A LOCAL TIME STEPPING METHOD FOR DISTRICT HEATING NETWORKS

RAUL BORSCHE, MATTHIAS EIMER, AND NORBERT SIEDOW

Abstract. In this article a new numerical solver for simulations of district heating networks is presented. The numerical method applies the local time stepping introduced in [11] to networks of linear advection equations. In combination with the high order approach of [4] an accurate and very efficient scheme is developed. In several numerical test cases the advantages for simulations of district heating networks are shown.

Keywords: District heating, high order scheme, ADER, advection, local time stepping, network

1. Introduction

In densely populated regions, district heating is a possible alternative to conventional heating systems. Heat is generated in central plants and distributed to private and commercial consumers, providing space heating and hot water. There are systems with a high energy efficiency and the pollution can be reduced compared to local boilers. In order to find an efficient control for such systems, fast and accurate simulations are needed.

Mathematically the distribution of the temperature in the district heating system can be modeled by a network of linear scalar balance laws [18]. Such very simple equations can be solved efficiently by various numerical methods for hyperbolic conservation laws [25, 20]. Among these methods high order schemes provide more accurate numerical results for comparable computational effort [16, 24, 8, 6]. Not all of these schemes can be directly extended onto networks of conservation laws, but members of the broad class of upwind type methods can be applied [13, 12, 7]. In order to increase the accuracy at fixed computational costs high order schemes on networks have been developed in [4, 22, 5, 19].

In all these methods the time steps are chosen according to CFL conditions. In case of flows on networks the time steps are synchronized in the whole system, i.e. Δt takes the smallest values of all edges in the network. In this paper we relax this limitation by applying the local time stepping approach of [11]. If the speeds on the edges vary strongly this alone can improve the numerical efficiency. But due to the simple structure of the linear equations the computational costs can be further reduced. As the time steps can be chosen independently for every edge, each Δt is selected such

Technische Universität Kaiserslautern, Department of Mathematics, Erwin-Schrödinger-Strasse, 67663 Kaiserslautern, Germany

Fraunhofer ITWM, Fraunhoferplatz 1, 67663 Kaiserslautern, Germany

E-mail addresses: borsche@mathematik.uni-kl.de.

that the cell values are just shifted in the direction of the flow. Thus the exact values in the pipes are obtained by almost no costs. This lowers the computation times for the entire simulation significantly. This algorithm can be combined with the high order approach of [4] to obtain an efficient tool for accurate and very fast simulations of district heating systems.

This paper is organized as follows: In section 2 the model for the flow of hot water in district heating systems is presented. In section 3 the numerical scheme is introduced. This includes a detailed description of the local time stepping and the fluxes across the junctions. Further the extension to a high order method is discussed. Finally, in section 4, three numerical test cases are shown. The numerical efficiency is investigated and the results of the local time stepping scheme are compared to those of high order ADER methods.

2. Modelling

A network of a district heating system is composed by several different components. First we describe the mathematical models of the flow in a single pipe. These can be joined to a network and connected to the houses via suitable coupling conditions. The modeling follows the approach presented in [18].

2.1. **A single Pipe.**

Consider a closed pipe filled completely with hot water. A mathematical model can be derived from the conservation of mass and the balance of momentum and energy, which is given by

$$
\begin{aligned}
\partial_t \rho + \partial_x q &= 0 \\
\partial_t q + \partial_x \left(\frac{q^2}{\rho} + p \right) &= -\frac{\lambda}{2d} \frac{q|q|}{\rho} - g(\partial_x b)\, \rho \\
\partial_t \left(c_p \rho T \right) + \partial_x \left(q c_p T \right) &= -\frac{4k}{d} \left(T - T_{ext} \right) \ .
\end{aligned} \tag{1}
$$

Here ρ denotes the density of the water, v its velocity in x direction, $q = \rho v$, p the pressure and T is the temperature of the fluid. λ is a friction coefficient for the Darcy-Weisbach friction formula and d the diameter of the tube. The term $g(\partial_x b)\, \rho$ takes the vertical elevation b into account, where g is the gravitational acceleration. The right hand side in the third equation is modeling the cooling related to the outer temperature T_{ext} with the thermal transmittance k.

To obtain a closed model often a closure relation for the pressure p is used. Instead of following this approach we directly provide a closure relation for the density ρ. As the water in the pipes is almost incompressible, but its density is still varying with the temperature, we assume the density to be a function just of T, i.e. $\rho = \rho(T)$.

Inserting this Ansatz into the energy equation and with help of the continuity equation (1) we obtain

$$\begin{aligned} \partial_t \left(c_p \rho T\right) + \partial_x \left(q c_p T\right) &= -\frac{4k}{d}\left(T - T_{ext}\right) \\ c_p \rho \partial_t T + c_p T \partial_t \rho + q c_p \partial_x T + c_p T \partial_x q &= -\frac{4k}{d}\left(T - T_{ext}\right) \\ \partial_t T + v \partial_x T &= -\frac{4k}{c_p d \rho}\left(T - T_{ext}\right) \ . \end{aligned} \tag{2}$$

By multiplying with $\partial_T \rho$ this results in

$$\partial_t \rho + v \partial_x \rho = -\frac{4k \partial_T \rho}{c_p d \rho}\left(T - T_{ext}\right) \ .$$

As the variations of the density due to the external temperature are small we can neglect the term on the right hand side. Using this and again the continuity equation leads to

$$\partial_x v = 0 \ . \tag{3}$$

Knowing that v is constant in space the momentum equation can be transformed into

$$\rho \partial_t v + \partial_x p = -\frac{\lambda}{2d} v|v|\rho - g(\partial_x b)\, \rho \ ,$$

which are the well known incompressible Euler equations [21].

These equations can be further simplified by integrating over the length of the pipe $[0, L]$. Aside from $v(t, x) = v(t)$ we obtain

$$\bar{\rho} \partial_t v + p(L) - p(0) = -\frac{\lambda}{2d} v|v|\bar{\rho} - \frac{1}{L} g(\partial_x b)\bar{\rho} \ , \tag{4}$$

where $\bar{\rho} = \frac{1}{L}\int_0^L \rho \, dx$ is the averaged density and we assumed that the vertical slope in the tube is constant.

On a single pipe the system composed by the equations (3),(4) and (2) has to be supplemented with boundary conditions. A common choice for $v > 0$ is to provide p and T at the left and q at the right end. For a pipe in a network these boundary conditions are replaced by coupling conditions in the junctions.

2.2. **Network.** In a district heating network the hot water is distributed to the houses via a system of pipes. A network of the same structure transports the colder water back to the power plant. Such networks can be modeled by connecting the equations for the pipes via suitable coupling conditions. In order to simplify the notation we assume that all pipes at a junction are orientated away from the junction. This can be realized by local variable transformations of all directed quantities. In both networks we consider the

following coupling conditions

$$\sum_{i=1}^{N} A_i q_i = 0 \tag{5}$$

$$p_i = p_j \quad i,j = 1,\dots,N,\ i \neq j \tag{6}$$

$$\sum_{i=1}^{N} c_p A_i q_i T_i = 0 \tag{7}$$

$$T_j = T_i \quad \text{for } i,j \text{ outgoing flows}\ , \tag{8}$$

where all quantities are evaluated at $x = 0$. The junction is assumed to connect N pipes and A_i denotes the cross section of the i-th pipe. Equation (5) states the conservation of mass and (7) the conservation of energy at the node, as the respective fluxes of (1) are balanced. The continuity of the pressure across the junction is a widely used condition, as e.g. in [1, 9, 10]. Additionally we assume a perfect mixing of flows at the junction, i.e. all outgoing flows have the same temperature (8).

