Reduced-Order Modeling of Thermal Dynamics in District Energy Networks using Spectral Clustering

Johan Simonsson\textsuperscript{12}, Khalid Tourkey Atta\textsuperscript{1}, Wolfgang Birk\textsuperscript{1}

Abstract—Simulation of thermal dynamics in city-scale district energy grids often becomes computationally prohibitive for long simulation runs. Current model order reduction methods offer limited interpretability with regards to the non-reduced system, and are not in general applicable for e.g., varying flow rates, multiple producers, or changing flow directions. This article presents a novel method based on graph theory that approximates the solution of an optimization problem that minimizes the local truncation error for heat transport in the grid. It is shown that the method can be used to reduce the thermal dynamic model of a city-scale energy grid, resulting in a coarser temporal and spatial resolution. The relative root mean square error was 2.3\% for the temperature in the evaluation scenario, comparing the reduced-order system with the non-reduced system at the instances of the coarser time-step.

I. INTRODUCTION

Simulation of district energy networks is a popular subject of scientific articles, where simulations are used e.g. for the planning of new district energy grids, optimization of production planning and thermal storage \cite{13}, and to explore various scenarios such as lowered supply temperatures and including waste heat from data centers.

There are several domain-specific commercial tools available for simulation of district energy networks, where most are focused on the steady-state analysis of pressure, flow, and temperatures \cite{10}. Dynamic simulation of the temperature in district energy grids on a city scale generally suffers from limited practical usability due to the computational demands of dynamic simulation \cite{12}. The computational performance is highly dependent on the scale of the problem and the resolution of interest – both the spatial and temporal resolution. The scale of the problem is usually given beforehand, but the resolution of interest is dependent on the use cases, and can also vary within a specific use case. It would thus be ideal to find methods that can adapt the spatial and time resolution of the simulation between simulation runs, or even dynamically during simulation.

Existing methods to reduce the computational demand can be roughly split into three categories. The first category consists of methods to simplify the model by merging nodes and pipes in a structured manner based on the physics of district energy grids. The most popular methods are called the Danish and the German method, respectively \cite{3}. However, both methods are limited in that they do not preserve the original structure of the grid, and thus the states are not directly interpretable with regards to the original system. Moreover, the methods cannot handle changing flow directions or multiple production units. A second category consists of model order reduction techniques for dynamic systems on state space form. Methods include Proper Orthogonal Decomposition (POD) and Hankel Model Order Reduction \cite{1}. Most of these methods rely on the system being linear and time-invariant, which is not true for the thermal dynamics of pipe flow with varying flow rates. Moreover, the POD method aims to find the most influential states, rather than averaging over (aggregating) many states. A third, somewhat related, category consists of methods for Computation Fluid Dynamics (CFD), e.g. various multi-grid methods \cite{14}. These methods are mostly focused on higher dimensions or higher accuracy than in this article, and are generally not applicable.

The article is structured as follows. First, the mathematical background for the differential equations of heat transport in a district energy pipe is presented, followed by a short introduction to the graph theory used in the article. Next, the two are combined to show how advection on a graph can be modeled. General reduced order models of a linear time-varying state space system for advection on a graph are then presented.

The article then proposes a novel method based on graph theory and the partial differential equations of advection with corresponding discretization methods, posing this as an optimization problem where the spatial resolution of the grid is optimized with regards to the time resolution of interest, minimizing the local truncation error. To avoid falling into the NP-hardness trap that is common for algorithms on graphs, the solution to the optimization problem is approximated by solving a generalized eigenvalue problem using the so-called graph Laplacian(s). While there are generic algorithms for clustering (sometimes called sparsening or coarsening of graphs) \cite{6}, these cannot be directly applied to a district energy grid without considering the underlying physics.

In the results section, the performance of the clustering is evaluated with regards to a non-reduced city-scale grid, and it is shown that the relative root mean square error is 2.3\% for the temperature of all the nodes of the original model when comparing 5s time steps with 600s time steps at the points of the coarser time step. The computational performance is improved with a speedup factor of $\approx 90$ for
The thermal dynamics of pipe flow

A necessary condition for stability of an explicit time integration schemes is that the volume of flow through a pipe during a time step of the solver does not exceed the integration scheme's time step.

