A Real-Time Capable PDE Model for an Industrial Heating Process

Ruven Weiss * **Moritz Diehl** ** **Johannes Reuter** *

* *Institute of System Dynamics, Hochschule Konstanz (e-mail: (rweiss, jreuter)@htwg-konstanz.de).*
** *Department of Microsystems Engineering, Universität Freiburg, (e-mail: moritz.diehl@imtek.uni-freiburg.de)*

Abstract: This paper presents a modeling approach of an industrial heating process where a stripe-shaped workpiece is heated up to a specific temperature by applying hot air through a nozzle. The workpiece is moving through the heating zone and is considered to be of infinite length. The speed of the substrate is varying over time. The derived model is supposed to be computationally cheap to enable its use in a model-based control setting. We start by formulating the governing PDE and the corresponding boundary conditions. The PDE is then discretized on a spatial grid using finite differences and two different integration schemes, explicit and implicit are derived. The two models are evaluated in terms of computational effort and accuracy. It turns out that the implicit approach is favorable for the regarded process. We optimize the grid of the model to achieve a low number of grid nodes while maintaining a sufficient amount of accuracy. Finally, the thermodynamical parameters are optimized in order to fit the model's output to real-world data that was obtained by experiments.

Keywords: PDE Systems, Fluidics and Thermodynamics, Control and Real-Time Applications

1. INTRODUCTION

There are plenty of applications in process engineering where some material is heated up to a specific temperature in order to obtain desired properties of the treated workpiece. In this paper a heating process is modeled, where hot air is applied through a nozzle to the surface of a moving substrate to increase its temperature up to the desired level. The substrate is considered to be of infinite length. The main focus is on deriving a model that is accurate on the one hand but also computationally cheap on the other hand, which is critical for being applicable in a real-time setting such as model-based temperature control. There exist a variety control-oriented approaches to model such or similar processes in different industrial applications. Especially in steel industry such examples can be found. However, in contrast to the process considered in this paper, most of these applications assume a different kind of work-piece transportation as in Steinboeck et al. (2013), where discontinuous movement in a pusher type reheating furnace is considered, or do not model heat conduction along the thickness of the heated material as in Strommer et al. (2018), where a strip annealing furnace is treated. The highest similarity can be found in continuous steel casting where both heat conduction inside the workpiece and time-varying processing speed are considered (Ivanova, 2013), (Wang et al., 2019).

In this paper, a stripe-shaped plastic substrate is heated, which is comparably thin (1 mm to 2 mm) and the velocity changes very dynamically, ranging from 0 m/min to 15 m/min. For the regarded process, it is crucial, that the surface temperature of the heated substrate has reached a certain value at a fixed location after passing the heating zone. Therefore, our aim is to develop a model that outputs this temperature given the trajectories of the volumetric flow rate of the heated air $\dot{V}(t)$ and the speed of the substrate $v(t)$.

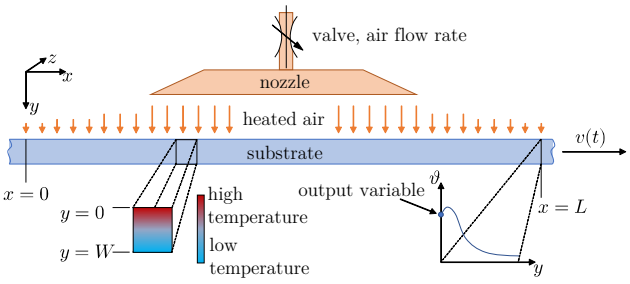

Fig. 1. Process sketch and coordinate system

Previous work on control-oriented modeling of this kind of process was already done in Weiss et al. (2018) and Weiss et al. (2021). In the former, the heat conduction inside the substrate was modeled by a reduced-order thermal-electrical analogy and the heat flux into the substrate was assumed to be an affine function of the volumetric air flow rate $\dot{V}(t)$. The transportation with variable speed was encountered by tracking multiple points on the substrate on their way through the heating zone separated by the distance that the substrate traveled in a fixed time step Δt. This resulted in a variable number of tracked points according to the speed of the substrate. In Weiss et al. (2021) the thermal-electrical analogy model obtained in Weiss et al. (2018) was also used to model the heat conduction inside the substrate but the modeling of the heat transfer was improved by considering the actual

temperature difference between the air and the substrate's surface temperature. Moreover, the substrate was divided in spatially fixed zones of constant width Δx to overcome the requirement to track a variable number of points on the substrate. The time-varying speed was incorporated by introducing a shifting scheme where the state vector of each zone was shifted one zone further each time the substrate has traveled the distance Δx.

In this paper we pursue a different approach. Instead of using a thermal-electrical analogy and order reduction for modeling the heat conduction we will stick with the governing Partial Differential Equation (PDE) and incorporate the time-varying speed by adding a transportation term to the PDE (1). We will then derive an explicit and an implicit integration scheme for the PDE and optimize the resulting model in terms of computational effort and accuracy. In contrast to the shifting-approach in Weiss et al. (2021), this allows for more flexibility in terms of sampling time, and it better represents the true nature of movement, which is continuous.