Note that this is expected to be the correct number of coupling conditions for the system of (3), (4) and (2). The equations (3) and (4) require one coupling condition independent of the orientation of the flow, which are provided by (5) and (6), while (7) and (8) set the values required by (2) for the outgoing flows.

An additional important component in a district heating system are the consumers. Each consumer is demanding a certain amount of thermal power $Q_k(t)$. Further the outgoing temperature T_{out} is assumed to be a fixed value and no mass is lost. This leads to the following equations

$$\begin{aligned} q_{in} &= q_{out} \\ Q_k(t) &= c_p A q_{in} \left(T_{in} - T_{out}\right)\ , \end{aligned} \tag{9}$$

where T_{in} is the temperature of the flow arriving at the house. These relations of the three quantities q_{in}, q_{out} and T_{in} are connecting the supplying network with the one for the return flow. Note that (9) can be viewed as the coupling conditions for the flow equations (3) and (4), the boundary value for the temperature is given by the fixed value of T_{out}.

3. Numerical Method

The complete mathematical model of district heating networks is a complex system system of algebraic and partial differential equations. In order to simplify the numerical computations we consider the flow equations and those for the temperature separately.

The flow is described by the equations (3), (4), (5), (6) and (9). To close the model the pressures of in- and outflow to the network are specified. The evolution of the temperature follows (2), (7) and (8). Additionally a temperature profile is provided at the entrance of the network and the outflow at the consumers is fixed to T_{out}. The coupling of these two sub systems is bidirectional. For the temperature equation (2) the velocity has to be provided by the flow solver. On the other hand the flow at the consumers depends on the local temperature (9) at these nodes.

Numerically this decomposition leads to a splitting algorithm [20], i.e. for a given time interval first the flow is computed and the resulting velocities are used to update the temperatures. Such a splitting reduces the accuracy of the coupled scheme to first order.

The equations for the flow can be solved efficiently using solvers which exploit the underlying structure of the network [14, 18, 15]. In the following we focus on numerical methods for the temperature equations. If a flow solver is needed the method described in [15] is used. The time step of the splitting is chosen according to the fastest waves in the temperature model.

The equations describing the evolution of the temperature form a network of linear scalar balance laws. For hyperbolic conservation laws many different numerical methods are available [20, 25]. For networks of hyperbolic PDEs mainly upwind type schemes should be considered, as the orientation of the waves can be analyzed individually [3]. For scalar equations often the Godunov method is used [7, 13, 2], which coincides with the Upwind scheme in the linear case [12]. Further there exist some recent approaches extending high order methods on networks of conservation laws [4, 22, 5, 19].

All of these explicit schemes have in common that they operate with the same time step on the whole network. This synchronized time step is determined by the minimal bound of the CFL conditions in all edges

$$\Delta t_{net} = \min_e \Delta t^e \ ,$$

where Δt^e is the maximal time step allowed on edge e. In the description of the numerical schemes each edge of the network is discretized with an equidistant grid with the spacing Δx. In the following it is assumed that Δx is identical on all the edges, but all methods can be used with variable Δx.

3.1. **The Upwind scheme.** The upwind scheme is one of the classical first order schemes for scalar conservation laws. If applied to an advection equation

$$\partial_t T + v\partial_x T = 0 \tag{10}$$

with initial condition $T(0,x) = T_0(x)$ it reads

$$\begin{cases} T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x}\left(vT_i^n - vT_{i-1}^n\right) & \text{if } v > 0 \ , \\ T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x}\left(vT_{i+1}^n - vT_i^n\right) & \text{if } v < 0 \ . \end{cases}$$

The values $T_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x+1/2} T(t^n,x)dx$ denote the cell averages on the cell $[x_{i-1/2}, x_{i+1/2}]$ at the time t^n.

It is well known that for $\Delta t = \frac{\Delta x}{v}$ it computes cell averages at the next time level without numerical error, since the exact solution is given by $T(t,x) = T_0(x - vt)$. In this case the cell averages are just shifted in the direction of the flow. The smaller Δt is compared to $\frac{\Delta x}{v}$ the more the numerical solution deviates from the exact values due to numerical diffusion.

At a junction only the values for the outgoing edges have to be specified. These share all the the same value due to (7) which is according to (8) given

by

$$T^k_{1,i} = \frac{1}{\sum_{j\in E_{out}} c_p A_j q_j} \sum_{i\in E_{in}} c_p A_i q_i T^k_{i,L_i} \ . \tag{11}$$

Here 1 is the index of the ghost cells at the left end of the edges and L_i is the index of the last cell at the right end of edge i. Here it is used that in the Upwind scheme for $v > 0$ the fluxes across the interfaces are given by the left cell. This notation assumes that $v > 0$ in all edges and the indices collected in E_{out} are those of the edges orientated away from the junction and E_{in} those ending there. This can be achieved by local transformations without changing the actual flows.

As mentioned above the time step is determined for all edges as the minimum of all involved CFL conditions. This leads to a Δt which might be significantly below $\frac{\Delta x}{v_i}$ for some edges i. This can induce strong numerical diffusion in those edges. Corresponding numerical results are provided in section 4.

3.2. **A local time stepping scheme.** Motivated by the above observation we will construct a procedure which decouples the time step in the network. Note that an exact CFL condition $\Delta t = \frac{\Delta x}{v_i}$ could also be achieved by choosing Δx in each edge independently such that equality is restored with a global time step. But if the v_i vary over time this would involve a remeshing after every update, which is costly, especially when higher order methods are considered.

In the following we describe an approach where the states in the edges are given at different time levels, each of them following its local time step. The time steps Δt for an edge are chosen adaptively according to

$$\int_t^{t+\Delta t} v(\tau)\mathrm{d}\tau = \Delta x \ . \tag{12}$$

This ensures a CFL number equal to 1, even for v varying in time.

As the solution T travels exactly one cell in each time step, no computations for the inner cells of the edge are needed. However the update of a given pipe can only be computed if the flux at the first interface, i.e. at the junction, is known.

According to (7) these values can only be computed, if the values of all ingoing edges are already determined. Thus the state of an edge can only be updated, if its time step is such that the new time level does not exceed the time level of the ingoing pipes. That means, if $(t^e, \Delta t^e)$ are the current time level and time step size of pipe e, the next step is performed if

$$t^e + \Delta t^e \leq t^k + \Delta t^k \tag{13}$$

is fulfilled for all ingoing pipes k. If (13) is satisfied the ingoing fluxes for pipe e can be computed and its state is updated to the next time level $t^e \to t^e + \Delta t^e$. Note that the ingoing pipes not necessarily are already at the time level $t^k + \Delta t^k$, but their flux at the final interface is determined.

