TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Werkzeugmaschinen und Fertigungstechnik am
Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*)

Adaption of Numerical Methods to the Specific Requirements of the Thermal Simulation in Laser Beam Melting

Christian Zeller

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Oskar J. Haidn
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Michael F. Zäh 2. Prof. Dr.-Ing. Rainer Callies

Die Dissertation wurde am 09.05.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 08.11.2019 angenommen.

"The highest reward for a person's toil is not what they get for it,
but what they become by it."
JOHN RUSKIN[1]

[1] **John Ruskin** (* February 8th, 1819 - † January 20th, 1900), British writer, painter, art historian and social philosopher.

Acknowledgment

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) der Technischen Universität München.

Herrn Prof. Dr.-Ing. Michael F. Zäh und Herrn Prof. Dr.-Ing. Gunther Reinhart, den Leitern dieses Instituts, gilt mein erster Dank für die wohlwollende Förderung und großzügige Unterstützung meiner Arbeit. Bei Herrn Prof. Dr.-Ing. Michael F. Zäh bedanke ich mich darüber hinaus besonders für die Betreuung meiner Arbeit. Vielen Dank vor allem für das, mir als Naturwissenschaftler, geschenkte Vertrauen zur Promotion in der Ingenieurwissenschaft.

Weiterhin gilt mein Dank Herrn Prof. Dr.-Ing. Oskar J. Haidn, Extraordinarius für Raumfahrtantriebe am Lehrstuhl für Flugantriebe der Technischen Universität München, für die Übernahme des Vorsitzes der Prüfungskommission. Für den Forschungsaufenthalt am Computational Manufacturing and Materials Research Lab der University of California at Berkeley bedanke ich mich zudem bei Herrn Ph.D. Prof. Tarek I. Zohdi.

Ein großer Dank gilt Herrn Prof. Dr.rer.nat. habil. Dr.-Ing. Rainer Callies vom Lehrstuhl für Numerische Mathematik der Technischen Universität München. Sie haben mich bereits in meinem Masterstudium begleitet und insbesondere meine Masterthesis betreut. Vielen Dank für Ihre stets offene Tür und die zahlreichen Diskussionen, nicht nur im Studium und bei der Betreuung meiner Masterthesis, sondern auch während der Anfertigung meiner Dissertation. Es ist mir eine besondere Freude, dass Sie das Koreferat dieser Arbeit übernommen haben.

Bei Herrn Dr.-Ing. Harald Krauss und Herrn Dr.rer.nat. Andreas Denner bedanke ich mich für die gründliche Durchsicht dieser Arbeit und die wertvollen Hinweise. Darüber hinaus bedanke ich mich bei allen Kolleginnen und Kollegen aus den Forschungsgruppen zur Additiven Fertigung am *iwb* und Fraunhofer IGCV für die freundschaftliche und konstruktive Arbeitsatmosphäre. Ein spezieller Dank geht an meinen ehemaligen Studenten Lukas Bante für seine weitreichende Unterstützung bei der Implementierung der entwickelten Methoden in CalculiX CrunchiX. Ganz besonders bedanke ich mich bei Herrn Fabian Bayerlein für die gemeinsame Zusammenarbeit in meinen 4 Jahren am Institut, für die vielen fachlichen Diskussionen, aber auch für die Gespräche über Themen abseits der Arbeit.

Ich bedanke mich auch bei meinem ehemaligen Lehrer Herrn Christian Böck. Mit Ihrer herausfordernden Art haben Sie meinen Ehrgeiz, aber auch meine Begeisterung, für die Mathematik geweckt.

Ein ganz besonderer Dank richtet sich an meine Familie, die mich auf meinem Weg immer unterstützt hat. Zuletzt gilt mein größter Dank Agnes Marz. Du warst mir in den vergangenen Jahren stets der wichtigste Rückhalt und mein verlässlicher Rückzugsort. Vielen Dank für Dein immerwährendes Verständnis für die Anfertigung dieser Arbeit, vor allem aber für die vielen wunderbaren gemeinsamen Momente.

Augsburg, den 15. November 2019 *Christian Zeller*

Contents

Acronyms and Abbreviations

AAM	AscentAM
ADE	Alternating Direction Explicit
ADI	Alternating Direction Implicit
AM	Additive Manufacturing
BFS	Breadth-First Search
ccx	CalculiX CrunchiX
DFS	Depth-First Search
DMLS	Direct Metal Laser Sintering
DoF	Degree of Freedom
EBM	Electron Beam Melting
FCM	Finite Cell Method
FEM	Finite Element Method
FSAL	First Same As Last
GLS	Galerkin/Least-Squares
GPGPU	General Purpose Computation on Graphics Processing Unit
GPU	Graphics Processing Unit
IGA	Isogeometric Analysis
IMEX	implicit-explicit
IVP	Initial Value Problem
LBM	Laser Beam Melting
LBM-DSS	LBM-specific Direct Solving Strategy
LBM-MM	LBM Multirate Method
LC	Layer Compound
LRZ	Leibniz Supercomputing Centre
M&S	Modeling and Simulation
ODE	Ordinary Differential Equation
PBF	Powder Bed Fusion
PDC	Penetration Depth Condition
PDE	Partial Differential Equation
PXFEM	Parameterized eXtended Finite Element Method

SFEM	Spectral Finite Element Method
SLM	Selective Laser Melting
SUPC	Streamline Upwind/Petrov-Galerkin
TGM	Temperature Gradient Mechanism
XFEM	eXtended Finite Element Method

List of Symbols

Latin Lower Symbols

Symbol	Unit	Description
a	$\mathrm{m^2\,s^{-1}}$	Thermal diffusivity
a_s	$\mathrm{m^2\,s^{-1}}$	s^{th} thermal diffusivity
$\boldsymbol{b}$	$\mathrm{kg\,m^2\,s^{-3}}$	Right-hand side vector
$\boldsymbol{b}^m$	$\mathrm{kg\,m^2\,s^{-3}}$	Right-hand side vector of the system of equations in time step m
b	-	Counting variable for scan fields
c	$\mathrm{m^2\,s^{-2}\,K^{-1}}$	Specific heat capacity
c_{fac}	-	Factor for increasing number of required iterations i_{total}
c_h	-	Number of layers from the current top layer before the first layer of the component Φ_h
c_{rate}	-	Rate of convergence
c_{scal}	-	Scaling variable
c_κ	-	Change in the local uncertainty κ_w^m due to the application of the LBM-MM
c_{Y}	-	Upper growth rate of the time steps from the set Y^m
d_{b}	m	Beam diameter
d_{ed}	m	Width of the area with initial energy deposition
$d_{\text{ed},s}$	m	Width of the area with initial energy deposition, s
d_{l}	m	Layer thickness
d_{LC}	m	Thickness of a LC
d_{PD}	m	Penetration depth δ of the first time step Δt^0
d_{refine}	-	Refinement factor
d_{sv}	m	Distance of neighbored scan vectors

Symbol	Unit	Description
d_Θ	m	Maximum occurring element size of a mesh Θ
$\boldsymbol{e}$	-	Eigenvector of $\boldsymbol{A}$
e	$\sqrt{\text{m}}$K	Error
$\boldsymbol{f}$	-	Vector-valued function
$\boldsymbol{f}_s$	-	Partial vector-valued function of function $\boldsymbol{f}$
f	-	Scalar-valued function
f_{aux}	-	Auxiliary function
$\hat{f}_{\text{aux}}$	-	Auxiliary function
$f_{q,c_{\text{scal}}}$	-	Enrichment function linearly scaled over c_{scal} elements and gradient determined by q
f_{t^m}	-	Transformation function from normalized equidistant time steps to normalized optimal time steps with respect to a minimal uncertainty κ
$\hat{f}_{t^m}$	-	Help function of f_{t^m}
f_{model}	-	Model function
f_{m_Y}	-	Correlation between required number of time steps m_Y and minimum uncertainty κ
h	-	Counting variable for Φ_h
h_m	-	Number of active components Φ_h in time step t^m
h_{mr}	-	Number of components Φ_h
i	-	Outer iteration variable
i_{total}	-	Number of iterations i
j	-	Inner iteration variable
k	-	Row of a vector or matrix
k_d	-	Bandwidth of $\boldsymbol{A}^m$
k_h	-	Calculation interval of Φ_h
l	-	Column of a matrix
m	-	Number of a time step
m_Y	-	Total number of time steps m
n	-	Counting variable for $\boldsymbol{r}_n$
n_1	-	Scalar
n_2	-	Scalar
n_3	-	Scalar

Symbol	Unit	Description
n'_3	-	Scalar
n_Ξ	-	Number of nodes $\boldsymbol{r}_n$
n'_Ξ	-	Number of nodes $\boldsymbol{r}_n$
$\boldsymbol{n}$	-	Exterior normal
o	-	Second counting variable for $\boldsymbol{r}_n$
p_s	-	Parameter s of the regression analyses
$\boldsymbol{q}$	$\mathrm{kg\,m^2\,s^{-3}}$	Load vector
q	-	Variable determining the gradient of the function $f_{q,c_{\mathrm{scal}}}$
$\dot{q}$	$\mathrm{kg\,s^{-3}}$	Heat flux density
$\dot{q}_{\Gamma_\mathrm{N}}$	$\mathrm{kg\,s^{-3}}$	Prescribed heat flux density on the boundary Γ_N
q_k	$\mathrm{kg\,m^2\,s^{-3}}$	Item k of load vector $\boldsymbol{q}$
$\boldsymbol{r}$	m	Position vector
$\boldsymbol{r}^m$	$\mathrm{kg\,m^2\,s^{-3}}$	Residual flux vector in time step m
$\boldsymbol{r}_n$	-	Node n
r_k	m	Item k of the position vector $\boldsymbol{r}$
s	-	General counting variable
t	s	Time
$\bar{t}$	-	Temporal factor
t^0	s	Initial time
t_0	s	Calculation time of T_0
t^m	s	Time at time step m
t^{m_Y}	s	End time of the interval Y
t_P	s	Calculation time of the parameter combination P
$\bar{t}_P$	-	Relative calculation time of the parameter combination P
Δt_PD	s	Time step length that satisfies the PDC with equality
Δt	s	Equidistant time step length
Δt_dwell	s	Duration of the dwell time in LBM
Δt_end	s	End cooling time from T_pre to room temperature in LBM
$\Delta t_\mathrm{heating}$	s	Duration of the liquid state for a point in LBM

Symbol	Unit	Description
Δt^m	s	Time step length between time step m and time step $m+1$
Δt_h^m	s	Local time step length for the component Φ_h at the time t^m
$\Delta \bar{t}_P$	-	Relative reduction of the calculation time $\bar{t}_P$ of the parameter combination P
Δt_{travel}	s	Travel time of the laser in LBM
u	-	Counting variable for ϵ_u
u_Θ	-	Number of elements ϵ_u
u_ζ	-	Number of enriched elements with PXFEM
$\boldsymbol{v}$	$\mathrm{m\,s^{-1}}$	Velocity
v	-	Test function
v_{b}	$\mathrm{m\,s^{-1}}$	Scan velocity of the beam focus
v_{init}	-	Initial function for the derivation of the enrichment functions
v_n	-	Lagrange shape function in node $\boldsymbol{r}_n$ of the function space V^s
$v_{n,o}$	-	Parameterized enrichment function
$\dot{v}_{n,o}$	-	Time derivative of the enrichment function $v_{n,o}$
$v_{n,o}^*$	-	Modified test function
w	-	Counting variable for LC
w_h	-	Layers included in Φ_h
w_Ψ	-	Number of layers
$\boldsymbol{x}$	-	Exemplary vector
$\boldsymbol{x}_i$	-	Approximation of the vector $\boldsymbol{x}$ in iteration i
x	m	Spatial variable
$\bar{x}$	-	Spatial factor
x_k	-	Item k of the vector $\boldsymbol{x}$
$\boldsymbol{y}$	-	Exemplary vector
$\dot{\boldsymbol{y}}$	-	Time derivation of the vector $\boldsymbol{y}$
$\boldsymbol{y}_s$	-	Partial vector of the vector $\boldsymbol{y}$
$\dot{\boldsymbol{y}}_s$	-	Partial vector of the vector $\dot{\boldsymbol{y}}$
$\boldsymbol{y}^0$	-	Initial vector
$\boldsymbol{y}^m$	-	Numerical approximation of $\boldsymbol{y}$ in time step t^m

Symbol	Unit	Description
y	m	Spatial variable
y_k	-	Item k of the vector $\boldsymbol{y}$
$\Delta\boldsymbol{y}_i^m$	K	Auxiliary vector for forward and backward substitution in iteration i of time step m of the LBM-DSS
$\boldsymbol{z}$	-	Exemplary vector
z_n	-	One-dimensional node n
z	m	Spatial variable

Latin Upper Symbols

Symbol	Unit	Description
$\boldsymbol{A}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Coefficient matrix
$\boldsymbol{A}_{\mathrm{const}}$	-	Exemplary matrix
$\boldsymbol{A}_k$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Block matrix of $\boldsymbol{A}$ on the diagonal
$\boldsymbol{A}_{kl}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Block matrix of $\boldsymbol{A}$ on the off-diagonal
$\boldsymbol{A}^m$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Coefficient matrix of the time step m
$\boldsymbol{A}^m_{\mathrm{const}}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Constant splitting matrix by the LBM-DSS
$\boldsymbol{A}^m_k$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Block matrix of $\boldsymbol{A}^m$ on the diagonal
$\boldsymbol{A}^m_{kl}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Block matrix of $\boldsymbol{A}^m$ on the off-diagonal
$A_{\mathrm{const},kl}$	-	Item of row k and column l of the matrix $\boldsymbol{A}_{\mathrm{const}}$
$A^m_{\mathrm{const},kl}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Item of row k and column l of the matrix $\boldsymbol{A}^m_{\mathrm{const}}$
A_{kl}	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Item of row k and column l of the matrix $\boldsymbol{A}$
A^m_{kl}	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Item of row k and column l of the coefficient matrix $\boldsymbol{A}^m$
$\Delta\boldsymbol{A}^m$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Remaining temperature-dependent part of $\boldsymbol{A}^m$ by the LBM-DSS
$\boldsymbol{C}$	$\mathrm{kg\,m^2\,s^{-2}\,K^{-1}}$	Capacitance matrix
$\boldsymbol{C}_h$	$\mathrm{kg\,m^2\,s^{-2}\,K^{-1}}$	Reduced capacitance matrix $\boldsymbol{C}$ by the LBM-MM
$\boldsymbol{C}^Z$	$\mathrm{kg\,m^2\,s^{-3}}$	Extended capacitance portion of the enrichment function
$\boldsymbol{C}^\zeta$	$\mathrm{kg\,m^2\,s^{-3}}$	Additional capacitance portion of the enrichment function