2. THERMAL MODELING

A sketch of the considered process as well as the coordinate system used throughout this paper is depicted in Fig. 1. We start by regarding the temperature distribution inside the substrate. Previous experiments have shown that the heat is predominately conducted in y-direction (into the substrate), whereas heat conduction in the x-direction (transportation direction) as well as in z-direction plays a minor role only and can be neglected (Weiss et al., 2018). The governing PDE describing the dynamics of the temperature field is thus given by

$$\frac{\partial \vartheta(x,y,t)}{\partial t} = a\frac{\partial^2 \vartheta(x,y,t)}{\partial y^2} - v(t)\frac{\partial \vartheta(x,y,t)}{\partial x} \tag{1}$$

with

$$a = \frac{\lambda}{c_p \cdot \rho}$$

where λ is the substrate's thermal conductivity, c_p its specific heat capacity and ρ its density. We consider convective heat transfer. Thus, the boundary conditions are given as

$$\lambda \cdot \frac{\partial}{\partial y}\vartheta(x,0,t) = -\dot{q}(x,0,t) \tag{2a}$$

$$\lambda \cdot \frac{\partial}{\partial y}\vartheta(x,W,t) = \dot{q}(x,W,t) \tag{2b}$$

with

$$\dot{q}(x,0,t) = \alpha_{\text{ref}}(x,\dot{V}(t)) \cdot \Big(\vartheta_{\text{ref}}(x,\dot{V}(t)) - \vartheta(x,0,t)\Big) \tag{3a}$$

$$\dot{q}(x,W,t) = \alpha_{\text{amb}} \cdot (\vartheta_{\text{amb}} - \vartheta(x,W,t)) \tag{3b}$$

where $\vartheta_{\text{ref}}(x,\dot{V}(t))$ and $\alpha_{\text{ref}}(x,\dot{V}(t))$ denote the air temperature and heat transfer coefficient on the heated side at $y = 0$. On the backside, at $y = W$, they are denoted ϑ_{amb} and α_{amb}. Note, both, the heat transfer coefficient α_{ref} as well as the air temperature ϑ_{ref} depend on the volumetric air flow rate $\dot{V}(t)$, which changes over time. They also differ along the spatial extent x according to the shape of the flow field. We account for this by using CFD-generated lookup tables, depicted in Fig. 2. On the backside, at $y = W$, we assume α_{amb} and ϑ_{amb} to be constant both

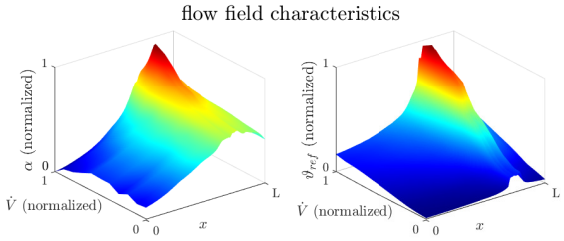

Fig. 2. Lookup tables for α_{ref} and ϑ_{ref} generated by CFD-simulation.

over time and along x. On the left boundary at $x = 0$, we assume ϑ_{start} to be constant, which corresponds to the entry temperature of the substrate, which can easily be measured in practice. For better readability the arguments of $\vartheta(x,y,t)$ are omitted in the sequel.

3. PDE DISCRETIZATION

In order to simulate the system on a computer, we need to discretize the PDE (1) on a spatial grid and in time. We use a rectangular grid and finite differences to approximate a solution of the PDE.

3.1 Grid

The temperature field is discretized using a grid as depicted in Fig. 3. Index i is used for discretizing the y-coordinate, whereas j is used for the x-coordinate. The grid size is chosen to be equidistant in x-direction whereas in y-direction the cells of the grid grow with a growth rate r. It was shown in Weiss et al. (2018) that the expanding grid in y-direction is beneficial as it yields higher approximation accuracy on the heated boundary of the workpiece. This is intuitive since the temperature changes more rapidly on the heated side than on the backside.

3.2 Derivative Approximations

The first order time derivative on the left hand side of (1) is approximated by a simple Euler scheme given by

$$\left.\frac{\partial \vartheta}{\partial t}\right|_{i,j}^{k} \approx \frac{\vartheta_{i,j}^{k+1} - \vartheta_{i,j}^{k}}{\Delta t}, \tag{4}$$

where Δt is the time between two subsequent time instants k and $k+1$. The left-hand side in (4) denotes the partial

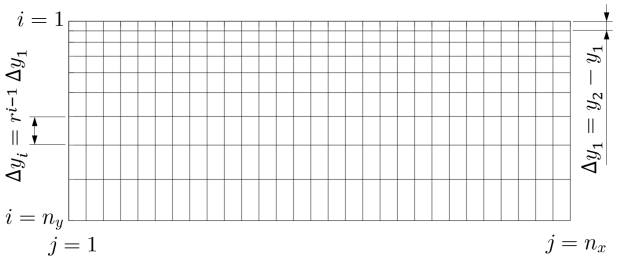

Fig. 3. Grid used for discretizing the PDE. Grid size is increased in y-direction and equidistant in x-direction. The parameters used for this plot are $n_x = 20$, $n_y = 10$, $r = 1.2$.