In the following we illustrate the procedure for a single junction , where all states are given at time t_1. With these values the fluxes across the last interfaces can be computed and the CFL condition specifies the time span

for which these values hold. Assume the minimal time span in the network is $t_2 - t_1$, i.e. the fluxes at least up to t_2 are known for every edge. Thus the two nodes adjacent to the edge for which the validity of the fluxes expires at t_2 can compute all their fluxes up to t_2. These fluxes are used to update the values of this edge to the new time level and to compute new fluxes with a new lifespan.

The next edge to be updated is now the one for which the flux expires next at t_3. But the contribution of the previous edge has to be composed by the flux for $t_2 - t_1$ and the one for $t_3 - t_2$. Therefore we introduce memory variables for each edge. These store the amount of thermal energy which should enter the corresponding pipe. Following (11) the thermal energy entering an edge i in the time step $t_2 - t_1$ is given by

$$\Delta Q^i = \left(\sum_{j \in E_{in}} q_j A_j c_p T_j \right) \quad \frac{q_i A_i}{\sum_{j \in E_{out}} q_j A_j} (t_2 - t_1) \ . \tag{14}$$

Accordingly the memory variable $\mathcal{MV}_i$ is filled via

$$\mathcal{MV}_i^{\mathrm{sign}(v_i)} = \mathcal{MV}_i^{\mathrm{sign}(v_i)} + \Delta Q^i \ . \tag{15}$$

Note that we introduce for each edge i two memory variables, one at the left end $\mathcal{MV}_i^1$ and another at the right end $\mathcal{MV}_i^{-1}$. This is needed if the orientation of the flow changes, i.e. there is always a memory variable at the inflow.

Whenever a pipe fulfils the update criterion (13), after the node update for both adjacent nodes, its cell values are updated. For positive (negative) flow velocity, the inner cells simply shift to the right (left). The energy stored in the last (first) cell exits and is transferred to other pipes. The first (last) cell value is computed as the average over all fluxes stored in $\mathcal{MV}_i^{|v_i|}$:

$$T = \frac{\mathcal{MV}_i^{|v_i|}}{q_i A_i c_p \Delta t_i} \ . \tag{16}$$

Since all thermal energy has entered the pipe, the value of $\mathcal{MV}_i$ is reset to zero.

The ordering of the local time steps is illustrated in figure 1 for a junction connecting three pipes with flow entering from the e_1 and exiting into e_2 and e_3. It is assumed that the CFL bounds of the edges restrict the local time steps to $\Delta t_1 = 1$, $\Delta t_2 = 0.5$ and $\Delta t_3 = \frac{2}{3}$ respectively. The vertical axis in the sub figures displays the time, possible time steps are drawn in light gray, completed time steps in dark gray.

Starting at $t = 0$, all states have to reach the final time level $t_{end} = 2$. In every step, the algorithm calculates the next time level for all pipes satisfying condition (13). Since $\Delta t_2 \leq \Delta t_i, i = 1, 3$ the first pipe to be updated is e_2, see figure 1a.

The energy flux over the junction can be computed up to $t = 0.5$. The corresponding amount of thermal energy is stored in the memory variables $\mathcal{MV}_2$ and $\mathcal{MV}_3$. $\mathcal{MV}_1$ stays void. As e_2 is updated the values in the internal cells are shifted to the right, the energy in $\mathcal{MV}_2$ is filled in the first cell according to (16). After its content was transported, the value of

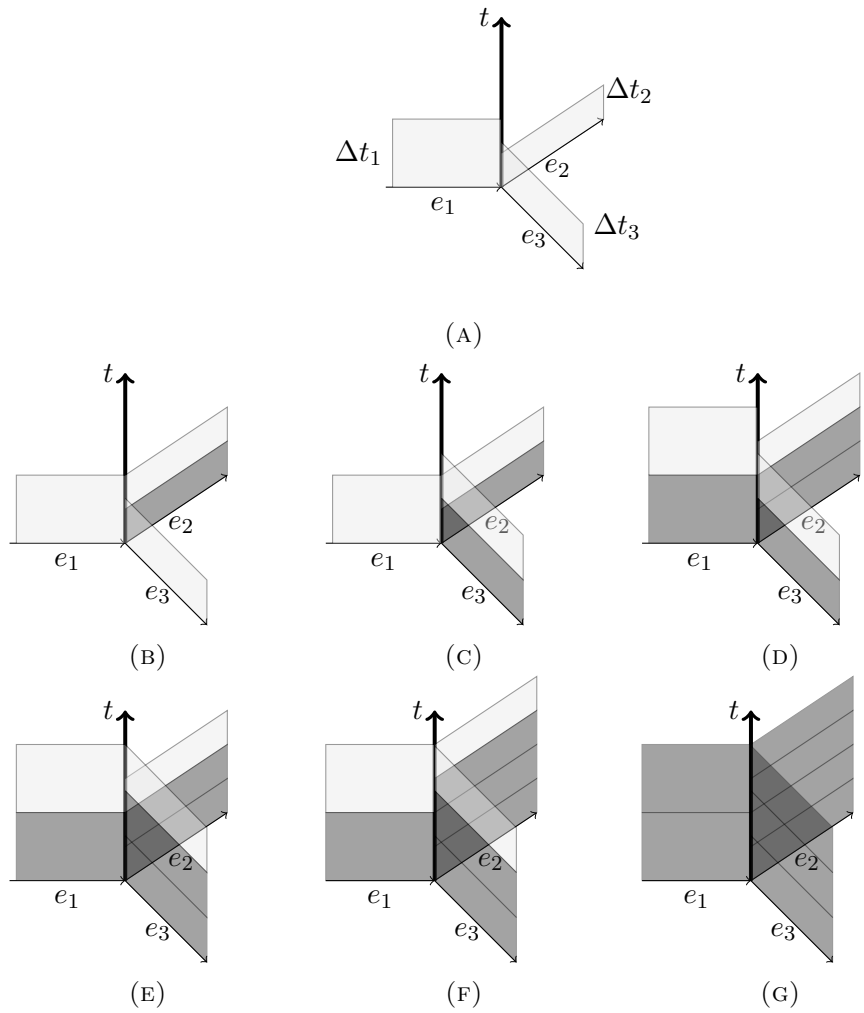

FIGURE 1. Illustration of the local time steps at a single node.

$\mathcal{MV}_2$ is set to zero. Finally the fluxes of the new state are computed, which completes the time step.

The next CFL bound is active in e_3 at $t = \frac{2}{3}$. Thus the fluxes in the junction from $t = 0.5$ to $t = \frac{2}{3}$ can be filled into the memory variables. The update of e_3 moves its time level from 0 to $t = \frac{2}{3}$. For the next time level the edges e_1 and e_2 can be updated simultaneously, as their fluxes expire at the exact same time.

Similarly the procedure continues until the final time $t_{end} = 2$ is reached. Note that, in contrast to this example, in general the last time steps do not reach t_{end} exactly but would end at a later time level. For this final step a classical upwind step is performed with $\Delta t \leq \frac{\Delta x}{v}$, what might introduce some numerical diffusion. In general this effect is only hardly visible, as it occurs in just a single iteration.