Symbol	Unit	Description
C_{kl}	$\mathrm{kg\,m^2\,s^{-2}\,K^{-1}}$	Item of row k and column l of the matrix $\boldsymbol{C}$
C^s	-	Function space of the s-fold continuously differentiable functions
C_k^Z	$\mathrm{kg\,m^2\,s^{-3}}$	Item k of the vector $\boldsymbol{C}^Z$
$\boldsymbol{D}$	-	Diagonal matrix
H^s	-	Sobolev space with partial derivatives up to degree s
$\boldsymbol{I}$	-	Identity matrix
I^{t^m}	-	Index set of the active components in time step t^m
$\boldsymbol{K}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Conductance matrix
$\boldsymbol{K}_C$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Convection portion of the conductance matrix $\boldsymbol{K}$
$\boldsymbol{K}_h$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Reduced conductance matrix $\boldsymbol{K}$ by the LBM-MM
$\boldsymbol{K}^Z$	$\mathrm{kg\,m^2\,s^{-3}}$	Extended conductance portion of the enrichment function
$\boldsymbol{K}_C^Z$	$\mathrm{kg\,m^2\,s^{-3}}$	Extended convection portion of the enrichment function
$\boldsymbol{K}^\zeta$	$\mathrm{kg\,m^2\,s^{-3}}$	Additional conductance portion of the enrichment function
$\boldsymbol{K}_C^\zeta$	$\mathrm{kg\,m^2\,s^{-3}}$	Additional convection portion of the enrichment function
$\boldsymbol{K}^*$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Augmented conductance matrix
$K_{C,kl}$	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Item of row k and column l of the matrix $\boldsymbol{K}_C$
K_{kl}	$\mathrm{kg\,m^2\,s^{-3}\,K^{-1}}$	Item of row k and column l of the matrix $\boldsymbol{K}$
$K_{C,k}^Z$	$\mathrm{kg\,m^2\,s^{-3}}$	Item k of the vector $\boldsymbol{K}_C^Z$
K_k^Z	$\mathrm{kg\,m^2\,s^{-3}}$	Item k of the vector $\boldsymbol{K}^Z$
$\boldsymbol{L}$	-	Lower triangular matrix
$\boldsymbol{L}_k$	-	Block matrix of $\boldsymbol{L}$ on the diagonal
$\boldsymbol{L}_{kl}$	-	Block matrix of $\boldsymbol{L}$ on the off-diagonal
$\boldsymbol{L}^m$	-	Cholesky decomposition of the coefficient matrix $\boldsymbol{A}^m$
$\boldsymbol{L}_k^m$	-	Block matrix of $\boldsymbol{L}^m$ on the diagonal
$\boldsymbol{L}_{kl}^m$	-	Block matrix of $\boldsymbol{L}^m$ on the off-diagonal
$\bar{L}$	m	Characteristic length

Symbol	Unit	Description
L_{kl}	-	Item of row k and column l of the lower triangular matrix $\boldsymbol{L}$
L_{kl}^{m}	-	Item of row k and column l of the Cholesky decomposition $\boldsymbol{L}^{m}$
$\boldsymbol{M}$	-	Iteration Matrix
$\boldsymbol{M}_{i}^{m}$	-	Iteration matrix of the LBM-DSS in time step m and iteration i
$\boldsymbol{N}$	-	Matrix
P	$\mathrm{m}^2\,\mathrm{kg}\,\mathrm{s}^{-3}$	Laser power
P	-	Parameter combination of the LBM-MM
P_s	-	Function space of Lagrange functions of order s
P^{*}	-	Fastest parameter combination P of the LBM-MM
$\boldsymbol{T}$	K	Vector of all nodal temperatures T_n
$\dot{\boldsymbol{T}}$	$\mathrm{s}^{-1}\,\mathrm{K}$	Time derivation of the vector $\boldsymbol{T}$
$\boldsymbol{T}^{m}$	K	Vector of all nodal temperatures T_n^m at time step m
$\dot{\boldsymbol{T}}^{m}$	$\mathrm{s}^{-1}\,\mathrm{K}$	Time derivation of the vector $\boldsymbol{T}$ at time step m
$\boldsymbol{T}_{i}^{m}$	K	Vector of all nodal temperatures T_n^m at time step m in iteration i
$\boldsymbol{T}_{i,j}^{m}$	K	Vector of all nodal temperatures T_n^m at time step m in sub-iteration j of iteration i
T	K	Temperature
T^{0}	K	Initial temperature
$T^{0,s}$	K	s^{th} initial temperature
T_0	K	Reference solution for the results obtained by the LBM-MM
T_{ana}	K	Analytical solution of the heat equation
T_{ana}^{s}	K	s^{th} analytical solution of the heat equation
T_{fix}	K	Prescribed temperature for $\boldsymbol{A}_{\text{const}}^{m}$
T^{m}	K	Approximated temperature at time step m
T_{i}^{m}	K	Approximated temperature at time step m in iteration i
T_{n}^{m}	K	Approximated temperature at time step m of node $\boldsymbol{r}_n$

Symbol	Unit	Description
T_n	K	Semi-discrete temperature at node $\boldsymbol{r}_n$
$\dot{T}_n$	$\mathrm{s^{-1}\,K}$	Time derivation of T_n
T_{pre}	K	Preheating temperature of the build platform
T_{ref}	K	Reference temperature
T_{sol}	K	Solidus temperature
T_w	K	Wall temperature
T_{w_1,k_2}	K	Result of the LBM-MM with the parameters w_1 and k_2
T_{Γ_C}	K	Prescribed ambient temperature T on the boundary Γ_C
T_{Γ_D}	K	Prescribed temperature T on the boundary Γ_D
$\Delta\boldsymbol{T}_i^m$	K	Incremental improvement in iteration i of $\boldsymbol{T}_{i-1}^m$
$\boldsymbol{U}$	-	Upper triangular matrix
V^s	-	Function space of the piecewise continuous polynomials of order s
V^ζ	-	Function space of the enrichment functions $v_{n,o}$
Z_n	-	Set of indices that form a pair in the set Λ with n

Greek Lower Symbols

Symbol	Unit	Description
α	$\mathrm{kg\,s^{-3}\,K^{-1}}$	Heat transfer coefficient
α_{side}	$\mathrm{kg\,s^{-3}\,K^{-1}}$	Heat transfer coefficient at the Γ_{side}
α_{top}	$\mathrm{kg\,s^{-3}\,K^{-1}}$	Heat transfer coefficient on top of the Ω_{bp}
δ	m	Penetration depth
ε	K	Prescribed threshold
ε_{w_1,k_2}	K	Maximal absolute appearing nodal temperature deviations per cooling cycle of the LBM-MM with regard to T_0 averaged over the number of layers w_Ψ with the parameters w_1 and k_2

Symbol	Unit	Description
$\varepsilon_{\mathrm{tol}}$	K	Tolerated temperature deviation
ϵ_u	-	Element u
$\boldsymbol{\zeta}$	-	Vector of all parameterization variables $\zeta_{n,o}$
$\dot{\boldsymbol{\zeta}}$	-	Time derivation of the vector $\boldsymbol{\zeta}$
ζ	-	Parameterization variable
$\zeta_{n,o}$	-	Parameterization variable for enrichment function $v_{n,o}$
$\dot{\zeta}_{n,o}$	-	Time derivation of the parameterization variable $\zeta_{n,o}$
ζ_{opt}	-	Optimal parameterization variable
$\zeta_{\mathrm{opt},n,o}$	-	Optimal value of $\zeta_{n,o}$
ζ'	-	Substituted parameterization variable ζ
η	-	Characteristic factor
θ	-	Parameter of the generalized trapezoidal rule
κ	K	Global uncertainty
κ^m	K	Local uncertainty in time step m
κ_w^m	-	Local uncertainty in time step m and layer w
$\hat{\kappa}_w^m$	K	Local uncertainty in time step m and layer w by LBM-MM
λ	$\mathrm{m\,kg\,s^{-3}\,K^{-1}}$	Thermal conductivity
λ	-	Eigenvalue of $\boldsymbol{A}$
λ^*	$\mathrm{m\,kg\,s^{-3}\,K^{-1}}$	Augmented thermal conductivity
μ_κ	K	Mean value of the local uncertainty κ^m over all time steps m
ξ	-	Variable of the remainder term in the Lagrange form of the Taylor polynomial
ρ	$\mathrm{m^{-3}\,kg}$	Density
ϱ	-	Spectral radius
σ_κ	K	Standard deviation of the local uncertainty κ^m over all time steps m
τ^m	K	Local truncation error in time step m
ω	-	Relaxation parameter

Greek Upper Symbols

Symbol	Unit	Description
Γ	-	Boundary of Ω
Γ_{C}	-	Boundary of Ω with Cauchy boundary condition
Γ_{N}	-	Boundary of Ω with Neumann boundary condition
Γ_{pre}	-	Boundary of the build plate with contact to the build platform
Γ_{side}	-	Boundary of the build plate with contact to the machine structure
Γ_w	-	Top boundary of Ω_w
$\Gamma_{w,b}$	-	Top boundary of $\Omega_{w,b}$
Γ_Φ	-	Additional boundary introduced by the LBM-MM
Γ_{D}	-	Boundary of Ω with Dirichlet boundary condition
Θ	-	Set of elements ϵ_u
Λ	-	Set of enriched node pairs (n, o)
Λ_w	-	Domain of LC w in the local coordinate systems
Ξ	-	Set of nodes $\boldsymbol{r}_n$
Π	-	Set of possible calculation intervals k_h
Υ	-	Time interval
Υ^m	-	Discrete time interval of Υ
Φ_h	-	Component h
Ψ	-	Set of possible numbers of layers c_h
Ψ_w	-	Set of all layers up to layer Ω_w and the build plate Ω_{bp}
$\Psi_{w,b}$	-	Set of Ψ_{w-1} and all scan fields of the layer Ω_w up to scan field $\Omega_{w,b}$
Ω	-	Domain
Ω_{bp}	-	Build plate
Ω_w	-	Domain of LC w
$\Omega_{w,b}$	-	Domain of scan field b in LC w

Special Symbols

Symbol	Unit	Description
$\varnothing$	-	Empty set
∞	-	Infinity
o	-	Asymptotic little-o notation
$\mathbb{N}$	-	Natural numbers
$\mathbb{R}$	-	Real numbers
$\boldsymbol{\nabla}$	-	Nabla operator

1 Introduction

1.1 Initial Situation and Need for Action

Additive Manufacturing (AM) technologies enable new possibilities in terms of both design and business management. According to MANYIKA et al. (2013) from the McKinsey Global Institute, AM technologies are one of the twelve potentially economic disruptive technologies whose economic impact is estimated at $ 230 to 550 billion by 2025. The main advantages are the realization of more complex designs, reduced lead times and increased production efficiency (STRATASYS DIRECT 2015). For example, the redesign of a fuel nozzle towards AM reduced the number of welds by 80 % (KELLNER 2014). There is less restriction on design, but at the same time higher complexity does not directly lead to higher costs (GIBSON et al. 2015). In the aerospace sector, the freedom of design makes it possible to increase the efficiency of the engines, resulting in reduced fuel consumption (−6.4 %) and CO_2 emissions (92.1×10^6 t to 215×10^6 t) by 2050 (HUANG et al. 2016). Additive Manufacturing thus makes a major and important contribution to climate targets in the aerospace industry.

The term AM covers several manufacturing processes that build parts through successive addition of material. The processes are further classified in seven categories (ISO/ASTM52900 2015). These differ according to the form of the raw material used and the technological principle of the solidification. The subcategory Powder Bed Fusion (PBF) contains all processes that use thermal energy to solidify selected areas in a powder bed. A layer of the powder material is applied on a build plate and selectively melted in the region of the desired geometry. The build platform is lowered, and a new layer of powder is spread across by a coater. The procedure of selectively melting, lowering the build platform and depositing powder is repeated until the entire part is manufactured. A laser beam (Laser Beam Melting (LBM)) or an electron beam (Electron Beam Melting (EBM)) is utilized as the energy source (WOHLERS et al. 2016).

In addition to the above-mentioned potentials, which arise as a result of design freedom, the manufacturing of parts by selective melting is also associated with challenges. Process-related deviations of the dimensional accuracy of the final part occur frequently. The high energy density of the energy sources used in LBM leads to high transient temperature gradients. Since the heat conduction is relatively slow in comparison, high thermal gradients in space occur as well

(KRUTH et al. 2004). The thermal stresses induced thereby sometimes exceed the yield strength and lead to plastic strains in the part, which remain after the build process (KNOWLES et al. 2012). The total dimensional deviations due to thermal gradients are attributable to three occurring effects during the buildup process (Figure 1.1). Distortions arising directly from the selective solidification process on the scale of the individual welding tracks are clustered as the primary effect. The reasons for this are described by the *Temperature Gradient Mechanism (TGM)*[2], and its impact is mainly influenced by the scanning strategy (KRUTH et al. 2004). The thermal expansion of a point under reduced strength in the liquid state and the subsequent thermal contraction during cooling in the solidified state result in compressive strains of the surrounding material. Generalizing the TGM to the scale of the complete layer leads to the secondary effect. The involved mechanism is known as *curling* and reflects the consequences of the delayed shrinkage of the layers (MERCELIS & KRUTH 2006; H.-J. YANG et al. 2002). In Addition to the mentioned effects, the solidification of selected areas with thermal energy results in a heat flux from the current top layer to the build plate. Depending on the geometry, heat accumulations may occur as a result, e. g. in filigree areas or when the transverse section is changing. The described dimensional deviations caused by the temperature gradients on the component scale are summarized as the tertiary effect.

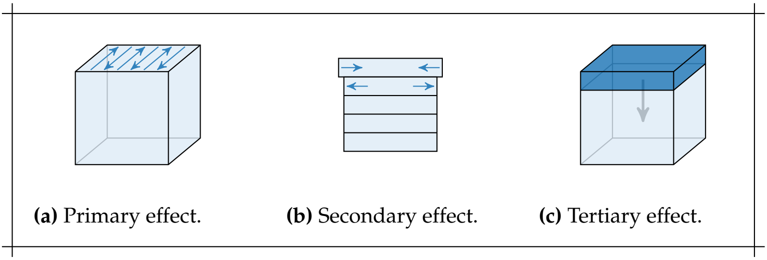

(a) Primary effect. **(b)** Secondary effect. **(c)** Tertiary effect.

Figure 1.1: Pictographic illustration of occurring effects that contribute to dimensional deviations in LBM.

Depending on the application, the dimensional accuracy of the finished part may be outside the tolerance range. The distortions sometimes even terminate the build job, if the dimensional deviations exceed the height of one layer during the buildup process and the warped part collides with the coater. To overcome these deviations, several countermeasures exist. The main strategies are preheat-

[2] Even though all three effects are due to thermal gradients, the term TGM is especially used for the resulting compressive strains on a local scale induced by heating a point or along a straight line.

ing, adjustment of the energy input, support adjustment and pre-deformation (MUNSCH et al. 2013). Aside from pre-deformation, all of these strategies address the causes of dimensional deviations, but also affect secondary quantities (BAYERLEIN et al. 2016). Pre-deformation influences only the symptoms of distortions but does not intervene in the process itself. Therefore, it is judged as especially suitable for PBF by BAYERLEIN et al. (2016).

The term pre-deformation describes the adjustment of the input geometry to compensate for the dimensional deviations occurring during the process. However, to know how the geometry must be adapted, a test build job with the current geometry must first be carried out and subsequently measured. The input geometry is then adjusted based on the measurement results. Besides the high labor costs for pre- and post-processing as well as measurement, expert knowledge is required to correctly adjust the geometry by the obtained measurement data (HELD & PFLIGERSDORFFER 2009). Even with expert knowledge, up to ten of those iterations are necessary to manufacture the geometry inside the tolerance bounds for aero applications, since the distortions in LBM are nonlinear. Thus, the procedure requires a high amount of machine time. In addition, an increase in the machine-hour rate is to be expected (SEIDEL 2016; WOHLERS et al. 2016). Furthermore, it is hardly possible to find the correct input geometry for complex parts by the described procedure. This is not at all possible for areas that are not metrologically accessible. For reasons of profitability, but also because of the mentioned limits of the previously described measurement-based pre-deformation, the buildup process is modeled and simulated to obtain the desired input geometry (BAYERLEIN et al. 2015). Currently, two Conceptual Models are mainly used to predict the process-related deviations in the buildup process: a *thermomechanical model* and a model that uses the *inherent strain method* (MEGAHED et al. 2016). The assumption of the main cause of the distortions differs in the models, but both have in common the simplification of several layers to a Layer Compound (LC).

