

Computational efficiency in numerical modeling of high temperature latent heat storage: Comparison of selected software tools based on experimental data

Harald Pointner^{a,*}, Alvaro de Gracia^b, Julian Vogel^a, N. H. S. Tay^c, Ming Liu^c,
Maike Johnson^a, Luisa F. Cabeza^d

^a Institute of Technical Thermodynamics, German Aerospace Center (DLR), Pfaffenwaldring 38-40,
70569 Stuttgart, Germany

^b CELIMIN, Universidad de Antofagasta, Av. Universidad de Antofagasta 02800, Antofagasta, Chile

^c Barbara Hardy Institute, University of South Australia, Mawson Lakes Boulevard, Mawson Lakes,
South Australia 5095, Australia

^d GREA Innovació Concurrent, Universitat de Lleida, Edifici CREA, Pere de Cabrera s/n, 25001 Lleida, Spain

*Corresponding author. Tel: +49 711 6862-8107. Fax: +49 711 6862 747
E-mail address: harald.pointner@dlr.de

ABSTRACT

In this article, four different numerical models for the investigation of phase change processes within latent heat storage are described and compared concerning accuracy, convergence behavior and computational efficiency. The models are based on different types of discretization, make use of different ways to model phase change and are implemented with C, MATLAB or ANSYS CFX. After a brief introduction into each investigated numerical model, the experimental reference setup is described. It consists of a flat plate latent heat storage with the eutectic mixture $NaNO_3(46wt\%) - KNO_3(54wt\%)$ with a measured melting temperature of $219.5\text{ }^\circ\text{C}$ as storage material. Based on the corresponding simulation model developed in this paper, the comparison of the numerical models is achieved. This methodology allows the investigation of the numerical performance of different software tools in the context of high temperature latent heat storage that was not achieved thus far. All four numerical models show good agreement to experimental results but differ significantly in speed and convergence behavior.

Keywords: latent heat storage; numerical simulation; efficiency; C; MATLAB; ANSYS CFX

Nomenclature

a_l	Thermal diffusivity at liquid phase (m^2/s)
a_s	Thermal diffusivity at solid phase (m^2/s)
C	Constant in equation (2) (—)
$c_{p,l}$	Specific heat capacity of PCM at liquid phase (J/kgK)
$c_{p,s}$	Specific heat capacity of PCM at solid phase (J/kgK)
D	Averaged temperature deviation from simulation to experiment ($^{\circ}C$)
g	Gravity acceleration (m/s^2)
k_{eff}	Effective thermal heat conductivity (W/mK)
k_l	Thermal heat conductivity of PCM at liquid phase (W/mK)
k_s	Thermal heat conductivity of PCM at solid phase (W/mK)
L	Heat of fusion (J/kg)
m	Constant in equation (2) (—)
n	Number of measuring points (—)
T	Temperature ($^{\circ}C$)
T_{exp}	Measured experimental temperature ($^{\circ}C$)
$T_{HTF,avg}$	Averaged HTF temperature between inlet and outlet ($^{\circ}C$)
$T_{HTF,in/out}$	HTF temperature at inlet (in) and outlet (out) ($^{\circ}C$)
T_m	Melting temperature ($^{\circ}C$)
T_{sim}	Temperature from simulation ($^{\circ}C$)
T_{TC18}	Temperature at position TC18 ($^{\circ}C$)
t	Time (s)
t_i	Time at measuring point $i = n$ (s)
$TC12, TC40, TC17, TC18$	Temperature measuring points (—)
X	Width of liquid PCM in horizontal direction (m)
x	Cartesian coordinate (—)
y	Cartesian coordinate (—)

Greek symbols

α	Heat transfer coefficient (W/m^2K)
β	Thermal expansion coefficient ($1/K$)
ΔT_{pc}	Temperature difference of melting range ($^{\circ}C$)
Δt	Time step (s)
θ	Uncertainty interval of experimental results ($^{\circ}C$)

μ_l	Dynamic viscosity of PCM at liquid phase (kg/ms)
ξ	Share of simulation results within uncertainty interval θ (%)
ρ_l	Density of PCM at liquid phase (kg/m^3)
ρ_s	Density of PCM at solid phase (kg/m^3)

Abbreviations

<i>CFD</i>	Computational fluid dynamics
<i>CHOLMOD</i>	Sparse Cholesky factorization and modification package
<i>CPU</i>	Central Processing Unit
<i>CVFDM</i>	Control volume based finite difference method
<i>DSC</i>	Differential scanning calorimetry
<i>FDM</i>	Finite difference method
<i>FVM</i>	Finite volume method
<i>HTF</i>	Heat transfer fluid
<i>KNO₃</i>	Potassium nitrate
<i>NaNO₃</i>	Sodium nitrate
<i>PCM</i>	Phase change material
<i>Ra</i>	Dimensionless Rayleigh number
<i>TC</i>	Thermocouple

1 Introduction

Increasing the share of fluctuating renewable energies in the worldwide energy mix is associated with the need for energy storage to smooth electricity consumption and production [1]. Latent heat storage in combination with direct steam generating solar thermal power plants promises high efficiencies for providing dispatchable and green electricity [2].

In latent heat storage, the isothermal phase change of the storage material is utilized to store thermal energy at a nearly constant temperature. The principles of latent heat storage are described in literature by Alexiades [3], Ziskind [4] and Mehling [5] among others. Since phase change processes base on highly non-linear equations, there are only analytical solutions available for some basic and specific problems [3], [6]. In order to understand and investigate such storage systems without analytical limitations, numerical calculation tools are of great importance. There is a huge variety of numerical models with varying approaches implemented in different software packages described in literature to overcome this problem.

To numerically investigate phase change processes, the commercial computational fluid dynamics (CFD) software packages ANSYS CFX and ANSYS Fluent are often used. Using ANSYS CFX, Tay simulated different latent heat storage systems including a high temperature shell and tube heat exchanger with surrounding phase change material (PCM) [7], additional pins and fins [8] and a latent heat storage concept named dynamic melting [9]. Jaworski [10] successfully investigated a combination of gypsum and microencapsulated PCM for low temperature building applications with ANSYS Fluent. Other authors also use this software package successfully for the examination of phase change processes [11]–[13].

Another common software tool for the simulation of PCM is MATLAB. Arzamendia Lopez [14] examined a PCM-to-air heat exchanger in the context of low temperature building applications using this tool. Borderon [15] used a coupled MATLAB model to examine a PCM/air module inside a ventilation system in order to improve the energy efficiency of buildings. Other authors also use MATLAB for PCM simulation [16]–[18].

A third and widely used programming language for modeling phase change processes is C. De Gracia [19], [20] examined the application of PCM in low temperature building applications with this methodology.

The use of commercial CFD code, MATLAB and C for accurate simulation results for phase change processes is proven. However, no comparative statements regarding the numerical efficiency of each tool can be found.

In addition to the different software tools, there are many approaches to modeling phase change: The three most important ones that deal with this challenge are the enthalpy method described by Alexiades [3], Ziskind [4] and other authors [21]–[24], the effective thermal heat capacity method described by Lamberg [25] and the source term method described by Voller [26], [27]. Adding up to the overall complexity, different numerical discretization schemes of the appearing differential equations are described in literature [28]–[30]. It is difficult to decide which path is best in order to achieve the highest performance for specific needs.

A recent approach to describe and examine the variety of different available numerical models with a systematic investigation was carried out by Al-Saadi [31]. Different numerical models are implemented into the MATLAB/SIMULINK environment and compared to each other in the context of annual simulations of low temperature PCM building applications. The comparison of the models is based on the implementation of different numerical models in this specific software package.

The paper presented here focuses on the examination of selected numerical models implemented in different software packages. All of these models base on the same experimental reference case. In this manner, the numerical efficiency of these models can be compared in detail in relation to the software package they are implemented in. Representative statements about the accuracy,

computational speed and convergence behavior of each specific program can be found. This kind of study, including the examination of these aspects in the context of high temperature latent heat storage, is carried out for the first time.

In the course of this paper, four different numerical models basing on the three above mentioned methods to model phase change are briefly outlined. These are coded and implemented in either C, MATLAB or the CFD tool ANSYS CFX.

In a second step, these numerical models are then validated and compared using original experimental data as a reference case. For this, the experimental setup and the corresponding simulation model with assumptions, boundary conditions and material properties are presented in detail.

The comparison of all described tools regarding accuracy, convergence behavior and computational efficiency helps the reader to decide which numerical method and software package best matches specific circumstances and requirements.

2 Examined numerical models

In the following section, the numerical models that are examined in this article are briefly described. All models base on the discretization of the two-dimensional heat conduction equation:

$$\frac{\partial T}{\partial t} = a_{l/s} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (1)$$

In equation (1), T stands for the temperature, t for the time, $a_{l/s}$ for the thermal diffusivity of the liquid or solid phase and x, y for the Cartesian coordinates of the two-dimensional plane.

Within the numerical models, convection effects are regarded by an effective thermal heat conductivity approach described in Section 3.2.

Each numerical model in this paper uses an iteration algorithm to calculate the results for each time step. This iteration process is stopped after a temperature tolerance criterion is reached. The criterion used is an absolute temperature difference from the actual iteration step to the last iteration step of $\leq 1 \cdot 10^{-5} K$. This criterion is implemented into all numerical models in this article. This makes the results of each model comparable to each other.

Moreover, the two numerical models with implemented effective heat capacity method, see Fig. 1, need a melting range in order to enable stable and reliable simulations. The melting range has to be implemented into these numerical models in order to avoid unsteady heat capacity curves in the region of the phase change temperature. To avoid numerical difficulties in these models, a melting

range of $\Delta T_{pc} = 4K$ is implemented. Within this melting range, the heat capacity curve has a triangle shape centered on the melting temperature T_m in order to model the phase change. This methodology was studied by Farid [32] and was found to be successful in describing the heat transfer in phase change materials [19].