\[ \frac{v_{sup} \Delta t}{\Delta z} = \frac{\dot{m}_{sup} \Delta t}{m} = \frac{\Delta t}{\tau} \leq 1 \iff \Delta t \leq \tau \]  

where \( \tau \) is the transport delay through the pipe. The Courant number also relates to the numerical accuracy of the discretization. A truncation error analysis using a Taylor expansion gives that the local truncation error for a finite volume upwind discretization is approximately

\[ T \approx \frac{v\Delta z}{2} \left(1 - \frac{v \Delta t}{\Delta z}\right) d^2 x / dx^2 + O(\Delta x^2), \]  

i.e., for \( C = 1 \) the truncation error due to discretization vanishes. Notably, the expression for the truncation error is similar to the diffusion equation, and the phenomenon is hence called numerical (or artificial) diffusion. While there exist many methods to minimize numerical diffusion and choose an optimal time step, for practical applications in district energy simulation, it is in most cases sufficient to aim for a maximal Courant number for the simulation that provides a reasonable trade-off between accuracy and stability for a given time-resolution.

B. The district energy grid as a graph

In Figure 1, a schematic view of a minimal district energy grid with three consumers and one producer is shown. The dashed circles represent nodes – connection points for junctions, consumers, or producers. The lines between the nodes are called edges and represent pipes. This minimal grid will be used as an example in the theoretical part, whereas a city-scale grid will be used in the results section to show the method's applicability.

The flows are assumed as balanced so that the supply flow equals the return flow for each node. A simplified schematic view of the same grid can be seen in Figure 2 where the labels \( n_i \) and \( e_i \) are added for nodes and edges. The grid can be represented by an incidence matrix, where the columns represent edges and rows represent nodes.

The flows can be represented by an incidence matrix, where the columns represent edges and rows represent nodes.

\[ M_w(i,j) = \begin{cases} \sqrt{w_{ij}}, & \text{if node } n_j \text{ is the target of } e_i \\ -\sqrt{w_{ij}}, & \text{if node } n_j \text{ is the source of } e_i \\ 0, & \text{elsewhere} \end{cases} \]
Fig. 2. Graph representation of a district energy grid with nodes weighted by \( n_i \) and edges by \( e_i \).

Consumers and producers are not included in the incidence matrix in this paper. When the weights are chosen as \( w_{ij} = 1 \), the incidence matrix is unweighted. For the example grid, the unweighted incidence matrix is

\[
M = \begin{bmatrix}
  n_1 & e_1 & e_2 & e_3 & e_4 \\
  n_2 & -1 & 0 & 0 & 0 \\
  n_3 & 1 & -1 & -1 & 0 \\
  n_4 & 0 & 1 & 0 & 0 \\
  n_5 & 0 & 0 & 1 & -1 \\
\end{bmatrix}
\]

The matrix \( L_w = M_n M_w^\top \) is called the weighted graph Laplacian, which is undirected, symmetric, and positive definite. The diagonal matrix with the diagonal elements from the weighted graph Laplacian is the degree matrix \( D_w \) and \( W = L_w - D_w \) is the weighted adjacency matrix. For unweighted matrices and vectors the subscript \( w \) is dropped in the notation.

For the model reduction performed in this paper, two different Laplacian matrices are used – the flow Laplacian \( L_f = M_f M_f^\top \) where the weights correspond to the maximum flow rate through each pipe, and the mass Laplacian \( L_m = M_m M_m^\top \) where the weights correspond to the mass of water within each pipe.

C. Advection on a graph

The graph itself can be interpreted as a finite volume discretization, where the edges represent mass flow rate and the nodes represent the masses of the volumes. Advection on graphs for the general case is described in [2]. The thermal dynamics of the supply and return pipes are modeled separately. To model advection, the modified incidence matrix

\[
M_v(i, j) = \begin{cases} 
1, & \text{if } j = i + 1 \\
0, & \text{elsewhere}
\end{cases}
\]

and the flow matrix \( U_d = \text{diag}(u_1, \ldots, u_m) \) where \( u_i \) is the flow on edge \( e_i \), are introduced. \( U_{do} \) is the diagonal matrix of flows exiting a node from the supply to the return pipe or vice versa. \( U_{di} \) the diagonal matrix of flows entering each node and \( x_{in} \) the temperature of flows entering each node. Equation (2) can then be written in matrix form as

\[
\dot{x}(t) = M_d^{-1}(MU_d(t)M_v^\top - U_{do}(t))x(t) + M_d^{-1}U_{di}(t)x_{in}(t)
\]

Equation (9) can be recognized as a time-varying state space model, that can be written on the form

\[
\dot{x}(t) = A(t)x(t) + Bu(t) \\
y(t) = Cx(t)
\]

with

\[
A(t) = M_d^{-1}(MU_d(t)M_o^\top - U_{do}(t)) \\
B = M_d^{-1}, \quad C = I 
\]

Since there are sometimes long pipes between nodes (i.e., no connected consumers for a long stretch of piping) in district energy grids, the graph can be oversampled by introducing intermediate nodes for edges corresponding to long pipes.