The first one determines discrete time steps of the appearing temperature field during the buildup process in a transient thermal analysis and then applies them as thermal load in a quasi-static mechanical analysis. It is referred to as *quasi-static* since each step in the mechanical calculation itself is static, but the discrete time steps of the temperature field are applied according to their chronological sequence. The retroactive effect of the resulting displacement on the temperature is neglected. In the causality of the model, the thermal analysis thus determines the cause, and the mechanical analysis merely shows the effect. Accordingly, the different gradations of the model take place over the differently detailed approximation of the heat source, more precisely the representation of the exposure strategy. A distinction is made between the consideration of the real exposure strategy and the simplification to scan fields (SEIDEL et al. 2014), as well as the simultaneous heat input over the entire LC (PAPADAKIS, LOIZOU, RISSE & SCHRAGE 2014). Due to the long, and therefore uneconomical, calculation time, the last gradation is mainly used in the application of the thermomechanical model (SEIDEL 2016).

The second model utilizes the inherent strain method, which was developed by UEDA et al. (1975) and is largely used in welding simulations (L. ZHANG et al. 2007). It is based on the main assumption that the plastic strains remaining after the solidification of one layer are the cause of the distortion of the entire part. Therefore, a detailed thermomechanical simulation is carried out on a smaller specimen in consideration of the exposure strategy and with as few simplifications as possible. Subsequently, the remaining plastic strains are averaged. This average is successively applied to the LCs as an inherent strain on the desired part in a pure elastic calculation (KELLER & PLOSHIKHIN 2014). Since only one calculation step is necessary for each LC, the calculation time for the second model with the inherent strain method is reduced in comparison with the thermomechanical analysis in the first model. The simulation to obtain the inherent strain must be done only once for each exposure strategy. Computation time therefore plays only a subordinate role, and the process is modeled as precisely as possible in the detailed thermomechanical simulation. However, to avoid additional modeling uncertainties, there are further experiment-based approaches to calibrate the inherent strain via the distortions of manufactured cantilevers (BUGATTI & SEMERARO 2018).

A decisive requirements for the use of simulation is that it is capable of determining the distortions with sufficient accuracy (COWLES et al. 2012). The presented models demonstrated suitable agreement with experimental data and thus reduce the necessary manufacturing iterations to achieve a pre-deformed geometry (BAYERLEIN & ZAEH 2016; KELLER & PLOSHIKHIN 2014; SEIDEL 2016). Nevertheless, none of them is able to predict the distortions at an accuracy that allows for a *first-time-right*[3] production (SCHAFSTALL et al. 2016). Regarding the mentioned effects leading to distortions, both models only consider two out of three effects from Figure 1.1. Both the thermomechanical model and the inherent strain model take into account the time-delayed shrinkage. The temperature history of a point during the buildup process is not represented in the inherent strain model. Conversely, the thermomechanical model does not include the effect of the exposure strategy, as the heat input is usually generalized to the whole LC due to calculation effort reasons. To completely replace the manufacturing-based pre-deformation cycle by simulation and to reach a first-time-right production, all distortion effects must be considered in the model. The consideration of the exposure strategy in the thermomechanical model represents such a model, but does currently not meet the requirements

[3] first-time-right describes the manufacturing of a part without preliminary test productions

in terms the calculation time (SCHAFSTALL et al. 2016). Therefore, the main motivation of this work is to make use of these already developed, more detailed thermomechanical models by reducing the computational effort. The fidelity of prediction of the simulation is increased by the application of higher-valued models (SEIDEL 2016) and the thesis contributes to the economic application of the buildup simulation of LBM in the production environment.

1.2 Field of Action and Objective

The described initial situation reveals the need for action to accelerate the simulation of LBM for using more detailed models and thus fully exploit their potential. According to OBERKAMPF et al. (2002), Modeling and Simulation (M&S) of a Physical System is subdivided into six phases:

1. Conceptual Modeling of the Physical System,
2. Mathematical Modeling of the Conceptual Model,
3. Discretization and Algorithm Selection for the Mathematical Model,
4. Computer Programming of the Discrete Model,
5. Numerical Solution of the Computer Program Model and
6. Representation of the Numerical Solution.

The first phase specifies the physical process, identifies the main causes for the considered effects and derives a *Conceptual Model*. From here on, the Conceptual Model in LBM always refers to the thermomechanical model. For the reasons described in the previous section, the inherent strain approach is not considered. In the next phase, the Conceptual Model is translated into Partial Differential Equations (PDEs) including the corresponding boundary and initial conditions. Phase three involves the selection of numerical methods to convert the obtained continuous *Mathematical Model* into a discrete one and to solve the resulting system of equations. Subsequently, the *Discrete Model* and solution procedures are implemented in computer code in phase four. The computation of the numerical solution and its representation remain for the last two phases.

When assigning existing publications in the M&S of the Physical System of LBM into the described phases, it is apparent that most of the work is devoted to the Conceptual and Mathematical Model (BRANNER 2011; KING et al. 2014; NEUGEBAUER et al. 2014; PAPADAKIS, LOIZOU, RISSE & SCHRAGE 2014; SEIDEL 2016; ZAEH & LUTZMANN 2010). All references have in common that they adapt the Conceptual Model in order to reduce calculation time. The associated loss of accuracy is accepted. The Discretization and Algorithm Selection for the Mathematical Model has hardly been considered thus far, as the reviews of SCHOINOCHORITIS et al. (2017) and SCHMIDT et al. (2017) imply. However, SEIDEL (2016) has pointed out the need to consider numerical methods. Moreover, NIKOUKAR et al. (2013) showed that the consideration of AM-specific characteristics in numerical methods is suitable to significantly speed up the simulation.

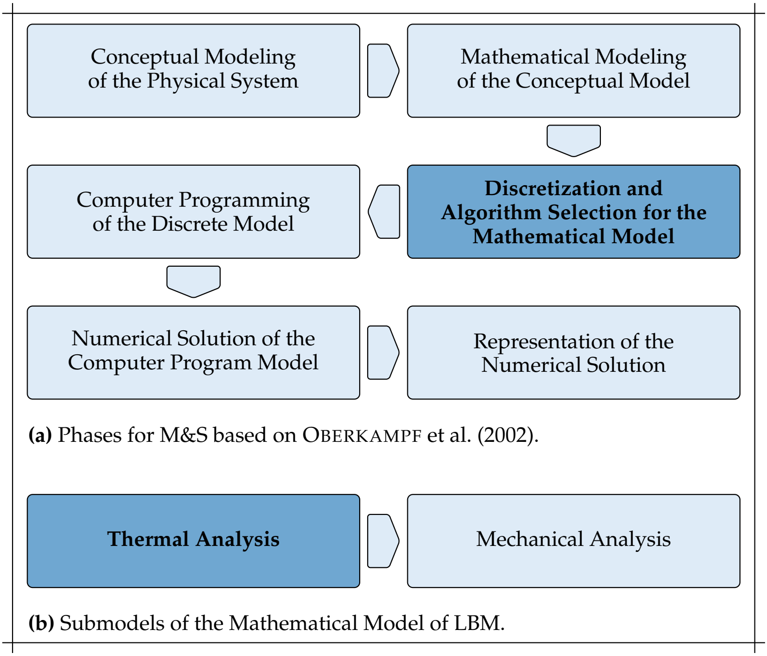
This thesis continues from existing Mathematical Models and considers the Discretization and Algorithm Selection for the Mathematical Model. Improvement of the Mathematical or even the Conceptual Models is not the objective, since several satisfactory models are already available and only their applications are currently not economical. Rather, the work of this thesis contributes to making these models usable. The accuracy is only taken into account in connection with the deviations resulting from the Discretization and Algorithm Selection of the Mathematical Model. Similarly, subsequent phases are considered only to the extent necessary to verify and validate the results.

The thermomechanical approach consists of a thermal analysis and a subsequently mechanical one. Since little preparatory work is available in the field of phase three, this thesis focuses on the first part, i. e. the thermal analysis. The Discretization and Algorithm Selection for the mechanical part are not considered. Moreover, developing numerical methods for the mechanical analysis requires the temperature field as input which is currently not available for more detailed heat sources due to the calculation effort. Therefore, this thesis intends to accelerate the thermal analysis and to lay the foundation for subsequent work in mechanical analysis with a more detailed temperature field. The fields of action of this thesis are graphically summarized in Figure 1.2.

1.3 Approach

Numerical methods are developed or adapted to the specific requirements of the Mathematical Model in LBM to achieve the goal of an accelerated thermal simulation. Knowledge of the process characteristics is integrated into the numerical methods and thus taken into account a priori in the simulation. The procedure selected for this is illustrated in Figure 1.3.

The Discretization and Algorithm Selection for the LBM of the heat equation, on which the thermal analysis is based, takes place in three fields. The PDE requires a spatial and temporal discretization, as well as the solving of the resulting system of equations (detailed fundamentals are provided in Chapter 2). The state of the art is discussed with regard to computationally intensive process characteristics of LBM, both in the specific area of the simulation of LBM and in the general subject area of numerics (cf. Chapter 3). Based on this, three separate numerical methods for the simulation of LBM are developed or adapted, respectively. The Parameterized eXtended Finite Element Method (PXFEM) supplements the function space in areas of high gradients with parameterized enrichment functions (cf. Chapter 4). These allow for the representation of steep local gradients taking into account the temporal change with few Degrees of


Conceptual Modeling of the Physical System
Mathematical Modeling of the Conceptual Model
Discretization and Algorithm Selection for the Mathematical Model
Computer Programming of the Discrete Model
Numerical Solution of the Computer Program Model
Representation of the Numerical Solution
Thermal Analysis
Mechanical Analysis

(a) Phases for M&S based on OBERKAMPF et al. (2002).

(b) Submodels of the Mathematical Model of LBM.

Figure 1.2: Fields of action (dark blue) of the presented thesis emphasized on the different levels of M&S of LBM.

Freedom (DoFs). In the case of temporal discretization, it is a multirate method for a growing system of equations (cf. Chapter 5). The system of equations is dynamically partitioned by the LBM Multirate Method (LBM-MM) into components with local time steps, taking advantage of the layered structure and taking into account the layers to be added by the buildup process. The LBM-specific Direct Solving Strategy (LBM-DSS) also considers the layered structure of the mesh (cf. Chapter 6). The system of equations is linearized and information from the previous solution step is adopted completely or partially.

Each method is verified and validated. According to *Systems and software engineering– System life cycle processes* (2015), verification is the "confirmation, (...), that the specified requirements have been fulfilled". Validation is defined as the "confirmation, (...), that the requirements for a specific intended use or application have been fulfilled". The intended use is to reduce the computing time under the requirement to receive the same result or smaller deviations. In the context of this work, verification is the comparison of the developed methods against the status quo with regard to the results obtained, and valida-

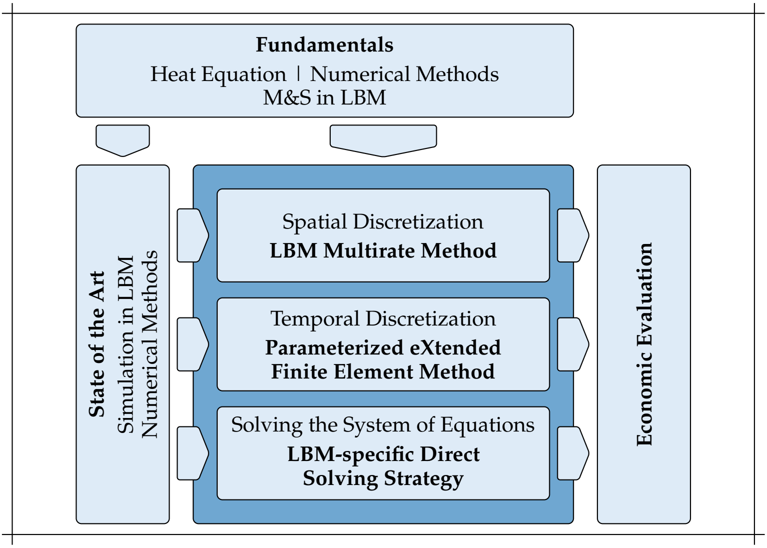


Figure 1.3: Procedure of the presented thesis.

tion is the comparison with regard to the required computing time and/or the computing effort. This thesis concludes with a summary and feasibility study of the developed numerical methods, followed by an outlook (cf. Chapter 7).

In general, the material parameters used are the ones of Inconel 718 (POTTLACHER et al. 2002; SPECIAL METALS CORPORATION 2007) and the machine data is of the manufacturing system EOS M270. The results obtained in terms of calculation accuracy and calculation time do not claim to be universally valid due to the strongly non-linear relationship between the results and the material parameters. However, the procedure used is generally valid and therefore transferable to arbitrary materials and manufacturing systems.

All scripts created and used in this thesis are available via mediaTUM (the media and publications repository of the Technical University of Munich) or via repository of the Leibniz Supercomputing Centre (LRZ) Gitlab as listed below and they are free to use.

1. ZELLER et al. (2019): MATLAB-based scripts used within the context of the PXFEM.
2. BANTE et al. (2019): Modified source code of the free open source software CalculiX CrunchiX (ccx) with the implemented LBM-MM and LBM-DSS.

3. *ascent*[4]: Python-based program for creating an input script for ccx to calculate the distortions occurring in LBM (repository for the codebase of the AscentAM (AAM) project, which has received funding from the Clean Sky Joint Undertaking under the European Union's Seventh Framework for research, technological development and demonstration Programme under grant agreement number 714 246).
4. ZELLER & ZAEH (2019): Remaining evaluation scripts used in this thesis.

[4] `https://gitlab.lrz.de/flayerbein/ascent.git`

2 Fundamentals for Discretization and Algorithm Selection in Simulation of Laser Beam Melting

2.1 Overview

This chapter describes the physical and numerical fundamentals required for the topics considered in this thesis. These include the PDE of the heat balance, general principles of numerics, and the numerical treatment of the heat equation. The explanations on the topics do not claim to be exhaustive, they merely serve to convey the necessary basics. Furthermore, this thesis deals with the Discretization and Algorithm Selection in the M&S of LBM. Therefore, the preceding phases according to OBERKAMPF et al. (2002), which serve as a basis, are presented. This includes the description of the Physical System of LBM, as well as the Conceptual Model and the Mathematical Model for the thermal simulation on the part scale.

At the beginning of this chapter, an explanation of the main mathematical notations used is given. Notations, as well as formula symbols, are based on the international standard DIN EN ISO 80000 (2013).

2.2 Main Mathematical Notations and Terms

As far as no notation is recommended for a quantity according to the international standard DIN EN ISO 80000 (2013), the general convention applies that

1. *scalars* are written in Latin lowercase letters, e. g. $n \in \mathbb{N}$ (natural numbers) and $c \in \mathbb{R}$ (real numbers),
2. *vectors* are represented by bold symbols in Latin lowercase letters, e. g. $\boldsymbol{b} \in \mathbb{R}^n$ with $n > 1$,
3. *matrices* are denoted by bold symbols in Latin capital letters, e. g. $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with $n > 1$,
4. *sets* are named with Greek capital letters, e. g. $\Omega \subseteq \mathbb{R}^3$, and
5. *functions* are written according to the notation of their value, e. g. $v : \Omega \rightarrow \mathbb{R}$.

Further specifications are indicated via subscript or superscript. Deviating from the standard, the individual items of a matrix $\boldsymbol{A}$ are labeled with A_{kl} for the corresponding row $1 \leq k \leq n$ and column $1 \leq l \leq m$. If the identifier of a matrix already contains a subscript, e. g. $\boldsymbol{A}_{\text{const}}$, the specification of the matrix items is separated by a comma, i. e. $A_{\text{const},kl}$. Although items are scalar quantities and should therefore be designated with Latin lowercase letters, the notation convention of Latin capital letters is retained, since they are induced by a matrix.

A square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is *symmetric* if

$$A_{kl} = A_{lk}. \tag{2.1}$$

and it is a *diagonal matrix* if

$$A_{kl} = 0, \qquad \text{for } k \neq l. \tag{2.2}$$

Also mentioned are the conditions for an *upper triangular matrix*

$$A_{kl} = 0, \qquad \forall k < l \tag{2.3}$$

and for a *strictly upper triangular matrix*

$$A_{kl} = 0, \qquad \forall k \leq l. \tag{2.4}$$

The transposed matrix $\boldsymbol{A}^\top$ is accordingly a *lower triangular matrix* or a *strictly lower triangular matrix*.

