Smart Energy 22 (2026) 100245

Contents lists available at ScienceDirect

Smart Energy

journal homepage: www.journals.elsevier.com/smart-energy

Tensor-based modeling framework for district heating pipes ☆

Nicholas Tedjosantoso [a,*], Arne Speerforck [b], Torben Warnecke [c], Hans Schäfers [a,d], Gerwald Lichtenberg [a,d]

[a] Competence Center for Energy Transition (CC4E), Hamburg University of Applied Sciences, Steindamm 96, Hamburg, 20099, Germany
[b] Institute of Engineering Thermodynamics, Hamburg University of Technology, Denickestr. 17, Hamburg, 21073, Germany
[c] Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, Hamburg, 22607, Germany
[d] Faculty of Life Sciences, Hamburg University of Applied Sciences, Ulmenliet 20, Hamburg, 21033, Germany

ARTICLE INFO

Keywords:
Multilinear modeling
Tensor decomposition
Simulation
Pipe model
District heating

ABSTRACT

District heating network simulation faces a fundamental computational challenge: traditional nonlinear models become intractable at large scales due to the curse of dimensionality, while linear models cannot accurately represent the nonlinear dynamics essential for district heating systems. Tensor-based methods have demonstrated effectiveness in modeling heating, ventilation, and air conditioning (HVAC) as well as local heating systems by providing a scalable compromise between accuracy and computational efficiency, yet their application to district heating networks is first described in this paper. This work applies multilinear time-invariant (MTI) modeling using a tensor-based framework for scalable representations of district heating networks.

Tensor and multilinear functions efficiently represent the governing equations and their nonlinear relationships, especially the quadratic pressure-loss relationships defined by the Darcy-Weisbach equation and nonlinear friction factors across flow regimes without causing an exponential growth in model complexity. Binary variables model discontinuous transitions between laminar and turbulent flow, maintaining computational tractability while preserving physical accuracy. The tensor structure inherently avoids the curse of dimensionality that constrains conventional approaches by factorization. Benchmarking against established models on a small network shows minimal deviations alongside considerable memory reductions, demonstrating the potential of tensor-based methods for efficient simulation and optimization of large-scale district heating networks and supporting the integration of renewable energy sources and advanced control strategies essential for modern energy-efficient systems.

1. Introduction

In this section, a brief review of existing approaches for simulation of a district heating network (DHN) (Section 1.1) is first presented. This is then followed by a brief assessment on the scalability challenge of these models (Section 1.2) and ends with tensor-based methods (Section 1.3) as a possible solution.

1.1. Literature review

Depending on the chosen modeling approach, a DHN model can be categorized by whether it represents steady-state, quasi-dynamic or dynamic behavior [1–3]. For steady-state analysis, the dynamic properties of both thermal and hydraulic processes are neglected. Quasi-dynamic models on the other hand, solve the thermal equations dynamically, while keeping hydraulic equations as steady state [2,4]. In dynamic models, the dynamic aspects of both thermal and hydraulic states are considered [2,5]. Moreover, depending on the control strategies, three commonly used approaches exist: Constant Flow Variable Temperature (CF-VT), Variable Flow Constant Temperature (VF-CT), and Variable Flow Variable Temperature (VF-VT) [1].

In recent years, Modelica-based tools have shown to be well suited for the simulation of DHNs, covering the integration of new heat sources [6,7] and storage options [8], co-planning approach [9], and network optimization [10,11], among others, due to its characteristics as an object-oriented, equation-based, acausal modeling language [1]. Furthermore, due to its acausality, Modelica is capable of modeling bidirectional flows. Several established libraries for DHN simulations are for example *ClaRa* [12], *TransiEnt* [13], *IBPSA* [14], and

☆ This article is part of a Special issue entitled: 'SESAAU2025' published in Smart Energy.

* Corresponding author.
E-mail address: Nicholas.Tedjosantoso@haw-hamburg.de (N. Tedjosantoso).

https://doi.org/10.1016/j.segy.2026.100245
Received 26 February 2026; Received in revised form 8 April 2026; Accepted 20 April 2026
Available online 25 April 2026

AixLib [15]. Despite the open-source nature of the libraries and of the Modelica programming language, these libraries are highly dependent and are tested on the commercial software Dymola [3]. To address this, a number of new libraries, e.g. *DHN4Control* [11], *n5GDHC_sim* [8] and a new concept shown in [3], have been introduced for the open-source simulation tool OpenModelica. The performance of OpenModelica, however, is still incomparable to Dymola, as shown in [3].

In recent years, a number of open-source Python libraries have also been developed for the simulation of DHNs. *DHNx* is a components-based simulation and optimization tool and is based on the open-source Open Energy Modeling Framework (*oemof*), which implements a steady-state modeling approach [16]. *DiGriPy* similarly builds upon *oemof* and has a similar area of application as *DHNx* [17]. The open-source package *pandapipes* offers a table-based steady-state tool for the simulation of gas and heating networks, with a focus on coupled simulations of power, gas and heat sectors [18]. In *PyDHN*, a quasi-dynamic simulation tool is presented, where a Lagrange-based modeling approach is used for the thermal dynamics [19], which extends the concept introduced by [20]. Similarly, *HeatNetSim* offers a quasi-dynamic simulation tool [5]. In *Jarvis*, a quasi-dynamic modeling approach based on distributed simulation architecture is presented. Despite its modern approach, its performance has not been systematically evaluated [21].

Beyond conventional simulation tools, recent works have explored analytical approaches such as frequency-domain transfer function [22] and Green's function [23], as well as data-driven surrogates based on graph neural networks [24,25], which target computational speedups.

1.2. The scalability challenge in district heating networks

Modeling and simulation of DHNs requires capturing both thermal and hydraulic dynamics across potentially hundreds or thousands of interconnected pipe segments. Large-scale networks present a fundamental computational challenge: the complexity of nonlinear models grows exponentially with system size [1,26], a phenomenon known as the curse of dimensionality [27]. While linear models offer computational efficiency, they cannot adequately represent the nonlinear dynamics inherent in fluid flow through pipes [1,28].

Existing simulation tools address this challenge through different compromises. Modelica-based approaches (*ClaRa*, *TransiEnt*, *IBPSA*, *AixLib*) provide dynamic, high-fidelity simulation but suffer from computational scalability limitations [3,29]. *DHNx*, *DiGriPy*, *pandapipes* improve computational efficiency by adopting steady-state approximations. *PyDHN*, on the other hand, is limited by its simulation speed [19]. Attempts to accelerate simulations through data-driven surrogates [24], transfer function [22] or Green's function [23] introduce their own limitations, such as dependence on data availability and constant flow. As a result, the fundamental tension between nonlinear model fidelity and computational tractability persists, particularly for VF-VT control strategies, which are essential for future district heating systems and the subsequent integration of renewable energy sources [1].

1.3. Tensor-based methods for scalable solutions

In other thermal-hydraulic domains, such as heating, ventilation, and air conditioning (HVAC) and local heating systems, tensor-based multilinear modeling approaches have been demonstrated to capture the system dynamics including inherent nonlinear behavior without the computational overhead of traditional nonlinear approaches [30,31]. Through structured tensor decomposition, these behaviors can be expressed by high-dimensional relationships in a factored way, such as canonical polyadic (CP) decomposition, and tensor train (TT) decomposition [27,32]. This structured representation dramatically reduces computational requirements without sacrificing accuracy, as the system's dynamics are captured through a smaller number of variables [27].

Although HVAC systems, local heating systems, and DHNs share similar governing equations, the considerably larger scale of DH systems poses distinct computational challenges. Despite the potential effectiveness of tensor methods in related thermal systems, their application to large-scale DHN modeling, where the curse of dimensionality is critical, remains unexplored. This gap is particularly significant given the urgent need for scalable simulation tools that balance accuracy and computational efficiency for large-scale systems [1].

The paper proceeds as follows. The fundamental physics of district heating pipes are presented in Section 2. Section 3 then introduces tensors and multilinearity and its subsequent implementation on the fundamental equations of district heating pipes. The results are then compared against established Modelica models in Section 4, followed by an analysis of the runtime performance of the model in Section 5 and finally ends with a conclusion in Section 6.

2. Modeling concept

This section presents the fundamental equations of district heating pipes. The conservation laws for mass (Section 2.1), energy (Section 2.2), and momentum (Section 2.3) form the basis of the proposed model.

2.1. Conservation of mass

The proposed model uses a static mass balance with no consideration of mass storage effects or density changes. This assumption of an incompressible fluid is often used in the context of a DHN [3,19,33]. The resulting mass balance between the inflow $q_{m,in}$ and outflow $q_{m,out}$ can be described by

$$q_{m,in} + q_{m,out} = 0. \tag{1}$$

2.2. Conservation of energy

Two main approaches exist in the literature to model the thermal behavior of a pipe [3,34]. The first approach discretizes a pipe spatially along the flow direction into $N > 1$ equal segments, whereas precision increases with the number N of segments. This approach is called lumped-parameter pipe model [34]. In contrast, the distributed-parameter pipe model, also called *PlugFlow*, determines the properties of the fluid at the inlet and outlet using the residence time of the fluid [3,35]. Mathematically, *PlugFlow* model can be derived from the lumped-parameter model using an infinitesimal segment volume, by going to the limit $N \to \infty$ [34]. Based on the assessment in [3], heat losses can be estimated more accurately using a lumped-parameter model. Moreover, the computation of heat losses using *PlugFlow* models is not valid for sudden temperature changes and flow reversal. Therefore, this work adopts the lumped-parameter approach.

The dynamic behavior of the discretized pipe model of length l is described by differential equations derived in [34], with boundaries at $i = 1$ and $i = N$. Using this notation, the distributed temperature of the pipe segments is denoted by $T_1, \dots, T_N$ with the corresponding ambient temperature given by $T_{a,1}, \dots, T_{a,N}$. The dynamics are further influenced by the thermal capacitance $C_{th,i}$, the specific heat capacity c_f, the mass flow q_m, and the combined thermal conductance $G_{th,i}$, which combines the convective, conductive, and radiative thermal conductance of the pipe wall and its insulation, as shown below

$$C_{\mathrm{th},i}\frac{\mathrm{d}T_i}{\mathrm{d}t} = -c_f q_m T_i + c_f q_m T_{i-1} - G_{\mathrm{th},i}T_i + G_{\mathrm{th},i}T_{a,i} \tag{2}$$
$$i = 1, 2, \dots, N.$$

The discretized pipe model is visualized in Fig. 1.