For the numerical models that use the enthalpy method and source term method to model the phase change, see Fig. 1, this melting range is not necessary. Both methodologies can handle specific and single melting temperatures. However, to obtain comparable results from all numerical models, the same melting range $\Delta T_{pc} = 4K$ is implemented into them. This is done for the FDM-enth-Mat model by enlarging the enthalpy interval that correlates to the area of phase change. The sensible heat caused by the temperature interval ΔT_{pc} sums to the heat of fusion. The resulting temperature of each node within this enthalpy interval is interpolated within ΔT_{pc} . For the CVFDM-sour-Mat model, the implementation of the melting range is realized by expanding the temperature interval corresponding to the phase change to $\Delta T_{pc} = 4K$, rather than to the single temperature T_m .

For further details about the different implemented approaches within each numerical model, e.g. for discretization and method of modeling phase change, the corresponding literature is given in each section of the corresponding numerical model. An overview over the different numerical models regarded within this article is shown in Fig. 1.

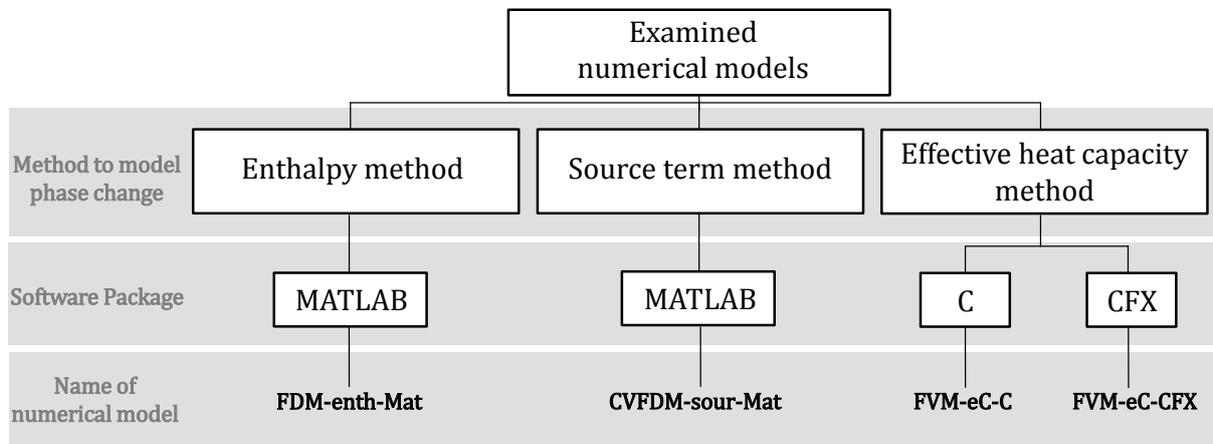


Fig. 1: Classification of the four numerical models studied in this article

2.1 FDM-enth-Mat

The two-dimensional discretization of the enthalpy form of equation (1) is done via the finite difference method (FDM) approach. Details about this method are given in [3], [28]. The main drawback of this method is the required uniform mesh in each direction. However, it is comparably simple to be implemented.

The phase change of the storage material here is modelled via the enthalpy method to overcome numerical issues regarding the constant temperature during phase change. This method is well

known and widely described in literature [3], [4]. A fully implicit approach is chosen in order to overcome stability criteria that would have to be complied with in explicit formulation. The need to fulfill explicit stability criteria would lead to small time steps that cause extensive overall calculation time of the model.

The iterations within each time step are carried out by the Gauss-Seidel iteration scheme without relaxation techniques as described in [3].

Sensible materials can easily be integrated into the simulation model, since the enthalpy method offers the possibility to introduce both sensible materials and PCM conveniently, without changes in discretization or model setup [3]. This makes the enthalpy method very flexible. Even moving parts inside the model can be handled efficiently by moving boundary conditions. For details see [33].

This model is implemented in the MATLAB R2012b software package. The code written is fully matrix and vector based. Additionally, the code is vectorized in order to avoid the extensive usage of loops in the code to enable computationally efficient calculations also for large simulation models with many discretization elements [34].

2.2 CVFDM-sour-Mat

The two-dimensional discretization of the numerical model described here is achieved via the control volume based finite difference method (CVFDM) approach. This method combines the advantages of the FVM with no restrictions of node spacing and element shape with the straightforward implementation of the FDM discretization method. Details about this method can be found in [29].

To model phase change, the source term method is applied in this numerical model. It can be implemented in any existing heat conduction algorithm by introducing or adjusting a latent heat source term. The source term can be linearized in order to transfer the non-linear equation system into a linear one. This reduces the iteration counts per time step and leads to a fast and robust convergence of the iterative calculations. Details about this implemented method are described by Voller [26], [27].

The resulting linear equation system for the temperatures of the next iteration step is solved in MATLAB R2013b with the function *mldivide* [35]. This function detects the resulting matrix as symmetric and sparse diagonal, and applies a direct solver (CHOLMOD: Sparse Cholesky factorization and modification package). The direct solver produces an accurate solution after one iteration only. Additional iterations are only needed due to remaining non-linearities, e.g. change of effective conductivity in the natural convection model or moving of the phase front from one control volume to another.

2.3 FVM-eC-C

The numerical model described here solves the energy equation using a fully implicit scheme in the two-dimensional Cartesian coordinates system. The discretization of the domain is done using the finite volume method (FVM) [29], [30].

The phase change is taken into account through an effective heat capacity method. Details can be found in [25]. Since this methodology tracks the effective heat capacity, which significantly increases for temperatures around the melting temperature in order to regard phase change, the time step within the simulation has to be small. Otherwise, the phase change could be partially or completely skipped.

The iterations within each time step are carried out by the Gauss-Seidel iteration scheme without relaxation [3] until convergence is achieved based on the resulting temperatures.

The outlined numerical model is implemented in the C programming language using the free-licensed software DevC++ 4.9.9.2.

2.4 FVM-eC-CFX

The discretization of the numerical model described here is realized by FVM methodology.

The phase change of the storage material is modeled by the effective heat capacity method as described in [25]. The problem of the need for comparably small time steps during simulation appears to be the same as described in the last section. All materials within the simulation model are created as solid domains. The PCM is defined as a single solid material with the specific heat capacity distribution depicted at the beginning of Section 2.

The CFX-PRE within version 15 of the academic research code ANSYS is utilized in this numerical model. Using this commercial code in order to solve phase change problems has already been found to be successful [7], [36].

3 Modeling the reference system

This section describes the transformation from an experimental setup to a corresponding simulation model. First, the existing experimental setup is described. These real circumstances are then idealized. Given all assumptions and methods, this procedure results in a simulation model that is the basis for the comparison of the numerical models examined in this paper.

3.1 Experimental setup

The experimental setup consists of DLR's flat plate heat storage with four PCM chambers [37], [38] as shown in Fig. 2. The eutectic mixture of technical grade quality $NaNO_3(46wt\%) - KNO_3(54wt\%)$

with a measured melting temperature onset of 219.5°C is the PCM. The thermal latent heat capacity of the storage is approximately 7.5 kWh . The inner two chambers have a width of 80 mm while the outer two PCM chambers are half of the width. These two smaller chambers minimize the boundary effects on the inner two chambers where the measurements are carried out. The height of the storage is 1010 mm . All PCM chambers are separated by channels through which the heat transfer fluid (HTF: Mobiltherm 603) flows. For construction details, see Fig. 3. For charging, the HTF is pumped from the upper flange downwards through the oil channels to the bottom flange. For discharging, the HTF flow is reversed. It enters the system at the bottom flange and is pumped upwards.

In order to be able to remove the PCM, each chamber is connected to a drain flange. The temperature inside the PCM chambers is measured via thermocouples of type K (Class 1) at various positions. Their measurement tolerances are given by the supplier with $\pm 1.5\text{ K}$.

Within this paper, one discharging cycle lasting 8 h is observed and compared to numerical results of the different models outlined in the last section. The temperature difference between the melting temperature of the PCM and the steady state HTF temperature thereby is 22.5 K .



Fig. 2: Insulated experimental setup of DLR's flat plate latent heat storage

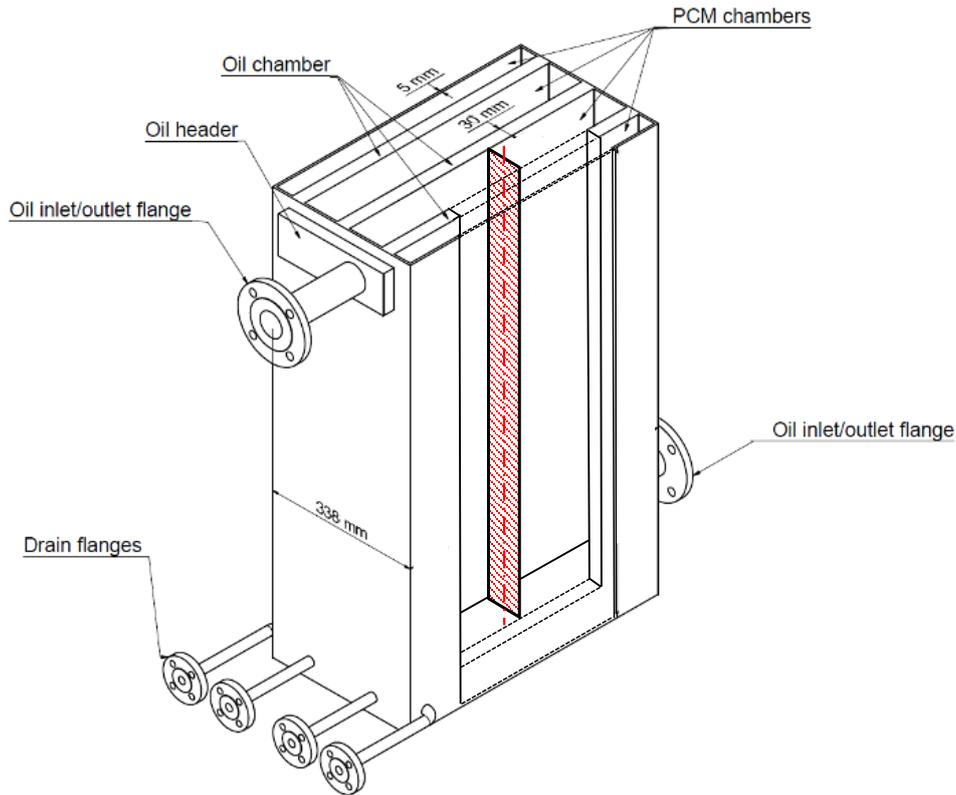


Fig. 3: Schematic of uninsulated storage with dimensions and the 2D-plane cut for simulation before applying a symmetry condition

3.2 Simulation model

For the simulation of the discharging process of the flat plate latent heat storage described in the last section, the problem has to be idealized.