The *rank* of a matrix denotes the dimension of the vector space generated by the linear mapping $\boldsymbol{x} \mapsto \boldsymbol{A} \cdot \boldsymbol{x}$ for all vectors $\boldsymbol{x} \in \mathbb{R}^n$. It is equivalent to the maximum number of linearly independent columns of $\boldsymbol{A}$.

The scalar $\lambda \in \mathbb{R}$ and the vector $\boldsymbol{e} \in \mathbb{R}^n$, $\boldsymbol{e} \neq 0$, are called the *eigenvalue* and *eigenvector* of a matrix $\boldsymbol{A}$, if

$$\boldsymbol{A} \cdot \boldsymbol{e} = \lambda \boldsymbol{e}. \tag{2.5}$$

In conjunction, the *spectral radius* $\varrho(\boldsymbol{A})$ of a matrix $\boldsymbol{A}$ is defined as the largest absolute value of eigenvalues of $\boldsymbol{A}$.

The two norms $\|\cdot\|_2$ and $\|\cdot\|_\infty$ are used in connection with functions and defined as

$$\|v\|_2 := \left(\int_\Omega |v|^2 \, \mathrm{d}\Omega \right)^{\frac{1}{2}} \tag{2.6}$$

$$\|v\|_\infty := \sup \{ |v(\boldsymbol{x})| : \boldsymbol{x} \in \Omega \} \tag{2.7}$$

For sums and unions over an empty index set, the convention applies that the sums are 0 and the unions are the empty set $\varnothing$, i. e.

$$\sum_{s \in \varnothing} \ldots = 0, \qquad \bigcup_{s \in \varnothing} \ldots = \varnothing. \tag{2.8}$$

2.3 Transient Heat Conductance in Solids

In this section the necessary basics in the field of transient heat conduction in solids, which are required in this thesis, are provided. This includes the PDE for the description of thermal processes with the associated initial and boundary conditions, special analytical solutions as well as their temporal propagation behavior.

2.3.1 Differential equation, initial and boundary conditions

For a domain $\Omega \subseteq \mathbb{R}^3$ and a time interval $\mathrm{Y} \subset \mathbb{R}$, the temperature T at point $\boldsymbol{r} \in \Omega$ and time $t \in \mathrm{Y}$ is defined as a function $T : \Omega \times \mathrm{Y} \to \mathbb{R}$. To provide a clearer notation, however, the arguments for the temperature are occasionally omitted, which means that they are not mentioned. The transient temperature distribution in the domain Ω during the time interval Y is expressed by the heat equation, a second order PDE. Its general homogeneous form for solids is given as

$$\rho c(T)\frac{\partial T}{\partial t} = \begin{pmatrix} \frac{\partial}{\partial r_1} \\ \frac{\partial}{\partial r_2} \\ \frac{\partial}{\partial r_3} \end{pmatrix} \cdot \left[\lambda(T) \begin{pmatrix} \frac{\partial T}{\partial r_1} \\ \frac{\partial T}{\partial r_2} \\ \frac{\partial T}{\partial r_3} \end{pmatrix} \right], \qquad \forall \boldsymbol{r} \in \Omega, t \in \mathrm{Y}, \tag{2.9}$$

with the temperature-dependent material values heat capacity $c : \mathbb{R} \to \mathbb{R}$ and thermal conductivity $\lambda : \mathbb{R} \to \mathbb{R}$. The material model of an incompressible body is used in the derivation of Equation 2.9. Small density changes due to the temperature and pressure changes are neglected since heat conduction in a solid is considered. Therefore, the density ρ is assumed to be constant, i. e. $\rho \in \mathbb{R}$. (Baehr & Stephan 2009)

A more compact form of Equation 2.9 is obtained by using the *nabla operator* $\boldsymbol{\nabla}$:

$$\rho c(T)\frac{\partial T}{\partial t} = \boldsymbol{\nabla} \cdot [\lambda(T)\boldsymbol{\nabla} T], \qquad \forall \boldsymbol{r} \in \Omega, t \in \mathrm{Y}. \tag{2.10}$$

The inhomogeneous form is not considered, as no internal heat sources exist in LBM and temperature changes result only from the boundary conditions.

Heat conduction in a solid is an initial-boundary value problem. For a well-posed problem and a unique solution, temporal *initial conditions* as well as spatial *boundary conditions* are required. The initial condition is a function $T^0 : \Omega \to \mathbb{R}$ that specifies a temperature in the entire domain Ω at a certain time t^0 of the interval Y, i. e.

$$T(\boldsymbol{r}, t^0) = T^0(\boldsymbol{r}), \qquad \forall \boldsymbol{r} \in \Omega. \tag{2.11}$$

Basically, the time t^0 is arbitrary within the interval Y. In this thesis, however, $t^0 = 0$ always applies, which also determines the start of the time interval Y. The end of the observation space is defined as t^{m_Y}; hence,

$$\mathrm{Y} = [0, t^{m_\mathrm{Y}}]. \tag{2.12}$$

The boundary condition has several possible types. A distinction is made between prescribed temperature, heat flux density and convection. Therefore, the boundary $\partial\Omega = \Gamma \in \mathbb{R}^3$ will be separated into pairwise disjoint sets

$$\Gamma = \Gamma_\mathrm{D} \,\dot{\cup}\, \Gamma_\mathrm{N} \,\dot{\cup}\, \Gamma_\mathrm{C}. \tag{2.13}$$

for the respective boundary conditions, even if not all conditions must apply at the same time (the empty set is disjoint to any set). A prescribed temperature on the boundary Γ_D by a function $T_{\Gamma_\mathrm{D}} : \Gamma_\mathrm{D} \times \mathrm{Y} \to \mathbb{R}$, i. e.

$$T(\boldsymbol{r}, t) = T_{\Gamma_\mathrm{D}}(\boldsymbol{r}, t), \qquad \forall \boldsymbol{r} \in \Gamma_\mathrm{D}, t \in \mathrm{Y}, \tag{2.14}$$

corresponds to a *Dirichlet*[5] *boundary condition*.

The *Neumann*[6] *boundary condition* is an applied heat flux density on the boundary Γ_N by a function $\dot{q}_{\Gamma_\mathrm{N}} : \Gamma_\mathrm{N} \times \mathrm{Y} \to \mathbb{R}^3$ as follows:

$$-\lambda(T)\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r}, t) = \dot{q}_{\Gamma_\mathrm{N}}(\boldsymbol{r}, t), \qquad \forall \boldsymbol{r} \in \Gamma_\mathrm{N}, t \in \mathrm{Y}, \tag{2.15}$$

where $\partial/\partial\boldsymbol{n}$ denotes the derivation in direction of the exterior normal $\boldsymbol{n} \in \mathbb{R}^3$ at $\boldsymbol{r} \in \Gamma_\mathrm{N}$. The choice $\dot{q}_{\Gamma_\mathrm{N}}(\boldsymbol{r}, t) = 0$ corresponds to an adiabatic boundary.

The last mentioned convection is a kind of mixed boundary condition in the form

$$-\lambda(T)\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r}, t) = \alpha \left[T(\boldsymbol{r}, t) - T_{\Gamma_\mathrm{C}}(\boldsymbol{r}, t)\right], \qquad \forall \boldsymbol{r} \in \Gamma_\mathrm{C}, t \in \mathrm{Y} \tag{2.16}$$

with a heat transfer coefficient $\alpha \in \mathbb{R}$ and a used function $T_{\Gamma_\mathrm{C}} : \Gamma_\mathrm{C} \times \mathrm{Y} \to \mathbb{R}$ to represent the ambient temperature. It is also referred to as the *Robin*[7] *boundary condition*.

The described boundary conditions are also referred to as first-, second- and third-order boundary conditions. In addition, there are further types, such as the heat transport over the boundary via radiation (EVANS 2015). However, these are neglected as they have no major significance in processes with rapidly moving heat sources (CLINE & ANTHONY 1977).

[5] **Peter Gustav Lejeune Dirichlet** (* February 13th, 1805 - † May 5th, 1859), German mathematician.

[6] **Carl Gottfried Neumann** (* May 7th, 1832 - † March 27th, 1925), German mathematician.

[7] **Victor Gustave Robin** (* May 17th, 1855 - † November 20th, 1987), French mathematician.

2.3.2 Analytical solutions

In most cases, no analytical solutions exist for the heat equation. This applies in particular for complex geometries and temperature-dependent material values (BAEHR & STEPHAN 2009). Under certain assumptions and simplifications, however, analytical solutions of the heat equation are obtained. In the case of the one-dimensional heat equation for a semi-infinite domain with constant material values

$$\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2}, \qquad \forall x \in [0, \infty), t \in \mathrm{Y}, \tag{2.17}$$

solutions exist for various initial and boundary conditions. The *thermal diffusivity $a \in \mathbb{R}$* is introduced in Equation 2.17, which is composed as follows:

$$a = \frac{\lambda}{\rho c}. \tag{2.18}$$

For the constant initial and boundary conditions

$$T(x, 0) = T^0, \qquad \forall x \in [0, \infty), \tag{2.19}$$

$$T(0, t) = T_{\Gamma_\mathrm{D}}, \qquad \forall t \in \mathrm{Y}, \tag{2.20}$$

the function

$$T_\mathrm{ana}(x, t) = \left(T^0 - T_{\Gamma_\mathrm{D}}\right) \mathrm{erf}\left(\frac{x}{\sqrt{4at}}\right) + T_{\Gamma_\mathrm{D}} \tag{2.21}$$

solves the heat equation for all x in $[0, \infty)$ and t in Y (CARSLAW & JAEGER 2008). The function $\mathrm{erf}(x)$ is the *error function*, also known as the *Gauss*[8] *error function*, and it is defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\tau^2}\, \mathrm{d}\tau. \tag{2.22}$$

There is also a solution for the balancing temperature distribution due to an initial energy deposition in an area of width d_ed at an adiabatic boundary (CARSLAW & JAEGER 2008). In formula notation, this means for the initial and boundary conditions

$$T(x, 0) = \begin{cases} T^{0,1}, & \forall x \in [0, d_\mathrm{ed}], \\ T^{0,2}, & \forall x \in (d_\mathrm{ed}, \infty), \end{cases} \tag{2.23}$$

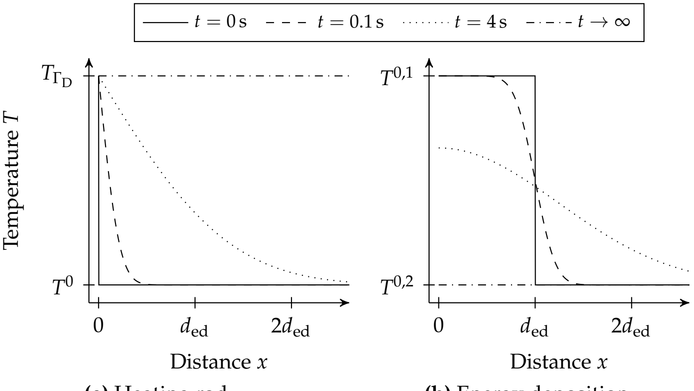
$$\frac{\partial T}{\partial \boldsymbol{n}}(0, t) = 0, \qquad \forall t \in \mathrm{Y}. \tag{2.24}$$

[8] **Johann Carl Friedrich Gauss** (* April 30th, 1777 - † February 23rd, 1855), German mathematician and physicist.

The corresponding analytical solution for all x in $[0, \infty)$ and t in Y is as follows:

$$T_{\text{ana}}(x,t) = \frac{(T^{0,1} - T^{0,2})}{2} \left[\operatorname{erf}\left(\frac{d_{\text{ed}} + x}{\sqrt{4at}}\right) + \operatorname{erf}\left(\frac{d_{\text{ed}} - x}{\sqrt{4at}}\right) \right] + T^{0,2}. \quad (2.25)$$

A graphical illustration of the two analytical solutions introduced is provided in Figure 2.1.



(a) Heating rod (Equation 2.21).

(b) Energy deposition (Equation 2.25).

Figure 2.1: Analytical solutions of the one-dimensional heat equation with different initial and boundary conditions for a semi-infinite domain with $a = 3.5 \times 10^{-6}\,\text{m}^2\,\text{s}^{-1}$.

Both solutions, Equation 2.21 and Equation 2.25, have the characteristic that temperature distributions are convertible into each other by temporal and spatial scaling. This is later used to take into account the number of combined layers within an LC. In the concrete case of the analytical solution with an initial energy deposition, this means that factors $\bar{x}$ and $\bar{t}$ exist, such that the following holds for two solutions T^1_{ana} and T^2_{ana} with different a and d_{ed}:

$$T^1_{\text{ana}}(\bar{x}x, \bar{t}t) = T^2_{\text{ana}}(x,t). \quad (2.26)$$

This is true if the equality for both arguments of the occurring error functions is fulfilled (cf. Equation 2.25). The first argument therefore requires the following:

$$\frac{d_{\mathrm{ed},1} + \bar{x}x}{\sqrt{4a_1\bar{t}t}} = \frac{d_{\mathrm{ed},2} + x}{\sqrt{4a_2 t}}, \tag{2.27}$$

$$\Rightarrow \quad \sqrt{\frac{a_1}{a_2}\bar{t}} = \frac{d_{\mathrm{ed},1} + \bar{x}x}{d_{\mathrm{ed},2} + x}, \tag{2.28}$$

$$\Rightarrow \quad \sqrt{\frac{a_1}{a_2}\bar{t}} = \bar{x}\frac{\frac{d_{\mathrm{ed},1}}{\bar{x}} + x}{d_{\mathrm{ed},2} + x}. \tag{2.29}$$

The equation is fulfilled with the choices

$$\bar{x} = \frac{d_{\mathrm{ed},2}}{d_{\mathrm{ed},1}}, \qquad \bar{t} = \frac{a_2}{a_1}\left(\frac{d_{\mathrm{ed},2}}{d_{\mathrm{ed},1}}\right)^2. \tag{2.30}$$

This procedure and the result also apply analogously to the argument of the second error function in Equation 2.25. However, for the permissibility of the transformations, the values are restricted to $x \neq d_{\mathrm{ed},2}$. In this case $\mathrm{erf}(0) = 0$ applies and the equality is directly satisfied independently of the values $\bar{x}$ and $\bar{t}$. The corresponding time interval of solution T^1_{ana} changes to $\mathrm{Y} = [0, \bar{t}t^{m_{\mathrm{Y}}}]$ with the introduction of the characteristic quantity $\bar{t}$, whereas the spatial interval is unaffected by the factor $\bar{x}$ due to its definition. With the choice $d_{\mathrm{ed},1} = d_{\mathrm{ed},2} = 1$ the results are transferable to the analytical solution with a constant Dirichlet boundary condition in Equation 2.21.