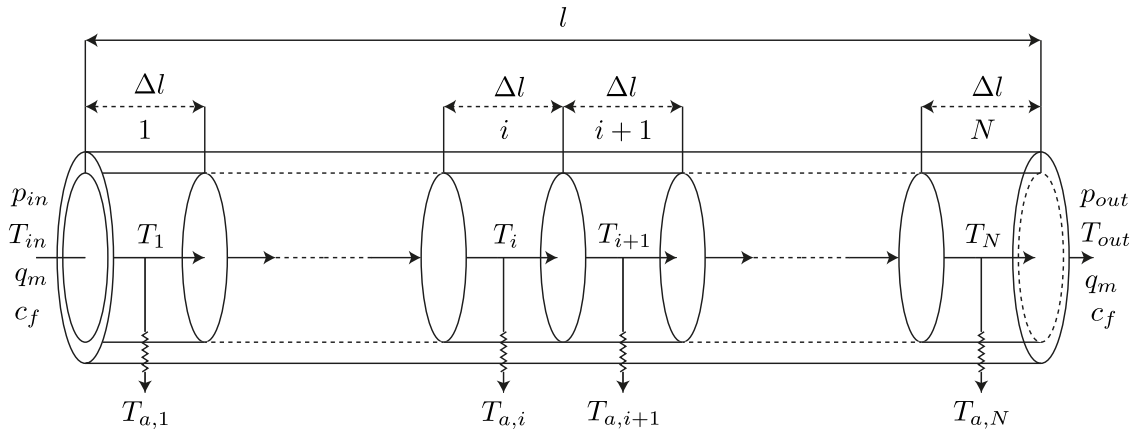


Fig. 1. Disaggregated pipe with constant discretization length Δl [34].

2.3. Conservation of momentum

Following [36], the overall pressure change Δp is influenced by the pressure loss from external forces Δp_{ext}, the dynamic pressure difference Δq, and the rate of change of mass flow q_m relative to its inertance L, which denotes the reluctance of the mass flow rate to change. The momentum balance can then be formulated as

$$\frac{d\,q_m}{dt}\,L + \Delta q = \Delta p - \Delta p_{ext}. \tag{3}$$

In this work, the difference in dynamic pressure Δq is neglected, as the acceleration of a particle in the fluid is neglected [3]. Similarly, the consideration of pressure loss from external forces Δp_{ext} is limited to pressure loss due to pipe friction. Furthermore, the effect of inertance is negligible for typical DHN operation [37]. These simplifications therefore lead to a purely algebraic equation

$$\Delta p = \Delta p_{ext}, \tag{4}$$

for the momentum balance.

To characterize the pressure loss Δp in a cylindrical pipe due to friction, the following Darcy-Weisbach pressure-loss equation

$$\frac{\Delta p}{l} = \lambda\,\frac{8\,{q_m}^2}{\pi^2\,\rho\,d^5} \tag{5}$$

is used [38]. This equation is influenced by the length l of the pipe segment, the dimensionless pipe friction factor λ, the hydraulic diameter d of the pipe, and the density ρ of the fluid. The equation above shows that the pressure loss is proportional to the square of the mass flow q_m. The friction factor λ is called Darcy-Weisbach friction factor and depends on both the characteristic of the pipe and of the fluid flow. Generally, the calculation can be divided into two different flow regimes: laminar and turbulent. For laminar flow regimes, the friction factor can be determined using

$$\lambda = \frac{64}{Re}, \tag{6}$$

where the Reynolds number

$$Re = \frac{4\,q_m}{\pi\,d\,\mu} \tag{7}$$

depends on the mass flow q_m, the dynamic viscosity μ and the hydraulic diameter d of the pipe [38].

For turbulent flow regimes, the friction factor λ can be described implicitly using the empirical Colebrook–White equation

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{2.51}{Re\sqrt{\lambda}} + \frac{\varepsilon}{3.71d}\right), \tag{8}$$

where the roughness ε depends on the material of the pipe [38].

3. Tensor and multilinear functions in district heating pipes

As shown in Section 2, the equations describing a pipe flow are nonlinear but have a defined structure with repeating terms, which can be exploited using defined mathematical methods. To bridge the gap between physical equations and mathematical abstractions, tensors and multilinear functions are introduced next.

A *tensor* is regarded as a multidimensional array or d-way array, with d representing the order of the tensor [32,39]. Under this convention, vectors and matrices are special cases of first- and second-order tensors, which defines a tensor as a natural generalization for higher-dimensional representations [32].

A function $f(x)$ with $x = (x_1, \dots, x_n)$ is called *multilinear* if it is linear with respect to each individual variable x_i, when all others are held constant [39,40]. Exemplary for $x = (x_1, x_2)$, a multilinear function may look like the following equation

$$f(x) = x_1 + x_2 + x_1\,x_2. \tag{9}$$

When all variables except one are constant, a multilinear function becomes a linear function. Taking (9) as an example and treating x_2 as a fixed constant $x_2 = \bar{x}_2$, it becomes

$$f(x_1) = (1 + \bar{x}_2)\,x_1 + \bar{x}_2. \tag{10}$$

This function is an affine function, which is a linear function followed by a translation [41]. Out of these definitions, polynomial functions of degree 2 or higher, such as

$$f(x) = x_1^2\,x_2, \tag{11}$$

are not multilinear functions.

Multilinear functions can furthermore be efficiently and compactly represented using tensors [39,42]. This sets the stage for the subsequent sections, where the inherent connection between tensor, multilinear function, and the underlying physics of district heating pipes are explored. A simplified representation of the underlying workflow is illustrated in Fig. 2.

3.1. Multilinearity of district heating pipes

Flow of fluid within a district heating pipe inherently exhibits multilinear relationships based on fundamental physical principles. To assess these relationships, consider the thermal balance

$$C_{\mathrm{th},i}\,\frac{\mathrm{d}T_i}{\mathrm{d}t} = -\,c_f\,q_m\,T_i + c_f\,q_m\,T_{i-1} - G_{th,i}\,T_i + G_{th,i}\,T_{a,i} \tag{2}$$
$$i = 1, 2, \dots, N,$$

where all time-varying variables are highlighted in red. Focusing on the convective heat term $c_f\,q_m\,T_i$, the heat transport involves the

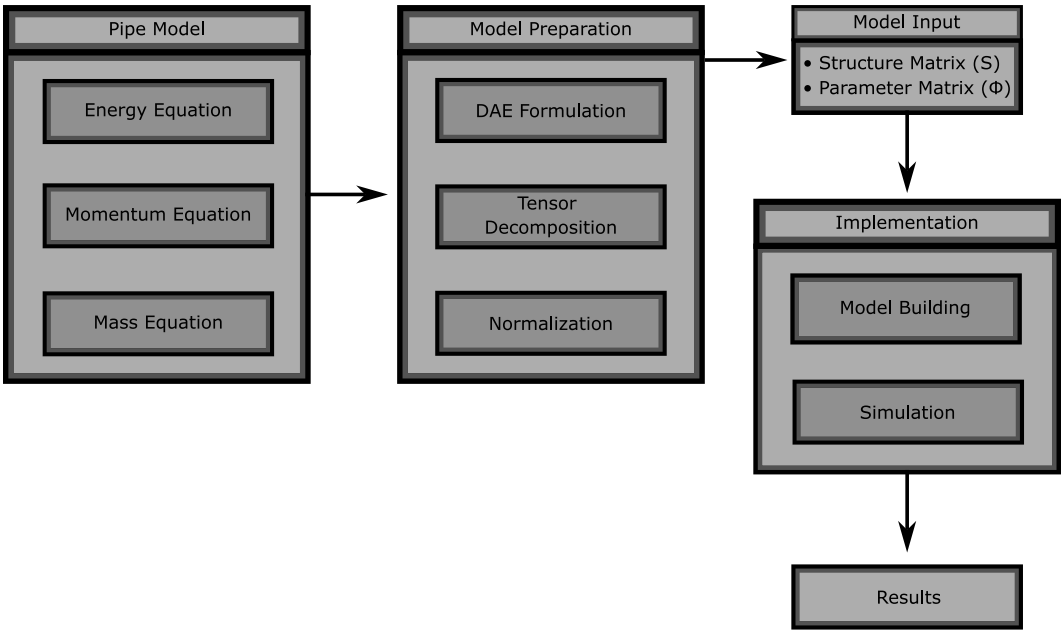


Fig. 2. Simplified workflow of this research.

product of two variables. Fixing one variable (e.g., T_i), the result is an affine function for q_m; conversely, fixing q_m, the product is affine for T_i. Such a relationship falls under the definition of a multilinear function.

For the pressure loss, the Darcy-Weisbach equation,

$$\frac{\Delta p}{l} = \lambda \frac{8\, {q_m}^2}{\pi^2\, \rho\, d^5}, \tag{5}$$

with time-varying λ and q_m is quadratic. Quadratic functions can however be reformulated into multilinear equation systems using auxiliary variables [43]. An implementation of this can be found in the following section (Section 3.2).

These inherent multilinearities motivate the adoption of multilinear time-invariant (MTI) modeling frameworks, which exploit tensor-based representations of these multiplicative variable interactions to achieve computational efficiency without sacrificing physical fidelity.

3.2. Multilinear approximation of nonlinear friction factor

The friction factor λ is not a constant but depends on the Reynolds number. Under the incompressible flow assumption (Section 2.1), with constant dynamic viscosity μ, friction factor λ is a function of mass flow rate alone: $\lambda = \lambda(q_m)$. This introduces a significant challenge: the pressure drop now couples a nonlinear, mass flow-dependent friction factor $\lambda(q_m)$ together with the quadratic mass flow term ${q_m}^2$, with the relationship differing depending on the flow regime.

For laminar flow regimes (Re $<$ 2300), $\lambda_{\text{lam}} = 64/Re$ can be substituted directly into the Darcy-Weisbach equation, resulting in a linear equation:

$$\frac{\Delta p_{\text{lam}}}{l} = \frac{512\, \mu\, q_m}{\pi\, \rho\, d^4}. \tag{12}$$

For turbulent flow regimes, however, friction factor λ is governed by the implicit Colebrook–White equation,

$$\frac{1}{\sqrt{\lambda}} = -2 \log_{10}\left(\frac{2.51}{Re\sqrt{\lambda}} + \frac{\varepsilon}{3.71 d}\right). \tag{8}$$

Looking at (8), the reformulation into a multilinear function is not possible, which prohibits its integration into the MTI framework. To address this, least-squares multilinear approximation is used. Such methods have been shown to be feasible to approximate hard empirical correlations in HVAC systems (e.g. Nusselt number) [44]. For simplicity, a sharp transition at $Re = 2300$ from laminar towards turbulent flow regime is used.

The methodology is as follows: for a given pipe geometry characterized by diameter d and roughness ε, the Colebrook–White equation is solved numerically across a range of mass flow rates q_m. A multilinear polynomial is then fitted to these values using least-squares fit, yielding an explicit algebraic approximation for $\lambda(q_m)$, which preserves the multilinear structure required for MTI representation. Considering the shape of the original function, and the possibility to represent a quadratic relationship using auxiliary variables, the auxiliary variable $y_{aux} = q_m$ [43] is implemented. By treating q_m and y_{aux} as separate variables, a multilinear polynomial of degree 3 can be used as an approximation function, which leads to the following fitting function,

$$0 = a_1\, \lambda\, q_m\, y_{aux} + a_2\, \lambda\, q_m + a_3\, \lambda\, y_{aux} + a_4\, \lambda + a_5\, q_m\, y_{aux} + a_6\, q_m + a_7\, y_{aux} + a_8. \tag{13}$$

As q_m and y_{aux} are tied together, the calculated fitting parameters are non-unique. In other words, different parameter sets can yield identical $\lambda(q_m)$ values.

Since DHNs operate using standardized pipe catalogs with discrete nominal diameter d with known roughness values ε [45], fitting parameters can be pre-computed for each pipe type and stored in a lookup table. The Example of a 263 mm steel pipe with $\varepsilon = 0.007$ mm [45] can be found in the appendix (Table 4).