First of all, symmetry is used to reduce the size of the model in order to save calculation time. Experiments during commissioning showed an equal distribution of HTF flow through all three HTF chambers [37]. For this reason, the simulation model regards only one PCM chamber instead of the two centered ones; see the red-shaded layer in Fig. 3. Simulation results are expected to be exemplary for both chambers, however. Since the two center PCM chambers are surrounded by HTF chambers of equal temperature on both sides, a symmetry condition at the vertical symmetry axis of the red-shaded plane can be applied. That is why only the half of the red-shaded area in Fig. 3 is regarded within simulations. See Fig. 4 for details.

The HTF changes temperature while flowing up through the storage from the bottom. Therefore, no symmetry condition can be applied in the direction of the HTF flow and the whole storage height has to be regarded in the simulation model. The resulting 2D-geometry of the simulation model can be seen in Fig. 4. It consists of the steel wall between the HTF and the PCM chamber and half of the PCM of one centered PCM chamber. At the bottom and the top of the model, adiabatic boundary conditions are assumed. On the right side of the model, the symmetry boundary condition applies.

Due to the HTF flowing on the left side, a convection boundary condition is integrated into the simulation model. The HTF itself is not considered in the model. Former experiments showed an average heat transfer coefficient between HTF and steel wall over the whole height of the storage of $\alpha = 200 \text{ W/m}^2\text{K}$ [37]. Since the HTF temperature changes while passing the steel wall, the temperature distribution along the steel wall is interpolated linearly between HTF inlet and outlet temperatures $T_{HTF,in/out}$. These temperatures are measured during experiments and implemented as time dependent boundary values into the model.

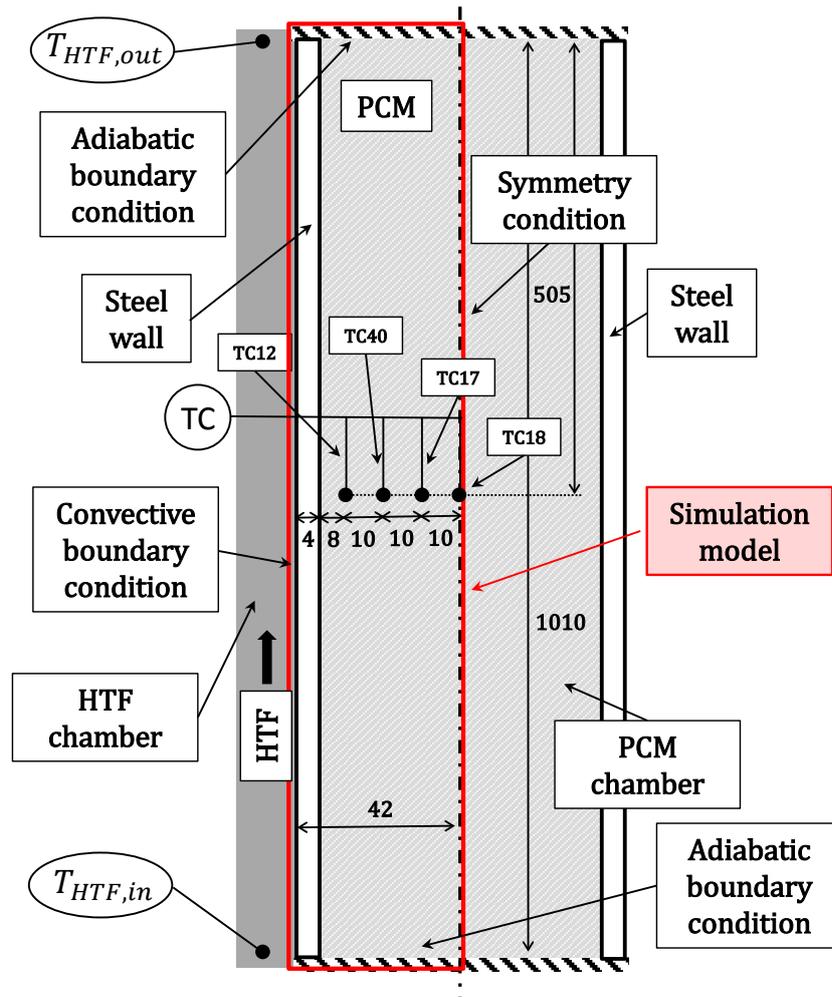


Fig. 4: Simulation model with boundary conditions and dimensions developed from 2D-plane in Fig. 3

Due to the narrow and high geometry of the storage, natural convection effects inside the liquid PCM have significant influence on the results. All numerical models described in this paper are heat conduction models only. Convection processes can be regarded using an effective thermal heat conductivity approach, however. In order to compare the models examined in this paper, the same convection approach is programmed into them. To regard convection effects within conduction models, the approach of Farid and Husian [39] is adapted. The effective thermal heat conductivity as a function of time and share of liquid PCM inside the chambers is calculated via equation (2).

$$k_{eff} = C k_l Ra^m \quad (2)$$

Whereby k_{eff} and k_l are the effective and liquid PCM heat conductivities, $C = 0.1$ and $m = 0.25$ are empirical constants dependent on the geometry of the storage (given in [39]) and the Rayleigh number Ra describing the influence of natural convection. Ra is calculated with equation (3):

$$Ra = \frac{g \beta \rho_l^2 c_{p,l} (T_{HTF,avg} - T_{TC18}) X^3}{\mu_l k_l} \quad (3)$$

Equation (3) contains $g = 9.81 \text{ m/s}^2$ as the gravitational acceleration, β as the thermal expansion coefficient, ρ_l as the density of the liquid PCM, $c_{p,l}$ and μ_l as the specific heat capacity and dynamic viscosity of the liquid PCM. $T_{HTF,avg}$ is the averaged HTF temperature between $T_{HTF,in/out}$ and T_{TC18} is the temperature of the PCM at the position of the symmetry condition, see Fig. 4. X is the width of the liquid PCM fraction along the horizontal direction of the PCM in the simulation model. At the beginning of discharging, all of the PCM is liquid and therefore $X = 0.038 \text{ m}$. The more PCM solidifies, the smaller X gets. Thereby, k_{eff} is limited by the material properties of the PCM and cannot drop below k_l ($k_{eff} \geq k_l$). Since $T_{HTF,avg}$, T_{TC18} and X change with time, the resulting k_{eff} varies at every time step. Ignoring convection effects in the numerical models leads to poor agreement between numerical and experimental results.

After giving the theory of the simulation model, the practical issues such as the material properties and process parameters are described in the following. The material properties used for simulation in all numerical models are given in Table 1 and are taken from [37], [38], [40]. The heat of fusion of the salt mixture used in the experimental setup was specifically measured for this study by differential scanning calorimetry (DSC) technique.

At the beginning of the discharging process ($t = 0$), the PCM, steel and HTF have a temperature of $246,2 \text{ }^\circ\text{C}$. At the same time, the starting HTF temperature distribution is applied to the outer steel wall. The models use this temperature distribution in combination with the heat transfer coefficient of $\alpha = 200 \text{ W/m}^2\text{K}$ between HTF and outer steel wall for calculating the heat flow induced by the temperature difference between steel/PCM and HTF. This heat flow leaving the system causes a temperature drop inside the PCM. The temperature curves of four different points (TC12/TC40/TC17/TC18) in the PCM are calculated and compared to experimental results of the corresponding thermocouple measurements. The location of the four different temperature measuring points can be seen in Fig. 4. The discharging process is observed for 8 hours, at which time all temperatures have reached a steady state.

Table 1: Material properties used in the simulation model

	Unit	Steel [1.0425]	PCM [$KNO_3 - NaNO_3(eu)$]
Density	$[\rho_{l/s}] = kg/m^3$	7800	2050 (s) 1959 (l)
Specific heat capacity	$[c_{p,l/s}] = J/kg K$	540	1350 (s) 1492 (l)
Thermal heat conductivity	$[k_{l/s}] = W/m K$	51	0.435 (s) 0.457 (l)
Dynamic viscosity	$[\mu_l] = kg/m s$	-	$5.8 \cdot 10^{-3}$ (l)
Thermal expansion coefficient	$[\beta] = 1/K$	-	$3.5 \cdot 10^{-4}$ (l)
Melting temperature/range	$[T_m] = ^\circ C$	-	219.5 – 215.5
Heat of fusion	$[L] = kJ/kg$	-	94

4 Results and discussion

In this section, the numerical models are evaluated and compared regarding their numerical performance. The time and mesh independency of the numerical models is discussed first. The accuracy of the models, their convergence behavior and their computational efficiency are then analyzed.