2.3.3 Penetration depth

As illustrated, the temperature change propagates from the origin (Figure 2.1a) or, in the case of the solution for the energy disposition (Figure 2.1b), from the point d_{ed}. The time-dependent distance $\delta : \mathrm{Y} \rightarrow \mathbb{R}$ of the cause of the temperature change to the most distant affected point is defined as the *penetration depth*. In general, the heat equation (Equation 2.10) violates the principles of relativity as information about temperature changes propagates with infinite speed (ECKERT & DRAKE 1972). This is negligible for metals (ALI & L. C. ZHANG 2005), but due to the infinitely rapid propagation of the temperature change, a threshold value must be specified from which a point is considered to be affected by the temperature change in relation to the penetration depth. MUNSON (2013) defines this as 1 % of the difference between the initial temperature and a wall temperature $T_w : \mathrm{Y} \rightarrow \mathbb{R}$ that causes the temperature change as established in fluid mechanics. According to X. YAN (2002), this definition results in the equation

$$\frac{T_w(t) - T(\delta(t), t)}{T_w(t) - T^0} = 0.99. \tag{2.31}$$

For the solution in Equation 2.21 and $T_w = T_{\Gamma_\mathrm{D}}$, it follows that

$$\operatorname{erf}\left(\frac{\delta(t)}{\sqrt{4at}}\right) = 0.99 \tag{2.32}$$

and its numerical solution is

$$\delta(at) = 3.642\,77\sqrt{at}. \tag{2.33}$$

The definition of the penetration depth of MUNSON (2013) actually refers only to temperature changes caused by boundary conditions. In the case of the solution from Equation 2.25, the initial condition within the domain causes the temperature change. However, the definition of the penetration depth and the procedure of the determination in Equation 2.31 from X. YAN (2002) are transferable to it. A direct solution depending on the variables a and t, as in Equation 2.33, is not possible. The equation is therefore solved for different explicit values at, and a regression analysis (cf. Section A.1) is performed for small values ($at \ll 3\,\mathrm{mm}^2$) as well as for higher values ($1\,\mathrm{mm}^2 \ll at$), resulting in

$$\delta(at) = \begin{cases} 2.86\sqrt{at} + 3.74 \times 10^{-1} at, & at \ll 3\,\mathrm{mm}^2, \\ 3.86\sqrt{at} + 8.12 \times 10^{-3} at + 6.60 \times 10^{-1}, & at \gg 1\,\mathrm{mm}^2. \end{cases} \tag{2.34}$$

A graphical representation of this is provided in Figure 2.2.

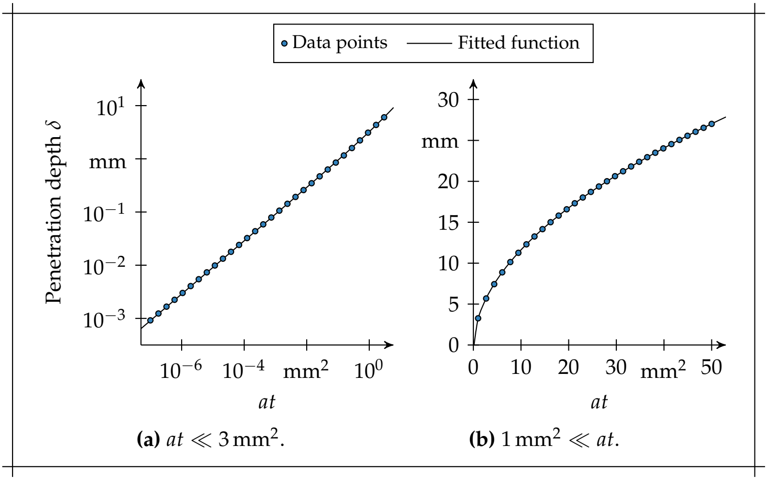
The penetration depth δ is derived from the heat equation of the semi-infinite domain. For short times, however, the analytical solution is also permissible for a simple domain with finite dimensions. According to POLIFKE & KOPITZ (2009), the *diffusive linear measure* must be significantly smaller than the *characteristic length* of the domain. The diffusive linear measure is an estimate of the temperature propagation and thus corresponds to the penetration depth derived above. In a one-dimensional case with directed heat flux, the characteristic length represents the dimension of the domain $\bar{L}$. In summary, it follows that in the case of

$$\frac{\delta(at)}{\bar{L}} \ll 1, \tag{2.35}$$

finite domains are allowed to be regarded as semi-infinite, and therefore the analytical solution is permissible as a short-term approximation.

2.4 Numerical Fundamentals

Numerical methods used to solve differential equations are presented. These are numerical integrators, which discretize the differential equation in time, as well as linear solvers.



(a) $at \ll 3\,\mathrm{mm}^2$.

(b) $1\,\mathrm{mm}^2 \ll at$.

Figure 2.2: Data points used in the regression analyses and the fitted functions obtained (cf. Equation 2.34) for small values *at* (left, log-scale) and greater values *at* (right).

2.4.1 Numerical integrator

Numerical integrators are used to solve Initial Value Problems (IVPs), i. e. an Ordinary Differential Equation (ODE) together with an initial condition. In general vector form, these correspond to

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \qquad \forall t \in \Upsilon, \tag{2.36}$$

$$\boldsymbol{y}\left(t^0\right) = \boldsymbol{y}^0 \tag{2.37}$$

with $\boldsymbol{y} : \Upsilon \rightarrow \mathbb{R}^{n_\Xi}$, $\boldsymbol{y}^0 \in \mathbb{R}^{n_\Xi}$ and $\boldsymbol{f} : \Upsilon \times \mathbb{R}^{n_\Xi} \rightarrow \mathbb{R}^{n_\Xi}$. To solve the IVP, the time interval Υ is discretized into $\Upsilon^m = \left\{t^0, t^1, \ldots, t^{m_\Upsilon}\right\}$ with

$$\Delta t^m = t^m - t^{m-1} \tag{2.38}$$

for $m \in \{1, \ldots, m_\Upsilon\}$. The discrete points in time are now used as integration points for the numerical integration of the derivation. Thereby, numerical integrators are classified according to the number of integration points used to calculate the derivation, i. e. *One-Step Methods* or *Multi-Step Methods*. This thesis is restricted to One-Step Methods, as they are more common due to their simplicity and flexibility. Among others, they are used in the program ccx (DHONDT 2004). A representative of this class is the *generalized trapezoidal* *rule* (T. J. R. HUGHES 1987, pp. 459). The approximated value $\boldsymbol{y}^m$ of $\boldsymbol{y}(t^m)$ is successively calculated for $m \in \{1, \dots, m_Y\}$ with the formula

$$\boldsymbol{y}^m = \boldsymbol{y}^{m-1} + (1-\theta)\,\Delta t^m\,\boldsymbol{f}\left(t^{m-1}, \boldsymbol{y}^{m-1}\right) + \theta\,\Delta t^m\,\boldsymbol{f}(t^m, \boldsymbol{y}^m) \tag{2.39}$$

and $\theta \in [0, 1]$. Depending on the choice of the parameter θ, different methods with various properties result. Important properties of numerical integrators are *numerical stability*, *consistency* and *convergence*. *Numerical stability* describes the robustness of the algorithm against disturbances in the input data. Several concepts exist in this context. Due to the scope of the concepts, these are not discussed, but reference is made to further literature (DEUFLHARD & BORNEMANN 2002). Only the consequence of *L-stability* is mentioned here. Methods with this property damp oscillations due to local disturbances in the data (DEUFLHARD & WEISER 2012). Assuming correct input data, i. e. $\boldsymbol{y}^{m-1} = \boldsymbol{y}(t^{m-1})$, *consistency* describes the quality of the solution, which is expressed by the *local truncation error*

$$\tau^m = \|\boldsymbol{y}(t^m) - \boldsymbol{y}^m\|\,. \tag{2.40}$$

A numerical method is *consistent*, if $\tau^m = o(\Delta t^m)$. *Convergence* finally characterizes the quality of the algorithm solutions under disturbed input data. Some concepts of stability combined with consistency imply convergence for the algorithm (DEUFLHARD & BORNEMANN 2002).

In this work, only the cases $\theta \in \{0, 0.5, 1\}$ of the generalized trapezoidal rule are briefly discussed. For properties of further numerical integrators, reference is made to the literature (T. J. R. HUGHES 1987; ZIENKIEWICZ & TAYLOR 2002). The choice $\theta = 0$ corresponds to the explicit *forward Euler*[9] *method*. The advantage of explicit methods is that the approximation of the next value is only calculated from the previous time step. However, the size of the increment must be limited to ensure stability (GRIFFITHS & HIGHAM 2010). For the two cases $\theta = 0.5$ and $\theta = 1$, implicit methods are obtained, i. e. a system of equations must be solved in each time step. The first case leads to the *Crank*[10]*-Nicolson*[11] *method*, and the second case to the *backward Euler method*. Both methods are consistent and stable. The Crank-Nicolson method has a higher order of consistency, but, in contrast to the backward Euler method, it is not *L*-stable (DEUFLHARD & WEISER 2012).

The distribution of the time steps strives for an optimal ratio between effort and error. This corresponds to an evenly distributed truncation error across all time steps (DEUFLHARD & BORNEMANN 2002). The temporal discretization is thus far specified in the considerations. Usually, however, the behavior of the

[9] **Leonhard Euler** (* April 15th, 1707 - † September 18th, 1783), Swiss mathematician and physicist.

[10] **John Crank** (* February 6th, 1916 - † October 3rd, 2006), British mathematician.

[11] **Phyllis Nicolson** (* September 21st, 1917 - † October 6th, 1968), British mathematician.

solution and thus the optimal distribution of the time steps is not known. To obtain an evenly distributed truncation error, time step control strategies are used which determine the next time step based on the current result or, in the case of an excessively large error, repeat the previous calculation with a smaller time step. Error estimators are required for this (GRIFFITHS & HIGHAM 2010).

2.4.2 Linear solvers

The most obvious way to obtain the solution $\boldsymbol{x} \in \mathbb{R}^{n_\Xi}$ of a system of linear equations

$$\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b} \tag{2.41}$$

with an invertible symmetrical *coefficient matrix* $\boldsymbol{A} \in \mathbb{R}^{n_\Xi \times n_\Xi}$ and a right-hand side vector $\boldsymbol{b} \in \mathbb{R}^{n_\Xi}$ is to determine the inverse $\boldsymbol{A}^{-1}$. However, this is very costly (required number of operations[12] ${n_\Xi}^3$) if $\boldsymbol{A}$ has no special structure (MARSDEN et al. 2008). For the determination of $\boldsymbol{x}$, two classes of linear solvers are available: *Direct Methods* and *Iterative Methods*.

Direct Methods decompose the coefficient matrix $\boldsymbol{A}$ and convert Equation 2.41 into two systems of equations that are easier to solve. The *LU factorization* and the *Cholesky*[13] *decomposition* are mentioned. In the case of the *LU* factorization, $\boldsymbol{A}$ is decomposed in a lower triangular matrix $\boldsymbol{L}$ and an upper triangular matrix $\boldsymbol{U}$:

$$\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{U}. \tag{2.42}$$

Accordingly, the systems of linear equations

$$\boldsymbol{L} \cdot \boldsymbol{y} = \boldsymbol{b}, \qquad \boldsymbol{U} \cdot \boldsymbol{x} = \boldsymbol{y} \tag{2.43}$$

are solved subsequently by *forward* and *backward substitution* with the additionally introduced vector $\boldsymbol{y} \in \mathbb{R}^{n_\Xi}$. The entries of $\boldsymbol{y}$ and $\boldsymbol{x}$ are determined successively starting with y_1 and x_{n_Ξ}, respectively. Due to the triangular structure, an entry depends only on the previous entries already determined.

The same applies for the Cholesky decomposition; however, the upper triangular matrix $\boldsymbol{L}$ is additionally the transposed one of the lower triangular matrix $\boldsymbol{U}$, i. e.

$$\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{L}^\top. \tag{2.44}$$

Therefore, only the matrix $\boldsymbol{L}$ must be determined. In addition to the symmetry, the matrix has to be *positive definite*, i. e. $\boldsymbol{z}^\top \cdot \boldsymbol{A} \cdot \boldsymbol{z} > 0$ for every non-zero vector $\boldsymbol{z} \in \mathbb{R}^{n_\Xi}$. The requirement of positive definiteness ensures that the root in

[12] only multiplications and divisions are counted

[13] **André-Louis Cholesky** (* October 15th, 1875 - † August 31st, 1918), French mathematician.

Equation 2.45, which appears later, is always non-negative. However, as the property is not otherwise relevant in this thesis, it is not discussed further.

The cost of the forward and backward substitution is ${n_\Xi}^2/2$ each. In general, the *LU* factorization requires ${n_\Xi}^3/3$ operations, and ${n_\Xi}^3/6$ operations are necessary for the Cholesky decomposition (Marsden et al. 2008). The decomposition of the coefficient matrix therefore clearly outweighs the substitutions. However, since the decomposition is independent of the vector $\boldsymbol{b}$, it is valid for any vector $\boldsymbol{b} \in \mathbb{R}^{n_\Xi}$.

Since the Cholesky decomposition requires half of the operations of the *LU* factorization, only the approach for this procedure is mentioned. The formulas

$$L_{ll} = \sqrt{A_{ll} - \sum_{s=1}^{l-1} {L_{ls}}^2}, \tag{2.45a}$$

$$L_{kl} = \frac{1}{L_{ll}} \left(A_{kl} - \sum_{s=1}^{l-1} L_{ks} L_{ls} \right), \qquad k > l \tag{2.45b}$$

apply to the entries of the matrix $\boldsymbol{L}$. In geometric terms, only the determined values to the left and above are required to determine a further entry to the matrix $\boldsymbol{L}$. Starting with L_{11}, the remaining matrix $\boldsymbol{L}$ is calculated either line by line or column by column.

Based on a starting value $\boldsymbol{x}_0$, Iterative Methods determine iterative improvements until these updates are smaller than a specified threshold value. A general statement about the operations required to determine a solution, as in the case of Direct Methods, is therefore not possible for Iterative Methods, since it depends on the number of iterations necessary. The methods are mainly subdivided into *Stationary Iterative Methods* and *Nonstationary Iterative Methods* (Barrett et al. 1994). The former are based on splitting the coefficient matrix into two matrices $\boldsymbol{M}$ and $\boldsymbol{N}$:

$$\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}. \tag{2.46}$$

Unlike Equation 2.41, the system of linear equations

$$\boldsymbol{M} \cdot \boldsymbol{x}_{i+1} = \boldsymbol{N} \cdot \boldsymbol{x}_i + \boldsymbol{b} \tag{2.47}$$

is solved in each iteration i. To obtain an advantage over the initial system of equations, an easily invertible matrix $\boldsymbol{M}$ is required. Basic methods are derived from the splitting of $\boldsymbol{A}$ into its cumulative portion of a strict lower triangular part $\boldsymbol{L}$, a strict upper triangular part $\boldsymbol{U}$ and a diagonal part $\boldsymbol{D}$ according to

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U}. \tag{2.48}$$

This results in the *Jacobi*[14] *method* ($\boldsymbol{M} := \boldsymbol{D}$) and the *Gauss-Seidel*[15] *method* ($\boldsymbol{M} := \boldsymbol{D} + \boldsymbol{L}$). The iteration in Equation 2.47 converges if and only if the spectral radius of the *Iteration Matrix* $\boldsymbol{M}^{-1} \cdot \boldsymbol{N}$, i. e. the largest eigenvalue in terms of amount denoted as $\varrho\big(\boldsymbol{M}^{-1} \cdot \boldsymbol{N}\big)$, is smaller than 1 (GOLUB & VAN LOAN 1983). In this case, linear convergence is obtained for the matrix splitting, i. e.

$$\lim_{i \to \infty} \frac{|\boldsymbol{x}_{i+1} - \boldsymbol{x}|}{|\boldsymbol{x}_i - \boldsymbol{x}|} = c_{\text{rate}} < 1 \tag{2.49}$$

The value c_{rate} is denoted as *rate of convergence* and describes the speed at which a sequence approaches its limit. Moreover, from the proof of convergence, it follows that

$$\frac{|\boldsymbol{x}_{i+1} - \boldsymbol{x}|}{|\boldsymbol{x}_i - \boldsymbol{x}|} < \varrho\Big(\boldsymbol{M}^{-1} \cdot \boldsymbol{N}\Big) \tag{2.50}$$

applies and that the rate of convergence is limited upwards by the spectral radius (SAAD 2003). Thus, the smaller the spectral radius is, the faster the sequence $(\boldsymbol{x}_i)_{i \in \mathbb{N}}$ converges.