The local time stepping has two advantages compared to the classical upwind scheme. First it is more accurate since numerical diffusion is only added at the junction. There the fluxes entering the node transport the exact

amount of thermal energy exiting the edge, but since it is assumed that the flux is constant in the corresponding time interval the resulting accuracy is only of first order. The second advantage is that it is less expensive from a computational point of view. The management of the time steps does not require significant computational time. In reward it reduces the total number of time steps in the network. Further, along the edges no actual computation has to be carried out, since with the exact CFL condition the cell values are just shifted into the direction of the flow. This reduces the computational effort significantly.

3.3. **High order coupling.** As pointed out above the local time stepping procedure combined with the upwind scheme is only first order accurate. This can be improved by using the ADER approach [4] at the coupling points, without loosing the advantages of the local time steps. In the following we summarize the modifications needed to obtain a method of arbitrary order of accuracy. Along the edges still the exact CFL condition holds, because the exact solution already is accurate enough. Thus only the fluxes entering the junction have to be computed with the ADER approach.

The aim is to obtain an accurate temporal evolution of the fluxes at the node. The first step is a polynomial reconstruction of the value at the last interface using the given cell averages. A detailed description of the WENO reconstruction is given in [16], a one sided reconstruction based on polynomials with increasing degrees is presented in [23]. In [17] a refinement of the weight of the first, i.e. constant, polynomial is introduced.

This reconstruction provides a spatial polynomial, which can be transformed via the Cauchy-Kowalevskaya or Lax-Wendroff procedure into a temporal one. Since the equation is linear the coefficients of the new polynomials up to order K are

$$\partial_t^k T = (-v)^k \partial_x^k T \quad k = 0, \dots, K \ . \tag{17}$$

With these the temporal evolutions the fluxes entering the junction can be computed. The approximation of the outgoing fluxes can be determined by the temporal derivatives of the coupling conditions

$$\begin{aligned} \sum c_p A_i q_i \partial_t^l T_i = 0 \qquad & \text{for } l = 0, \dots, K, \\ \partial_t^l T_j = \partial_t^l T_i \qquad & \text{and } i, j \text{ outgoing flows} \ . \end{aligned}$$

This is a linear system of equations, which gives the temporal derivatives of the outgoing flows, depending on the ingoing ones. Once the evolution in time of the outgoing fluxes is known, their contribution for the time step Δt

$$\Delta Q^j = \Delta t \rho c_p \sum_{l=0}^{k} \frac{\Delta t^l}{(l+1)!} \partial_t^l T_{out}$$

can be added to the memory variable.

3.4. **Extension to time dependent velocity field.** In the previous section the local time stepping scheme was discussed for velocities constant in time. The numerical methods still can be applied if $v = v(t)$ changes over time, i.e. if (10) is replaced by

$$\partial_t T + v(t) \partial_x T = 0 \ . \tag{18}$$

In the following we assume $v(t)$ to be known and regular enough to apply higher order methods.

The CFL condition (12) was already formulated for variable v. Also the further steps of the time step management remain unchanged. When a higher order method is used the Cauchy-Kowalevskaya procedure is more involved than (17), due to the product rule. The resulting lengthy expressions can be determined using packages for symbolic manipulations. Further in the integration of the thermal fluxes at the nodes v is involved

$$\Delta Q = \int_t^{t+\Delta t} \rho c_p v(\tau) T_{out}(\tau) \mathrm{d}\tau \ . \tag{19}$$

The update inside the pipes remains unchanged, since in (12) Δt is chosen exactly such that the values can be shifted by one cell.

3.5. **Source terms.** Finally we discuss the incorporation of the source term in the local time stepping scheme. Note that the the source term on the right hand side of (2) is linear and not varying in space. These two features can be exploited to further reduce the computational costs. For (2) the exact solution even with source term can be given explicitly, i.e. in each time step the cell values are not just shifted in the direction of the flow but also adjusted according to the source term. If the values of T are only needed at the outflow and not used in the interior of the pipes, the computational costs can be further reduced by applying the source term only at the outflow. Therefore inside the pipe just the advection according to the homogeneous PDE (10) is considered. For the source term we keep track of the time span which the cell value at the outflow stayed in the pipe. Then the outflow is changed by applying the source term for this complete time interval. This is only possible, if both the PDE and the source term are linear and the source term independent of x.

4. Numerical Examples

In this section several numerical examples are presented in order to investigate the properties of the local time stepping scheme. These test cases include a comparison with the high order ADER scheme in terms of accuracy and efficiency, as well as a case study on a larger realistic network.

4.1. **Split network.** As first examples a simplified supply branch of a district heating network is considered. As depicted in figure 2 the network is composed of 6 pipes and 4 junctions. The heat is supplied at the node A, a

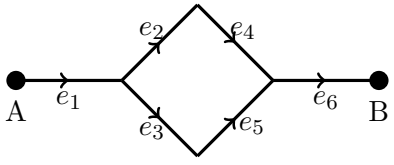

Figure 2. Graph of a small supply network.

single consumer is located at B. The lengths of the pipes are $L_1 = L_6 = 1$ and $L_2 = L_3 = L_4 = L_5 = 0.5$. The velocities along the edges do not change

over time but have the following different values $v_1 = v_6 = 1$, $v_2 = v_4 = \frac{1}{3}$ and $v_3 = v_5 = \frac{2}{3}$. Furthermore no external temperature is considered.

The initial temperature is zero in all pipes, the inflow profile at node A is given by

$$T_A(t) = \begin{cases} \sin(\pi x)^4, & t < 1 \\ 0, & t \geq 1 \end{cases}. \tag{20}$$

The quality of the schemes is compared via the temperature at the final node $T_B(t)$, as this is a major quantity of interest when operating district heating systems.

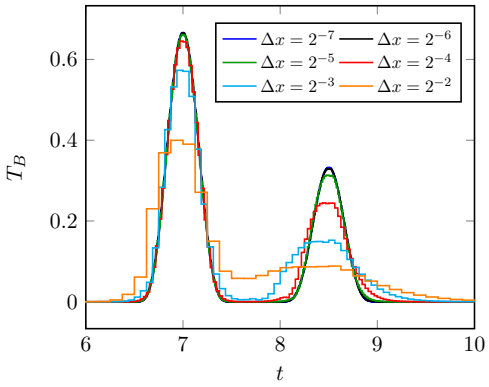

FIGURE 3. T_B for different spatial grid resolutions computed with the third order ADER scheme (left) and the third order local timestepping scheme (right).

In figure 3 the values of T_B in the time interval $[6, 10]$ are plotted. The picture on the left hand side shows the solution obtained with the third order ADER scheme for different spatial resolutions in all edges. For a coarse grid with only two or four cells per pipe the strong numerical diffusion lowers the peaks of the temperature significantly. As the spatial, and with it also the temporal, resolution is increased, the two signals separate and become more pronounced. The solutions seem to converge to a given profile, but only for the last two resolutions the curves in the picture can not be distinguished any more.

The values obtained with the local time stepping scheme are shown on the right hand side. Even for the coarsest resolution the two peaks are separated. Their heights are comparable with those of the ADER scheme on a grid with twice as many cells. Already for a resolution of $\Delta x = 2^{-4}$ the final shape is reached.