4.1 Time and mesh independency

To avoid results being influenced by the size of time steps and the size of the numerical discretization mesh, an independency study of these two parameters is performed. Within this comparative study, the combinations of two different mesh sizes and two different time steps are investigated. In order to get meaningful results, large variations of the parameters are chosen. Time steps of $\Delta t = 1 s$ and $\Delta t = 5 s$ are investigated together with mesh sizes with different numbers of elements 20×21 (*Height x Width*) and 505×21 (*Height x Width*). Each combination of these values is used for the calculation of the measuring points TC12, TC40, TC17 and TC18 for each numerical model. The results of these simulations are given for TC40 in Fig. 5 as an example. TC40 was chosen, due to its central position within the simulation model. All other points show a similar behavior.

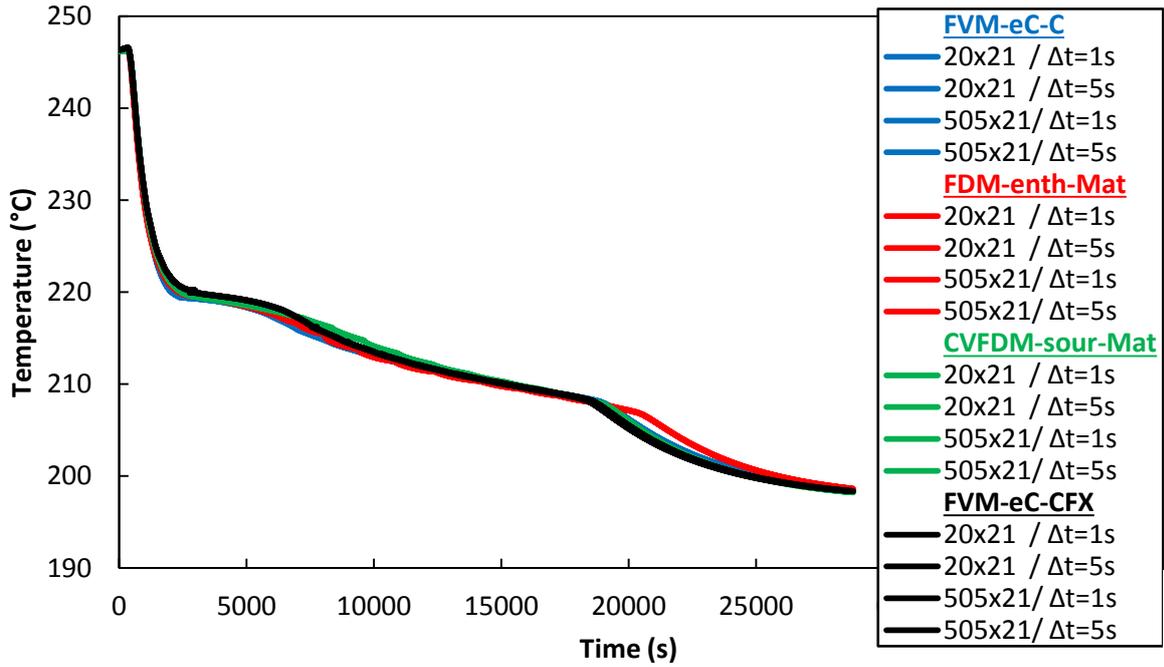


Fig. 5: Results of the time and mesh independency study at position TC40 for all numerical models

Each color in Fig. 5 stands for one numerical model and its results of the four combinations within the time and mesh independency study. It can clearly be seen that results of the different numerical models slightly differ between the models, but not within one model due to varying time step and mesh size. All investigated combinations can therefore be treated as both time step and mesh independent.

4.2 Accuracy

In order to evaluate the accuracy of the investigated numerical models, the difference between the calculated temperatures T_{sim} and the corresponding measured temperatures of the experimental setup T_{exp} during the discharging is determined. The resulting absolute temperature differences are accumulated and divided by the number of measurements including the starting condition $n + 1$. This gives the averaged deviation D of the simulation results compared to experimental results with the unit $^{\circ}C$. The calculation of D is shown in equation (3).

$$D = \frac{\sum_{i=0}^{i=n} |T_{sim}(t_i) - T_{exp}(t_i)|}{n + 1} \quad (3)$$

There is a variety of uncertainties during the measurements and regarding material properties. For example, the positioning of the thermocouples inside the PCM was accomplished as accurately as possible. But due to the alteration of density of the PCM while changing phase, the thermocouples can move slightly and change position. This effect cannot be excluded completely, because strong

mounting devices in the sensor area of the thermocouples would influence heat transfer significantly. Simulations with varying the position of the thermocouples $\pm 1\text{ mm}$ in horizontal direction show an average deviation of $\pm 0.74\text{ K}$ compared to results with the exact assumed position.

Material properties also affect results significantly. Varying the integral value of the heat of fusion by $\pm 5\%$ results into averaged deviations of $\pm 0.41\text{ K}$ in the simulations.

Assuming the positioning uncertainty, the measuring variance of the heat of fusion together with the temperature measurement error of the thermocouples ($\pm 1.5\text{ K}$) to be the most influencing parameters, the overall uncertainty adds up to $\theta = \pm 2.65\text{ K}$. This value is shown via error bars in Fig. 6 - Fig. 9 and gives an error approximation for the experimental data.

In order to compare the number of simulation points that are within θ of the experimental data, the number ξ is introduced. ξ has the unit % and describes the ratio between the number of simulation points that are inside θ , divided by the total number of simulation points $n + 1$. ξ can be interpreted as a classification number to describe the quality of agreement between the curve shape of the experimental and simulation results. Fig. 6 - Fig. 9 show both the experimental and the simulation results of the four measuring points TC12, TC40, TC17 and TC18 during a discharging process of 8 h. The specific location of the TC measurements can be seen in Fig. 4.

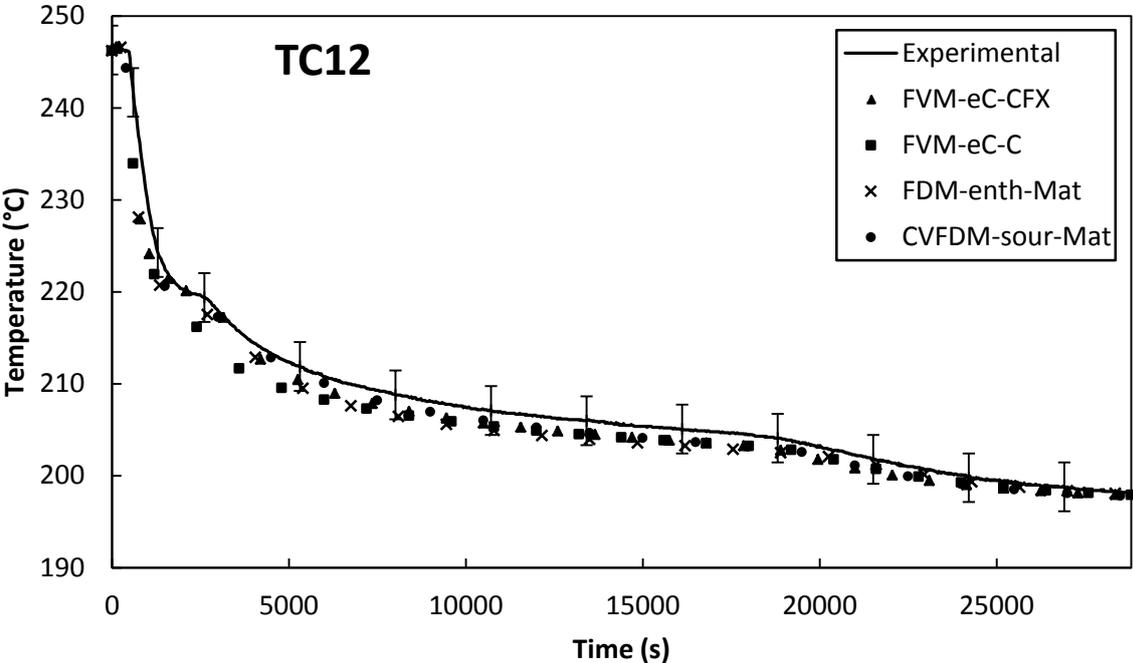


Fig. 6: Results of the numerical models compared to experimental results at measuring point TC12. Dotted lines represent the simulation results; the continuous line represents the experimental results.

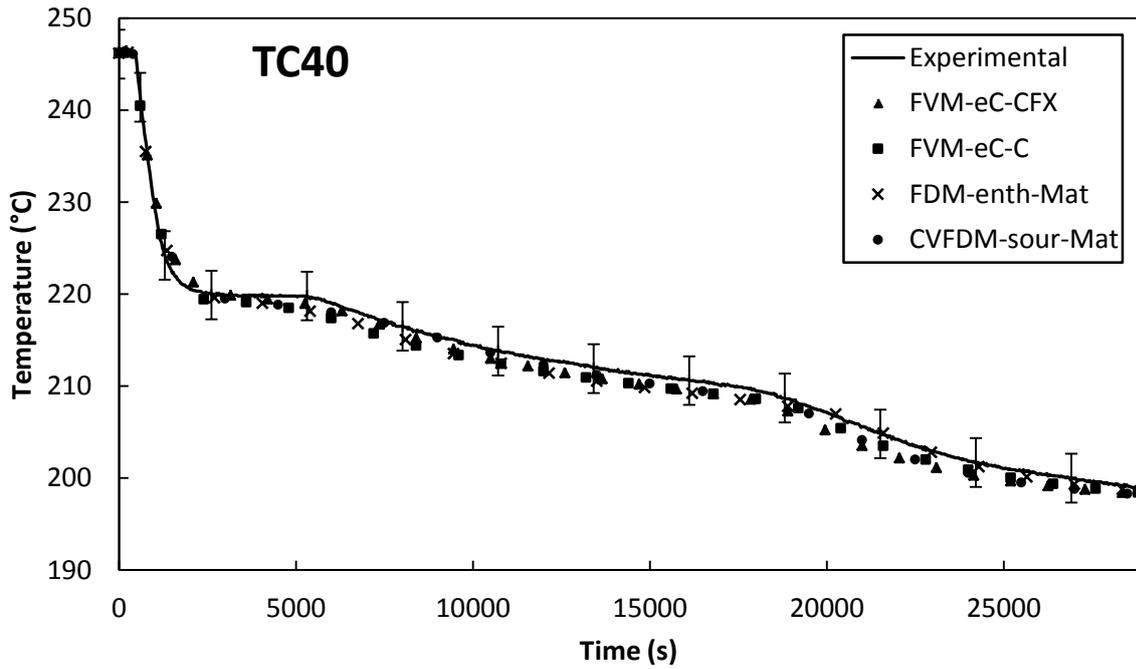


Fig. 7: Results of the numerical models compared to experimental results at measuring point TC40. Dotted lines represent the simulation results; the continuous line represents the experimental results.