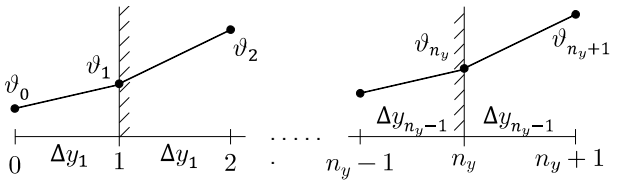

Fig. 4. Grid boundary at $y = 0/i = 1$ and $y = W/i = n_y$

derivative of the temperature w.r.t. time t, evaluated at grid point (i, j) at time instant k. The expression with second derivative in (1) is approximated at a given grid point (i, j) using a central difference scheme:

$$a \left.\frac{\partial^2 \vartheta}{\partial y^2}\right|_{i,j}^{k} \approx a \frac{\left.\frac{\partial \vartheta}{\partial y}\right|_{i+\frac{1}{2},j}^{k} - \left.\frac{\partial \vartheta}{\partial y}\right|_{i-\frac{1}{2},j}^{k}}{\frac{1}{2}\left(\Delta y_{i-1} + \Delta y_i\right)} \tag{5}$$

Using approximations for the first order derivatives in the numerator according to

$$\left.\frac{\partial \vartheta}{\partial y}\right|_{i+\frac{1}{2},j}^{k} \approx \frac{\vartheta_{i+1,j}^{k} - \vartheta_{i,j}^{k}}{\Delta y_i} \tag{6}$$

$$\left.\frac{\partial \vartheta}{\partial y}\right|_{i-\frac{1}{2},j}^{k} \approx \frac{\vartheta_{i,j}^{k} - \vartheta_{i-1,j}^{k}}{\Delta y_{i-1}} \tag{7}$$

and further using the relationship of the expanding grid (Fig. 3)

$$\Delta y_i = r^{i-1} \Delta y_1 \tag{8}$$

finally yields the approximation for the second order derivative:

$$a \left.\frac{\partial^2 \vartheta}{\partial y^2}\right|_{i,j}^{k} \approx a \frac{r\vartheta_{i-1,j}^{k} - (1+r)\vartheta_{i,j}^{k} + \vartheta_{i+1,j}^{k}}{\frac{1}{2} r^{2i-3}(1+r)\Delta y_1^2} \tag{9}$$

The transportation term in (1) is discretized using an upwind scheme as described for example in Ferziger et al. (2008). Considering the fixed transportation direction of the edge band (in positive x-direction) the upwind discretization is given by

$$v(t) \left.\frac{\partial \vartheta(x,y,t)}{\partial x}\right|_{i,j}^{k} \approx v_k \frac{\vartheta_{i,j}^{k} - \vartheta_{i,j-1}^{k}}{\Delta x}, \tag{10}$$

where v_k denotes the speed at time instant k and Δx is the distance between two grid points along the x-axis. Putting it all together, we obtain an explicit expression for the temperature at grid point i, j at time instant $k+1$ given the temperatures at time instant k as

$$\begin{aligned}\vartheta_{i,j}^{k+1} = {} & 2M_i r \vartheta_{i-1,j} + \left(1 - 2M_i(1+r) - v_k \frac{\Delta t}{\Delta x}\right)\vartheta_{i,j}^{k} \\ & + 2M_i \vartheta_{i+1,j}^{k} + v_k \frac{\Delta t}{\Delta x} \vartheta_{i,j-1}^{k},\end{aligned} \tag{11}$$

with

$$M_i = \frac{a \Delta t}{r^{(2i-3)}(1+r)\Delta y_1^2}.$$

At the upper and lower boundaries, the grid size is not further expanded outside the substrate (Fig. 4). Following Baehr and Stephan (2013), discretizing the boundary conditions (2) and (3) yields

$$\vartheta_{0,j}^{k} = \vartheta_{2,j}^{k} - 2\mathrm{Bi}_{1,j}^{k}\left(\vartheta_{1,j}^{k} - \vartheta_{\mathrm{ref},j}^{k}(x, \dot{V}^k)\right) \tag{12a}$$

$$\vartheta_{n_y+1,j}^{k} = \vartheta_{n_y-1}^{k} - 2\mathrm{Bi}_{n_y}\left(\vartheta_{n_y,j}^{k} - \vartheta_{\mathrm{amb}}\right) \tag{12b}$$

with Bi_1^k and Bi_{n_y} being the Biot-number at time instant k for the upper and lower boundary respectively:

$$\mathrm{Bi}_{1,j}^{k} = \alpha_{\mathrm{ref},j}^{k}(x, \dot{V}^k)\frac{\Delta y_1}{\lambda} \tag{13a}$$

$$\mathrm{Bi}_{n_y} = \alpha_{\mathrm{amb}} \frac{\Delta y_{n_y-1}}{\lambda} \tag{13b}$$