3.3. Implicit multilinear models

As introduced in Section 3, tensors are multidimensional arrays encompassing vectors and matrices as special cases [32,39]. Using the concept of monomial vector $m(x)$, all combinations (variable products) in a multilinear function can be systematically enumerated via the Kronecker product. Any multilinear function can then be expressed as the inner product of a coefficient vector c^T and the monomial vector $m(x)$ [27]. In the following, an example is shown using a bilinear function (x_1 and x_2) with monomial vector

$$m(x_1, x_2) = \begin{bmatrix} 1 \\ x_1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ x_2 \\ x_1 \\ x_1 x_2 \end{bmatrix} \tag{14}$$

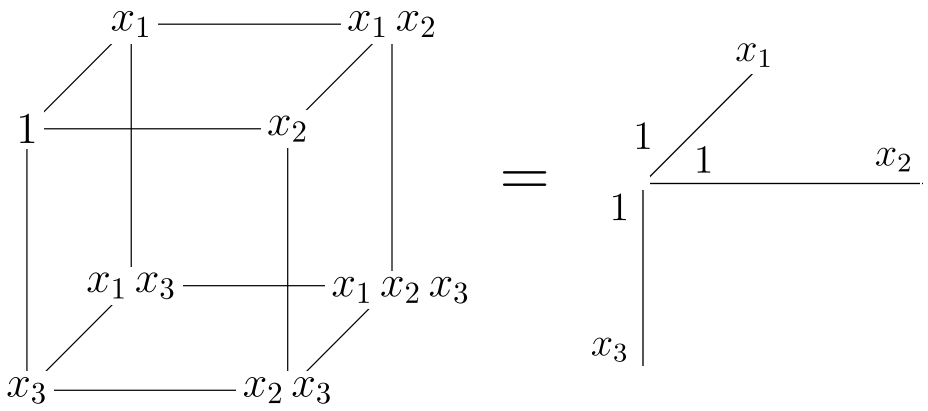


Fig. 3. Example of Canonical Polyadic (CP) decomposition.

and coefficient vector

$$c = \left[c_0, c_1, c_2, c_3\right], \tag{15}$$

which then leads to the following formulation

$$f(x_1, x_2) = c^T m(x_1, x_2) = c_0 + c_1\,x_2 + c_2\,x_1 + c_3\,x_1\,x_2. \tag{16}$$

Using the monomial vector, the implicit multilinear time-invariant (iMTI) state–space model in [27]

$$F m(\dot{x}, x, u, y) = 0 \tag{17}$$

has a state vector $x \in \mathbb{R}^n$, inputs $u \in \mathbb{R}^m$, outputs $y \in \mathbb{R}^p$, and model matrix $F \in \mathbb{R}^{2^{(2n+m+p)}\times e}$ [27]. The parameter matrix F of the model has rows corresponding to the number e of equations and columns corresponding to the monomial m. When this approach is abstracted to higher dimension, the contracted product of two tensors can be used to define multilinear vector functions [27,39]:

$$\langle H | M(\dot{x}, x, u, y)\rangle = 0 \tag{18}$$

which is analogous to the representation (17) in vector form.

While the tensor representation via the monomial vector and the contracted product (18) provides a structured mathematical framework for multilinear models, the complete tensor encounters fundamental scalability challenges for large-scale systems [27]. This issue stems from the exponential growth of the monomial tensor $M(x)$, which includes all the possible products of the variables. For a system with n state variables, m inputs, and p outputs, a total of 2^{2n+m+p} different combinations has to be stored. $M(x)$ therefore grows with the scale of $O(2^{2n+m+p})$. As the dimension of $M(x)$ also determines the dimension of H, the need for decomposition methods to reduce the dimension of the tensors is prevalent.

3.3.1. Tensor decomposition and normalization

By representing the coefficients of a multilinear function as a tensor, it becomes possible to apply tensor decomposition methods, which have proven to be useful in a range of application domains [27,32,39]. This work utilizes the canonical polyadic (CP) decomposition method as described in [27]. In its simplest sense, CP decomposition refers to the factorization of a tensor into a sum of its rank-1 tensors [32,39].

To exemplify the CP decomposition, the decomposition of a 3-way tensor $X \in \mathbb{R}^{2\times2\times2}$ is shown in Fig. 3. The full tensor is decomposed into a single rank-1 component. As the tensor is decomposed into a single rank-1 tensor, the decomposed tensor rank R is 1. Mathematically, the three-dimensional tensor X can now be concisely expressed as [39]:

$$X = M(x_1, x_2, x_3) = \begin{pmatrix}1\\x_1\end{pmatrix} \circ \begin{pmatrix}1\\x_2\end{pmatrix} \circ \begin{pmatrix}1\\x_3\end{pmatrix} \tag{19}$$

where the operator $\circ$ denotes the outer product and x_1, x_2, and x_3 correspond to each of the three modes of the tensor.

As previously mentioned, each monomial term corresponds to a tensor, leading to a total storage complexity of $O(2^i)$ in the unreduced case, with the number i of variables (states, inputs, and outputs). Further reduction towards $O(i)$ is also possible due to the monomial tensor being of rank 1. In the CP-decomposed format, the storage requirement of the tensor M is therefore reduced to $O(i)$, as the number of parameters that must be stored depends only on the rank of the tensor M. This reduction in scaling from exponential to linear is especially significant in large-scale models, where many variables are present in the system.

To further reduce the storage requirement, the CP-decomposed tensor H can be further reduced by normalization. Within this work, the Manhattan norm (1-norm) is implemented [27,46]. The calculation steps to compute the CP-normalized (CPN) are presented in [46]. By applying the CPN to the model tensor $H \in \mathbb{R}^{\overbrace{2\times2\times\cdots\times2}^{2n+m+p\ \text{times}}\times R}$, the multilinear model can be represented using two matrices, namely the structure matrix $S \in \mathbb{R}^{(2n+m+p)\times R}$ and the parameter matrix $\Phi \in \mathbb{R}^{e\times R}$ [27,46]. The set of multilinear functions can now be expressed using the 1-norm by

$$f_j(\mathbf{v}(t)) = \sum_{k=1}^{R} \Phi_{j,k} \prod_{i=1}^{2n+m+p} \left(1 - |S_{i,k}| + S_{i,k} v_i(t)\right), \tag{20}$$

where $v = (\dot{x}, x, u, y) \in \mathbb{R}^{2n+m+p}$ for all $j \in \{1, 2 \ldots, e\}$. Using this CPN format, the storage requirement can further be reduced by approximately a factor of two [27].

3.3.2. Tensor representation of the energy equation

For the implicit tensor representation of the energy equation, (2) has to be reformulated to its implicit form. To illustrate the reformulation, a pipe model with $N = 2$ is used as an example, leading to

$$C_{\text{th},1}\frac{\mathrm{d}T_1}{\mathrm{d}t} + c_f q_m T_1 - c_f q_m T_0 + G_{\text{th},1} T_1 - G_{\text{th},1} T_{a,1} = 0, \tag{21}$$

$$C_{\text{th},2}\frac{\mathrm{d}T_2}{\mathrm{d}t} + c_f q_m T_2 - c_f q_m T_1 + G_{\text{th},2} T_2 - G_{\text{th},2} T_{a,2} = 0. \tag{22}$$

Considering the standardized pipe catalogs and the assumptions of incompressible flow, the parameters $G_{th,i}$, $C_{th,i}$, and c_f are assumed to be constant for the complete length of the pipe. As district heating pipes are typically built underground, the ambient temperature T_a is assumed constant along the pipe length [47]. The used assumptions lead to the following signals:

- State variable $x = [T_1, T_2] \rightarrow n = 2$
- Inputs $u = [q_m, T_0, T_a] \rightarrow m = 3$
- Outputs $y = [\,] \rightarrow p = 0$

Using the notation of states and inputs, (21) and (22) can now be reformulated into:

$$C_{th}\,\dot{x}_1 + c_f\,u_1\,x_1 - c_f\,u_1\,u_2 + G_{th}\,x_1 - G_{th}\,u_3 = 0 \tag{23}$$

$$C_{th}\,\dot{x}_2 + c_f\,u_1\,x_2 - c_f\,u_1\,x_1 + G_{th}\,x_2 - G_{th}\,u_3 = 0 \tag{24}$$

From these implicit equations, 8 distinct terms can be identified (e.g. $\dot{x}_1$, $u_1\,u_2$, …). Using the CPN format defined in (20), the full tensor can now be decomposed into $S \in \mathbb{R}^{7\times 8}$, which represents the 8 terms out of the 7 variables $\{\dot{x}_1, \dot{x}_2, x_1, x_2, u_1, u_2, u_3\}$

$$S = \begin{array}{c} \dot{x}_1 \\ \dot{x}_2 \\ x_1 \\ x_2 \\ u_1 \\ u_2 \\ u_3 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

and $\Phi \in \mathbb{R}^{2\times 8}$ containing the coefficients from the energy equations

$$\Phi = \begin{bmatrix} C_{th} & 0 & G_{th} & 0 & -c_f & c_f & 0 & -G_{th} \\ 0 & C_{th} & 0 & G_{th} & 0 & -c_f & c_f & -G_{th} \end{bmatrix}. \tag{26}$$

This representation compactly captures the two energy balance equations compared to the full tensor representation. For this example, storing 56 values, from $\mathbb{R}^{7\times 8}$, in the CPN format is equal to the $2^7 = 128$ values required in the full tensor. With increasing discretization amounts of N, the reduction effect is even more pronounced.

3.3.3. Tensor representation of the momentum equation

The pressure drop across a pipe segment is governed by the Darcy-Weisbach Eq. (5). As discussed in Section 2.3, λ itself depends on the flow regime and based on the assumption of incompressible flow, the Reynolds number Re solely depends on q_m. In this work, the transition between laminar and turbulent flow is modeled at $Re = 2300$. As shown in (12) the iMTI framework can directly be implemented for laminar regime due to the linear relationship between Δp and q_m. In the turbulent regime, λ must be approximated first using (13). Due to the discontinuous nature of the system, a single implicit equation cannot simultaneously represent both regimes. This can be overcome by the use of binary variables in the iMTI model [27]. Using a binary variable z, a single equation can now be used to summarize the pressure drop from both flow regimes.

In the implicit form, the pressure drop of a laminar flow (12) can be reformulated as

$$\Delta p_{\text{lam}} - q_m\,\frac{512\,\mu\,l}{\pi\,\rho\,d^4} = 0. \tag{27}$$

For the turbulent flow regime, auxiliary variable $y_{aux} = q_m$ is implemented to preserve the multilinearity from the original quadratic equation (5), yielding

$$\Delta p_{turb} - \lambda\,q_m\,y_{aux}\,\frac{8\,l}{\pi^2\,\rho\,d^5} = 0. \tag{28}$$

From the multilinear approximation (13) of λ, this function can now be factorized into its CPN equivalent formulation (20). Within the implicit multilinear framework, λ is represented as:

$$\begin{aligned} 0 =& \Phi_{11}(1 - |S_{11}| + S_{11}\,y_{aux})(1 - |S_{21}| + S_{21}\,\lambda)(1 - |S_{31}| + S_{31}\,q_m) + \\ & \Phi_{12}(1 - |S_{12}| + S_{12}\,y_{aux})(1 - |S_{22}| + S_{22}\,\lambda)(1 - |S_{32}| + S_{32}\,q_m). \end{aligned} \tag{29}$$

As the discontinuous change between laminar and turbulent regime occurs at $Re = 2300$, (7) can be rearranged to calculate the value of the transitional mass flow $q_{m,\text{crit}}$ at $Re = 2300$, leading to

$$q_{m,\text{crit}} = \frac{2300\,\pi\,d\,\mu}{4}. \tag{30}$$

Using the value $q_{m,\text{crit}}$, the value of the binary variable z can be used to determine the flow regime for a particular q_m, leading to the following equation

$$(2z - 1)(q_m - q_{m,\text{crit}}) \le 0. \tag{31}$$

This automatically enforces $z = 0$ for laminar flow ($q_m < q_{m,\text{crit}}$) and $z = 1$ for turbulent flow ($q_m > q_{m,\text{crit}}$). At the transition $q_m = q_{m,\text{crit}}$, both regimes could satisfy the constraint and z is chosen by the solver based on the previous timestep [48].