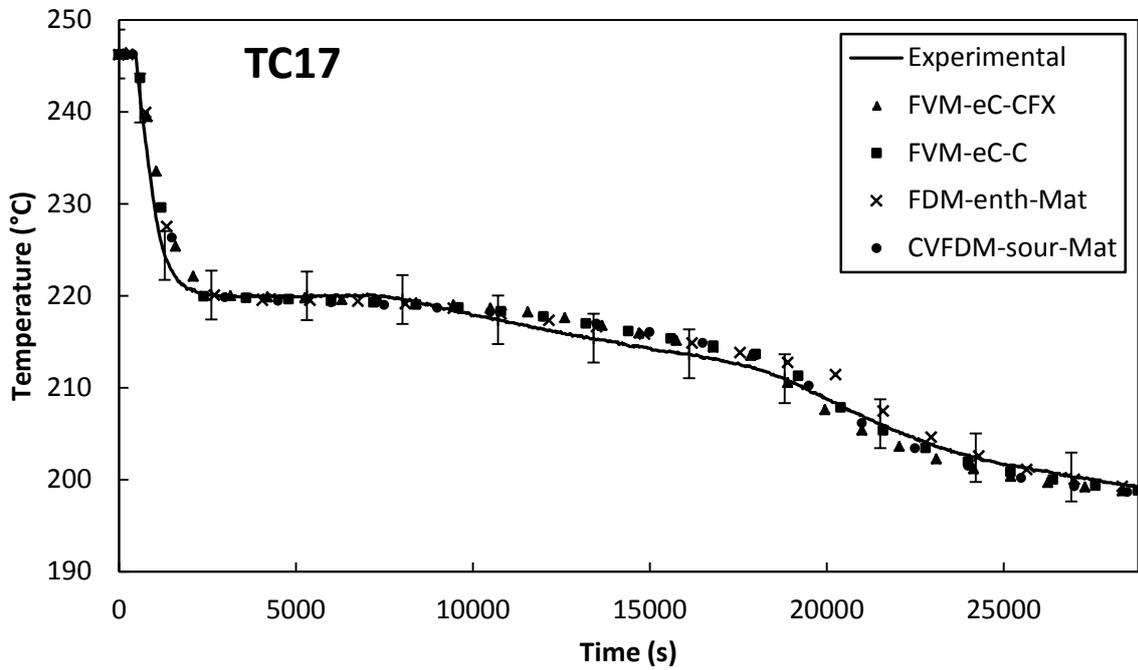


Fig. 8: Results of the numerical models compared to experimental results at measuring point TC17. Dotted lines represent the simulation results; the continuous line represents the experimental results.

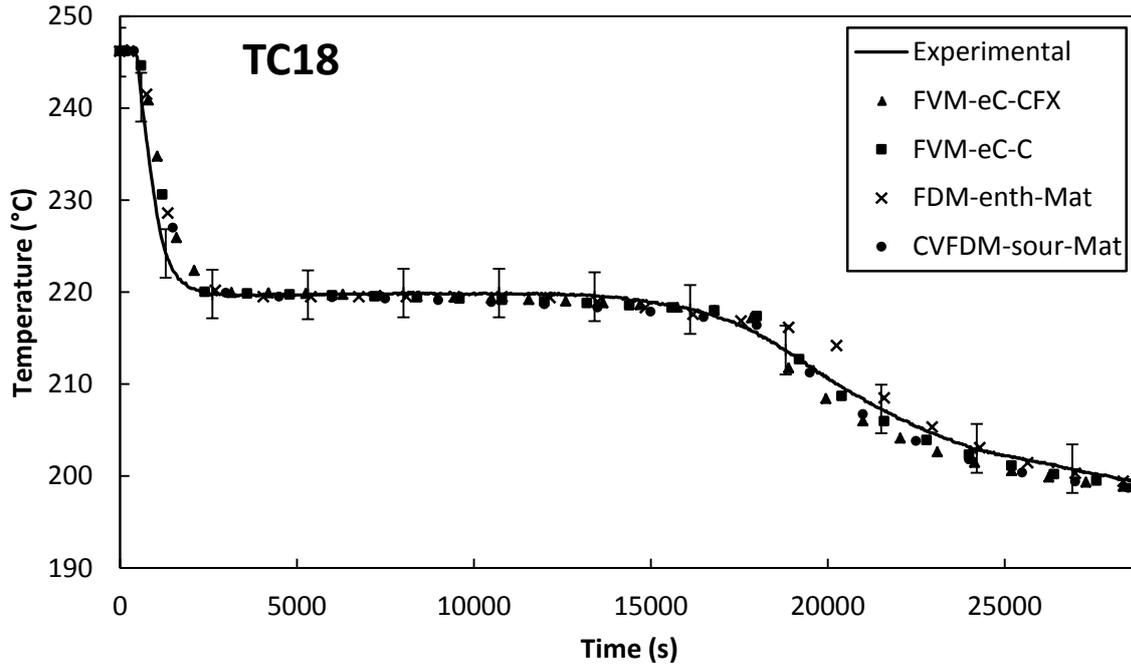


Fig. 9: Results of the numerical models compared to experimental results at measuring point TC18. Dotted lines represent the simulation results; the continuous line represents the experimental results.

Table 2 gives the single values of D and ξ of the investigated numerical models at the four measurement points and the values averaged over these four points. It can be concluded that all numerical models show a good agreement to experimental results. 97.11 % of the results from the commercial code FVM-eC-CFX can be found within the uncertainty interval θ , whereas its averaged overall deviation from experimental data is the highest of the compared models with $D = 1.16 \text{ }^\circ\text{C}$. The CVFDM-sour-Mat model has the smallest deviation ($D = 1.07 \text{ }^\circ\text{C}$) from all analyzed models. But generally, differences in D between all the numerical models are small.

Higher deviations reducing the averaged value for $\xi = 94.28 \text{ } \%$ of the model FVM-eC-C are mainly due to deviations after starting the discharging process in the area of sensible heat transfer before reaching the melting temperature, especially at position TC12, see Fig. 6. Within the FDM-enth-Mat model, the comparably low value of $\xi = 94.77 \text{ } \%$ is mainly due to a slower temperature drop at the end of the phase change process compared to experimental results, e.g. see the end of the latent phase in Fig. 9. The accuracy of the FDM-enth-Mat model in this context is the lowest of the here regarded numerical models. However, all numerical models have a satisfactory accuracy, taking the general uncertainties regarding measurements in latent heat storage and the assumptions in the simulation model into account. For the measuring point TC40, all simulation results of all models are within the uncertainty interval θ , which can be seen in Fig. 7 and Table 2.

The next section will analyze the number of necessary iterations per time step for the evaluation of the convergence behavior of the numerical models.

Table 2: Temperature deviation D and percentage of results within uncertainty interval ξ for all models, measuring points and averaged values

Model	TC12		TC40		TC17		TC18		Average	
	D ($^{\circ}C$)	ξ (%)								
FVM-eC-CFX	1.30	97.40	1.08	100	1.12	95.84	1.12	95.21	1.16	97.11
FVM-eC-C	1.75	83.87	1.09	100	0.88	97.09	0.81	96.15	1.13	94.28
FDM-enth-Mat	1.61	96.36	0.89	100	1.00	93.44	0.92	89.28	1.11	94.77
CVFDM-sour-Mat	1.17	96.88	0.86	100	1.11	95.84	1.13	95.21	1.07	96.98

4.3 Convergence behavior

Within this section, the convergence behavior of the investigated numerical models is examined. Due to the non-linear physical equations of the phase change process and phase dependent material properties, all numerical models use an iteration process to obtain their results. The kind of iteration system that is used by each model is described in the corresponding part for each model in Section 2. The number of required iterations per time step influences the necessary calculation time and therefore is an important result. Two different time steps over the whole discharging process are analyzed. Each time step converges after the temperature change of all mesh elements is less than a tolerance value of $1 \cdot 10^{-5} K$ from one iteration step to the next, see Section 2. The number of necessary iterations is recorded for each time step $\Delta t = 1 s$ and $\Delta t = 5 s$, accumulated and averaged by dividing the total amount of iterations by the number of time steps. The results are summarized in Table 3.

Table 3: Averaged number of iterations per time steps of $\Delta t = 1s$ and $\Delta t = 5s$ for the numerical models

Model	Σ Iterations / $\Sigma(\Delta t = 1s)$	Σ Iterations / $\Sigma(\Delta t = 5s)$	Increase (%)
FVM-eC-CFX	1.24	5.01	304.03
FVM-eC-C	11.91	54.20	355.08
FDM-enth-Mat	10.25	41.60	305.85
CVFDM-sour-Mat	1.80	2.17	20.56

The models FVM-eC-CFX and CVFDM-sour-Mat show the smallest amount of necessary iterations compared to the other two models. The commercial FVM-eC-CFX code uses an iterative solver with an optimized iteration algorithm aiming for high efficiency. The CVFDM-sour-Mat model uses a direct solver from the MATLAB internal function *mldivide* for solving the equation system that has been previously linearized. This internal MATLAB function is optimized for solving different types of equation systems efficiently. The execution and decision how the equation system is solved most efficiently is achieved automatically by MATLAB. The deployment of optimization algorithms within these two models results in a fast convergence behavior.