In figure 4 the numerical solutions with a fixed spatial grid with $\Delta x = 2^{-5}$ are shown. The curves in red are the ADER schemes of order one up to order 5, the blue curves display the solutions of the local time stepping scheme for an increasing order of accuracy. For both approaches the accuracy increases with the order of the method. The ADER schemes can capture the height of the first peak for order three or more, but for the second peak the full height is never reached. In contrast to this, the local time stepping scheme provides more accurate solutions. For all orders both peaks have the correct

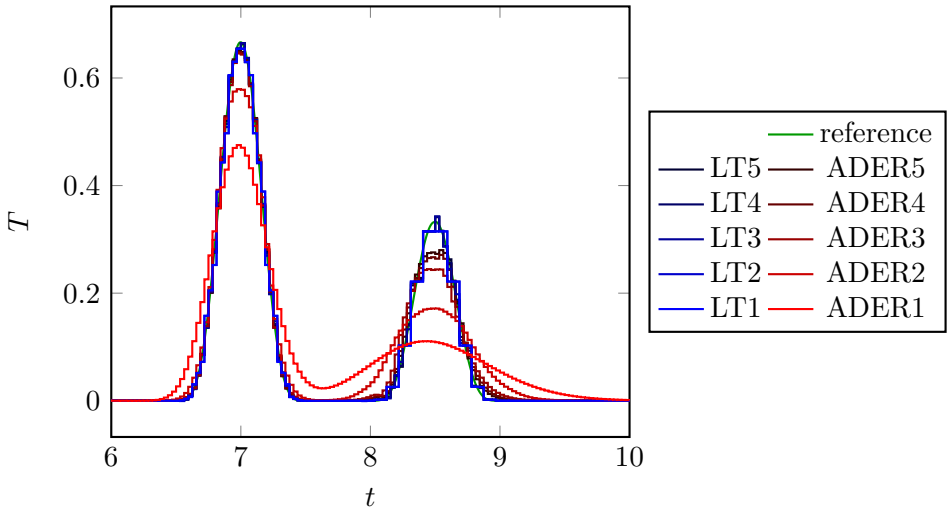

FIGURE 4. Schemes of different order on the same grid with $\Delta x = 2^{-5}$.

height, but the values of the second peak seem to be on a coarser grid. This is related to the coarse resolution and the varying velocities in the network. The signal T_A send into the e_1 is split at the first node into one peak traveling along e_2, e_4 and another one along e_3, e_5. Since the flow in edge two and four is significantly slower, the arriving peak is compressed into very few grid cells. This loss of resolution can not be recuperated when the flows merge again, nor excluded by the present high order approach. Thus the coarser signal is transferred to the faster flow in e_6.

The tables 1 and 2 show the L^1-errors, rates of convergence and computational times for the ADER and the local time stepping schemes. The L^1 errors are computed in comparison with a numerical reference solution obtained with the fifth order ADER scheme on a grid with $\Delta x = 2^{-10}$. Note that the fifth order ADER scheme can not be used on the coarsest grid, as there are not sufficient cells for the polynomial reconstruction available.

For all schemes the solution converge to the reference solution when the spatial grid is refined. With the ADER methods this is paid by quadratically increasing computational costs. For the local time stepping schemes this increase is only linear. Here the numerical effort is almost independent of the number of cells and just scales with the increase of the number of time steps.

The orders of convergence are visualized in figure 5. For fine enough grids all schemes achieve the predicted order of convergence. Except for the fifth order schemes, the local time stepping schemes are about one order of magnitude more accurate compared to the ADER methods.

In figure 6 the efficiency of the schemes is studied. The local time stepping scheme needs significantly less time to obtain accurate numerical results, e.g. the fifth order local time stepping scheme on the finest grid with $\Delta x = 2^{-9}$ takes as long as the first order upwind scheme with $\Delta x = 2^{-5}$, but the error is 3.266e-9 in contrast to 2.104e-1.

	ADER1			ADER3		
Δx	L^1 error	order	time	L^1 error	order	time
1.25e-1	3.477e-1	-	0.1	1.713e-1	-	0.8
6.25e-2	2.840e-1	0.29	0.2	8.495e-2	1.01	1.0
3.13e-2	2.104e-1	0.43	0.6	2.791e-2	1.61	3.8
1.56e-2	1.451e-1	0.54	1.6	3.577e-3	2.96	14.5
7.81e-3	9.276e-2	0.64	5.4	3.818e-4	3.23	55.9
3.91e-3	5.446e-2	0.77	19.7	4.226e-5	3.18	221
1.95e-3	2.890e-2	0.91	74.6	4.460e-6	3.24	880

	ADER5		
Δx	L^1 error	order	time
1.25e-1	-	-	0.9
6.25e-2	2.413e-2	-	2.4
3.13e-2	7.892e-4	4.93	6.0
1.56e-2	2.728e-5	4.85	22.8
7.81e-3	8.280e-7	5.04	90.0
3.91e-3	2.580e-8	5.00	355
1.95e-3	9.715e-10	4.73	1411

TABLE 1. L^1-errors, rates of convergence and computational times for the ADER schemes.

	LT1			LT3		
Δx	L^1 error	order	time	L^1 error	order	time
1.25e-1	1.157e-1	-	0.1	1.035e-1	-	0.1
6.25e-2	5.508e-2	1.07	0.2	2.556e-2	2.02	0.2
3.13e-2	2.507e-2	1.14	0.4	3.996e-3	2.68	0.4
1.56e-2	1.202e-2	1.06	0.8	5.269e-4	2.92	0.7
7.81e-3	5.877e-3	1.04	1.4	6.669e-5	2.98	1.5
3.91e-3	2.876e-3	1.08	2.7	8.328e-6	3.00	3.4
1.95e-3	1.363e-3	1.33	5.9	9.988e-7	3.06	6.7

	LT5		
Δx	L^1 error	order	time
1.25e-1	1.133e-1	-	0.1
6.25e-2	3.612e-2	1.65	0.2
3.13e-2	2.260e-3	4.00	0.4
1.56e-2	9.352e-5	4.59	0.8
7.81e-3	3.227e-6	4.86	1.6
3.91e-3	1.042e-7	4.95	3.2
1.95e-3	3.266e-9	5.00	7.0

TABLE 2. L^1-errors, rates of convergence and computational times for the local time stepping schemes.

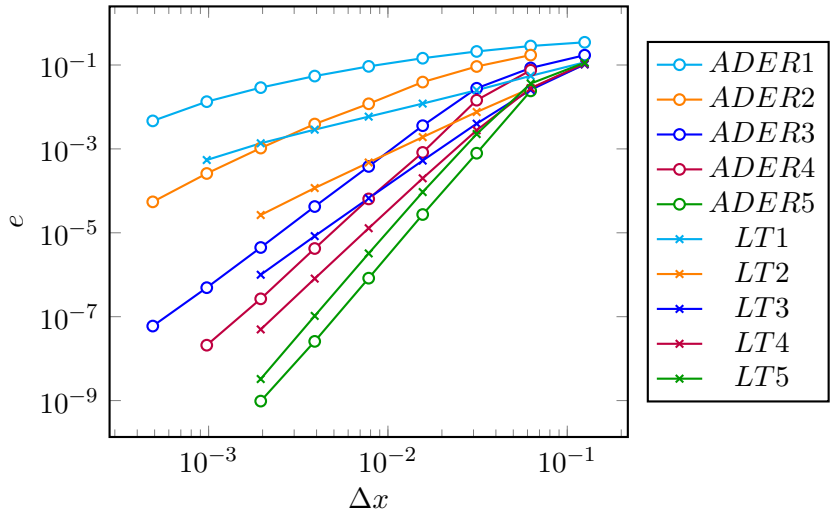

FIGURE 5. Convergence plots for the different schemes.