The overall pressure loss due to friction from both regimes can now be unified into a single equation

$$p_{in} - p_{out} + (z - 1)\,\Delta p_{lam} - z\,\Delta p_{turb} = 0. \tag{32}$$

Like the energy equations, the variables can be classified as inputs, state, binary and output variables:

- State $x = [\,] \rightarrow n = 0$
- Inputs $u = [q_m, p_{in}] \rightarrow m = 2$
- Outputs $y = [y_{aux}, \lambda, p_{out}] \rightarrow p = 3$
- Binary $z = [z] \rightarrow q = 1$

By substituting and inserting the variables into the equations, the following set of equations can now be used to describe the complete momentum equation.

$$\begin{aligned} & \Phi_{11}(1 - |S_{11}| + S_{11}\,y_1)(1 - |S_{21}| + S_{21}\,y_2)(1 - |S_{31}| + S_{31}\,u_1) + \\ & \Phi_{12}(1 - |S_{12}| + S_{12}\,y_1)(1 - |S_{22}| + S_{22}\,y_2)(1 - |S_{32}| + S_{32}\,u_1) = 0 \end{aligned} \tag{33}$$

$$u_1 - y_1 = 0 \tag{34}$$

$$u_2 - y_3 - u_1\,\frac{512\,\mu\,l}{\pi\,\rho\,d^4} + z_1\,u_1\,\frac{512\,\mu\,l}{\pi\,\rho\,d^4} - z_1\,u_1\,y_1\,y_2\,\frac{8\,l}{\pi^2\,d^5} = 0 \tag{35}$$

$$2\,z_1\,u_1 - 2\,z_1\,q_{m,\text{crit}} - u_1 + q_{m,\text{crit}} \le 0 \tag{36}$$

From these implicit equations, 10 distinct terms can be identified (e.g. u_1, y_1, $z_1\,u_1$, …) out of 6 distinct variables. Using the CPN format (20), the full tensor can now be decomposed into $S \in \mathbb{R}^{6\times 10}$, leading to

$$S = \begin{array}{c} y_1 \\ y_2 \\ u_1 \\ u_2 \\ y_3 \\ z_1 \end{array} \begin{bmatrix} S_{11} & S_{12} & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ S_{21} & S_{22} & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ S_{31} & S_{32} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{37}$$

and $\Phi \in \mathbb{R}^{4\times 10}$, with

$$\Phi^T = \begin{bmatrix} \Phi_{11} & 0 & 0 & 0 \\ \Phi_{12} & 0 & 0 & 0 \\ 0 & -1 & -\frac{512\,\mu\,l}{\pi\,\rho\,d^4} & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & \frac{512\,\mu\,l}{\pi\,\rho\,d^4} & 2 \\ 0 & 0 & -\frac{8\,l}{\pi^2\,\rho\,d^5} & 0 \\ 0 & 0 & 0 & -2\,q_{m,\text{crit}} \\ 0 & 0 & 0 & q_{m,\text{crit}} \end{bmatrix} \tag{38}$$

containing the coefficients from the momentum equations. The last row of Φ corresponds to the inequality constraint of the equation set. $S_{i,k}$, and $\Phi_{i,k}$ (for $i = 1, 2, 3$ and $k = 1, 2$) are determined from the multilinear approximation (13) and its subsequent CPN formulation (20) for λ.

3.3.4. Tensor representation of continuity equations

For the modeling of a pipe, the continuity equations can now be combined into one unified CPN formulation. On top of the energy and momentum equations, the mass Eq. (1) adds another equation. Notably, the variable q_m is shared across all three equation sets. In the CPN format, using $N = 2$ as an example, the following variables can be defined

$$
S = \begin{array}{c} \\ y_1 \\ y_2 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ y_3 \\ y_4 \\ \dot{x}_1 \\ \dot{x}_2 \\ x_1 \\ x_2 \\ z_1 \end{array}
\begin{array}{c}
\begin{array}{ccccccccccccccccccc} \dot{x}_1 & \dot{x}_2 & x_1 & x_2 & z_1 & y_1 & y_2 & y_3 & y_4 & u_1 & u_2 & u_3 & u_4 & \dot{x}_1 & \dot{x}_2 & x_1 & x_2 & z_1 & \end{array} \\
\left[\begin{array}{ccccccccccccccccccc}
S_{11} & S_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
S_{21} & S_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
S_{31} & S_{32} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 &
\end{array}\right]
\end{array}
\tag{39}
$$

Box I.

- State $x = [T_1, T_2] \rightarrow n = 2$
- Inputs $u = [q_m, T_0, T_a, p_{in}] \rightarrow m = 4$
- Outputs $y = [y_{aux}, \lambda, p_{out}, q_{m,out}] \rightarrow p = 4$
- Binary $z = [z] \rightarrow q = 1$

The combined structure matrix $S \in \mathbb{R}^{13\times19}$, with S in Box I, encompasses all the variable combinations from the set of equations, while the combined parameter matrix $\Phi \in \mathbb{R}^{7\times19}$

$$
\Phi^T = \begin{bmatrix}
\Phi_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\
\Phi_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & C_{th} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & C_{th} & 0 & 0 & 0 & 0 \\
0 & G_{th} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & G_{th} & 0 & 0 & 0 & 0 \\
0 & -c_f & 0 & 0 & 0 & 0 & 0 \\
0 & c_f & -c_f & 0 & 0 & 0 & 0 \\
0 & 0 & c_f & 0 & 0 & 0 & 0 \\
0 & -G_{th} & -G_{th} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & -\frac{512\,\mu\,l}{\pi\,\rho\,d^4} & -1 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{512\,\mu\,l}{\pi\,\rho\,d^4} & 0 & 2 \\
0 & 0 & 0 & 0 & -\frac{8\,l}{\pi^2\,\rho\,d^5} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & q_{m,\mathrm{crit}} \\
0 & 0 & 0 & 0 & 0 & 0 & -2\,q_{m,\mathrm{crit}} \\
0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\tag{40}
$$

provides all coefficients of the multilinear equations. Using the iMTI framework, both S and Φ can be used to simultaneously determine all states, inputs, outputs and binary variables. This unified representation enables modeling of transient thermal-hydraulic dynamics while explicitly handling laminar-to-turbulent flow regime transitions.

3.4. Implementation

The CPN formulation of the system is implemented using the MTI-Toolbox [49], a MATLAB-based framework for multilinear time-invariant (MTI) systems. In the current version (2.1), this toolbox makes use of the sparsity pattern of the equation system by reordering it to a block lower triangular matrix and subsequently splitting it to distinct sub-sets and sub-problems [48,50]. The system is then solved numerically using MATLAB's ode15i implicit ODE solver [51,52]. In Fig. 4, a simplified flowchart of the framework implementation is shown.

4. Validation

This section validates the proposed iMTI model against established benchmarks. Temperature profile validation appears in Section 4.1, followed by pressure profile validation in Section 4.2.

4.1. Validation of the temperature profile

To assess the validity of the proposed methodology, a comparison is made between the temperature profile of a single DN80 pipe of $l = 100\,\mathrm{m}$ with technical parameters taken from [28]. As references, the L4 pipe model from the *ClaRa* [12], the *DynamicPipe* model from Modelica Standard Language [53], and the *PlugFlow* pipe model from *AixLib* are used [15]. The simulations in Modelica are done in OpenModelica v1.24.3 with the DASSL solver [54,55]. The comparison is made for different discretization steps of $N = 5, 10, 20, 50, 100$. For the simulation, the temperature of the different pipe segments T_i are initialized at 20 °C, with a constant ambient temperature $T_a = 9\,°\mathrm{C}$. The inflow temperature T_0 starts at 20 °C and is increased stepwise to 40 °C at $t = 500\,\mathrm{s}$. The results of this comparison can be seen in Fig. 5.

Visually, the results of the iMTI, *ClaRa* L4, and *DynamicPipe* models are comparable for all discretization steps. The significant difference to the *PlugFlow* temperature profile occurs due to the different modeling approach, as *PlugFlow* model makes use of residence time instead of discretizing the length of the pipe into multiple segments. The results shown in this work align with results shown in literature [3,34].

In addition to the single pipe test, a second validation is made using a section of the DHN in Pongau shown in Fig. 6 [28]. To assess the model performance, visual comparison as well as numerical error quantification (Appendix B) are made [56]. The datasets required for this comparison are freely available as part of *AixLib* library [15]. For the calculation, iMTI is compared against *DynamicPipe*, *PlugFlow*, and measurement values, with discretization of $1\,\mathrm{segment/m}$ to better match *PlugFlow* behavior.

As discussed in [28], the first 6 h of the simulation are highly likely affected by model initialization and are therefore omitted. Temperature profiles for substations 2 and 3 are shown in Fig. 7. At both substations, iMTI and *DynamicPipe* show high alignment with *PlugFlow*. Deviations from measured values reflect measurement uncertainty discussed in [28].

For substation 4 (Fig. 8), substantial deviations appear between models and measured values in zero-flow regions (highlighted in orange). iMTI and *DynamicPipe* show identical behavior in these periods, deviating from *PlugFlow* and measurements. In all other operating conditions, the models show acceptable results. These zero-flow deviations reflect known modeling limitations discussed in detail in [28].