D. Low rank approximations

A low rank approximation of the state space system means that the calculations can be performed in a lower dimensional space \( \mathbb{R}^k \) with \( k < n \) where \( n \) is the number of states (nodes in the graph). The resulting number of edges from the reduction is \( l \). The reduction matrices for a graph need to act on both the nodes and the edges by contracting nodes and removing edges within the contracted set, and thus two transformation matrices \( P_n, P_e \), of sizes \( k \times n, m \times l \) respectively, are needed. For the reduced state space model, using the notation from Equation (9) and dropping the time index for more compact notation, the reduced versions of respective vectors and matrices become

\[
\tilde{M}_d = P_n M_d P_n^\top \\
\tilde{M}_o = P_n M_o P_e^\top \\
\tilde{U}_d = P_n U_d P_e^\top, \quad \tilde{U}_{do} = P_n U_{do} P_n^\top \\
\tilde{B} = P_n BP_e^\top, \quad \tilde{u} = P_n u
\]

Some special care needs to be taken for the reduced states \( \tilde{x} \) with regards to energy conservation, since the temperature of the reduced state should be the average temperature of the nodes scaled by the mass of each node

\[
\tilde{x}_i = \frac{\sum_{j \in C_i} m_j x_j}{\sum_{j \in C_i} m_j},
\]

or in matrix form

\[
\tilde{x} = (P_n M_d P_n^\top)^{-1} P_n M_d x.
\]

The reduced system can thus be written as

\[
\dot{\tilde{x}}(t) = \tilde{A}(t)\tilde{x}(t) + \tilde{B}\tilde{u}(t) \\
y(t) = P_n^\top \tilde{x}(t)
\]

where

\[
\tilde{A}(t) = \tilde{M}_d^{-1}(\tilde{M}_o U_d(t)\tilde{M}_o^\top - \tilde{U}_{do}(t))
\]

Note that since \( y(t) \) is the lifted state vector in \( \mathbb{R}^n \), there is a direct interpretation with regards to the states of the original graph.
III. SPECTRAL GRAPH CLUSTERING

Coarsening the spatial discretization of the grid corresponds to contracting nodes by adding the masses of the contracted nodes and removing the edges connecting contracted nodes. This can be seen as a graph clustering problem – the original nodes of the graph should be assigned to the cluster $N_i$ where the temperature is averaged within the cluster. The clustering of the reduced grid should minimize the truncation error for the advection equation with some stability margin with regards to the CFL condition, corresponding to choosing a target maximum Courant number $C_T$. The number of clusters needs to be approximated or chosen iteratively. When the number of clusters is decided, the minimization problems equals minimizing the sum of Courant numbers for the reduced problem

$$C_{\text{tot}} = \Delta t \sum_{i}^{k} C_i = \Delta t \sum_{i \leftrightarrow j} \left( \frac{\bar{m}_{ij}}{m_i} + \frac{\bar{m}_{ij}}{m_j} \right) \tag{17}$$

where $i \leftrightarrow j$ denotes that the clusters are connected and $\Delta t$ is the wanted step size.

To deal with the problem, some additional notation is introduced. For a cut in a graph $G = (N, E)$ consisting of nodes $N$ and edges $E$, the weight of the cut is defined as the sum of the weights of the edges separating $N_1, N_2$

$$\text{cut}(N_1, N_2) = \sum_{i \in N_1, j \in N_2} w_{ij} \tag{18}$$

Nodes are weighted with $m_i$, and thus the total mass of a cluster is

$$\text{mass}(N_i) = \sum_{i \in N_i} m_i. \tag{19}$$

To illustrate the clustering problem for advection on a graph, it is clarifying to look at the clustering of the graph to two clusters $N_1, N_2$ – i.e. the cut is a bisection. Starting from the previously introduced graph of a district energy grid, edge weights corresponding to maximum flow rate through each pipe, and node weights corresponding to the mass of each node are added, as can be seen in Figure 3.

