E. Onyebuchi, A. Kolios, D. P. Hanak, C. Biliyok, and V. Manovic, “A systematic review of key challenges of CO2 transport via pipelines”, Renewable and Sustainable Energy Reviews, vol. 81, part 2, 2018, pp. 2563-2583.

[14] H. Lu, X. Ma, K. Huang, L. Fu, and M. Azimi, “Carbon dioxide transport via pipelines: A systematic review”, Journal of Cleaner Production, vol. 266, 2020, 121994.

[15] 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.

[16] T. Clees, I. Nikitin, and L. Nikitina, “Making Network Solvers Globally Convergent”, Advances in Intelligent Systems and Computing, vol. 676, 2018, pp. 140-153.

[17] 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.

[18] 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.

[19] 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, pp. 138-152, 2023.

[20] J. Nikuradse, “Laws of flow in rough pipes”, NACA Technical Memorandum 1292, Washington, 1950.

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

[22] J. Mischner, H. G. Fasold, and K. Kadner, System-planning basics of gas supply, Oldenbourg Industrieverlag GmbH, 2011 (in German).

[23] 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.

[24] 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.

[25] 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, 2015.
[26] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C, Cambridge University Press, 1992.
[27] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[28] E. L. Allgower and K. Georg, Introduction to Numerical Continuation Methods, SIAM, 2003.
[29] J. Katzenelson, "An algorithm for solving nonlinear resistor networks", Bell System Technical J., vol. 44, 1965, pp. 1605-1620.
[30] M. J. Chien and E. S. Kuh, "Solving piecewise-linear equations for resistive networks", Int. J. of Circuit Theory and Applications, vol. 4, 1976, pp. 1-24.
[31] A. Griewank, J.-U. Bernt, M. Radons, and T. Streubel, "Solving piecewise linear systems in abs-normal form", Linear Algebra and its Applications, vol. 471, 2015, pp. 500-530.
[32] T.-P. Azevedo-Perdicoúlis, F. Perestrelo, and R. Almeida, "A note on convergence of finite differences schemata for gas network simulation", in Proc. of the 22nd International Conference on Process Control, pp. 274-279, IEEE, 2019.
[33] 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.
[34] 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.
[35] O. Schenk and K. Gärtner, "PARDISO", in: D. Padua (eds) Encyclopedia of Parallel Computing, Springer, 2011.