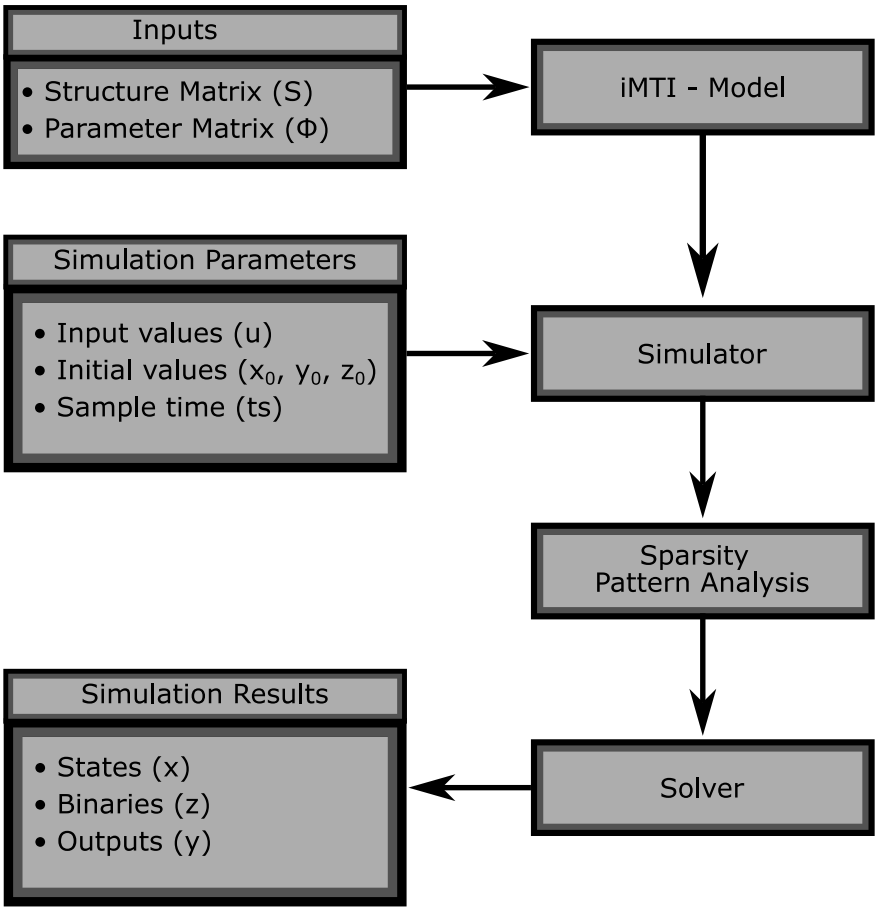


Fig. 4. Simplified representation of the implementation workflow.

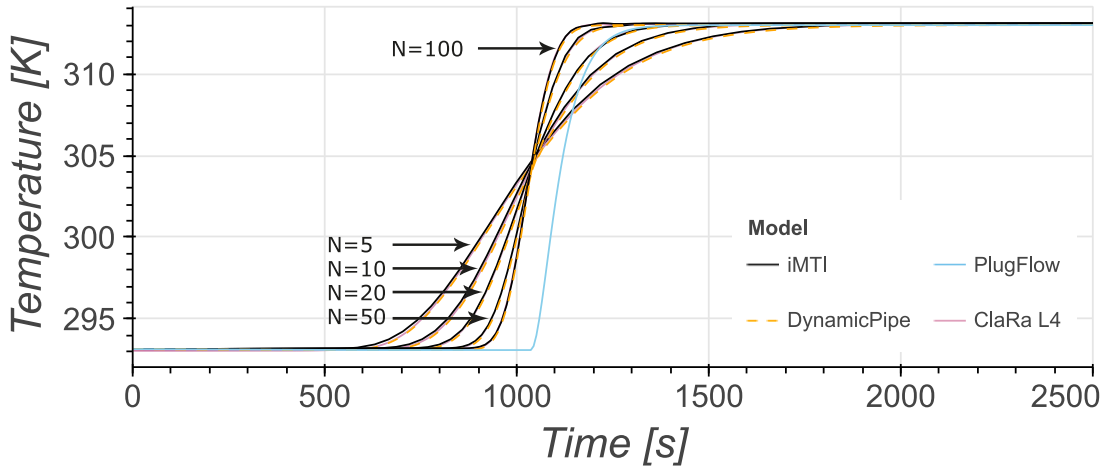


Fig. 5. Temperature profiles for DN80 pipe following 20 K step input at $t = 500$ s, comparing iMTI, *ClaRa* L4, *DynamicPipe*, and *AixLib PlugFlow* models.

The numerical error indicators MAE, RMSE, and NRMSE for substations 2, 3, and 4 are shown in Table 1. Due to the known limitation surrounding zero-flow regions [28], the analysis for substation 4 is done once on values outside zero-flow regions and again for the complete range. From Table 1, it is shown that the performance of iMTI and *DynamicPipe* is similar and tends to be slightly worse than *PlugFlow*. Even within zero-flow regions, the performance of iMTI is shown to be adequate.

4.2. Validation of pressure profile

To check the validity of the proposed pressure model, a comparison is made using the *DynamicPipe* model from Modelica. In the setup, steady state momentum balance with a detailed pipe flow model is used to quantify the pressure losses due to friction. For the simulation, a DN80 pipe of $l = 20$ m with $\varepsilon = 2.5 \cdot 10^{-5}$ m is used. The fluid is assumed to be incompressible water with constant properties of $\rho = 995.6$ kg/m^3 and $\mu = 466 \cdot 10^{-6}$ Pa s. The simulation runs for 3600 s with the mass flow q_m ramping from 0.01 kg/s to 3 kg/s.

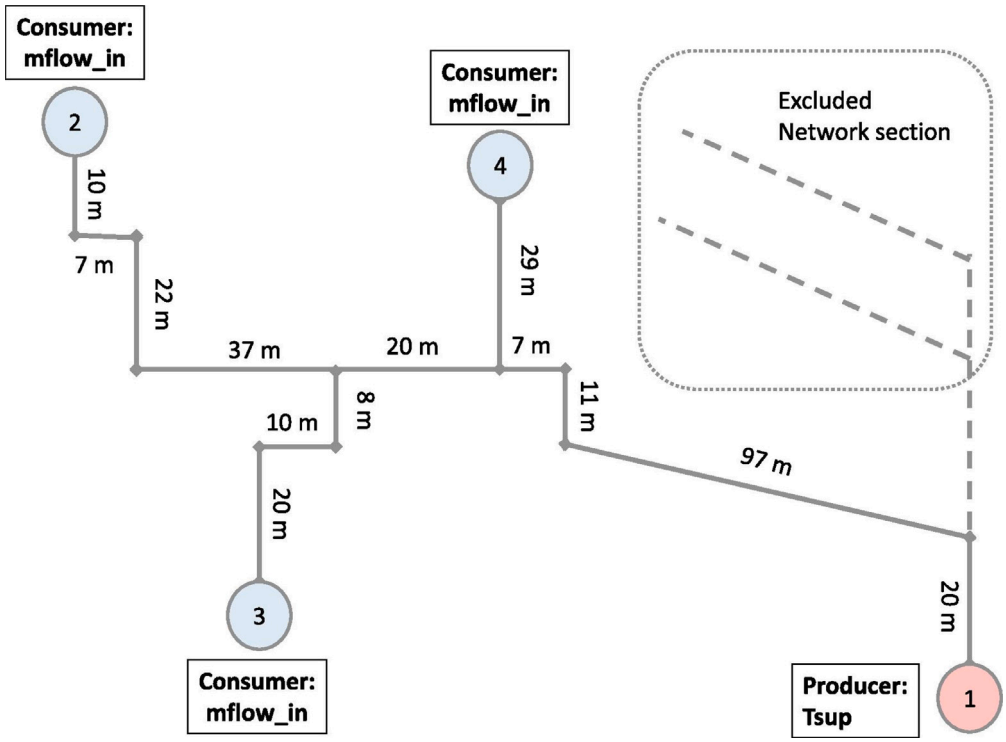


Fig. 6. Schematic representation of the district heating network in Pongau from [28].

Table 1
Numerical error quantification for different substations of DHN Pongau.

	iMTI	DynamicPipe	PlugFlow
Substation 2			
MAE [K]	1.38	1.43	1.00
RMSE [K]	1.70	1.77	1.30
NRMSE [%]	19.37	20.16	14.80
Substation 3			
MAE [K]	1.23	1.29	0.97
RMSE [K]	1.66	1.72	1.38
NRMSE [%]	11.85	12.28	9.89
Substation 4 - Outside Zero-Flow Region			
MAE [K]	2.72	2.73	2.58
RMSE [K]	8.54	8.56	8.36
NRMSE [%]	10.05	10.07	9.84
Substation 4 - Complete range			
MAE [K]	5.60	5.53	3.15
RMSE [K]	11.01	10.85	8.77
NRMSE [%]	12.96	12.77	10.32

The results in Fig. 9 show strong overall alignment between the iMTI and *DynamicPipe*. However, detailed analysis reveals regime-dependent behavior. To better understand the sources of agreement and deviation, the results are examined separately for the laminar, transition, and turbulent regimes.

In the laminar regime (Fig. 10), the values match closely up to the tail end of the laminar region. This occurs due to the friction model implemented in Modelica's *DynamicPipe*, where the laminar–turbulent transition region begins below $Re_{crit} = 2300$ threshold (typically around $Re \approx 2000$) [53].

The main difference occurs in the transition region ($2025 \leq Re \leq 4000$), as shown in Fig. 11. In *DynamicPipe*, this region is modeled using Cubic Hermite interpolation, which smoothly blends laminar and turbulent friction losses. This work, in contrast, employs a sharp binary switch at $Re = 2300$. At $Re = 2300$, *DynamicPipe* blending assigns approximately 94.7% weight to the laminar component, whereas the discontinuous transition creates the observed overshoot.

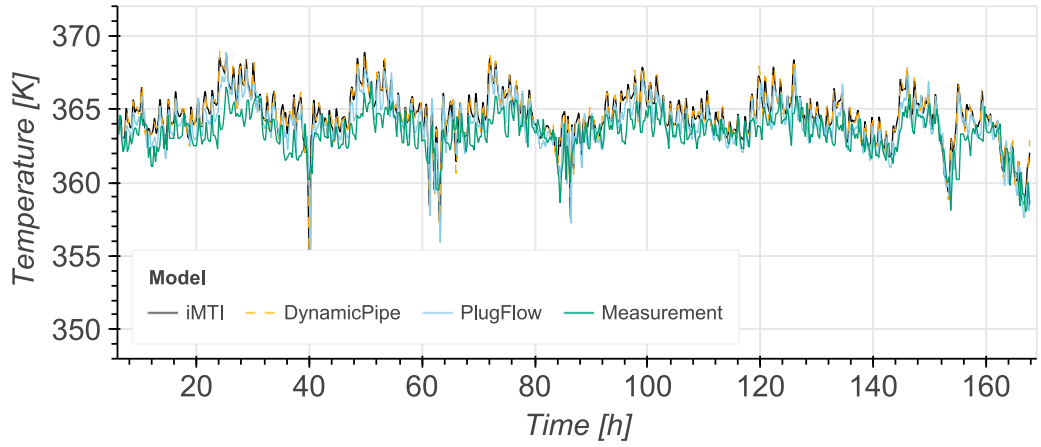
This leads to a significant overestimation of friction losses in the transition zone. However, this is operationally benign, as sizing of district heating pipes is typically done for turbulent flow regime [16,28,57,58] and DHNs typically operate at $Re > 10^4$ [59–62]. Subsequently, in the turbulent regime (Fig. 12), both models show matching results.

To quantify model performance, RMSE (Pa), NRMSE (%), and R^2 are used (see Appendix B). For the complete operational range, the iMTI framework achieves RMSE = 0.53 Pa, NRMSE = 0.070% and $R^2 = 0.999994$. The turbulent regime shows RMSE = 0.53 Pa, NRMSE = 0.072% and $R^2 = 0.999994$, while the laminar regime achieves RMSE = 0.0041 Pa, NRMSE = 0.946% and $R^2 = 0.9989$ (max absolute error 0.019 Pa). The highest deviation is within the transition zone ($2025 \leq Re \leq 4000$), with RMSE = 0.24 Pa, NRMSE = 12.340% and $R^2 = 0.8469$, reflecting the different modeling approach.