As already outlined in section 2, the time variant quantities α_{ref} and ϑ_{ref} depend on x and are therefore indexed with j, whereas on the backside of the substrate, ϑ_{amb} and α_{amb} are assumed to be constant. Plugging (12) into (11) yields

$$\begin{aligned}\vartheta_{1,j}^{k+1} = {} & \left(1 - 2M_1(1+r) - v_k\frac{\Delta t}{\Delta x} - 4rM_1\mathrm{Bi}_{1,j}^{k}\right)\vartheta_{1,j}^{k} \\ & + 2M_1(1+r)\vartheta_{2,j}^{k} + v_k\frac{\Delta t}{\Delta x}\vartheta_{1,j-1}^{k} \\ & + 4rM_1\mathrm{Bi}_{1,j}^{k}\vartheta_{\mathrm{ref},j}^{k}\end{aligned} \tag{14}$$

for the upper boundary (at $y = 0/i = 1$) and

$$\begin{aligned}\vartheta_{n_y,j}^{k+1} = {} & 2M_{n_y}(1+r)\vartheta_{n_y-1,j}^{k} \\ & + \left(1 - 4M_{n_y}\mathrm{Bi}_{n_y} - 2M_{n_y}(1+r) - v_k\frac{\Delta t}{\Delta x}\right)\vartheta_{n_y,j}^{k} \\ & + v_k\frac{\Delta t}{\Delta x}\vartheta_{n_y,j-1}^{k} + 4M_{n_y}\mathrm{Bi}_{n_y}\vartheta_{\mathrm{amb}}\end{aligned} \tag{15}$$

for the lower boundary (at $y = W/i = n_y$) of the substrate. This yields the following equation for propagating the temperature field from time instant k to $k+1$:

$$\xi^{k+1} = \mathbf{A}_{\mathrm{ex}}^{k}\xi^{k} + \mathbf{B}_{\mathrm{ex}}^{k}\vartheta_{\mathrm{ref}}^{k} + \mathbf{C}_{\mathrm{ex}}\vartheta_{\mathrm{amb}} + \mathbf{D}_{\mathrm{ex}}\vartheta_{\mathrm{start}} \tag{16}$$

where

$$\begin{aligned}&\mathbf{A}_{\mathrm{ex}}^{k} \in \mathbb{R}^{n_y n_x \times n_y n_x}, && \xi^{k} \in \mathbb{R}^{n_y n_x \times 1} \\ &\mathbf{B}_{\mathrm{ex}}^{k} \in \mathbb{R}^{n_y n_x \times n_x}, && \vartheta_{\mathrm{ref}} \in \mathbb{R}^{n_x \times 1} \\ &\mathbf{C}_{\mathrm{ex}} \in \mathbb{R}^{n_y n_x \times 1}, && \vartheta_{\mathrm{amb}} \in \mathbb{R} \\ &\mathbf{D}_{\mathrm{ex}} \in \mathbb{R}^{n_y n_x \times 1}, && \vartheta_{\mathrm{start}} \in \mathbb{R}\end{aligned}$$

The index "ex" highlights, that these are matrices of the explicit formulation in contrast to the implicit one derived in section 3.3. ξ^k is referred to as the state vector at time instant k and contains all temperatures of the field defined by the grid:

$$\xi^{k} = \begin{pmatrix} \xi_1^k \\ \vdots \\ \xi_j^k \\ \vdots \\ \xi_{n_x}^k \end{pmatrix} \quad \text{with } \xi_j^k = \begin{pmatrix} \vartheta_{1,j}^k \\ \vartheta_{2,j}^k \\ \vdots \\ \vartheta_{n_y,j}^k \end{pmatrix} \tag{17}$$

The resulting matrices have sparsity patterns that are depicted in Fig. 5. In matrix $\mathbf{A}_{\mathrm{ex}}$ each of the indicated blocks corresponds to one column of the grid, i.e. one particular ξ_j. On the top left of each block (green dot) we have:

$$1 - 4rM_1\mathrm{Bi}_{1,j}^{k} - 2M_1(1+r) - v_k\frac{\Delta t}{\Delta x}$$

Note, that these are different for each j because the Biot number (13) depends on $\alpha_{\mathrm{ref}}(x, \dot{V})$. The same holds for the value in the bottom right corner of each block (green cross) which is given by:

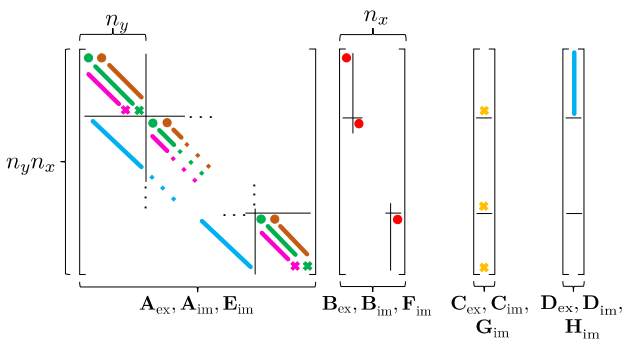

Fig. 5. Sparsity patterns of matrices. Only the non-zero elements are colored.