With the introduction of a *relaxation parameter* ω, e. g. like

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \omega \cdot (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_i)\,, \tag{2.51}$$

for the *modified Richardson*[16] *iteration* ($\boldsymbol{M} := \frac{1}{\omega}\boldsymbol{I}$, $\omega \neq 0$), further methods are created. The *Successive over-relaxation method* ($\boldsymbol{M} := \frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}$, $\omega \neq 0$) follows from a similar procedure. A clever choice of ω reduces the spectral radius and accelerates the convergence.

Nonstationary Iterative Methods derive information from the current iteration (BARRETT et al. 1994). The best-known representative of them is the *Conjugated gradient method*. It takes advantage of the fact that the solution of Equation 2.41 is the unique minimum of the function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\top} \cdot \boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{x}^{\top} \cdot \boldsymbol{b}. \tag{2.52}$$

The name already implies that the gradient is included in each iteration, but this is done on a subspace. As in the case of Stationary Iterative Methods, additional conditions are necessary to ensure convergence. More information on the methods is provided in GOLUB & O'LEARY (1989).

[14] **Carl Gustav Jacob Jacobi** (* December 10th, 1804 - † February 18th, 1851), German mathematician.

[15] **Philipp Ludwig von Seidel** (* October 23rd, 1821 - † August 13th, 1896), German mathematician.

[16] **Lewis Fry Richardson** (* October 11th, 1881 - † September 30th, 1953), English mathematician, physicist and meteorologist.

2.5 Simulation of Heat Transfer

Since analytical solutions are only available in special cases, the heat equation is solved numerically to obtain solutions even for a complex geometry. First, the weak formulation is prepared. After it has been discretized in both space and time, the resulting system of equations is solved. The present section presents the procedure according to the *Method of Lines*, in which the discretization is carried out first in space and then in time (cf. Figure 2.3). The *Galerkin*[17] *method* is used for the spatial discretization. The entire procedure is also implemented for the transient thermal analysis in the software ccx (DHONDT 2004).

2.5.1 Weak formulation of the heat equation

The regularity requirements for the solution of Equation 2.10 are often too strong so that a solution does not exist (BATHE 2006). Therefore, the equation is weak in the sense that it no longer needs to be fulfilled point by point, but only in the integral form

$$\int\limits_{\Omega} \rho c(T) \frac{\partial T}{\partial t} v \, \mathrm{d}\Omega - \int\limits_{\Omega} \boldsymbol{\nabla} \cdot [\lambda(T) \boldsymbol{\nabla} T] v \, \mathrm{d}\Omega = 0, \qquad \forall t \in \mathrm{Y} \qquad (2.53)$$

for arbitrary test functions $v \in C^2(\Omega)$ that vanish on the Dirichlet boundary Γ_D. Using *Green's*[18] *identities*, the equation is transformed for all t in Y to

$$\int\limits_{\Omega} \rho c(T) \frac{\partial T}{\partial t} v \, \mathrm{d}\Omega + \int\limits_{\Omega} \lambda(T) \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} T \, \mathrm{d}\Omega = \int\limits_{\Gamma} \lambda(T) v \boldsymbol{n} \cdot \boldsymbol{\nabla} T \, \mathrm{d}\Gamma. \qquad (2.54)$$

By dividing the integral over the boundary according to Equation 2.13 and inserting the corresponding boundary conditions, the following equation is obtained:

$$\begin{aligned} \int\limits_{\Omega} \rho c(T) \frac{\partial T}{\partial t} v \, \mathrm{d}\Omega + \int\limits_{\Omega} \lambda(T) \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} T \, \mathrm{d}\Omega + \int\limits_{\Gamma_\mathrm{C}} \alpha v T \, \mathrm{d}\Gamma & \\ = \int\limits_{\Gamma_\mathrm{C}} \alpha v T_{\Gamma_\mathrm{C}} \, \mathrm{d}\Gamma - \int\limits_{\Gamma_\mathrm{N}} v \dot{q}_{\Gamma_\mathrm{N}} \, \mathrm{d}\Gamma, & \qquad \forall t \in \mathrm{Y}. \qquad (2.55) \end{aligned}$$

Now, significantly lower requirements are demanded concerning the solution T and the functions v compared to Equation 2.10, i. e. T and v only need to belong

[17] **Boris Grigoryevich Galerkin** (*March 4th, 1871 - †July 12th, 1945), Soviet mathematician and engineer.

[18] **George Green** (*July 14th, 1793 - †May 31st, 1841), British mathematical physicist.

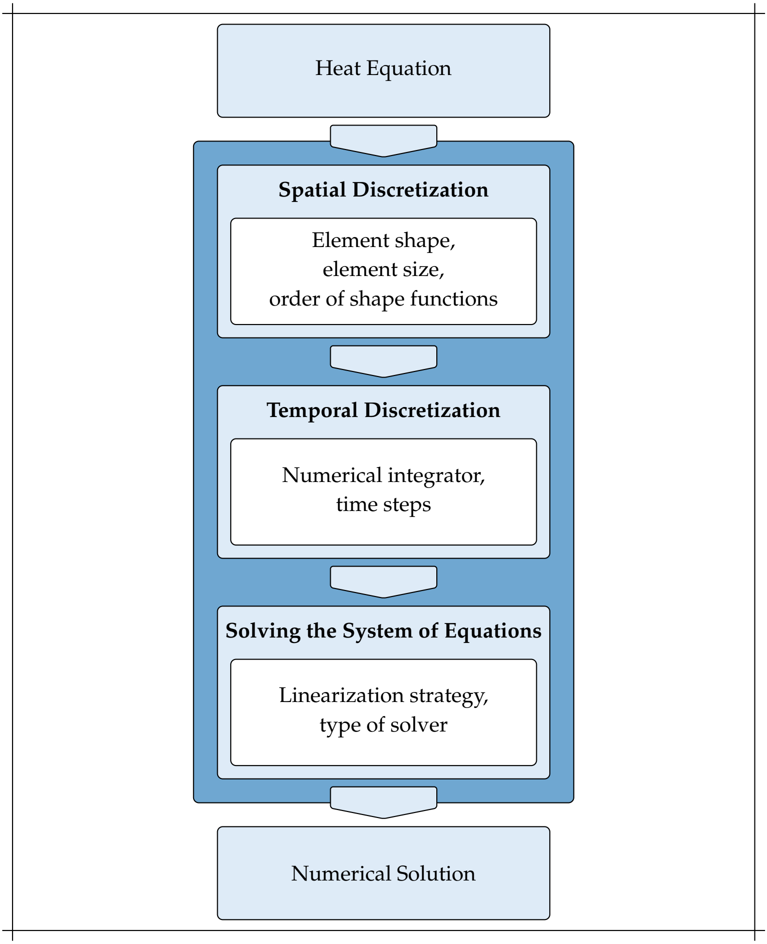


Figure 2.3: Steps required for numerically solving the heat equation with the Method of Lines and the necessary settings.

to the *Sobolev*[19] *space* $H^1(\Omega)$. Sobolev spaces are functional spaces of weak differentiable functions and form the basis of the solution theory of PDEs. They are mentioned here for the sake of completeness. For more detailed information, however, reference is made to EVANS (2015).

The condition that a solution $T \in H^1(\Omega)$ must satisfy Equation 2.55 for all $v \in H^1(\Omega)$ is referred to as the *weak formulation* of Equation 2.10 (DEUFLHARD & WEISER 2012). The functions T and v do not necessarily have to belong to the same function space. At this point, however, the later use of the Galerkin method is already anticipated, in which both functions are based on the same function space.

2.5.2 Spatial discretization

The requirements for the solution are reduced with the weak formulation in Equation 2.55. However, since $H^1(\Omega)$ is an infinite dimensional function space, a solution is still not numerically determinable. Therefore, the set Ω is first decomposed into *elements* ϵ_u with $u \in (1, \dots, u_\Theta)$. Common shapes in the three-dimensional case are tetrahedrons and hexahedrons. The term *element size* used in this context corresponds to the respective element diameter. The resulting partition of the spatial domain

$$\Theta = \left\{ \epsilon_u \,\middle|\, u \in (1, \dots, u_\Theta) \right\} \tag{2.56}$$

is called a *grid* or a *mesh*, and the maximum occurring element size is represented by the parameter d_Θ. The corners $\boldsymbol{r}_n$ of the elements are called *nodes*, with the corresponding set of nodes specified as

$$\Xi = \left\{ \boldsymbol{r}_n \,\middle|\, n \in (1, \dots, n_\Xi) \right\} \tag{2.57}$$

and $n_\Xi = |\Xi|$. An *edge* is the connection of two nodes. In the case that nodes of an element lie on an edge of an adjacent element, Θ is called a *nonconformal* grid; otherwise, it is a *conformal* grid.

Lagrange[20] *shape functions* are defined on each element $\epsilon_u \in \Theta$ using its corresponding nodes. These are interpolation polynomials that are equal to 1 at node $\boldsymbol{r}_n$ and equal to 0 at all other nodes. This provides linear shape functions per element. For higher-order shape functions, the set of nodes is extended

[19] **Sergei Lvovich Sobolev** (* October 6th, 1908 - † January 3rd, 1989), Soviet mathematician.

[20] **Joseph-Louis de Lagrange** (* January 25th, 1736 - † April 10th, 1813), Italian mathematician and astronomer.

by nodes on the element edges or inside. The function space formed by the Lagrange functions of order s on an element ϵ_u is specified as $P_s(\epsilon_u)$.

From the above definitions, the finite dimensional, discrete function space

$$V^s(\Omega) = \left\{ v \,\middle|\, v \in C^0(\Omega)\,, v|_{\epsilon_u} \in P_s(\epsilon_u), v|_{\Gamma_\mathrm{D}} = 0, \epsilon_u \in \Theta \right\} \tag{2.58}$$

is derived where $v|_{\epsilon_u}$ denotes the restriction of v to the element ϵ_u, i. e. $v : \epsilon_u \to \mathbb{R}$. The demand for a solution $T \in V^s(\Omega)$ that satisfies the equation

$$\begin{aligned}\int\limits_{\Omega} \rho c(T) \frac{\partial T}{\partial t} v \, \mathrm{d}\Omega + \int\limits_{\Omega} \lambda(T) \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} T \, \mathrm{d}\Omega + \int\limits_{\Gamma_\mathrm{C}} \alpha v T \, \mathrm{d}\Gamma \\ = \int\limits_{\Gamma_\mathrm{C}} \alpha v T_{\Gamma_\mathrm{C}} \, \mathrm{d}\Gamma - \int\limits_{\Gamma_\mathrm{N}} v \dot{q}_{\Gamma_\mathrm{N}} \, \mathrm{d}\Gamma, \qquad \forall v \in V^s(\Omega)\,. \end{aligned} \tag{2.59}$$

at all times t in Υ leads to the *semi-discrete weak formulation* of the heat equation, which is also the basis of the Finite Element Method (FEM) (ZIENKIEWICZ & TAYLOR 2002). A solution for Equation 2.59 has the form

$$T(\boldsymbol{r}, t) = \sum_{n=1}^{n_\Xi} T_n(t)\, v_n(\boldsymbol{r})\,, \qquad v_n \in V^s(\Omega) \tag{2.60}$$

with the semi-discrete temperature values

$$T_n(t) = T(\boldsymbol{r}_n, t)\,. \tag{2.61}$$

This approach is inserted into Equation 2.59. Since the terms T_n are spatially independent, they are dragged in front of the integrals. Due to the finite dimension of $V^s(\Omega)$, the equations are explicitly specifiable for each $v \in V^s(\Omega)$. As the choice of test functions is also arbitrary, the vectoral representation

$$\boldsymbol{C}(T) \cdot \dot{\boldsymbol{T}}(t) + [\boldsymbol{K}(T) + \boldsymbol{K}_C(T)] \cdot \boldsymbol{T}(t) - \boldsymbol{q}(t) = 0, \qquad \forall t \in \Upsilon \tag{2.62}$$

of Equation 2.59 is obtained with

$$\boldsymbol{T}(t) = (T_1(t)\,, \ldots, T_{n_\Xi}(t))\,, \tag{2.63}$$

$$\dot{\boldsymbol{T}}(t) = \frac{d\boldsymbol{T}(t)}{dt} \tag{2.64}$$

and the appearing terms are calculated as follows:

$$C_{kl}(T) = \int_{\Omega} \rho c(T) v_k v_l \,\mathrm{d}\Omega, \tag{2.65a}$$

$$K_{kl}(T) = \int_{\Omega} \lambda(T) \boldsymbol{\nabla} v_k \cdot \boldsymbol{\nabla} v_l \,\mathrm{d}\Omega, \tag{2.65b}$$

$$K_{\mathrm{C},kl}(T) = \int_{\Gamma_{\mathrm{C}}} \alpha v_k v_l \,\mathrm{d}\Gamma, \tag{2.65c}$$

$$q_k(t) = \int_{\Gamma_{\mathrm{C}}} \alpha v_k T_{\Gamma_{\mathrm{C}}}(t) \,\mathrm{d}\Gamma - \int_{\Gamma_{\mathrm{N}}} v_k \dot{q}_{\Gamma_{\mathrm{N}}}(t) \,\mathrm{d}\Gamma. \tag{2.65d}$$

$\boldsymbol{C}$ is the *capacity* matrix and $\boldsymbol{K}$ is the *conductance* matrix. $\boldsymbol{K}_{\mathrm{C}}$ is the portion of the conductance matrix as a result of the convection boundary condition. Although it has more similarities to the capacity matrix, it is derived from the same integral as the conductance matrix when transformed with Green's identities. Usually, it is not directly assigned to the conductance matrix, since it is only present if convection boundary conditions are also present. In the case of the software ccx, it is assigned to the driving flux (DHONDT 2004).

The various integrals are determined by a numerical integration formula on a reference element. The temperature-dependent material parameters are evaluated accordingly at the integration points used to determine the integral. Afterwards, the results are projected onto the elements ϵ_u and finally assembled into global matrices. For the detailed procedure, refer to T. J. R. HUGHES (1987).

Any curved boundary Γ of the domain Ω results in geometric approximation errors due to the presented subdivision into elements with straight edges. The element size must therefore be reduced locally if necessary to keep these errors negligibly small. For the creation of meshes and the necessary requirements for use in simulation, reference is made to FREY & GEORGE (2000) and ZIENKIEWICZ & TAYLOR (2002). In what follows, conformal grids with negligibly small geometric approximations errors are always assumed.

2.5.3 Temporal discretization

In Equation 2.62, the problem is already reduced to an IVP. However, the equation is still continuous in time and contains a temporal derivation. The time interval $[t^0, t^{m_\mathrm{Y}}]$ is thus discretized into $\mathrm{Y}^m = \{t^0, t^1, \ldots, t^{m_\mathrm{Y}}\}$ with

$$t^m = t^{m-1} + \Delta t^{m-1}, \qquad m \in \{1, \ldots, m_\mathrm{Y}\}. \tag{2.66}$$

The generalized trapezoidal rule from Equation 2.39

$$\boldsymbol{T}^m = \boldsymbol{T}^{m-1} + (1-\theta)\,\Delta t^m \dot{\boldsymbol{T}}^{m-1} + \theta\,\Delta t^m\,\dot{\boldsymbol{T}}^m \tag{2.67}$$

is used as numerical integrator. The approximations $\boldsymbol{T}^m$ of $\boldsymbol{T}(t^m)$ for $m = 1, \ldots, m_\Upsilon$ are finally obtained by solving the system

$$\begin{aligned}&\left[\frac{1}{\theta\,\Delta t^m}\boldsymbol{C}(T^m) + \boldsymbol{K}(T^m) + \boldsymbol{K}_\mathrm{C}(T^m)\right]\cdot \boldsymbol{T}^m \\ &\qquad = \frac{1}{\theta\,\Delta t^m}\boldsymbol{C}(T^m)\cdot\left[\boldsymbol{T}^{m-1} + (1-\theta)\,\Delta t^m\,\dot{\boldsymbol{T}}^{m-1}\right] + \boldsymbol{q}(t^m)\end{aligned} \tag{2.68}$$

of nonlinear equations. The notation

$$T^m(\boldsymbol{r}) = \sum_{n=1}^{n_\Xi} T_n^m v_n(\boldsymbol{r}), \qquad\qquad v_n \in V^s(\Omega) \tag{2.69}$$

is introduced with the nodal temperature values T_n^m of the approximation $\boldsymbol{T}^m$.