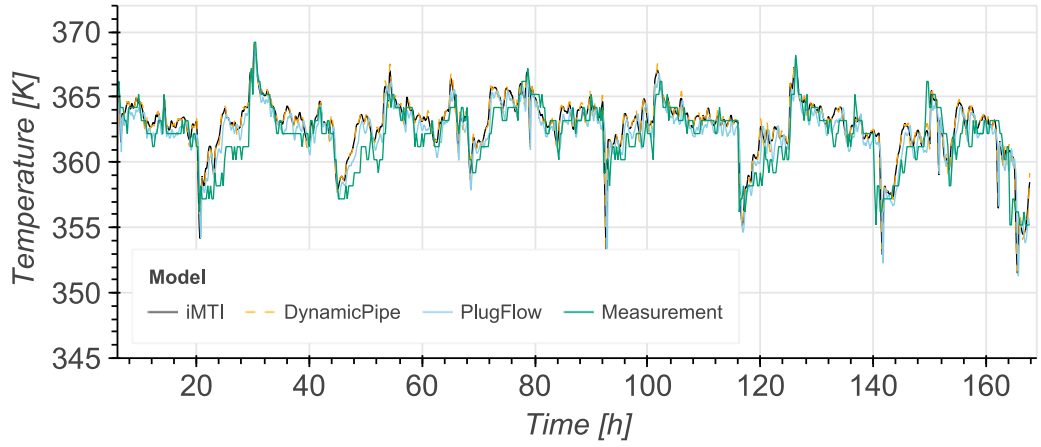
5. Runtime performance

To compare runtime performance of Modelica models and the proposed iMTI model in MATLAB, the fundamental difference has to be highlighted. Modelica in itself is a modeling language rather than a conventional programming language [53]. First, Modelica models must be translated into a sequence of instructions (translation) and then compiled (compilation) into an executable file, where the simulation finally occurs. MATLAB, in contrast, is an interpreted language, where code is executed line-by-line during runtime [51].

Performance is measured using the built-in Windows 11 tool *Perfmon* (Performance Monitor) [63], which logs indicators for individual processes (e.g. translation, compilation, and simulation). Three performance indicators are measured. Private Bytes represent total memory allocated by a process, while Working Set represents the portion actively used at any given moment. In practice, Private Bytes typically exceed Working Set as unused portions may be stored temporarily on disk instead of the physical memory. Alongside the memories, CPU usage is also measured for the duration of the task [63]. As CPU usage is monitored on each logical core, values over 100% may occur, if the process utilizes multiple cores simultaneously. These metrics are then recorded across the test cases described below.



(a) Substation 2



(b) Substation 3

Fig. 7. Temperature profile of Substation 2 and Substation 3 for iMTI, *DynamicPipe*, *PlugFlow* and measurement values.

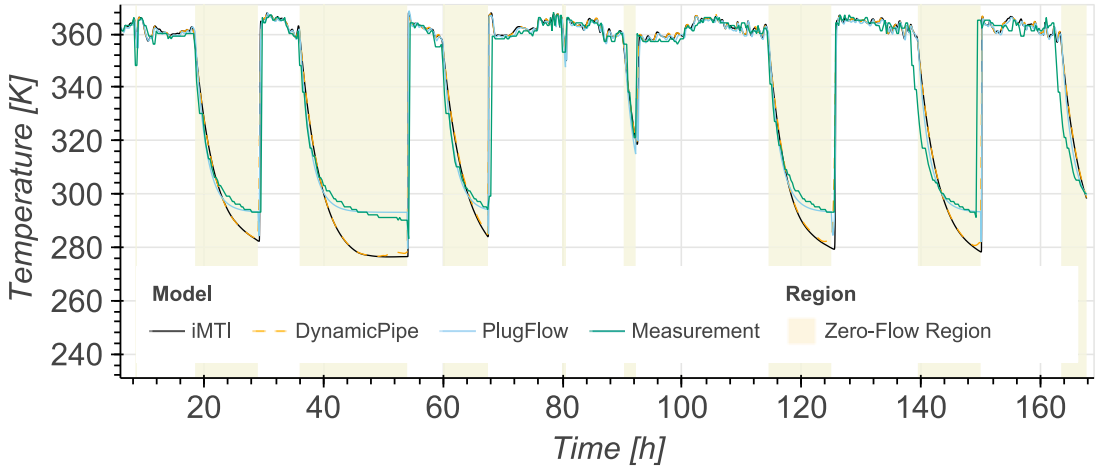


Fig. 8. Temperature profile of Substation 4 for iMTI, *DynamicPipe*, *PlugFlow* and measurement values.

To test the performance, the network shown in Fig. 6 is modeled using iMTI, *DynamicPipe*, and *PlugFlow*. To reduce variability, the task is repeated 5 times. In Table 2 the results are presented as mean and standard deviation across the 5 runs. For the CPU performance, Modelica models exhibit significant spikes during compilation, while MATLAB maintains a more consistent average processor load. The spikes from Modelica models coincide with multiple concurrent compiler instances. From the runtime perspective, iMTI is significantly faster than *DynamicPipe*, while *PlugFlow* is expectedly the fastest due to its reduced equation count. From Table 2 it is further shown that iMTI demonstrates the most efficient memory management. Notably, *DynamicPipe* and *PlugFlow* showed larger gaps between their allocated

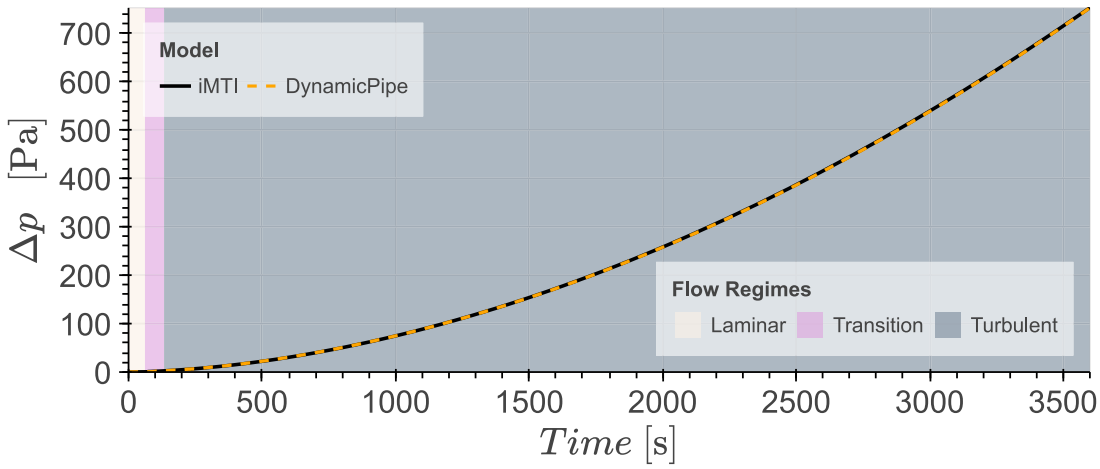


Fig. 9. Comparison of pressure loss due to friction for iMTI and *DynamicPipe*.

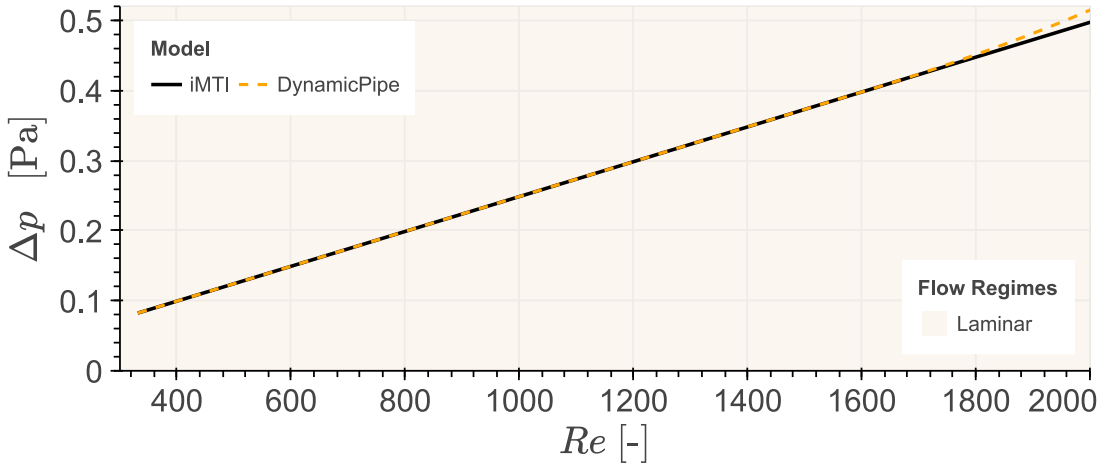


Fig. 10. Comparison of pressure loss due to friction in the laminar flow regime for iMTI and *DynamicPipe*.

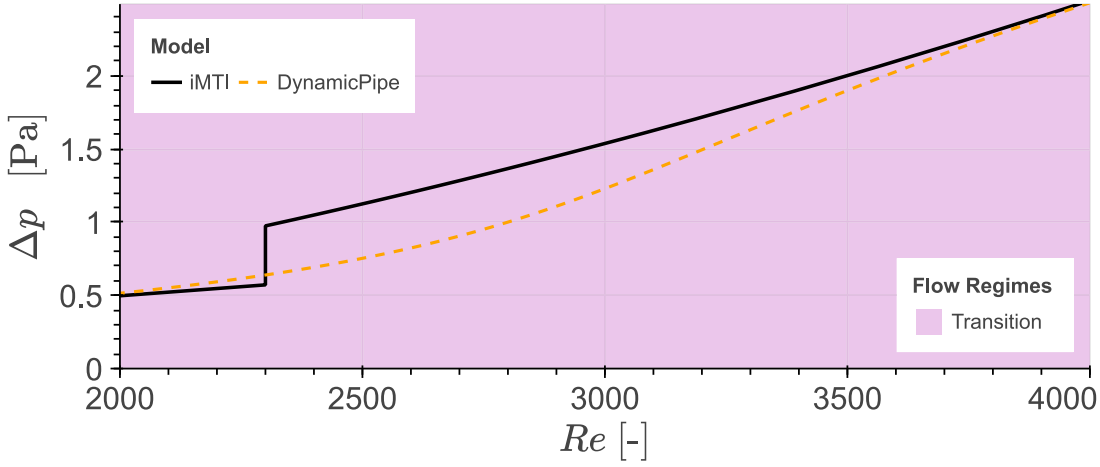


Fig. 11. Comparison of pressure loss due to friction in the transition flow regime for iMTI and *DynamicPipe*.

(Private Bytes) and active (Working Set) memory, indicating less efficient memory utilization. iMTI maintained the lowest variability across all metrics, hinting at stable and predictable performance.

Despite the promising results, the direct comparison between iMTI, *DynamicPipe*, and *PlugFlow* does not consider the inherent overhead of the typical Modelica implementation, as Modelica models generally make use of the medium model for the computation of the fluid properties. Additionally, Modelica models also allow the modeling of multiple substances or phases within a single flow, which is outside the scope of this work [53]. As these overheads inherently increase the model complexity and subsequently computational load, the comparison shown in Table 2 cannot be directly used to indicate whether the memory saving of iMTI occurs due to the modeling overhead or due to the tensor benefits.