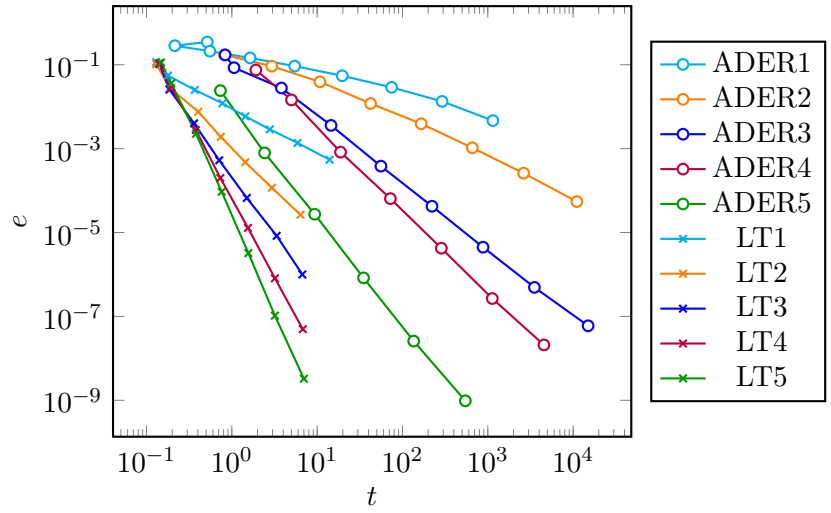

FIGURE 6. Efficiency plots for the different schemes.

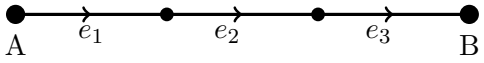

FIGURE 7. Three pipes in a row with slightly varying velocities.

4.2. **Slightly varying velocities.** In this simple test case three pipes in a line are considered, as depicted in figure 7. The pipes are all of the same length 1, but there is a small variation in the width of the tubes. The diameters are $d_1 = d_3 = 10.00$ and $d_2 = 10.01$. Thus the velocities resulting from the constant flow $10^2\pi$ are given by $v_1 = v_3 = 1$ and $v_2 = \frac{1}{1.001}$, i.e. the small variation in the width directly transfers to the velocities. The temperature of the inflow is again given by (20).

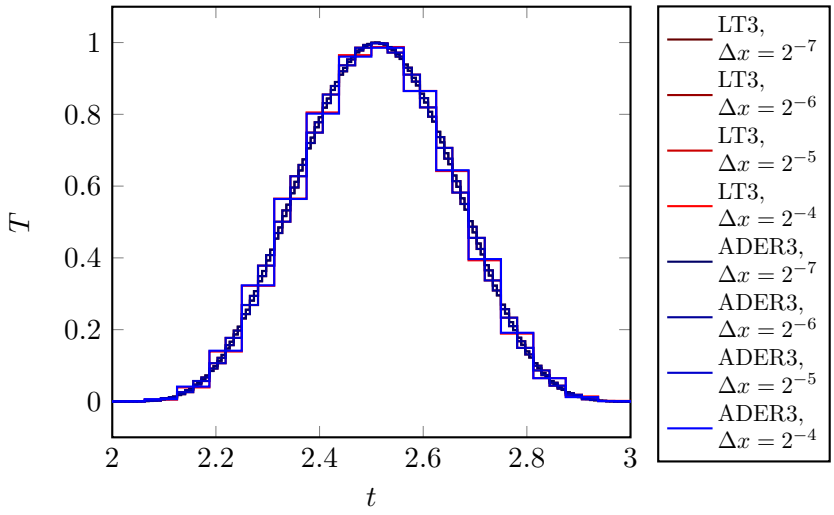

FIGURE 8. Temperature in node B computed with third order schemes for different grid sizes.

In figure 8 the numerical results for the temperature at the outflow of the third pipe is shown. As expected all numerical schemes provide accurate approximations. For this setting, there is no notable advantage for the local time stepping in terms of the accuracy. This is also confirmed by the errors given in table 3. For the fifth order schemes the errors of the local time stepping method is slightly bigger due to the larger Δts.

	ADER3		LT3	
Δx	l_1 error	time	l_1 error	time
6.25e-2	3.62e-04	0.3	1.39e-04	0.1
3.13e-2	2.45e-05	0.9	8.54e-06	0.3
1.56e-2	1.82e-06	3.5	4.37e-07	0.6
7.81e-3	1.35e-07	13	2.80e-08	1.2
3.91e-3	8.48e-09	48	2.66e-09	2.6
1.95e-3	3.69e-10	194	2.34e-10	5.5

TABLE 3. L^1 errors and computational times of the third order schemes for three pipes in a row.

But the computational costs, listed in table 3, are significantly smaller when the local time stepping is applied. This is achieved by avoiding numerical computations along the pipes due to the exact CFL condition. Note that the computational savings considered in [11] are only due to the reduced number of time steps, which are negligible in the present test case. Here the linearity of the considered equation allows to further exploit the advantages of the exact time steps.

4.3. **A district heating network.** In the final test case we study the numerical algorithms on a more realistic setting for a district heating network. The layout of the system is shown in figure 9. The supply branch (red)

and the return branch (blue) consist of nine edges each. Each pipe has a diameter 0.1 and the lengths are $L_1 = L_{18} = L_6 = L_{13} = 500$, $L_2 = L_{17} = L_5 = L_{14} = L_7 = L_{12} = L_9 = L_{10} = 282$, $L_3 = L_{16} = L_8 = L_{11} = 400$ and $L_4 = L15 = 424$.

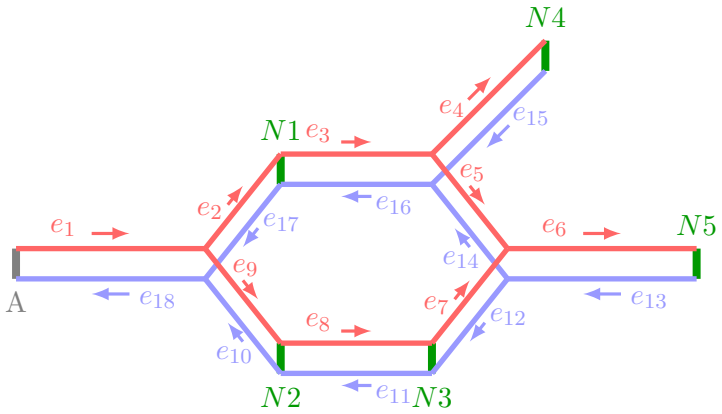

FIGURE 9. Graph of a district heating network.