The models FVM-eC-C and FDM-enth-Mat do not make use of any optimized iteration schemes. The pure Gauss-Seidel-Iteration algorithm without relaxation is implemented and used for calculations presented in this article. Using pure Gauss-Seidel leads to a much higher necessary number of iterations until the convergence criterion is fulfilled. This leads to higher computational effort.

The impressive efficiency of using the direct solver in the above mentioned internal MATLAB function together with linearization of the system of equations can be seen by comparing the necessary amount of iterations of both time steps. The first three models depicted in Table 3 do not use this method. By increasing the time step from $\Delta t = 1 \text{ s}$ to $\Delta t = 5 \text{ s}$, the number of necessary iterations increases by 304.03 % – 355.08 %, see Table 3. On the other hand, for the CVFDM-sour-Mat model with a direct solver and linearized source term method, the rise is only 20.56 %. This promises high computational efficiency, even for large time steps and simulation models with a high number of elements.

Nevertheless, the overall computation time does not only depend on the number of iterations, but also on the program the models are implemented in and which programming language is used for that. These aspects are evaluated within the following section.

4.4 Computational efficiency

In order to evaluate the computational efficiency of the numerical models, four different mesh sizes with a significantly increasing number of total elements and a time step of $\Delta t = 5 \text{ s}$ are calculated with each numerical model, see Table 4.

The CPU specific absolute time to complete each calculation is timed for each numerical model. Evaluating these calculation times for each model and mesh size allows for the investigation of computational efficiency for small and large simulation models with few and many discretization elements. Based on these results, the correlation between the number of elements and resulting calculation time for the numerical models is studied.

Different computers are used for calculations and time measurements. Therefore, the absolute time measurements cannot be compared to each other directly. As a first step, the results are normalized and their analysis is carried out dimensionlessly. This is achieved by dividing the specific time results from each model for all mesh sizes by the specific calculation time of the smallest mesh size. Results of this procedure can be seen in Table 4 in the lines “Normalized time”. These are visualized in Fig. 10. Due to the normalization process, the curves of all numerical models start with a value of “1”.

The FVM-eC-C model shows a significantly higher increase than the other models. It increases from the smallest to the largest model by a factor of 1788.79. The two numerical models programmed in MATLAB (FDM-enth-Mat and CVFDM-sour-Mat) show a similar behavior to each other and have an increase by a factor of 37.59 and 42.63, respectively. This is much less than the C-based FVM-eC-C

model and higher than the ANSYS-CFX based FVM-eC-CFX model. Here, the normalized calculation time increases by a factor of 5.81. Since MATLAB specializes on handling vectors and matrices, its code is highly optimized for linear algebra. MATLAB automatically depicts computationally efficient algorithms and provides an automatic efficient memory management. To enhance performance in the FVM-eC-C model, similar algorithms would have to be implemented by coding them manually or by implementing comparable input packages. Since this was not done within the FVM-eC-C model, the handling of huge matrices within the two MATLAB-based models is more efficient than the FVM-eC-C model. ANSYS-CFX’s solver treats the basing equation systems and matrices efficiently, since FVM-eC-CFX shows the smallest increase of normalized time of all investigated numerical models.

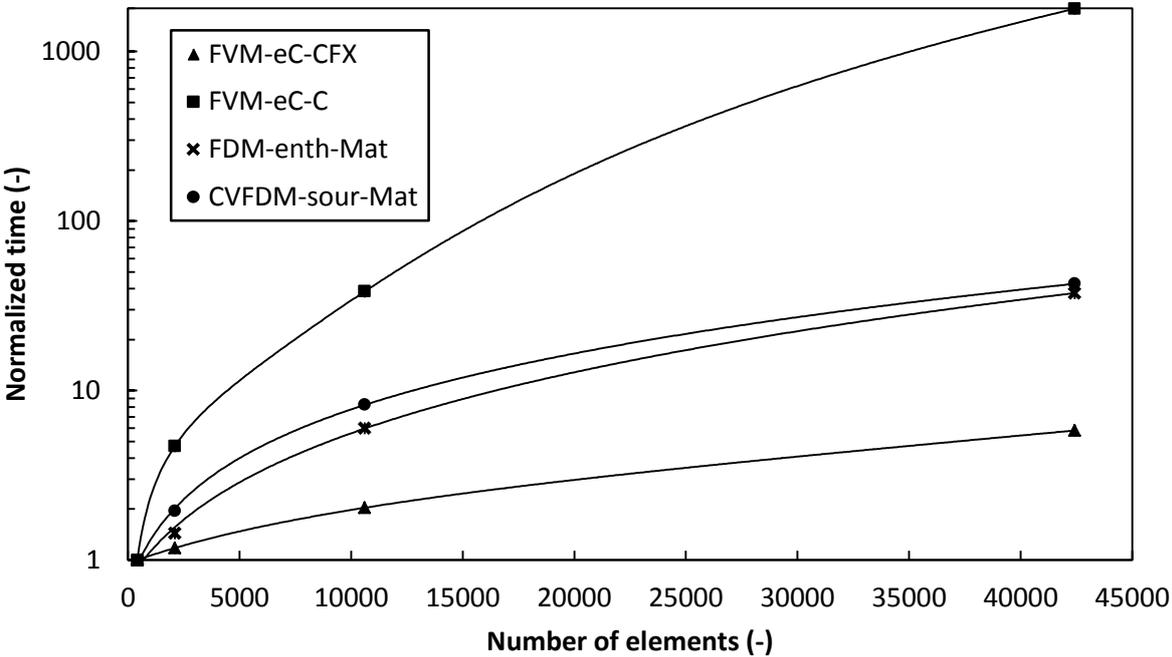


Fig. 10: Normalized calculation time over the number of elements for the investigated numerical models

The comparison of the computational efficiency was carried out by normalized values in this first step. In the next step, the specific CPU time is scaled to make the absolute calculation time comparable between the different computers. Actual benchmark tests for each CPU used within this investigation can be found at [41] with a daily update. The benchmarks given are based on a test algorithm that is run on a high number of samples of different computers with the corresponding CPU. The CPU benchmarks for each CPU are comparable to each other and are given in Table 4. The model FVM-eC-CFX is run on a computer using all eight logical cores of its CPU, whereas the other models are restricted to the use of only one core. To regard this circumstance, the CPU benchmark value for the FVM-eC-CFX model, the “Averaged CPU Mark” calculated for the whole CPU including all cores is used. For the other models, the value for “Single Thread Rating” goes into the

calculations. Being aware of the correlated uncertainties, a high overall error of $\pm 30\%$ is assumed. As is shown later, even errors of this size do not influence the results significantly and a tendency is still clear.

To scale the computational time of each model, the specific CPU time is multiplied by a scaling factor. This factor is calculated by dividing the CPU benchmark of all computers by the biggest one; in this case, by the CPU benchmark of the FVM-eC-CFX model. All necessary information such as the different CPU types, benchmarks, scaling factors and scaled calculation times of each model can be found in Table 4. The results of the scaled calculation time together with the error bars of $\pm 30\%$ are given in Fig. 11.

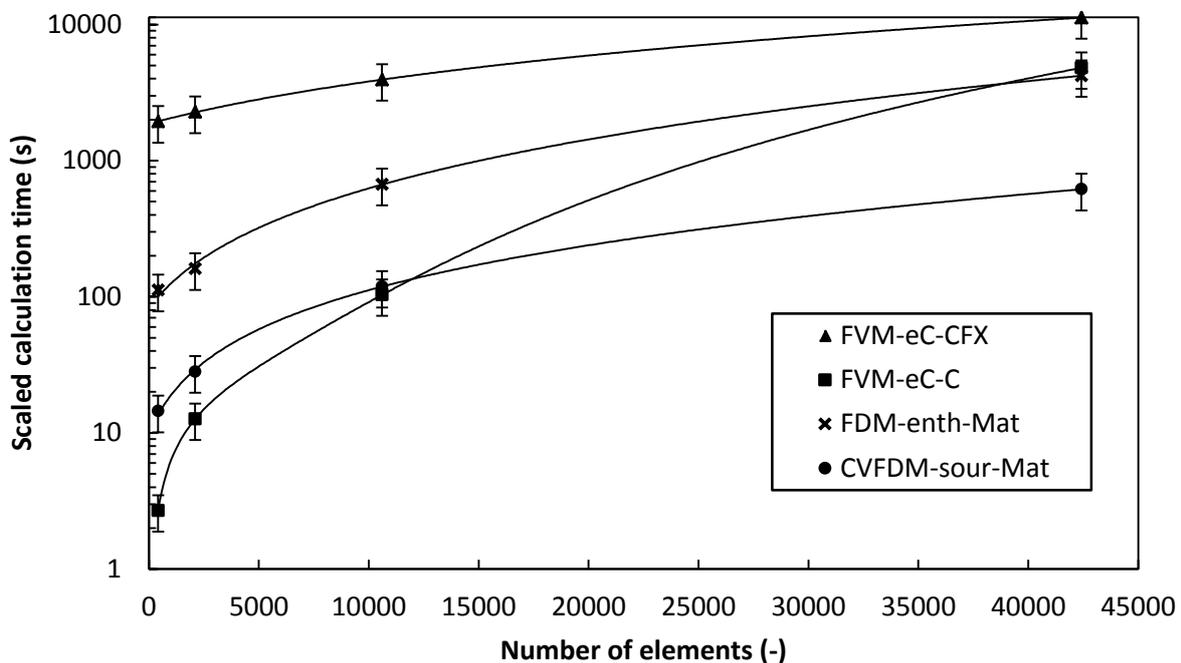


Fig. 11: Scaled calculation time over the number of elements for the investigated numerical models

Comparing Fig. 11 with Fig. 10, the forms of the curves are the same. In Fig. 11, the curves do not all start at 1. Even though the FVM-eC-CFX has a comparably flat shape, it starts at a high scaled calculation time. Even for the biggest model, with 42400 elements, the small number of iterations and the flat curve cannot compensate the overall high calculation time.