$$1 - 4M_{n_y}\text{Bi}_{n_y} - 2(1+r)M_{n_y} - v_k\frac{\Delta t}{\Delta x}$$

The main diagonal elements of $\mathbf{A}_{\text{ex}}$ (green) are given by:

$$1 - 2M_i(1+r) - v_k\frac{\Delta t}{\Delta x}.$$

The lower diagonal elements (pink) are given by $2rM_i$ and the last respective element of each block (pink crosses) by:

$$2M_{n_y}(1+r)$$

The upper diagonal elements (brown) are given by $2M_i$ and the first respective element of each block (brown dots) by:

$$2M_1(1+r)$$

The elements introduced by the upwind discretization of the transportation term in (1) (light blue) are given by

$$v_k\frac{\Delta t}{\Delta x}$$

The non-zero elements in $\mathbf{B}_{\text{ex}}$ (red dots) and $\mathbf{C}_{\text{ex}}$ (yellow crosses) are given by $4rM_1\text{Bi}_{1,j}^k$ and $4M_{n_y}\text{Bi}_{n_y}$ respectively. The computational effort can tremendously be reduced using highly efficient sparcity exploiting libraries.

3.3 Implicit formulation via Crank-Nicolson

To improve numerical stability and accuracy, an implicit Crank-Nicolson time stepping scheme is derived in the sequel for the discretized system (Baehr and Stephan, 2013), (Ferziger et al., 2008). The derivatives are approximated in between two time instants using the average of the derivatives at time instant k and $k+1$. For the second order derivative in (1) this yields:

$$\left.\frac{\partial^2\vartheta}{\partial y^2}\right|_{i,j}^{k+\frac{1}{2}} \approx \frac{1}{2}\left(\left.\frac{\partial^2\vartheta}{\partial y^2}\right|_{i,j}^{k} + \left.\frac{\partial^2\vartheta}{\partial y^2}\right|_{i,j}^{k+1}\right) \tag{18}$$

The same is applied to the transport term in (1):

$$\left.\frac{\partial\vartheta(x,y,t)}{\partial x}\right|_{i,j}^{k+\frac{1}{2}} \approx \frac{1}{2}\left(\left.\frac{\partial\vartheta}{\partial x}\right|_{i,j}^{k} + \left.\frac{\partial\vartheta}{\partial x}\right|_{i,j}^{k+1}\right) \tag{19}$$

Plugging (9) into (18) and (10) into (19) yields an implicit expression for the temperature at grid point i,j at time instant $k+1$. The system is now given by:

$$\begin{aligned}&\mathbf{E}_{\text{im}}^{k+1}\xi^{k+1} + \mathbf{F}_{\text{im}}^{k+1}\vartheta_{\text{ref}}^{k+1} + \mathbf{G}_{\text{im}}\vartheta_{\text{amb}} + \mathbf{H}_{\text{im}}\vartheta_{\text{start}}\xi^{k+1}\\ &= \mathbf{A}_{\text{im}}^{k}\xi^{k} + \mathbf{B}_{\text{im}}^{k}\vartheta_{\text{ref}}^{k} + \mathbf{C}_{\text{im}}\vartheta_{\text{amb}} + \mathbf{D}_{\text{im}}\vartheta_{\text{start}}\end{aligned} \tag{20}$$

The structure of the matrices is depicted in Figure 5. The main diagonal elements of $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (green) are given by

$$1 - M_i(1+r) - v_k\frac{\Delta t}{2\Delta x}$$

for $\mathbf{A}_{\text{im}}$ and

$$1 + M_i(1+r) + v_k\frac{\Delta t}{2\Delta x}$$

for $\mathbf{E}_{\text{im}}$ respectively. On the top left of each block in $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (green dot) we have

$$1 - M_1(1+r) - v_k\frac{\Delta t}{2\Delta x} - 2rM_1\text{Bi}_{1,j}^k$$

for $\mathbf{A}_{\text{im}}$ and

$$1 + M_1(1+r) + v_{k+1}\frac{\Delta t}{2\Delta x} + 2rM_1\text{Bi}_{1,j}^{k+1}$$

for $\mathbf{E}_{\text{im}}$. On the bottom right of each block in $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (green cross) we have

$$1 - M_{n_y}(1+r) - v_k\frac{\Delta t}{2\Delta x} - 2M_{n_y}\text{Bi}_{n_y}$$

for $\mathbf{A}_{\text{im}}$ and

$$1 + M_{n_y}(1+r) + v_{k+1}\frac{\Delta t}{2\Delta x} + 2M_{n_y}\text{Bi}_{n_y}$$