$$\sum_{i}^{k} C_i = \sum_{i \leftrightarrow j} \left( \frac{\bar{m}_{ij}}{m_i} + \frac{\bar{m}_{ij}}{m_j} \right) \tag{17}$$

Following the procedure in [11], the minimization can be expressed in a more compact form as

$$\min_{v} \frac{v^\top L_f v}{v^\top M_d v} = \frac{\sum_{ij} \bar{m}_{ij} (v_i - v_j)^2}{\sum m_i v_i^2}. \tag{22}$$

That the expression minimizes the Courant number can be shown for $k = 2$ by introducing the indicator vector $v$ where $v_i = 1$ if the corresponding node $n_i \in N_1$ and $v_i = -1$ if $n_i \in N_2$, in the expression above. The right hand side of Equation (22) can then be rewritten as

$$\frac{\sum_{ij} \bar{m}_{ij} (v_i - v_j)^2}{\sum m_i v_i^2} = 4 \frac{\bar{m}_{12}}{m_{N1}} + 4 \frac{\bar{m}_{21}}{m_{N2}} = \frac{4}{\Delta t} (C_1 + C_2), \tag{23}$$

i.e. the sum of Courant numbers scaled by a constant. This combinatorial optimization problem is known to be NP-hard [7], but the optimum can be approximated by relaxing the problem so that $v$ contains real numbers, and the partitions are chosen from the sign of $v$. The expression on the left hand side of Equation (22) is the generalized Rayleigh quotient and is minimized by the smallest eigenvector of the generalized eigenvalue problem $L_f v = \lambda M_d v$. The first eigenvector is trivially $v_1 = 1/\sqrt{\bar{m}}$, but the second eigenvector $v_2$ called the Fiedler vector, that is orthogonal to $v_1$, solves the minimization problem.

Following the procedure in [7], the method can be extended to $k$ clusters by including the first $k$ vectors of the generalized eigenvalue problem, forming the $n \times k$ matrix.
of eigenvectors $V_k = [v_1 \ldots v_k]$ and performing k-means clustering on the rows – i.e. viewing the nodes as points in the $k$-dimensional space spanned by the rows of $V_k$. The maximal Courant number of the reduced-order system can be calculated as

$$\tilde{C}_{\text{max}} = \max(\text{diag}(0.5 \Delta t M_d^{-1} \tilde{L}_f)),$$

where the number of clusters should be adjusted so that $C_{\text{max}} \approx C_T$. The clustering process is summarized in Algorithm 1.

**Algorithm 1 Spectral graph clustering of district energy network**

- $k \leftarrow$ user specified number of clusters
- Calculate graph Laplacians $L_f, L_m$
- $M_d \leftarrow \text{Diagonal(diag}(L_m))$
- Solve the generalized eigenvalue problem $L_f V = \lambda M_d V$
- $V_k \leftarrow [v_1, \ldots, v_k]$
- Let $y_i = V_k(i,:)$
- Assign each $y_i$ to cluster $N_j$ by performing a k-means clustering on the rows of $V_k$
- Adjust the number of clusters if $C_{\text{max}} \neq C_T$

The reduction matrices $P_n \in \mathbb{R}^{k \times n}$, $P_e \in \mathbb{R}^{m \times l}$ calculated from a given clustering serve two separate purposes, namely 1) merging the nodes of each cluster, and 2) removing the edges within each cluster, where

$$P_n(i,j) = \begin{cases} 1, & n_j \in N_i \\ 0, & \text{elsewhere} \end{cases} \quad (25)$$

and

$$P_e(i,j) = \begin{cases} 1, & e_i \in N_j \rightarrow N_k \neq j \\ 0, & \text{elsewhere} \end{cases} \quad (26)$$

The reduced-order system matrices in Equation (12) can then be computed.

**A. A note on numerical methods**

The computational bottlenecks for large grids in Algorithm 1 are the calculation of the eigenvalues and the memory requirements of the matrices and corresponding algorithms. However, only the first $k$ eigenvalues and corresponding eigenvectors are needed, and the graph Laplacian matrix is sparse. Storing the graph Laplacian matrices in a sparse format and using iterative algorithms such as Lanczos algorithm allows for efficient calculations of the first $k$ eigenvalues and corresponding eigenvectors when the number of clusters is sufficiently small. For reference, on a standard laptop with an 8th generation i7 Intel processor and 16GB of RAM, some elapsed CPU times for the clustering algorithm are presented in Table I.