The five green vertical lines indicate the locations of the consumers. Each of them has a given energy demand $Q_1(t) = Q_2(t) = Q_3(t) = 3000$, $Q_4(t) = 6000$ and $Q_5(t) = 9000$, which enters into the mathematical model via equation (9). The thermal energy is provided by the boundary conditions for e_1 at the node A by

$$T_A(t) = \begin{cases} 120 & t < 3600 \\ 80 & t > 3600 \end{cases}.$$

At the same location the pressure of the supply branch is $p_1(t) = 3$ and $p_{18}(t) = 2$. In the houses thermal energy is consumed such that the temperature is cooled down to $T_{out} = 60$. The external cooling the parameters are $k = 0.1$, $\lambda = 0.006$ and $T_inf = 20$.

In the numerical integrations spatial discretizations from $\Delta x = 40$ up to $\Delta x = 2.5$ are considered. The cell along the edges are equally spaced, i.e. Δx is slightly decreased if the length of the pipe is not a multiple of the given value. This leads to 8 up to 13 cells for the coarsest grids and 118 up to 200 cells on the fines level. The simulation is run up to the time $t_{end} = 36000$.

In figure 10 the evolution of the temperature in the node $N1$ and $N4$ are shown. As no flows mix, a part of the input signal passes almost unchanged. Just the slight cooling due to the outer temperature can be observed in $N4$. All numerical schemes give accurate results.

More complex is the behaviour of the temperature at node $N5$. In figure 11 the results for the local time stepping scheme are compared to those of the ADER methods on a grid with $\Delta x = 20$. The temperature varies over time, since it is a mixture of waves passing the upper branch e_2, e_3 and e_5 with those of the lower branch e_9, e_8 and e_7. The first order ADER scheme is not able to resolve these features. For the local time stepping method even the first order is more accurate than the fifth order ADER scheme.

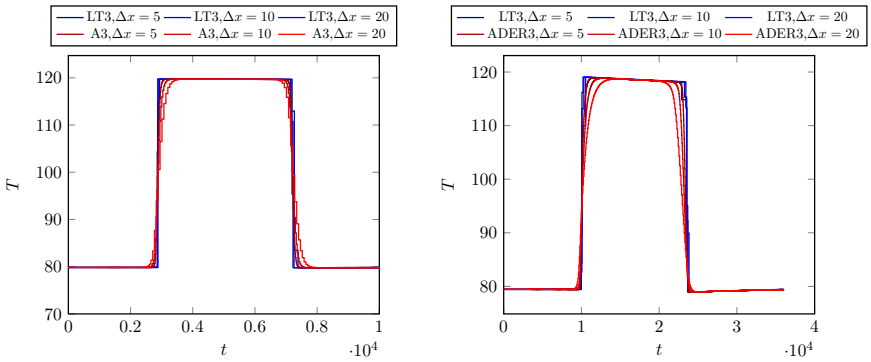

FIGURE 10. Temperature at the node $N1$ (left) and $N4$ (right).

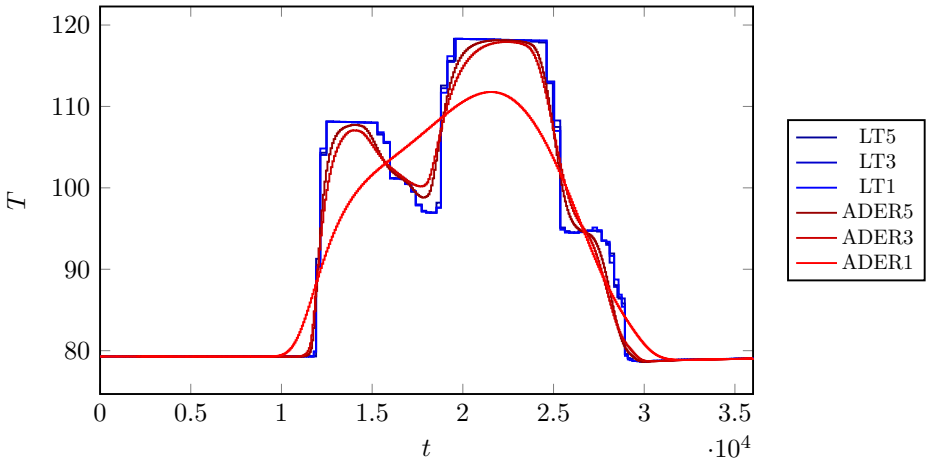

FIGURE 11. Temperature in $N5$ computed with schemes of different orders with $\Delta x = 20$.

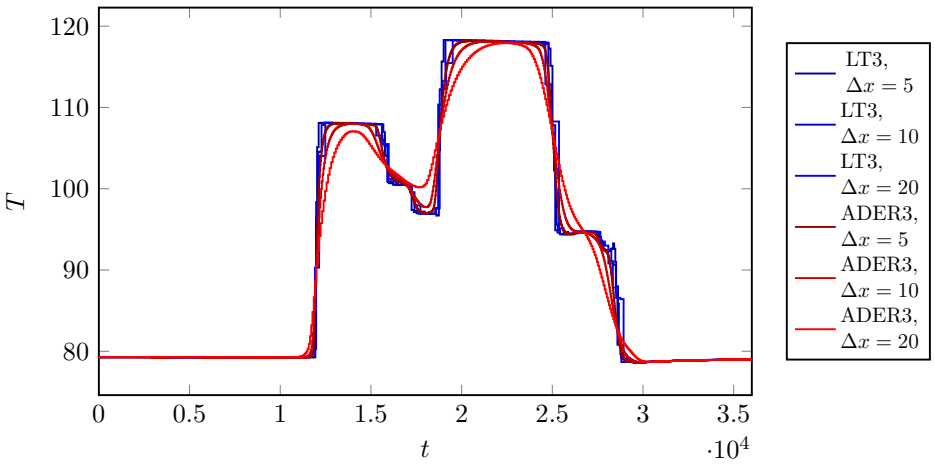

FIGURE 12. Temperature in $N5$ computed with schemes of order 3 for different spatial resolutions.

	ADER1		ADER3		ADER5	
Δx	L^1 error	time	L^1 error	time	L^1 error	time
40	4.76	4	2.80	6	-	-
20	4.01	9	1.75	16	1.28	20
10	3.26	19	0.98	45	0.66	57
5	2.54	44	0.52	126	0.32	185
2.5	1.88	109	0.26	446	0.14	664

TABLE 4. L^1-errors and computational times for the ADER schemes.

	LT1		LT3		LT5	
Δx	L^1 error	time	L^1 error	time	L^1 error	time
40	1.17	7	1.15	7	1.16	8
20	0.61	13	0.65	13	0.63	14
10	0.30	24	0.30	25	0.28	28
5	0.19	47	0.18	48	0.17	54
2.5	0.11	96	0.10	97	0.09	118

TABLE 5. L^1-errors and computational times for the LT schemes.

Figure 11 shows that all methods converge to the same solution as the grid is refined.