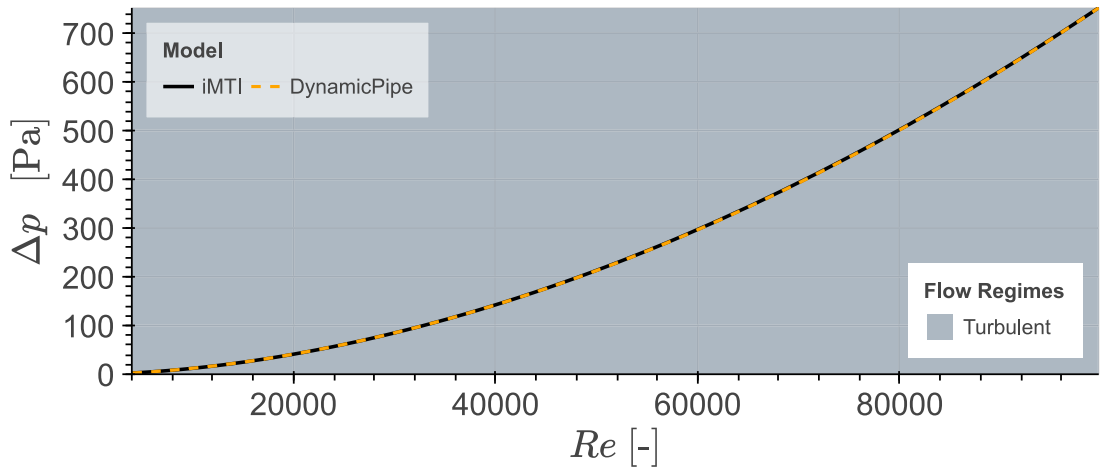


Fig. 12. Comparison of pressure loss due to friction in the turbulent regime for iMTI and *DynamicPipe*.

Table 2
Performance comparison of the different frameworks.

	iMTI	DynamicPipe	PlugFlow
Runtime [s]	80.40 ± 2.61	143.99 ± 0.70	53.80 ± 0.84
Max			
Working Set [MB]	1020.42 ± 1.25	3349.94 ± 112.35	2765.95 ± 47.10
Private Bytes [MB]	877.77 ± 15.53	7759.91 ± 7.39	7191.97 ± 6.69
CPU Usage [%]	369.54 ± 20.56	841.52 ± 45.72	645.63 ± 24.15
Mean			
Working Set [MB]	920.46 ± 6.99	1800.06 ± 17.52	1386.19 ± 14.85
Private Bytes [MB]	805.27 ± 6.01	5058.17 ± 25.68	6034.15 ± 61.00
CPU Usage [%]	207.44 ± 3.65	147.93 ± 2.01	121.71 ± 2.10

Table 3
Performance comparison against minimal model.

	iMTI	DirectEq	Simscape
Runtime [s]	80.40 ± 2.61	17.80 ± 0.84	40.00 ± 6.74
Max			
Working Set [MB]	1020.42 ± 1.25	1629.80 ± 32.14	1852.57 ± 3.07
Private Bytes [MB]	877.77 ± 15.53	6056.98 ± 1.12	1477.08 ± 3.22
CPU Usage [%]	369.54 ± 20.56	667.07 ± 101.45	234.92 ± 15.78
Mean			
Working Set [MB]	920.46 ± 6.99	591.63 ± 9.93	1476.88 ± 62.70
Private Bytes [MB]	805.27 ± 6.01	3657.47 ± 127.28	1082.34 ± 17.16
CPU Usage [%]	207.44 ± 3.65	178.24 ± 11.64	104.09 ± 14.63

For this purpose, a minimal model of the DHN is once again generated using solely the set of equations. A direct equation implementation (*DirectEq*) is created in OpenModelica to exactly match iMTI's implementation. This approach isolates framework efficiency from modeling overhead, as both *DirectEq* and iMTI solve identical equations without component library abstractions. Nonetheless, this approach still does not consider the inherent difference between compiled (Modelica) and interpreted (MATLAB) language. To account for the language difference, a simplified Simscape model of the DHN is created. Simscape is a physical modeling language within MATLAB, which is mainly used for equation-based physical modeling [64]. As the model of the DHN (Fig. 6) neglects the pressure calculation, standard Simscape libraries are not directly usable [65]. To mitigate this, a causal model with manually defined inputs and outputs is generated. However, explicit causality in Simscape may reduce computational burden compared to iMTI, introducing a potential bias in the comparison.

From the results in Table 3, it is clear that the simplified models are significantly faster than the iMTI framework. The minimal Modelica implementation minimizes the standard library translation overhead by directly expressing the DHN equations in Modelica code, reducing the need for object-to-equation conversion. This explains the faster runtime compared to iMTI, which solves the same equations through matrix operations. The iMTI infrastructure, however, shows more efficient memory allocation. Interestingly, the memory actively used by *DirectEq* is significantly lower than its allocated memory. When compared to Simscape, a conventional component-based implementation within the same programming language (MATLAB), the tensor-based framework demonstrates substantial memory efficiency gains (877 MB vs 1477 MB), which reflects inherent tensor benefits. It is however important to highlight that the implementations of the simplified models in Modelica and Simscape are solely for benchmarking and do not reflect current practical standards in DHN modeling. Nonetheless, the chosen approaches cannot produce exact comparisons but represent the practical effort toward fair model evaluation. The relevant code snippets for the benchmark are available in the Supplementary Materials.

6. Conclusion

This work presents the first application of tensor-based implicit multilinear time-invariant (iMTI) modeling to DHNs, addressing a critical scalability bottleneck in simulating large-scale thermal-hydraulic systems. The proposed framework successfully captures the governing equations of DHN pipes including nonlinear friction factors and quadratic pressure-loss relationships through compact tensor representations while maintaining computational tractability.

Benchmarking against established Modelica-based implementations demonstrates substantial efficiency gains: up to 8.8× memory reduction compared to full-featured component libraries (*DynamicPipe*), and 1.7× reduction compared to the simplified Simscape framework, showing that these advantages likely stem from inherent tensor benefits rather than merely systematic overhead. The simplification around the transition zones ($2025 \leq Re \leq 4000$), however, introduces approximation errors with observed NRMSE up to 12.340%. This is operationally benign because: (a) DHNs are sized for the turbulent regime and (b) oversizing provides inherent safety margin.

Future work should extend the framework to integrate other components (e.g., valves, pumps, and heat exchangers), modeling concepts (e.g. *PlugFlow*, Cubic Hermite interpolation), meshed networks, and bidirectional flows. Additionally, applications on large-scale networks should further assess real-world applicability. With these improvements, the compact tensor representations and potential scaling of the iMTI framework may offer a viable path toward practical deployment in simulation and control contexts, particularly towards VF-VT operation, which may support the integration of distributed renewable sources essential for future district heating systems.

CRediT authorship contribution statement

Nicholas Tedjosantoso: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Arne Speerforck:** Writing – review & editing, Supervision, Software, Methodology, Funding acquisition. **Torben Warnecke:** Writing – review & editing, Software, Methodology. **Hans Schäfers:** Writing – review & editing, Supervision, Funding acquisition. **Gerwald Lichtenberg:** Writing – review & editing, Supervision, Software, Methodology, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Perplexity AI in order to suggest grammatical and wording improvements to the manuscript text. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been supported by the Federal Ministry of Research, Technology and Space in the framework of the research project "go-2-prof:in - Gewinnung und Entwicklung von professoralem Personal an der HAW Hamburg" (FKZ 03FHP223A).

Appendix A. Pre-computed lookup table for friction factor λ

See Table 4.

Appendix B. Model validation metrics

Mean Absolute Error (MAE).

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \hat{y}_i)| \tag{B.1}$$

Root Mean Square Error (RMSE).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{B.2}$$

Normalized Root Mean Square Error (NRMSE).

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\max(y) - \min(y)} \times 100\% \tag{B.3}$$

Table 4
Pre-computed lookup table for a 263 mm steel pipe with $\varepsilon = 0.007$ mm [45].

Fitting parameter	Value
a_1	$7.09 \cdot 10^{-4}$
a_2	$1.98 \cdot 10^{-3}$
a_3	$1.54 \cdot 10^{-2}$
a_4	$4.31 \cdot 10^{-2}$
a_5	$4.95 \cdot 10^{-3}$
a_6	$1.45 \cdot 10^{-2}$
a_7	$1.06 \cdot 10^{-1}$
a_8	$3.13 \cdot 10^{-1}$

Coefficient of determination (R^2).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{B.4}$$

Appendix C. Supplementary materials

Supplementary data to this article are available as `Supplementary_Material.zip` at https://doi.org/10.5281/zenodo.19454131.

Data availability

Data will be made available on request. The relevant datasets and code for the runtime performance comparison (Section 5) are available in the Supplementary Material.

References

[1] Kuntuarova S, Lickleder T, Huynh T, Zinsmeister D, Hamacher T, Perić V. Design and simulation of district heating networks: A review of modeling approaches and tools. Energy 2024;305:132189.
[2] Valentini LV, Forndran S, Ochs F, Thür A, Streicher W. District heating modeling tools: A review based on design and simulation of renewable district heating networks. Energy Convers Manag: X 2026;29:101414.
[3] Westphal J, Brunnemann J, Speerforck A. Enabling the dynamic simulation of an unaggregated, meshed district heating network with several thousand substations. Energy 2025;322:135434.
[4] Li C, Prasad S, Bai Y, Turkeri C, Wang J. A quasi-dynamic model and comprehensive simulation study of district heating networks considering temperature delay. Energy 2025;318:134855.
[5] Dibos S, Pesch T, Benigni A. HeatNetSim: An open-source simulation tool for heating and cooling networks suitable for future energy systems. Energy 2024;312:133588.
[6] Stock J, Arjuna F, Xhonneux A, Müller D. Modelling of waste heat integration into an existing district heating network operating at different supply temperatures. Smart Energy 2023;10:100104.
[7] Angelidis O, Zinsmeister D, Ioannou A, Friedrich D, Thomson A, Falcone G. 5Th generation district heating and cooling modelica models for prosumer interaction analysis. In: Proceedings of the 15th international modelica conference 2023, aachen, October 9-11. Modelica 2023, Linköping University Electronic Press; 2023, http://dx.doi.org/10.3384/ecp204607.
[8] Kollmar M, Bürger A, Bohlayer M, Altmann-Dieses A, Braun M, Diehl M. A detailed simulation model for fifth generation district heating and cooling networks with seasonal latent storage evaluated on field data. Appl Energy 2025;401:126757.
[9] Vieth J, Westphal J, Speerforck A. District heating network topology optimization and optimal co-planning using dynamic simulations. Adv Appl Energy 2025;19:100233.
[10] Friedrich P, Huynh T, Niessen S. Optimizing district heating operations: Network modeling and its implications on system efficiency and operation. Smart Energy 2025;18:100175.
[11] Nigro L, La Bella A, Casella F, Scattolini R. Control-oriented modeling, simulation, and predictive control of district heating networks. IEEE Trans Autom Sci Eng 2025;22:7064–79.
[12] Vojacek A, Brunnemann J, Hanke T, Marx-Schubach T, Eiden J. Status of the clara library: Detailed transient simulation of complex energy systems. In: Proceedings of the 15th international modelica conference 2023, aachen, October 9-11. Modelica 2023, Linköping University Electronic Press; 2023, p. 617–26. http://dx.doi.org/10.3384/ecp204617.
[13] Gillner M, Westphal J, Wiegel B, Steffen T, Urbansky J, Hagemeier A, Ruppert S, Heyer A, Benthin J, Hanke T, Brunnemann J, Becker C, Speerforck A. Status of the TransiEnt library: Transient simulation of complex integrated energy systems. In: Proceedings of the 16th international modelica&fMI conference, September 8 - 10, 2025, lucerne university of applied sciences and arts, vol. 218, Linköping University Electronic Press; 2025, http://dx.doi.org/10.3384/ecp218989.
[14] Wetter M, van Treeck C, Helsen L, Maccarini A, Saelens D, Robinson D, Schweiger G. IBPSA project 1: BIM/GIS and modelica framework for building and community energy system design and operation - ongoing developments, lessons learned and challenges. IOP Conf Ser: Earth Environ Sci 2019;323(1):012114.
[15] Maier L, Jansen D, Wüllhorst F, Kremer M, Kümpel A, Blacha T, Müller D. AixLib: an open-source modelica library for compound building energy systems from component to district level with automated quality management. J Build Perform Simul 2023;17(2):196–219.
[16] Röder J, Meyer B, Krien U, Zimmermann J, Stührmann T, Zondervan E. Optimal design of district heating networks with distributed thermal energy storages - method and case study. Int J Sustain Energy Plan Manag 2021;31.