for $\mathbf{E}_{\text{im}}$. The lower diagonal elements of $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (pink) are given by rM_i for $\mathbf{A}_{\text{im}}$ and $-rM_i$ for $\mathbf{E}_{\text{im}}$ respectively. On the bottom right of each lower diagonal in $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (pink cross) we have $M_{n_y}(1+r)$ for $\mathbf{A}_{\text{im}}$ and $-M_{n_y}(1+r)$ for $\mathbf{E}_{\text{im}}$. The upper diagonal elements of $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (brown) are given by M_i for $\mathbf{A}_{\text{im}}$ and $-M_i$ for $\mathbf{E}_{\text{im}}$. On the top left of each upper diagonal in $\mathbf{A}_{\text{im}}$ and $\mathbf{E}_{\text{im}}$ (brown dot) we have $M_1(1+r)$ for $\mathbf{A}_{\text{im}}$ and $-M_1(1+r)$ for $\mathbf{B}_{\text{im}}$. The elements introduced by the upwind discretization of the transport term in (1) (light blue) in $\mathbf{A}_{\text{im}}$, $\mathbf{E}_{\text{im}}$, $\mathbf{D}_{\text{im}}$ and $\mathbf{H}_{\text{im}}$ are given by

$$v_k\frac{\Delta t}{2\Delta x}$$

for $\mathbf{A}_{\text{im}}$ and $\mathbf{D}_{\text{im}}$ and

$$-v_{k+1}\frac{\Delta t}{2\Delta x}$$

for $\mathbf{E}_{\text{im}}$ and $\mathbf{H}_{\text{im}}$. The non-zero elements in $\mathbf{B}_{\text{im}}$ and $\mathbf{F}_{\text{im}}$ (red dots) are given by $2rM_1\text{Bi}_{1,j}^k$ for $\mathbf{B}_{\text{im}}$ and $-2rM_1\text{Bi}_{1,j}^{k+1}$ for $\mathbf{F}_{\text{im}}$. The non-zero elements in $\mathbf{C}_{\text{im}}$ and $\mathbf{G}_{\text{im}}$ (yellow crosses) are given by $2M_{n_y}\text{Bi}_{n_y}$ for $\mathbf{C}_{\text{im}}$ and $-2M_{n_y}\text{Bi}_{n_y}$ for $\mathbf{G}_{\text{im}}$.

4. MODEL SELECTION AND OPTIMIZATION

To be used in a real-time setting such as model-based control, the model needs to be computationally as cheap as possible while at the same time deliver accurate simulations of the real-world process. To achieve this, we want to choose the discretization grid as coarse as possible while maintaining a sufficient amount of accuracy.

In section 3 we derived an explicit (16) and an implicit (20) integration scheme for the governing PDE (1). A comparison of the two approaches shows that of course one simulation step of the explicit model takes less time than one of the implicit, but we have to choose very small time steps to maintain numerical stability, whereas in the implicit approach we can choose the time step arbitrarily large in terms of numerical stability and only need to focus on accuracy. Figure 6 shows a runtime comparison of the two approaches. In both cases a physical time of 50 s was simulated and realistic trajectories of the volumetric

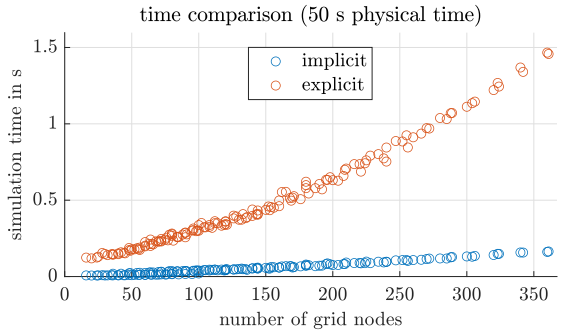

Fig. 6. Comparison of computation time. Simulation was carried out over 50 s physical time which corresponds to 50000 simulation steps in the explicit model and 1250 steps with the implicit model. The simulation was carried out in MATLAB on an Intel Core™ i7-8850H @ 2.60GHz.

air flow $\dot{V}(t)$ and speed $v(t)$ were used, similar to the one depicted in Fig. 10. The time step was chosen to be 40 ms in the implicit formulation, whereas in the explicit approach we are forced to choose a time step not larger than 1 ms in order to maintain numerical stability. The overall simulation time therefore is lower with the implicit approach by a factor of roughly 10 compared to the explicit one (Fig. 6). We also evaluate the loss of accuracy that comes with using a coarse grid. Like before, we simulate a realistic trajectory for 50 s physical time with both models (coarse grid and fine grid) and compare the obtained results. Likewise we use the resulting surface temperature after passing the heating zone as comparative value. The root-mean-square error (RMSE)

$$J(n_x, n_y, r) = \frac{1}{N}\sqrt{\sum_{k=1}^{N}\left(\vartheta_{1,n_{x,\text{fine}}}^{k,\text{fine}} - \vartheta_{1,n_x}^{k,\text{test}}(n_x, n_y, r)\right)^2} \tag{21}$$

serves as a measure for the accuracy loss, where N denotes the number of time steps. The grid is fully determined by the parameters n_x and n_y (number of grid points in each direction) as well as the grid expansion factor r. For the reference simulation with a fine grid, which serves as the ground truth we chose $n_{x,\text{fine}} = 50$, $n_{y,\text{fine}} = 30$ and $r_{\text{fine}} = 1.05$, which results in a rather big system with 1500 grid points. We use the implicit integration scheme (20) with a time step of 40 ms. For a given number of grid points n_x and n_y we can still freely choose the expansion factor r resulting in a better or worse accuracy. To ensure the optimal setting, we minimize the RMSE over r for each regarded combination of n_x and n_y by solving the following optimization problem:

$$\begin{aligned} J^\star(n_x, n_y) = \; & \min_r J(n_x, n_y, r) \\ & s.t. \quad 1 \le r \le 2 \end{aligned} \tag{22}$$