**IV. RESULTS**

To evaluate the clustering performance, a partial model of the district energy grid of Luleå, Sweden, was compared with the same model where the model order reduction provided in the paper was applied. The model comparison only includes the thermal dynamics of the supply piping. The advection equation and flows are similar on the return side, with an opposite flow direction, but the return temperatures of the consumers need to be modeled to get realistic results for the return piping, which is beyond the scope of the paper.

The models were automatically generated from grid data, where the spatial discretization of the reference model corresponds to that each pipe was discretized with a uniform segment length $\Delta z$ when the pipe length exceeds $\Delta z$, elsewhere the pipe length provided by data was used. The original model consisted of about 3700 pipe sections and 1500 consumers. After some initial preprocessing, where nodes connected by pipes shorter than 10m were merged, the resulting model included 2715 pipes.

Both models were generated as Julia code and run using BDF solvers provided by the package DifferentialEquations.jl [8]. In this case, and the general case, there was no measured ground truth available – the reference instead corresponded to what would be the result of directly modeling the grid from available geographical data with pipe models that are uniformly discretized using finite volumes.

The simulation scenario used was an increased outdoor temperature, with an increased demand of the consumers and a linearly increasing supply temperature. The demands of the consumers included morning and afternoon peaks, with random perturbations so that the demand pattern resembled what could be expected from real-life data. An example of the demand for 100 different consumers is seen in Figure 5. The simulation time was set to 24h, where the reference model used a time-step of 5s and the reduced model a time-
step of 600s. While the model was optimized with respect to the time-step, this was not necessarily the internal time step of the solver. The reduced model’s speedup factor was around 90, resulting in the reduced-order model running at about 4000 times the real-time speed on a standard laptop computer. To evaluate how close the reduced model tracked the reference model, the states (temperatures) $x$ of the original model are compared with the lifted states $\mathbf{P}_n^T \tilde{x}$, at each successful time step of the larger step size, using the relative root mean square error (rRMSE)

$$\text{rRMSE} = 100 \frac{1}{\bar{x}} \sqrt{\left\| \mathbf{P}_n^T \tilde{x} - x \right\|_2^2}.$$  \hspace{1cm} (27)

The reduced-order model was generated using the algorithm provided in Algorithm [1] and the procedure described in the paper. An example of the clustering found by the algorithm for $k = 150$ clusters is seen in Figure 6. The number of clusters was chosen so that $C_{\text{max}} \approx 0.9$ for the time step size and maximum flow rate, corresponding to about 150 clusters for the reduced model.

The first comparison used idealized pipe models only including the advection equation for thermal dynamics so that e.g., (non-numerical) diffusion and thermal losses were neglected. The simulation comparison resulted in a rRMSE of $\approx 0.7\%$, corresponding to an absolute error within $1^\circ C$. A heat map of the absolute error can be seen in Figure 7.

From the truncation error analysis, it can be deduced that the reduced-order model does not necessarily have a larger truncation error so that the reduced model might be better in this regard. However, the dynamics of the more fine-grained model between the 600s time steps is lost in the comparison, that might be of interest for specific use cases.

In the second comparison shown in Figure 8 a more realistic model with heat loss to the outside was used, otherwise using the same setup as in the preceding comparison. The resulting rRMSE was $2.3\%$ for this case, and as expected, some individual nodes – typically for low consumption consumers – were relatively far off due to averaging the temperature over the cluster.

In Table II a comparison between the elapsed CPU time and the rRMSE for some different choices for the number of clusters for 24h of simulation time, compared at the instances of 600s time steps is shown.

| Clusters $k$ | Step size | CPU time | rRMSE |
|-------------|-----------|----------|-------|
| No clustering | 600s | 2042s | 2.9% |
| 500 | 600s | 508.9s | 2.9% |
| 150 | 600s | 23.1s | 2.3% |
| 50 | 600s | 5.9s | 3.0% |

Notably, more clusters did not necessarily result in a better rRMSE for a given time step, consistent with the theoretical truncation error.
V. SUMMARY AND CONCLUSIONS
This paper presents a novel method for reduced-order modeling of a district energy grid’s thermal dynamics. The method preserves the total energy of the water in the system, and the states can be directly interpreted with the states of the original system. It is shown that the reduced model can run with a speedup factor of ≈ 90 with a corresponding rRMSE of 0.7% for all nodes in the non-reduced grid for the advection only, and 2.3% when including heat losses to surroundings, compared at the instances of the longer time step.