On the other hand, the FVM-eC-C model has a steep slope of the calculation time curve over the number of elements. For models with a small number of elements, it is the fastest one. However, with the number of model elements increasing, the computational effort increases more than the other models. The curve first intersects the curve from the CVFDM-sour-Mat model shortly after the model size of 10605 elements. Before the simulation model with most elements, it also intersects the curve of the FDM-enth-Mat model. Comparing the two MATLAB-based models FDM-enth-Mat

and CVFDM-sour-Mat to each other, the influence of linearization of the non-linear equation system and the use of a direct solver in MATLAB's internal *mldivide* function can be observed. This brings time savings of about a factor of 5 - 7. Additionally, it can be seen that the assumed error resulting from the CPU benchmark tests, illustrated by error bars in Fig. 11, do not affect the tendency of the results.

Table 4: Information about used CPUs, specific calculation times, scaled times and scaling factors for the investigated mesh sizes and models with the time step $\Delta t = 5s$

Model and CPU		Mesh size	20x21	100x21	505x21	1010x42
		Number of elements	420	2100	10605	42420
FVM-eC-CFX	Intel® Core™ i7-860 @2.8GHz	CPU specific time (s)	1933.03	2277.01	3933.66	11228.46
		Normalized time (-)	1.00	1.18	2.03	5.81
		CPU benchmark (-) [41]	5101 (Average CPU Mark; 1528 Samples)			
		Scaling factor (-)	1.00			
		Scaled time (s)	1933.03	2277.01	3933.66	11228.46
FVM-eC-C	AMD Phenom™ II X6 1055T @2.8GHz	CPU specific time (s)	12.31	57.94	472.94	22020.00
		Normalized time (-)	1.00	4.71	38.42	1788.79
		CPU benchmark (-) [41]	1114 (Single Thread Rating; 1208 Samples)			
		Scaling factor (-)	0.22			
		Scaled time (s)	2.71	12.75	104.05	4844.40
FDM-enth-Mat	Intel® Core™ i7-2600 @3.4GHz	CPU specific time (s)	298.01	428.12	1786.47	11201.98
		Normalized time (-)	1.00	1.44	5.99	37.59
		CPU benchmark (-) [41]	1914 (Single Thread Rating; 3088 Samples)			
		Scaling factor (-)	0.38			
		Scaled time (s)	113.24	162.69	678.86	4256.75
CVFDM-sour-Mat	Intel® Core™ i7-3770 @3.4 GHz	CPU specific time (s)	35.60	69.40	293.30	1517.70
		Normalized time (-)	1.00	1.95	8.24	42.63
		CPU benchmark (-) [41]	2070 (Single Thread Rating; 5373 Samples)			
		Scaling factor (-)	0.41			
		Scaled time (s)	14.60	28.45	120.25	622.26

The resulting overall calculation time of the mesh size 20x21 depending on two time steps $\Delta t = 1s$ and $\Delta t = 5s$ can be seen in Table 5. Here, the model-specific and, according to the data given in Table 4, scaled calculation time for the whole discharging cycle of $t = 28800s$ is shown for the cases with time steps of $\Delta t = 1s$ and $\Delta t = 5s$. With an increase in the time step from $\Delta t = 1s$ to $\Delta t = 5s$, the necessary number of time steps within the simulations decreases by a factor of five from 28800 to 5760. The overall calculation time of all numerical models, however, decreases less. Table 5 gives the specific decrease of the overall calculation time for all numerical models regarded within this

study. Both the FVM-eC-C and the FDM-enth-Mat show a moderate reduction of the overall calculation time by 27.93% and 30.00%, respectively. These models base on the Gauss-Seidel iteration scheme without an optimization algorithm and therefore show a similar behavior. The overall calculation time for the FVM-eC-CFX is reduced by 54.50% while increasing the time step from $\Delta t = 1s$ to $\Delta t = 5s$, s. Table 5. The CVFDM-sour-Mat model's overall calculation time is reduced by 77.00%. As a consequence, the overall calculation time is reduced for all numerical models regarded in this article by increasing the time step. However, the difference of this reduction between the numerical models is less than it can be expected from Table 3. Here, the number of necessary iterations per time step for the CVFDM-sour-Mat model increases from $\Delta t = 1s$ to $\Delta t = 5s$ by 20.56%. On the other hand, this increase is much higher for the other regarded models with 304.03% – 355.08%, see Table 3. This leads to the finding, that the *mldivide* function of MATLAB implemented into the CVFDM-sour-Mat model is a very efficient solver, but realizing one iteration takes more computational effort than e.g. compared with the computational effort necessary for one Gauss-Seidel iteration. That is why this model shows its strength by minimizing the amount of necessary time steps through step enlargement.

Table 5: Comparison of the scaled overall calculation time of the mesh size 20x21 depending on the time steps $\Delta t = 1s$ and $\Delta t = 5s$. For scaling parameters, see Table 4.

Model	Overall calculation time, $\Delta t = 1s$ (s)	Overall calculation time, $\Delta t = 5s$ (s)	Decrease (%)
FVM-eC-CFX	4248.43	1933.03	54.50
FVM-eC-C	3.76	2.71	27.93
FDM-enth-Mat	161.77	113.24	30.00
CVFDM-sour-Mat	63.47	14.60	77.00

It can be concluded from this section that different programs have their specific advantages and disadvantages regarding computational efficiency for modeling phase change processes. The program and numerical model have to be chosen according to the specific size of a simulation model and to the duration of the process and the corresponding necessary number of time steps in order to achieve a highest possible performance. There is not one fastest program for all needs.

5 Conclusions

In this article, four numerical models with three different discretization methods (FDM, CVFDM, FVM) and three different approaches to modeling phase change (enthalpy method, source term method, effective heat capacity method) were implemented into three different software packages

(MATLAB, C, ANSYS CFX) with corresponding programming languages. The goal of the study was the comparison of these four numerical models regarding their numerical performance. To make the numerical models comparable to each other, the same features, e.g. a melting range of 4 K, the same boundary conditions and the same mesh sizes and time steps were implemented. To validate each numerical model, a simulation model of DLR's flat plate latent heat storage was developed. The idealization of the real problem to the simulation model including an effective thermal heat conductivity model to regard convection inside the PCM was described. The numerical results of each model were then compared with experimental results. In a further step, the performance of each numerical model was evaluated regarding accuracy, convergence behavior and computational efficiency.

For the regarded case, all numerical models show a good agreement with the experimental results. The numerical models based on ANSYS-CFX and MATLAB with a linearized equation system have the lowest number of necessary iterations per time step due to their efficient solvers. However, the overall calculation time for a coarse grid with a small number of elements is the lowest for the C-based model, even though a higher number of iterations caused by the implemented Gauss-Seidel iteration method is necessary. With an increase of the element number of the simulation model, the overall calculation time of the C-based model increases greatly since the matrix operations and memory management are not optimized as in MATLAB or commercial codes. The increase of overall calculation time of the MATLAB-based models with increasing the model size is less than the C-based model, due to MATLAB's capability to solve equation systems efficiently. For bigger models with many elements, the MATLAB-based numerical models show a less time consuming behavior than the C-based model. The ANSYS-CFX model has the highest overall calculation time of the numerical models regarded in this article. Nevertheless, realizing one iteration with MATLAB's *mldivide* function implemented into the CVFDM-sour-Mat model requires a higher computational effort than e.g. one Gauss-Seidel iteration. That is why this model tendentially demonstrates its strength for simulation models with a high number of elements with big time steps. The C-based model FVM-eC-C is recommended out of the numerical models regarded here for small simulation models with small time steps.

Acknowledgments

This work was partially funded by the nextPCM project supported by the German Federal Ministry for the Environment, Nature Conservation and Nuclear Safety. Moreover, the work was partially funded by the Spanish government (ENE2011-22722 and ULLE10-4E-1305). The authors Alvaro de Gracia and Luisa F. Cabeza would like to thank the Catalan Government for the quality accreditation given to

their research group (2014-SGR-123). The research leading to parts of these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement PIRSES-GA-2013-610692 (INNOSTORAGE). Alvaro de Gracia would like to thank the Education Ministry of Chile for Grant PMI ANT1201.