The search space is bounded by imposing constraints on r. The optimization was carried out by brute force, testing r in the considered range with a distance of 0.01 for any given n_x, n_y combination. It is worth mentioning that for some combinations of n_x, n_y and r the explicit model became numerically unstable.

However, on the tested points it was always possible to find a value for r for which the simulation was stable. The

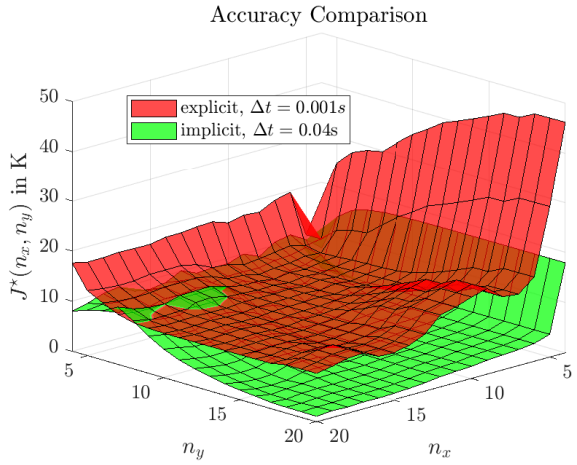

Fig. 7. Comparison of accuracy, explicit vs. implicit model. Simulation was carried out over 50 s physical time using realistic trajectories of the input quantities $\dot{V}(t)$ and $v(t)$.

results are summarized in Fig. 7. It can easily be seen that also in terms of accuracy the implicit formulation outperforms the explicit model. Thus, we can get a higher level of accuracy with less computational effort using the implicit rather than the explicit approach. We therefore choose to continue with the implicit model in the sequel. Given the achievable accuracy for each setting (Fig. 7) we need to choose the grid parameters n_x and n_y.

For achieving a good trade-off between accuracy and a coarse grid, four promising grid settings according to Fig. 8 are selected. From the plot it can be seen that there are some sweet spots that yield comparably high accuracy (low RMSE) for a relatively coarse grid. For the remaining part of this paper we will use the setting number 3 from Fig. 8, which is defined by the parameters $n_x = 10$, $n_y = 10$, $r^\star = 1.25$ with $r^\star$ being the arg min of (22).

5. PARAMETER FITTING AND EXPERIMENTAL RESULTS

The values of thermodynamical parameters, like density ρ, thermal conductivity λ and the specific heat capacity c_p are not exactly known. Thus, measurement data is used

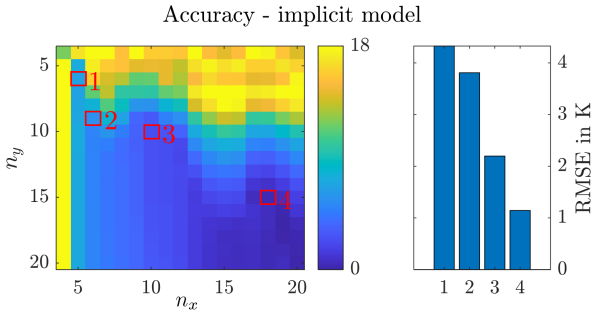

Fig. 8. Accuracy of different grid settings. Left: dark blue corresponds to high accuracy (low RMSE), yellow corresponds to low accuracy (high RMSE). The selected points are marked in red. Right: The RMSE is plotted for the selected grid settings on the left.

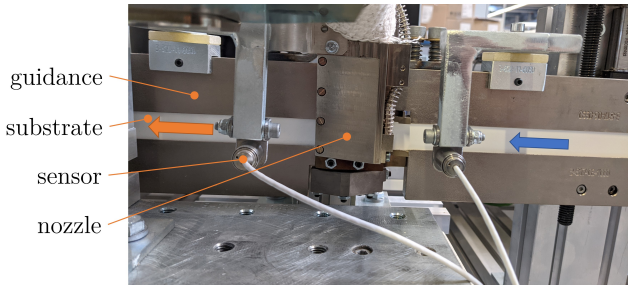

Fig. 9. Experimental setup

to estimate these parameters by minimizing the difference between the simulated and measured temperature. Since λ and ρ only enter the model multiplicatively in the combined parameter a (1), they cannot be estimated independently. Instead, the combined parameter $\Theta = c_p \cdot \rho$ is introduced. Note, that λ does enter separately through the Biot-number (13). The optimization problem is formulated and solved for obtaining those parameter values that lead to an accurate fit between simulated and actual temperatures measured on the real process (Fig. 9). The used trajectory is depicted in Fig. 10. It represents a realistic scenario for the regarded process. Summarizing the parameters in a vector Ψ