While it is a promising result for district energy simulation, some theoretical background, bounds, and choices of parameters such as the number of clusters are in large parts omitted due to the format of the article and could be expanded on in the future.

Moreover, the pressure and flow calculations are only briefly mentioned in the article but are of considerable interest for many use cases. While these calculations can be assumed to be static for the time resolutions assumed in this paper, e.g., pump characteristics, thermal storages, and consumer behavior needs to be included for a complete model – a challenging problem in its own right.

ACKNOWLEDGMENT
The authors want to thank Luleå Energi AB and especially Fredrik Udén for discussions and making data available.

REFERENCES
[1] Athanasios C. Antoulas. Approximation of Large-Scale Dynamical Systems. Advances in Design and Control. Society for Industrial and Applied Mathematics, Jan. 2005. ISBN: 978-0-89871-529-3. DOI: 10.1137/1.9780898718713
[2] Airlie Chapman and Mehran Mesbahi. “Advection on graphs”. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference. ISSN: 0743-1546. Dec. 2011. pp. 1461–1466. DOI: 10.1109/CDC.2011.6161471
[3] Basak Falay et al. “Enabling large-scale dynamic simulations and reducing model complexity of district heating and cooling systems by aggregation”. en. In: Energy 209 (Oct. 2020). p. 118410. ISSN: 0360-5442. DOI: 10.1016/j.energy.2020.118410
[4] B. van der Heijde et al. “Dynamic equation-based thermo-hydraulic pipe model for district heating and cooling systems”. en. In: Energy Conversion and Management 151 (Nov. 2017). pp. 158–169. ISSN: 0196-8904. DOI: 10.1016/j.enconman.2017.08.072
[5] Randall J. LeVeque. Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press, 2002. ISBN: 978-0-521-00924-9. DOI: 10.1017/CBO9780511791253
[6] Andreas Loukas. “Graph Reduction with Spectral and Cut Guarantees”. In: Journal of Machine Learning Research 20.116 (2019). pp. 1–42. ISSN: 1533-7928.
[7] Ulrike von Luxburg. “A tutorial on spectral clustering”. en. In: Statistics and Computing 17.4 (Dec. 2007). pp. 395–416. ISSN: 0960-3174, 1573-1375. DOI: 10.1007/s11222-007-9033-z
[8] Christopher Rackauckas and Qing Nie. “DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia”. en. In: Journal of Open Research Software 5.1 (May 2017). Number: 1 Publisher: Ubiquity Press. p. 15. ISSN: 2049-9647. DOI: 10.5334/jors.151
[9] Kevin Sartor, David Thomas, and P Dewallef. “A comparative study for simulating heat transport in large district heating networks”. In: International Journal of Heat and Technology 36.1 (Mar. 2018). pp. 301–308. ISSN: 03928764. DOI: 10.18280/ijht.360140.
[10] Gerald Schweiger et al. “District energy systems: Modelling paradigms and general-purpose tools”. en. In: Energy 164 (Dec. 2018). pp. 1326–1340. ISSN: 0360-5442. DOI: 10.1016/j.energy.2018.08.193
[11] Jianbo Shi and Jitendra Malik. “Normalized Cuts and Image Segmentation”. en. In: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 22.8 (2000), p. 18.
[12] Johan Simonsson et al. “Experiences from City-Scale Simulation of Thermal Grids”. en. In: Resources 10.2 (Feb. 2021). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute. p. 10. DOI: 10.3390/resources10020010 URL: https://www.mdpi.com/2079-9276/10/2/10 (visited on 01/25/2021).
[13] Ezio Todini and Lewis A. Rossman. “Unified Framework for Deriving Simultaneous Equation Algorithms for Water Distribution Networks”. en. In: Journal of Hydraulic Engineering 139.5 (May 2013). pp. 511–526. ISSN: 0733-9429, 1943-7900. DOI: 10.1061/(ASCE)HY.1943-7900.0000703
[14] Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schuller. Multigrid. English. 1st edition. Amsterdam Heidelberg: Academic Press, Dec. 2000. ISBN: 978-0-12-701070-0.
[15] Annely Vandermeulen, Bram van der Heijde, and Lieve Helsen. “Controlling district heating and cooling networks to unlock flexibility: A review”. In: Energy 151 (2018). Publisher: Elsevier Ltd. pp. 103–115. ISSN: 03605442. DOI: 10.1016/j.energy.2018.03.034