2.5.4 Solving the system of nonlinear equations

Equation 2.68 is a system of nonlinear equations due to the temperature-dependent matrices. Direct Methods for solving systems of nonlinear equations are only available for special cases. Therefore, it is linearized and solved iteratively for each time step m. Starting with the value of the previous time step $\boldsymbol{T}_0^m = \boldsymbol{T}^{m-1}$, an incremental improvement $\Delta\boldsymbol{T}_i^m$ for the value $\boldsymbol{T}_{i-1}^m$, i. e.

$$\boldsymbol{T}_i^m = \boldsymbol{T}_{i-1}^m + \Delta\boldsymbol{T}_i^m, \tag{2.70}$$

is determined in each iteration i until $\Delta\boldsymbol{T}_i^m < \varepsilon$ applies for a specified threshold ε. The value received in the last iteration i is accepted as $\boldsymbol{T}^m = \boldsymbol{T}_i^m$.

Several procedures are available for this (BERGHEAU & FORTUNIER 2008). A common one is to use the temperature values of the previous iteration for the matrices and to update them iteratively (DHONDT 2004; TANNEHILL et al. 1997). In each iteration i, the system of linear equations

$$\begin{aligned}&\left[\frac{1}{\theta\,\Delta t^m}\boldsymbol{C}(T_{i-1}^m) + \boldsymbol{K}(T_{i-1}^m) + \boldsymbol{K}_\mathrm{C}(T_{i-1}^m)\right]\cdot \boldsymbol{T}_i^m \\ &\qquad = \frac{1}{\theta\,\Delta t^m}\boldsymbol{C}(T_{i-1}^m)\cdot\left[\boldsymbol{T}^{m-1} + (1-\theta)\,\Delta t^m\,\dot{\boldsymbol{T}}^{m-1}\right] + \boldsymbol{q}(t^m)\,.\end{aligned} \tag{2.71}$$

is solved with the methods from Subsection 2.4.2 to obtain an incremental improvement $\Delta\boldsymbol{T}_i^m$. Using the previous temperature values is referred to as *lagging the coefficients* (TANNEHILL et al. 1997). The properties listed above ensure that the system of linear equations is unambiguously solvable. Due to the same function space for solution and test functions, the coefficient matrix obtained is also symmetrical, as indicated in Equations 2.65.

2.6 Modeling and Simulation of Laser Beam Melting

This section is the basis for the later investigations on the Discretization and Algorithm Selection in the simulation of LBM. The Physical System of LBM, the Conceptual Model and finally the Mathematical Model are presented.

2.6.1 Physical System

Laser Beam Melting (LBM) belongs to the class of Powder Bed Fusion (PBF) processes. This class encompasses all AM processes that use thermal energy to selectively fuse areas of a powder bed (WOHLERS et al. 2016). The terms Direct Metal Laser Sintering (DMLS) and Selective Laser Melting (SLM) are used synonymously for LBM.

The processing principle of LBM consists of three repeating steps (ZAEH 2013). The powder of the desired material is applied with a *coater* to a *build plate*, which is screwed to the *build platform* of the LBM machine. Corresponding areas of the part to be produced in the coated powder layer are solidified with a laser using an F-Theta lens and galvanometer scanners. Accordingly, the current top layer is always referred to as the *build surface*. The energy source fuses the layer with underlying layers along scan vectors arranged in a predefined *scanning strategy*. Subsequently, the build platform is lowered by the height of one layer. The steps of powder deposition, solidification and lowering of the build platform are carried out under a circulating shield gas atmosphere in a closed *build chamber* and are repeated until the entire part is generated.

The build plate serves not only to fix the part, but also to dissipate thermal energy. To reduce process-induced residual stresses, the build platform is heated to a defined temperature T_{pre} (SOCHALSKI-KOLBUS et al. 2015). In the preparation of the system set-up, the build plate is screwed to the build platform once the preheating temperature is reached. After the build cycle, the part and the build plate first cool down to room temperature before they are freed from the powder and removed from the build chamber.

Depending on the part and its orientation on the build plate, *support structures* are required. These are structures that are built up simultaneously with the part but do not actually belong to the part and are removed afterwards. Among other things, support structures are used to prevent overhangs from sinking into the powder bed, as well as also to reduce the cause and the effect of process-related distortions (KROL et al. 2012). Furthermore, support structures contribute to heat transport and thus prevent heat accumulation. At the same time, they also represent a mechanical fixation of the components and counteract distortions.

For the described buildup process, it is necessary that the part's information, as well as information on the support structures, is available in layers. The necessary data preparation is accompanied by the orientation of the part on the build platform and the associated determination of the buildup direction. Among other things, this influences manufacturability, construction time, necessary post-processing steps and surface quality (GIBSON et al. 2015). A general optimal orientation does not usually exist, as this depends on the weighting of the various target values.

The properties of parts manufactured by LBM are meanwhile comparable to conventionally produced parts (GEBHARDT 2013). However, this depends on the process parameters used. The significant parameters are laser power P, scanning velocity of the beam focus v_b, beam diameter in the process zone d_b, distance of neighbored scan vectors d_sv and layer thickness d_l (GIBSON et al. 2015; KRAUSS 2016). The process parameters vary for the materials to be processed.

The build time for an exemplary part with a height of 200 mm in buildup direction is in the range of hours or even days and, according to the processing principle, consists mainly of the travel time of the laser Δt_travel, the lowering of the build platform and the cover time for the coater. The exact duration of the laser travel time depends on the size of the component cross-sections in the buildup direction. The other two, collectively referred to as Δt_dwell, are constant for each layer and do not depend on the geometry but only on the respective manufacturing system.

2.6.2 Conceptual Model

Two coordinate systems are introduced for the Conceptual Model. The *global machine coordinate system* is centrally fixed on the surface of the build plate. The positive x-direction is defined from left to right from a frontal view of the build chamber. The z-direction corresponds to the normal of the build surface. To obtain a right-hand side system, the y-direction runs from the front to the back of the build chamber. In addition, a *local part coordinate system* is defined whose origin is shifted from the machine coordinate system in the z-direction to the current build surface. The z-axis points in the direction of the part, and, accordingly, the direction of the y-axis is reversed compared to the global machine coordinate system to get a right-hand system again.

Dimensional accuracy in LBM is a thermomechanical problem, as the cause of the distortions are the process-related temperature gradients (KRUTH et al. 2004). Since the appearing dimensional deviations in LBM are relatively small, the retroactive effect of the distortions on the temperature and the transient mechanical effects are neglected (L. ZHANG et al. 2004). Therefore, a sequential coupling of a transient thermal analysis and a quasi-static mechanical analysis is considered (cf. Figure 2.4a).

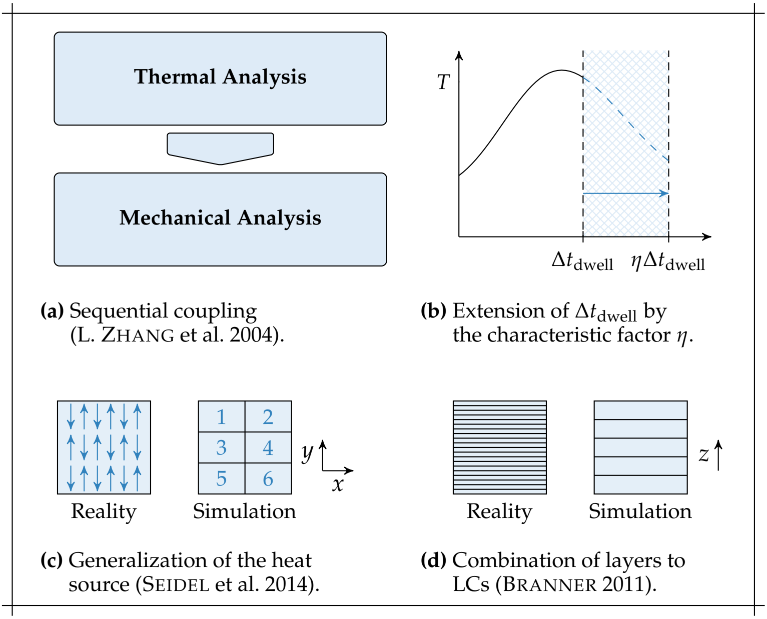
The considered geometry is reduced to the part and the base plate (DENLINGER 2017; N'DRI et al. 2015; SEIDEL & ZAEH 2018). Scanner optics and coater are neglected, as their arrangement and dimensions have no influence on the deformations. The build chamber is taken into account in that it specifies the dimensions for the build plate and limits the size of the part. The temperature of the build platform is completely determined by the preheating temperature T_{pre} and is therefore only included in the model as a temperature boundary condition for the build plate. Non-consolidated powder has a thermal insulating effect on the part (ALKAHARI et al. 2012; ROMBOUTS et al. 2005). Surfaces with adjacent powder are therefore assumed to be adiabatic. A convection boundary condition is assumed for the side surfaces of the build plate, as these are in contact with the machine structure, which is affected by preheating. The remaining free build surface experiences convection by the process gas flow, which is assumed to be homogeneous for the entire surface, and the temperature of which corresponds to the preheating temperature. Heat transport via radiation is neglected as it has little importance in processes with fast heat sources (CLINE & ANTHONY 1977).

The modeling of LBM is divided into the iterative buildup of the part and a final cooling to room temperature. The preheating of the base plate is not considered, as it expands free of stress due to the lack of fixation. However, if the build plate is screwed to the build platform before the preheating process, it must be taken into account. The Conceptual Model does not distinguish between lowering the build platform and powder application. Decisive for the distortion is the accumulated time Δt_{dwell} of both steps, in which the thermal energy from the melting of the current layer is distributed in the part (DENLINGER et al. 2015). In contrast to the three phases in the real process, the modeled processing principle consists only of two phases: 'Heating' and 'Cooling'. The term 'Heating' is chosen because the actual melting and solidification processes are not regarded. Solid material is assumed directly without any preceding phase transformations, since the liquid state of aggregation does not generate any internal stresses. Up to this state, however, thermal energy is already introduced into the part during the melting of the powder particles, as well as in the phase of the liquid state. There are two approaches for this 'Heating' phase: a temperature-based approach (MA & BIN 2007; PAPADAKIS, LOIZOU, RISSE & SCHRAGE 2014; SCHILP et al. 2014; SEIDEL & ZAEH 2018), which is also used in this thesis, and a heat flux-based approach (CONTUZZI et al. 2011; MATSUMOTO et al. 2002; VAN BELLE et al. 2012). The temperature-based approach sets the top layer to solidus temperature T_{sol} corresponding to the duration $\Delta t_{\text{heating}}$ of the liquid state of a point in the real process. To determine $\Delta t_{\text{heating}}$, the solution of ROSENTHAL (1946) is used (SEIDEL 2016). The heat flux-based approach applies a constant thermal energy input until the solidus temperature is reached. For both approaches, there are different gradations in the mapping of the scanning strategy (KROL et al. 2013; PAPADAKIS, LOIZOU, RISSE, BREMEN, et al. 2014; SEIDEL et al. 2014). Within a layer, scan trajectories are merged into coarser scan fields (cf. Figure 2.4c).

Several layers are combined into a so-called Layer Compound (LC) (BRANNER 2011) (cf. Figure 2.4d). This is based on the assumption that solidification of adjacent layers leads to similar temperature fields for the part, and, consequently, to similar deformations. Accordingly, it is only necessary to consider certain layers that lead to changes in the deformations. However, this interpretation on its own would still require a local resolution of the heat source according to the layer height. At the same time, the total heat balance of the build cycle would not be depicted, and, consequently, the correct representation of the reheating of already solidified layers would not be guaranteed due to the melting of further layers. To maintain the energy balance, the expansion of the heat source is therefore extended to the height of the LC. However, this now means that a significantly larger amount of thermal energy needs to be dissipated during the cooling time $\Delta t_{\mathrm{dwell}}$. Depending on the height d_{LC} of the LC and the real layer height d_{l}, a characteristic factor η is therefore necessary which transforms the time accordingly to ensure the correct reheating of already solidified layers (cf. Figure 2.4b).

The appearing temperatures range from solidus temperature to room temperature. Using the example of Inconel 718 with a solidus temperature T_{sol} of 1528 K (POTTLACHER et al. 2002), this means temperature differences of up to 1235 K for an assumed room temperature of 293 K. Therefore, temperature-dependent material parameters are used (SCHOINOCHORITIS et al. 2017).

With the assumptions and simplifications made, the following procedure results. Only the build plate is initially present. The part is assembled in two nested loops. Starting from the lowest LC, all scan fields in the inner loop are activated, i. e. the calculation domain is extended accordingly. The load is applied instantaneously to the entire scan field when it is activated. Afterwards, the load is replaced by convection on the build surface for the activation of the remaining scan fields. After all scan fields have been activated, cooling takes place over the time $\Delta t_{\mathrm{dwell}}$. Due to the introduction of the scan fields, the travel time of the laser, and thus also the cooling time per layer, is no longer correctly represented. To preserve the gradients in the xy-plane between the scan fields, the missing travel time $\Delta t_{\mathrm{travel}}$ is not considered between the activation of the scan fields, but is added to the cooling time $\Delta t_{\mathrm{dwell}}$. The outer loop repeats the 'Heating' and 'Cooling' for all LCs. Once all LCs have been created, the boundary condition of the preheating temperature is deleted and the part cools down to room temperature over the time Δt_{end}, which then also determines the



(a) Sequential coupling (L. ZHANG et al. 2004).

(b) Extension of Δt_{dwell} by the characteristic factor η.

(c) Generalization of the heat source (SEIDEL et al. 2014).

(d) Combination of layers to LCs (BRANNER 2011).

Figure 2.4: Main assumptions and simplifications of the Conceptual Model.

reference temperature for the convection. A schematic flow of the Conceptual Model is provided in Figure 2.5

2.6.3 Mathematical Model

Initially, several notations are introduced for the Mathematical Model. The LCs are named with Ω_w and numbered by w in ascending order, starting from the lowest LC with Ω_1 according to the global machine coordinate system. Corresponding scan fields of Ω_w are labeled by the additional index b, i. e. $\Omega_{w,b}$. The top boundary of a layer Ω_w or scan field $\Omega_{w,b}$ is denoted as Γ_w or $\Gamma_{w,b}$, respectively. The build plate is defined as the set Ω_{bp}. Based on this, the boundary of the build plate which has contact to the build platform is denoted with Γ_{pre}, and the surfaces in contact with the machine structure are labeled as Γ_{side}. An illustration of the defined sets is provided in Figure 2.6.

For notational simplification, the additional sets Ψ_w of all layers up to layer Ω_w and the build plate Ω_{bp} are introduced, as well as the sets $\Psi_{w,b}$ containing

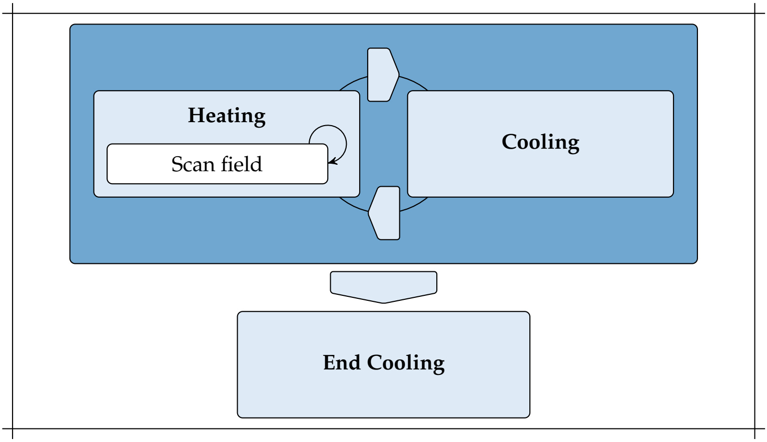


Figure 2.5: Schematic flow for the thermal analysis of the Conceptual Model of LBM.