[17] Vorspel L, Bücker J. District-heating-grid simulation in python: DiGriPy. Computation 2021;9(6):72.
[18] Lohmeier D, Cronbach D, Drauz SR, Braun M, Kneiske TM. Pandapipes: An open-source piping grid calculation package for multi-energy grid simulations. Sustainability 2020;12(23):9899.
[19] Boghetti R, Kämpf JH. Verification of an open-source Python library for the simulation of district heating networks with complex topologies. Energy 2024;290:130169.
[20] Dénarié A, Aprile M, Motta M. The importance of an accurate numerical model for the simulations of new generation district heating systems. J Phys: Conf Ser 2024;2893(1):012030.
[21] Kernstock P, Lorenzen P, Tilmann P, Schäfers H. Jarvis - Ein verteilter Simulator für hoch aufgelöste Systeme des Energiesektors. In: Proceedings of the 9th regenerative energy technology conference in nordhausen, nordhausen, February 12-13, 2026, nordhausen university of applied sciences. 2026.
[22] Ruan Y, Chen J, Wu Z, Xu T, Meng H, Li Z, Yao Y, Wang C, Liu W, Xia Y. A transfer function method for thermal transient modeling in district heating networks: Optimizing accuracy and computational efficiency. Energy 2026;140834.
[23] Xu K, Ai D, Sun C, Qi Y, Wang J, Yang F, Ren H. Green's function approach for simulating district heating networks. Energies 2025;18(10):2627.
[24] Boghetti R, Odobez J-M, Kämpf JH. Graph neural network-based surrogate modeling for fast and scalable simulations of meshed district heating networks. Energy AI 2026;24:100721.
[25] Boussaid T, Rousset F, Sciturici V-M, Clausse M. Enabling fast prediction of district heating networks transients via a physics-guided graph neural network. Appl Energy 2024;370:123634.
[26] Lu Y, Yang K. An accurate and efficient quasi-dynamic simulation method of electricity-heat multi-energy systems. Front Energy Res 2023;11.
[27] Lichtenberg G, Pangalos G, nez CCY, Luxa A, Jöres N, Schnelle L, Kaufmann C. Implicit multilinear modeling. At - Autom 2022;70(1):13–30, URL: https://doi.org/10.1515/auto-2021-0133.
[28] van der Heijde B, Fuchs M, Ribas Tugores C, Schweiger G, Sartor K, Basciotti D, Müller D, Nytsch-Geusen C, Wetter M, Helsen L. Dynamic equation-based thermo-hydraulic pipe model for district heating and cooling systems. Energy Convers Manage 2017;151:158–69.
[29] Agosta G, Casella F, Cattaneo D, Cherubin S, Leva A, Scuttari M, Terraneo F. MARCO: An experimental high-performance compiler for large-scale modelica models. In: Proceedings of the 15th international modelica conference 2023, aachen, October 9-11. Modelica 2023, vol. 204, Linköping University Electronic Press; 2023, p. 13–22. http://dx.doi.org/10.3384/ecp20413.
[30] Pangalos G. Model-based controller design methods for heating systems [Ph.D. thesis], Technische Universität Hamburg; 2016, http://dx.doi.org/10.15480/882.1324.
[31] Warnecke T, Lichtenberg G. Hybrid implicit multilinear modeling of complex HVAC systems. In: Wagner G, Werner F, De Rango F, editors. Simulation and modeling methodologies, technologies and applications. Cham: Springer Nature Switzerland; 2024, p. 79–104.
[32] Ballard G, Kolda TG. Tensor decompositions for data science. Cambridge University Press; 2025.
[33] Jiang M, Speetjens M, Rindt C, Smeulders D. A data-based compact pipe model for district-heating-and-cooling networks with variable flow conditions. Energy Build 2025;330:115321.
[34] Liu X, Strunz K. Lumped and distributed-parameter pipe model framework for thermal transients: State-space and transfer function theory and application to multi-energy systems. IET Renew Power Gener 2022;17(1):199–215.
[35] Benthin J, Hagemeier A, Heyer A, Huismann P, Krassowski J, Settgast C, Wortmann BD, Görner K. Integrierte Betrachtung von Strom-, Gas- und Wärmesystemen zur modellbasierten Optimierung des Energieausgleichs- und Transportbedarfs innerhalb der deutschen Energienetze : Gemeinsamer Abschlussbericht des Forschungsvorhabens : Laufzeit: 01.07.2016-30.06.2019. 2020, http://dx.doi.org/10.2314/KXP:1743191693.
[36] Zimmer D. Robust object-oriented formulation of directed thermofluid stream networks. Math Comput Model Dyn Syst 2020;26(3):204–33.
[37] Nussbaumer T, Thalmann S, Jenni A, Ködel J. Handbook on planning of district heating networks. Bern: Swiss Federal Office of Energy; 2020,
[38] White FM. Fluid mechanics, In: McGraw-hill series in mechanical engineering, 7th ed. New York [u.a.]: McGraw-Hill; 2011, Auf der DVD-ROM-Beil. (Student Resource DVD): Engineering Equation Solver (EES).
[39] Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Rev 2009;51(3):455–500.
[40] Lang S. Algebra. Springer New York; 2002, http://dx.doi.org/10.1007/978-1-4613-0041-0.
[41] Sloughter DC. The calculus of functions of several variables. Furman University; 2011.
[42] Batselier K. Low-rank tensor decompositions for nonlinear system identification: A tutorial with examples. IEEE Control Syst 2022;42(1):54–74.
[43] Kramer B, Willcox KE. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. AIAA J 2019;57(6):2297–307.
[44] Warnecke T, Lichtenberg G. Implicit multilinear modeling of air conditioning systems. In: Proceedings of the 13th international conference on simulation and modeling methodologies, technologies and applications. SCITEPRESS - Science and Technology Publications; 2023, p. 440–7. http://dx.doi.org/10.5220/0012138200003546.
[45] isoplus. Product catalogue. 2024, URL: https://www.isoplus.group/fileadmin/products/1_Product_catalogue_EN_1124.pdf.
[46] Jöres N, Kaufmann C, Schnelle L, nez CY, Pangalos G, Lichtenberg G. Reduced CP representation of multilinear models. In: Proceedings of the 12th international conference on simulation and modeling methodologies, technologies and applications - volume 1: SIMULTECH. SciTePress, INSTICC; 2022, p. 252–9. http://dx.doi.org/10.5220/0011273100003274.
[47] Rumohr S. Dimensionierung und Betrieb von Erdwärmesonden – Erläuterungen vor dem Hintergrund missverständlicher Darstellungen. In: ARGE GMIT, editor. Geowissenschaftliche Mitteilungen - GMIT 94. Bonn: ARGE GMIT; 2023, p. 7–16. http://dx.doi.org/10.23689/FIDGEO-5845.
[48] Warnecke T, Lichtenberg G. Approach for the mode switching problem in piecewise smooth implicit multilinear IVPs. In: Proceedings of the 15th international conference on simulation and modeling methodologies, technologies and applications. SCITEPRESS - Science and Technology Publications; 2025, p. 349–56. http://dx.doi.org/10.5220/0013578200003970.
[49] Gerwald Lichtenberg and Team (HAW / Fraunhofer / DESY). MTI-toolbox 2.0. 2025, Software toolkit. URL: http://mti.systems.
[50] Warnecke T, Lichtenberg G. Hybrid implicit multilinear simulation using difference algebraic equations reordering by sparsity patterns*. In: 2024 10th international conference on control, decision and information technologies. 2024, p. 2578–83. http://dx.doi.org/10.1109/CoDIT62066.2024.10708218.
[51] MathWorks. MATLAB. 2025, Version R2025a. URL: https://www.mathworks.com/products/matlab.html.
[52] MathWorks. Ode15i: Solve fully implicit differential equations. 2025, MATLAB Documentation. URL: https://www.mathworks.com/help/matlab/ref/ode15i.html.
[53] Modelica Association. Modelica standard library 4.0.0. 2023, Open-source library. URL: https://github.com/modelica/ModelicaStandardLibrary.
[54] Open Source Modelica Consortium. OpenModelica v1.24.3. 2024, Open-source Modelica-based modeling and simulation environment. URL: https://openmodelica.org/.
[55] Petzold LR. Description of DASSL: a differential/algebraic system solver. Livermore, CA (USA: Sandia National Labs; 1982, URL: https://www.osti.gov/biblio/5882821.
[56] Kuś J, Mika Ł, Żurawski M. Validation strategies for district heating network models. Energies 2025;18(18):5012.
[57] Nussbaumer T, Thalmann S. Influence of system design on heat distribution costs in district heating. Energy 2016;101:496–505.
[58] nPro Energy. Pipe dimensioning for heat networks in npro. 2024, URL: https://www.npro.energy/main/en/help/pipe-dimensioning. (Accessed 25 November 2025).
[59] Meibodi SS, Rees S, Loveridge F. Modeling district heating pipelines using a hybrid dynamic thermal network approach. Energy 2024;290:130107.
[60] Duquette J, Rowe A, Wild P. Thermal performance of a steady state physical pipe model for simulating district heating grids with variable flow. Appl Energy 2016;178:383–93.
[61] Li H, Svendsen S. Energy and exergy analysis of low temperature district heating network. Energy 2012;45(1):237–46.
[62] EMD International. Method of hydraulic transmission calculation in energyPRO. 2025, URL: https://energyprodocs.emd.dk/Methods/07Hydraulictransmission/. (Accessed 23 March 2026).
[63] Yosifovich P, Russinovich ME, Solomon DA, Ionescu A. Windows internals, part 1: System architecture, processes, threads, memory management, and more (7th Edition). 7th ed. USA: Microsoft Press; 2017.
[64] MathWorks. What is the simscape language?. 2024, MATLAB & Simulink Documentation. URL: https://www.mathworks.com/help/simscape/lang/what-is-the-simscape-language.html.
[65] MathWorks. Modeling thermal liquid systems. 2026, URL: https://www.mathworks.com/help/simscape/ug/thermal-liquid-modeling-workflow.html.