Literature

- [1] A. Castillo and D. F. Gayme, "Grid-scale energy storage applications in renewable energy integration: A survey," *Energy Convers. Manag.*, vol. 87, pp. 885–894, 2014, DOI: 10.1016/j.enconman.2014.07.063.
- [2] D. Laing, C. Bahl, T. Bauer, D. Lehmann, and W.-D. Steinmann, "Thermal energy storage for direct steam generation," *Sol. Energy*, vol. 85, no. 4, pp. 627–633, Apr. 2011, DOI: 10.1016/j.solener.2010.08.015.
- [3] V. Alexiades and A. D. Solomon, *Mathematical Modeling of Melting and Freezing Processes*. Washington DC: Hemisphere Publishing Corporation, 1993.
- [4] G. Ziskind, "Advances in Thermal Energy Storage Systems," in *Advances in Thermal Energy Storage Systems: Methods and Applications*, L. F. Cabeza, Ed. Cambridge, Waltham, Kidlington: Woodhead Publishing, 2015, pp. 307–323.
- [5] H. Mehling and L. F. Cabeza, *Heat and Cold Storage with PCM - An up to date introduction into basics and applications*. Berlin, Heidelberg: Springer-Verlag, 2008.
- [6] T. Bauer, "Approximate analytical solutions for the solidification of PCMs in fin geometries using effective thermophysical properties," *Int. J. Heat Mass Transf.*, vol. 54, no. 23–24, pp. 4923–4930, Nov. 2011, DOI: 10.1016/j.ijheatmasstransfer.2011.07.004.
- [7] N. H. S. Tay, F. Bruno, and M. Belusko, "Experimental validation of a CFD model for tubes in a phase change thermal energy storage system," *Int. J. Heat Mass Transf.*, vol. 55, no. 4, pp. 574–585, 2012, DOI: 10.1016/j.ijheatmasstransfer.2011.10.054.
- [8] N. H. S. Tay, F. Bruno, and M. Belusko, "Comparison of pinned and finned tubes in a phase change thermal energy storage system using CFD," *Appl. Energy*, vol. 104, pp. 79–86, 2013, DOI: 10.1016/j.apenergy.2012.10.040.
- [9] N. H. S. Tay, M. Belusko, M. Liu, and F. Bruno, "Investigation of the effect of dynamic melting in a tube-in-tank PCM system using a CFD model," *Appl. Energy*, vol. 137, pp. 738–747, 2015, DOI: 10.1016/j.apenergy.2014.06.060.
- [10] M. Jaworski, P. Łapka, and P. Furmański, "Numerical modelling and experimental studies of thermal behaviour of building integrated thermal energy storage unit in a form of a ceiling panel," *Appl. Energy*, vol. 113, pp. 548–557, 2014, DOI: 10.1016/j.apenergy.2013.07.068.

- [11] A. Joulin, Z. Younsi, L. Zalewski, S. Lassue, D. R. Rouse, and J. P. Cavrot, "Experimental and numerical investigation of a phase change material: Thermal-energy storage and release," *Appl. Energy*, vol. 88, no. 7, pp. 2454–2462, 2011, DOI: 10.1016/j.apenergy.2011.01.036.
- [12] a. C. Marques, G. F. Davies, J. a. Evans, G. G. Maidment, and I. D. Wood, "Theoretical modelling and experimental investigation of a thermal energy storage refrigerator," *Energy*, vol. 55, pp. 457–465, 2013, DOI: 10.1016/j.energy.2013.03.091.
- [13] W. Wang, H. Li, S. Guo, S. He, J. Ding, J. Yan, and J. Yang, "Numerical simulation study on discharging process of the direct-contact phase change energy storage system," *Appl. Energy*, vol. 150, pp. 61–68, 2015, DOI: 10.1016/j.apenergy.2015.03.108.
- [14] J. P. Arzamendia Lopez, F. Kuznik, D. Baillis, and J. Virgone, "Numerical modeling and experimental validation of a PCM to air heat exchanger," *Energy Build.*, vol. 64, pp. 415–422, 2013, DOI: 10.1016/j.enbuild.2013.04.017.
- [15] J. Borderon, J. Virgone, and R. Cantin, "Modeling and simulation of a phase change material system for improving summer comfort in domestic residence," *Appl. Energy*, vol. 140, pp. 288–296, 2015, DOI: 10.1016/j.apenergy.2014.11.062.
- [16] X. Sun, Q. Zhang, M. a. Medina, and S. Liao, "Performance of a free-air cooling system for telecommunications base stations using phase change materials (PCMs): In-situ tests," *Appl. Energy*, vol. 147, pp. 325–334, 2015, DOI: 10.1016/j.apenergy.2015.01.046.
- [17] E. Tumilowicz, C. L. Chan, P. Li, and B. Xu, "An enthalpy formulation for thermocline with encapsulated PCM thermal storage and benchmark solution using the method of characteristics," *Int. J. Heat Mass Transf.*, vol. 79, pp. 362–377, 2014, DOI: 10.1016/j.ijheatmasstransfer.2014.08.017.
- [18] D. Zhao and G. Tan, "Numerical analysis of a shell-and-tube latent heat storage unit with fins for air-conditioning application," *Appl. Energy*, vol. 138, pp. 381–392, 2015, DOI: 10.1016/j.apenergy.2014.10.051.
- [19] A. De Gracia, E. Oró, M. M. Farid, and L. F. Cabeza, "Thermal analysis of including phase change material in a domestic hot water cylinder," *Appl. Therm. Eng.*, vol. 31, no. 17–18, pp. 3938–3945, 2011, DOI: 10.1016/j.applthermaleng.2011.07.043.
- [20] A. De Gracia, L. Navarro, A. Castell, and L. F. Cabeza, "Numerical study on the thermal performance of a ventilated facade with PCM," *Appl. Therm. Eng.*, vol. 61, no. 2, pp. 372–380, 2013, DOI: 10.1016/j.applthermaleng.2013.07.035.
- [21] A. Ayasoufi, "Numerical Simulation of Heat Conduction with Melting and/or Freezing by Space-Time Conversation Element and Solution Element Method," University of Toledo, 2004.
- [22] A. B. Crowley, "Numerical Solution of Stefan Problems," *Int. J. Heat Mass Transf.*, vol. 21, pp. 215–219, 1978, DOI: 10.1016/0017-9310(78)90225-9.
- [23] Y. Dutil, D. R. Rouse, N. Ben Salah, S. Lassue, and L. Zalewski, "A review on phase-change materials: Mathematical modeling and simulations," *Renew. Sustain. Energy Rev.*, vol. 15, no. 1, pp. 112–130, Jan. 2011, DOI: 10.1016/j.rser.2010.06.011.

- [24] V. Voller and M. Cross, "Accurate Solutions of Moving Boundary Problems Using the Enthalpy Method," *Int. J. Heat Mass Transf.*, vol. 24, pp. 545–556, 1981, DOI: 10.1016/0017-9310(81)90062-4.
- [25] P. Lamberg, R. Lehtiniemi, and A. M. Henell, "Numerical and experimental investigation of melting and freezing processes in phase change material storage," *Int. J. Therm. Sci.*, vol. 43, pp. 277–287, 2004, DOI: 10.1016/j.ijthermalsci.2003.07.001.
- [26] V. R. Voller and C. R. Swaminathan, "Eral Source-Based Method for Solidification Phase Change," *Numer. Heat Transf. Part B Fundam.*, vol. 19, no. 2, pp. 175–189, 1991, DOI: 10.1080/10407799108944962.
- [27] V. R. Voller, "Fast Implicit Finite-Difference Method for the Analysis of Phase Change Problems," *Numer. Heat Transf.*, vol. 17, no. Part B, pp. 155–169, 1990.
- [28] J. Peiro and S. Sherwin, "Finite Difference , Finite Element and Finite Volume Methods for Partial Differential Equations," in *Handbook of Materials Modeling*, vol. I, S. Yip, Ed. Dordrecht, Berlin, Heidelberg, New York: Springer, 2005.
- [29] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. Washington, New York, London: Hemisphere Publishing Corporation, 1980.
- [30] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*. Berlin Heidelberg: Springer, 2001.
- [31] S. N. Al-Saadi and Z. (John) Zhai, "Systematic evaluation of mathematical methods and numerical schemes for modeling PCM-enhanced building enclosure," *Energy Build.*, vol. 92, pp. 374–388, 2015, DOI: 10.1016/j.enbuild.2015.01.044.
- [32] M. M. Farid, "A new approach in the calculation of heat transfer with phase change," in *9th International Congress on Energy and Environment*, 1989.
- [33] H. Pointner, W.-D. Steinmann, and M. Eck, "Introduction of the PCM Flux Concept for Latent Heat Storage," *Energy Procedia*, vol. 57, pp. 643–652, 2014, DOI: 10.1016/j.egypro.2014.10.219.
- [34] S. J. Chapman, *Essentials of MATLAB Programming*, 2nd ed. Stamford: CENGAGE Learning, 2009.
- [35] D. J. Hartfield, *Matrix Theory and Applications with MATLAB*. Texas A&M University: CRC Press Inc., 2001.
- [36] N. H. S. Tay, F. Bruno, and M. Belusko, "Experimental validation of a CFD and an ϵ -NTU model for a large tube-in-tank PCM system," *Int. J. Heat Mass Transf.*, vol. 55, no. 21–22, pp. 5931–5940, 2012, DOI: 10.1016/j.ijheatmasstransfer.2012.06.004.
- [37] M. Johnson, M. Fiss, T. Klemm, and M. Eck, "Test and Analysis of a Flat Plate Latent Heat Storage Design," *Energy Procedia*, vol. 57, pp. 662–671, 2014, DOI: 10.1016/j.egypro.2014.10.221.

- [38] J. Vogel, M. Johnson, M. Eck, and D. Laing, "Numerical analysis of natural convection in a latent heat thermal energy storage system containing rectangular enclosures," in *Eurotherm Seminar N99 - Advances in Thermal Energy Storage*, 2014.
- [39] M. M. Farid and R. M. Husian, "An electrical storage heater using the phase-change method of heat storage," *Energy Convers. Manag.*, vol. 30, no. 3, pp. 219–230, 1990, DOI: 10.1016/0196-8904(90)90003-H.
- [40] T. Bauer, D. Laing, and R. Tamme, "Overview of PCMs for Concentrated Solar Power in the Temperature Range 200 to 350C," *Adv. Sci. Technol.*, vol. 74, pp. 272–277, Oct. 2010, DOI: 10.4028/www.scientific.net/AST.74.272.
- [41] "PassMark® Software Pty Ltd." [Online]. Available: <http://www.cpubenchmark.net/>. [Accessed: 20-Aug-2015].