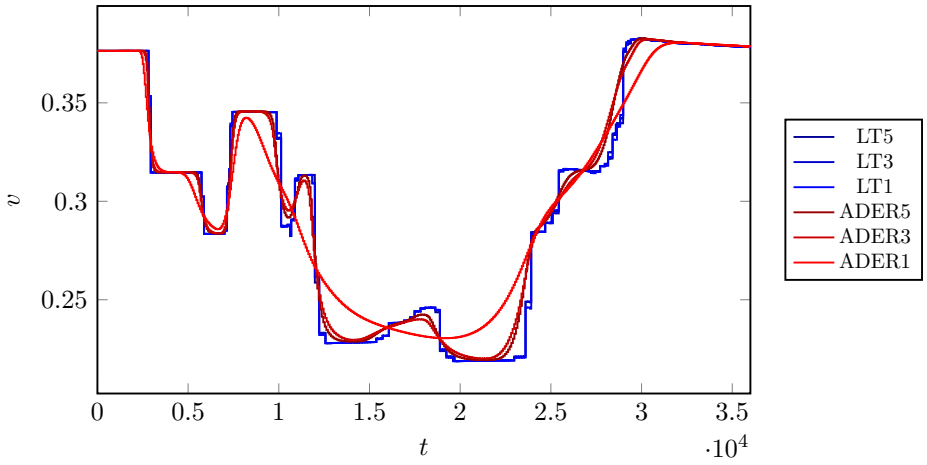

FIGURE 13. The velocity on edge e_1 for $\Delta x = 20$

Note that with differing values in T also influence the flow solver, i.e. also the velocity profile changes. In figure 13 the velocity on edge e_1 is plotted. The values fluctuate over time as the inflow has to adjust according to the needs of all consumers simultaneously. Similar to the temperature, the first order ADER scheme can not capture all features of the solution, while the local time stepping methods provide very accurate results.

In the tables 4 and 5 the L^1 errors and the computation times for the ADER and the local time stepping methods are shown. Here the reference

solution is computed with the local time stepping scheme of order five on a grid with $\Delta x = 0.625$. The errors are calculated after transforming the time interval to $[0, 1]$.

For all numerical schemes the errors decrease and the computation times increase when the grid is refined. The errors of the local time stepping schemes are below those of the fifth order ADER method. Further the results are obtained in much shorter time. Note that the gain in terms of computation time is not as big as in the previous examples. This is due to the flow solver, which is identical for all methods. It occupies about 30 seconds on the finest grid. These costs could be reduced by exploiting the structure of the network or considering stationary version of (4) as in [18].

5. Conclusion

In this work we constructed a numerical method for simulating the transport of thermal energy in district heating networks. By applying a local time stepping, the advection equations along the edges can be solved exactly. This increases the numerical accuracy and simultaneously reduces the computational costs, such that it is even more efficient than high order ADER schemes. The developed method is restricted to linear problems, but extensions to systems of linear equations, e.g. the wave equation, will be subject to future research.

Acknowledgements

This research was supported by the *Stiftung Rheinland-Pfalz für Innovation* with the project 961 − 386261/1172. The first author is supported by the Deutsche Forschungsgemeinschaft (DFG) grant BO 4768/1.

References

[1] Mapundi K. Banda, Michael Herty, and Axel Klar. Coupling conditions for gas networks governed by the isothermal Euler equations. *Netw. Heterog. Media*, 1(2):295–314, 2006.

[2] S. Blandin, D. Work, P. Goatin, B. Piccoli, and A. Bayen. A general phase transition model for vehicular traffic. *SIAM J. Appl. Math.*, 71(1):107–127, 2011.

[3] Raul Borsche. Numerical schemes for networks of hyperbolic conservation laws. *Appl. Numer. Math.*, 108:157–170, 2016.

[4] Raul Borsche and Jochen Kall. ADER schemes and high order coupling on networks of hyperbolic conservation laws. *J. Comput. Phys.*, 273:658–670, 2014.

[5] Raul Borsche and Jochen Kall. High order numerical methods for networks of hyperbolic conservation laws coupled with ODEs and lumped parameter models. *J. Comput. Phys.*, 327:678–699, 2016.

[6] Walter Boscheri, Raphaël Loubère, and Michael Dumbser. Direct arbitrary-lagrangian–eulerian ader-mood finite volume schemes for multidimensional hyperbolic conservation laws. *Journal of Computational Physics*, 292:56 – 87, 2015.

[7] G. Bretti, R. Natalini, and B. Piccoli. Numerical algorithms for simulations of a traffic model on road networks. *J. Comput. Appl. Math.*, 210(1-2):71–77, 2007.

[8] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (mood). *Journal of Computational Physics*, 230(10):4028 – 4050, 2011.

[9] R. M. Colombo and M. Garavello. On the Cauchy problem for the p-system at a junction. *SIAM J. Math. Anal.*, 39(5):1456–1471, 2008.

[10] Pia Domschke, Oliver Kolb, and Jens Lang. Adjoint-based error control for the simulation and optimization of gas and water supply networks. *Appl. Math. Comput.*, 259:1003–1018, 2015.

[11] Michael Dumbser, Martin Käser, and Eleuterio F. Toro. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – V. Local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.

[12] S. Göttlich, M. Herty, and A. Klar. Network models for supply chains. *Commun. Math. Sci.*, 3(4):545–559, 2005.

[13] Michael Herty and Axel Klar. Modeling, simulation, and optimization of traffic flow networks. *SIAM Journal on Scientific Computing*, 25(3):1066–1087, 2003.

[14] Michael Hinze, Ralf Köcher, and Jens Pfafferott. Zur numerischen simulation von wärmenetzen. *Forschung im Ingenieurwesen*, 62(11):301–314, Nov 1996.

[15] Lennart Jansen and Jonas Pade. Global unique solvability for a quasi-stationary water network model. *Preprint series: Institut für Mathematik, Humboldt-Universität zu Berlin*, 2013-11, 2013.

[16] G. Jiang and Chi-wang Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126(1):202–228, jun 1996.

[17] Jochen Kall. *ADER Schemes for Systems of Conservation Laws on Networks*. PhD thesis, Technische Universität Kaiserslautern, 2016. Verlag Dr. Hut.

[18] Ralf Köcher. *Beitrag zur Berechnung und Auslegung von Fernwärmenetzen*. PhD thesis, Technische Universität Berlin, 2000. http://dx.doi.org/10.14279/depositonce-180.

[19] T. Köppl, B. Wohlmuth, and R. Helmig. Reduced one-dimensional modelling and numerical simulation for mass transport in fluids. *Internat. J. Numer. Methods Fluids*, 72(2):135–156, 2013.

[20] Randall J LeVeque. *Finite-Volume Methods for Hyperbolic Problems*, volume 31. Cambridge university press, 2002.

[21] Andrew J. Majda and Andrea L. Bertozzi. *Vorticity and incompressible flow*, volume 27 of *Cambridge Texts in Applied Mathematics*. Cambridge University Press, Cambridge, 2002.

[22] Lucas O Müller and Pablo J Blanco. A high order approximation of hyperbolic conservation laws in networks: Application to one-dimensional blood flow. *Journal of Computational Physics*, 300:423–437, 2015.

[23] Sirui Tan and Chi-Wang Shu. Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. *Journal of Computational Physics*, 229(21):8144–8166, 2010.

[24] E. F. Toro, R. C. Millington, and L. A. M. Nejad. Towards very high order Godunov schemes. In *Godunov methods (Oxford, 1999)*, pages 907–940. Kluwer/Plenum, New York, 2001.

[25] Eleuterio F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 2009.