$$\Psi = [\lambda, \Theta]^\top \tag{23}$$

we can formulate the optimization problem as follows:

$$\Psi^\star = \arg\min_{\Psi} F(\Psi) \tag{24a}$$

$$s.t. \quad \lambda_{\min} \leq \lambda \leq \lambda_{\max} \tag{24b}$$

$$c_{p,\min} \cdot \rho_{\min} \leq \Theta \leq c_{p\max} \cdot \rho_{\max} \tag{24c}$$

With F being the RMSE between the final surface temperature at $x = L$ of the measured and simulated trajectories:

$$F(\Psi) = \frac{1}{N} \sqrt{\sum_{k=1}^{N} \left(\vartheta^{k,\text{meas}} - \vartheta^{k,\text{sim}}_{1,n_x}(\Psi) \right)^2} \tag{25}$$

The lower and upper bounds are constraining the parameters to stay in a physically reasonable range. To solve the optimization problem (24) we use CasADi (Andersson et al. (2019)) with the interior point solver IPOPT.

Figure 10 shows a comparison between the simulated temperature with optimized parameters and the measured temperature obtained by experiment. It can be seen that they are in good agreement. However, there is still room for improvement as discussed in the next section. The mean difference between the measured and the simulated temperature is 1 K and the standard deviation is 10.6 K.

6. CONCLUSION AND FUTURE WORK

The model developed in this paper delivers accurate results that are in good agreement with the measured data as depicted in Fig. 10. However, a delay between the measured and the simulated data can be observed. This can be traced back to the communication delays in the experimental setup, which are not considered in the model. Furthermore, it is obvious that the stationary temperatures in some operating points are not exactly matched. This mismatch most certainly stems from two different effects that were not considered in our model. First, the temperature of the

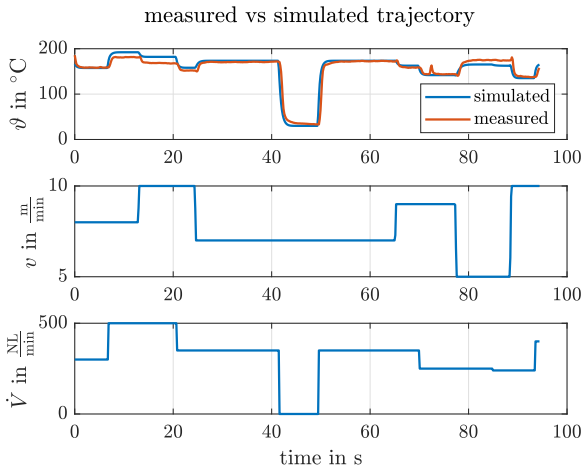

Fig. 10. Trajectories of test scenario.

heated air varies slowly over time, which obviously influences the substrate's temperature. Second, the nozzle used in the experimental setup cannot be completely sealed, resulting in some air leakage especially in regimes with high air pressure. Future work might therefore focus on developing an adaptive approach, where some parameters are estimated online in order to obtain even more accurate simulation results in a real-time online setting.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36. doi: 10.1007/s12532-018-0139-4.

Baehr, H.D. and Stephan, K. (2013). *Wärme-und stoffübertragung*, volume 8. Springer.

Ferziger, J.H., Perić, M., and Street, R.L. (2008). *Numerische Strömungsmechanik*, volume 1. Springer.

Ivanova, A. (2013). Model predictive control of secondary cooling modes in continuous casting. In *Proc. of 22nd Conference on Metallurgy and Materials METAL, Brno*, volume 6.

Steinboeck, A., Wild, D., and Kugi, A. (2013). Nonlinear model predictive control of a continuous slab reheating furnace. *Control Engineering Practice*, 21(4), 495–508.

Strommer, S., Niederer, M., Steinboeck, A., and Kugi, A. (2018). Hierarchical nonlinear optimization-based controller of a continuous strip annealing furnace. *Control Engineering Practice*, 73, 40–55. doi: 10.1016/j.conengprac.2017.12.005.

Wang, Y., Luo, X., and Wang, H. (2019). Gpu-based model predictive control of nonlinear parabolic partial differential equations system and its application in continuous casting. *IEEE Access*, 7, 79337–79353. doi: 10.1109/ACCESS.2019.2921582.

Weiss, R., Albrecht, L., Reuter, J., and Gehan, O. (2018). A control-oriented thermal model for a moving dual-layer substrate. *IFAC-PapersOnLine*, 51(2), 325–330.

Weiss, R., Diehl, M., Rieger, H., and Reuter, J. (2021). Modeling and temperature control of a moving substrate. *IFAC-PapersOnLine*, 54(6), 212–217.