Ψ_{w-1} and all scan fields of the layer Ω_w up to scan field $\Omega_{w,b}$:

$$\Psi_w = \Omega_{\text{bp}} \cup \bigcup_{s=1}^{w} \Omega_s, \tag{2.72}$$

$$\Psi_{w,b} = \Psi_{w-1} \cup \bigcup_{s=1}^{b} \Omega_{w,s}. \tag{2.73}$$

In addition to the notation Ω_w, a local notation Λ_w for the LCs is introduced based on the local part coordinate system. It includes only the currently existing LCs and numbers them from the build surface to the base plate; i. e., the current top LC always corresponds to Λ_1.

Despite the introduction of LCs, the characteristic factor maintains the correct reheating of deeper areas. Deviating from STRANTZA et al. (2018), however, there is no linear scaling of the time according to the combined layers to preserve the total process time. The property of the linear heat equation from Equation 2.30 is used for this purpose. By introducing LCs, the characteristic length measure is changed from the layer height d_{l} to the height d_{LC} of an LC. To obtain similar temperature curves with respect to the new characteristic length measure, the time is scaled according to the characteristic factor

$$\eta = \left(\frac{d_{\text{LC}}}{d_{\text{l}}}\right)^2. \tag{2.74}$$

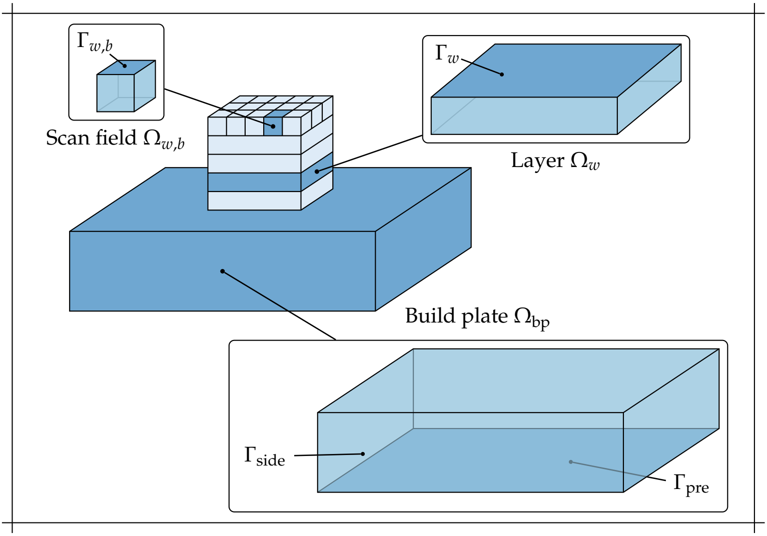


Figure 2.6: Graphical illustration of the defined sets for build plate, layer and scan field.

The characteristic factor corresponds to nothing more than a transformation in the sense of the similarity theory. The scaling of the time with the characteristic factor η is equivalent to the relative scaling of the spatial variables from the height d_{LC} of an LC to the layer height d_{l}. Therefore, temperature curves of the real process constellations are obtained independent of the number of combined layers for an LC. From here, the terms layer and LC are used synonymously in the context of M&S. Nevertheless, it should be mentioned that the layers of the part were summarized in z-direction, but with the characteristic factor scaling in all spatial directions and in all areas is associated. Correspondingly, this also concerns the build plate. As a result, the cooling effect starts earlier due to the boundary condition of the preheating on the bottom of the build plate. However, this only affects the period of time when the penetration depth has reached the bottom of the build plate.

The Conceptual Model of LBM consists of the three phases 'Heating', 'Cooling' and 'End Cooling'. For the very first phase of 'Heating', the initial condition

$$T^0(\boldsymbol{r}) = T_{\mathrm{pre}}, \qquad \forall \boldsymbol{r} \in \Omega_{\mathrm{bp}} \tag{2.75}$$

applies. In the subsequent phases, the initial condition $T^0(\boldsymbol{r})$ is the temperature distribution at the end of the previous observation period. Isolating Neumann boundary conditions apply to boundaries not listed below. The run variable

i represents the outer loop over all layers. The inner loop over all scan fields of the respective layer is carried out with the run variable j. This results in the following PDEs with initial and boundary conditions for the respective phases.

Heating

$$\rho c(T)\,\frac{\partial T}{\partial t} = \boldsymbol{\nabla}\cdot[\lambda(T)\,\boldsymbol{\nabla} T], \qquad \forall \boldsymbol{r}\in\Psi_{i,j}, t\in\left[0,\eta\,\Delta t_{\text{heating}}\right] \tag{2.76}$$

$$T(\boldsymbol{r},0) = T^0(\boldsymbol{r}), \qquad \forall \boldsymbol{r}\in\Psi_{i,j-1} \tag{2.77}$$

$$T(\boldsymbol{r},t) = T_{\text{sol}}, \qquad \forall \boldsymbol{r}\in\Omega_{i,j}, t\in\left[0,\eta\,\Delta t_{\text{heating}}\right] \tag{2.78}$$

$$T(\boldsymbol{r},t) = T_{\text{pre}}, \qquad \forall \boldsymbol{r}\in\Gamma_{\text{pre}}, t\in\left[0,\eta\,\Delta t_{\text{heating}}\right] \tag{2.79}$$

$$-\lambda(T)\,\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r},t) = \alpha_{\text{top}}\left[T(\boldsymbol{r},t)-T_{\text{ref}}\right], \qquad \forall \boldsymbol{r}\in\Gamma_{i,j-1}, t\in\left[0,\eta\,\Delta t_{\text{heating}}\right] \tag{2.80}$$

$$-\lambda(T)\,\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r},t) = \alpha_{\text{side}}\left[T(\boldsymbol{r},t)-T_{\text{ref}}\right], \qquad \forall \boldsymbol{r}\in\Gamma_{\text{side}}, t\in\left[0,\eta\,\Delta t_{\text{heating}}\right] \tag{2.81}$$

If all scan fields of a layer are activated, the part cools down over the side surfaces and the build surface, as well as the boundary condition of the preheating. Expressed in equations, this corresponds to the following.

Cooling

$$\rho c(T)\,\frac{\partial T}{\partial t} = \boldsymbol{\nabla}\cdot[\lambda(T)\,\boldsymbol{\nabla} T], \qquad \forall \boldsymbol{r}\in\Psi_{i}, t\in[0,\eta\,\Delta t_{\text{dwell}}] \tag{2.82}$$

$$T(\boldsymbol{r},0) = T^0(\boldsymbol{r}), \qquad \forall \boldsymbol{r}\in\Psi_{i} \tag{2.83}$$

$$T(\boldsymbol{r},t) = T_{\text{pre}}, \qquad \forall \boldsymbol{r}\in\Gamma_{\text{pre}}, t\in[0,\eta\,\Delta t_{\text{dwell}}] \tag{2.84}$$

$$-\lambda(T)\,\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r},t) = \alpha_{\text{top}}\left[T(\boldsymbol{r},t)-T_{\text{ref}}\right], \qquad \forall \boldsymbol{r}\in\Gamma_{i}, t\in[0,\eta\,\Delta t_{\text{dwell}}] \tag{2.85}$$

$$-\lambda(T)\,\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r},t) = \alpha_{\text{side}}\left[T(\boldsymbol{r},t)-T_{\text{ref}}\right], \qquad \forall \boldsymbol{r}\in\Gamma_{\text{side}}, t\in[0,\eta\,\Delta t_{\text{dwell}}] \tag{2.86}$$

Once all layers have been created, the part cools down to room temperature.

End Cooling

$$\rho c(T)\,\frac{\partial T}{\partial t} = \boldsymbol{\nabla}\cdot[\lambda(T)\,\boldsymbol{\nabla} T], \qquad \forall \boldsymbol{r}\in\Psi_{i}, t\in[0,\eta\,\Delta t_{\text{end}}] \tag{2.87}$$

$$T(\boldsymbol{r},0) = T^0(\boldsymbol{r}), \qquad \forall \boldsymbol{r}\in\Psi_{i} \tag{2.88}$$

$$-\lambda(T)\,\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r},t) = \alpha_{\text{top}}\left[T(\boldsymbol{r},t)-T_{\text{ref}}\right], \qquad \forall \boldsymbol{r}\in\Gamma_{i}, t\in[0,\eta\,\Delta t_{\text{end}}] \tag{2.89}$$

$$-\lambda(T)\,\frac{\partial T}{\partial \boldsymbol{n}}(\boldsymbol{r},t) = \alpha_{\text{side}}\left[T(\boldsymbol{r},t)-T_{\text{ref}}\right], \qquad \forall \boldsymbol{r}\in\Gamma_{\text{side}}, t\in[0,\eta\,\Delta t_{\text{end}}] \tag{2.90}$$

Concrete values for the parameters of the Mathematical Model in the case of Inconel 718 and an EOS M270 are provided in Figure 2.7.

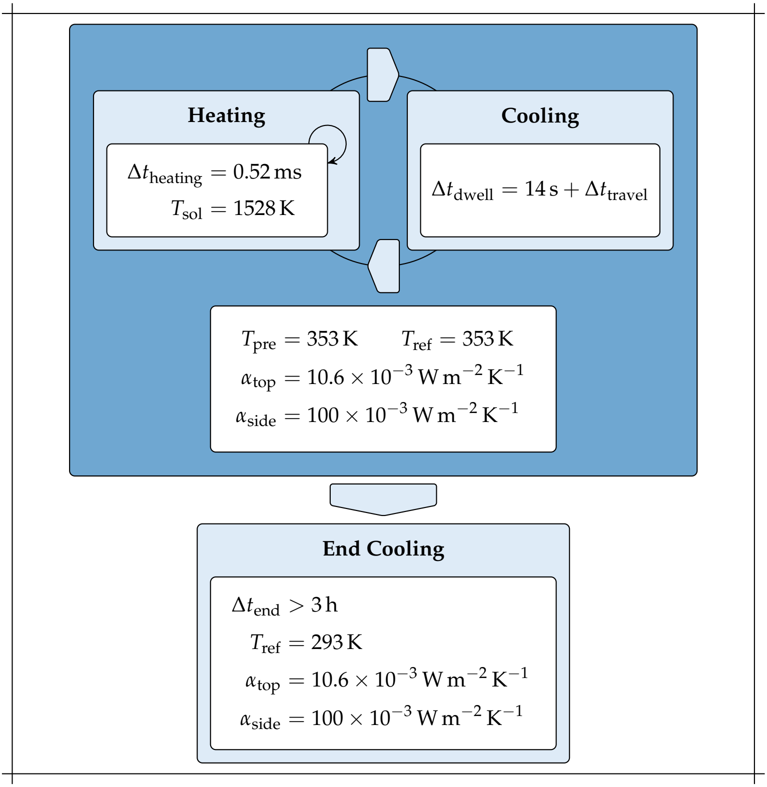

Heating
$\Delta t_{\text{heating}} = 0.52\,\text{ms}$
$T_{\text{sol}} = 1528\,\text{K}$
Cooling
$\Delta t_{\text{dwell}} = 14\,\text{s} + \Delta t_{\text{travel}}$
$T_{\text{pre}} = 353\,\text{K}$
$T_{\text{ref}} = 353\,\text{K}$
$\alpha_{\text{top}} = 10.6 \times 10^{-3}\,\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$
$\alpha_{\text{side}} = 100 \times 10^{-3}\,\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$
End Cooling
$\Delta t_{\text{end}} > 3\,\text{h}$
$T_{\text{ref}} = 293\,\text{K}$
$\alpha_{\text{top}} = 10.6 \times 10^{-3}\,\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$
$\alpha_{\text{side}} = 100 \times 10^{-3}\,\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$

Figure 2.7: Parameters for the Mathematical Model in the case of Inconel 718 and an EOS M270 (BRANNER 2011; POTTLACHER et al. 2002; SEIDEL 2016).

3 State of the Art in Discretization and Algorithm Selection

3.1 Overview

The purpose of this chapter is to summarize the topic-relevant findings from science regarding numerical methods for accelerating the simulation of LBM, as well as to present computation-efficient methods from the general field of numerics. The requirements from the Mathematical Model for the selection of Discretization and Algorithm Selection are determined, and, in this respect, the state of the art in the field of LBM simulation is presented. Subsequently, fields of action that will arise for the spatial and temporal discretization, as well as the solving of the system of equations when using a more detailed Conceptual Model are derived. With regard to these, computationally efficient methods from the general field of numerics are discussed.

3.2 Discretization and Algorithm Selection in the Simulation of Additive Manufacturing

The goal of determining the temperature field for the entire part during the LBM buildup process is pursued. With regard to the state of the art, however, the progress made in shortening the computing time on smaller scales is also investigated to gain insight from this. The same applies to the different process classes of AM. They are generally considered, since the different process variants have several characteristics in common. Methods are considered for the various fields displayed in Figure 2.3 with regard to short calculation times. The specifications of particular values always refer to the material Inconel 718 and the AM machine EOS M270.

Many activities in the field of AM simulation aim at reducing the solution effort. It is apparent from the reviews of BIKAS et al. (2016), LUO & ZHAO (2018), SCHMIDT et al. (2017), SCHOINOCHORITIS et al. (2017), URIONDO et al. (2015), Z. YAN et al. (2018), and ZENG et al. (2012) that the Conceptual Model is largely adapted to decrease computing time. This is partly due to the fact that many models are implemented in commercial FEM programs (KROL et al. 2012; KUNDAKCIOGLU et al. 2016; PAPADAKIS, LOIZOU, RISSE, BREMEN, et al. 2014; SEIDEL et al. 2014; STATHATOS & VOSNIAKOS 2018). The list in Z. YAN et al. (2018) of further simulation models for AM and the platforms used illustrate this, in which only 6 of the 35 listed are based on their own code. In addition, these are assigned to only four different projects. Due to the closed source code, the numerical methods are therefore only modifiable within the framework of the options provided by the software and usually only include established adaptations. In most cases, an adaptation of the numerical methods beyond that is not possible or is at least cumbersome to achieve, which is therefore inefficient.

3.2.1 Element shape

With the procedure for spatial discretization in Subsection 2.5.2, only the FEM has been introduced, as this is the most suitable method for LBM (Zeng et al. 2012). For the shape of the elements, there is a choice between tetrahedron and hexahedron. Hexahedrons provide qualitatively better solutions for larger elements, but meshing of a complex geometry is very costly or impossible at all (Knupp 2001). Although there are specific approaches for LBM to generate hexahedral meshes (Neugebauer et al. 2014), these may lead to highly distorted elements in the boundary areas. Therefore, tetrahedrons are used as elements, because especially complex geometries demonstrate the potential of AM compared to conventional manufacturing methods.

Stathatos & Vosniakos (2018) have achieved significantly shorter calculation times by replacing the solid elements with shell elements to reduce the number of DoFs. However, the experiments were only performed on a simplified geometry and on one laser trajectory. The extension to the calculation of complete components is limited by the creation of meshes from shell elements for complex geometries. Nevertheless, Krol et al. (2011) uses shell elements in the area of support structures and thus for complex geometries. However, support structures have a repeating structure, allowing the mesh generation to be reduced to a representative volume element.

3.2.2 Element size, order of shape functions and numerical integrator

Steep temporal and spatial gradients appear in LBM due to the high energy input on the build surface. If the choice of the element size is too large in relation to the time step size, spatial oscillations occur in the numeric solution as a result. The FEM attempts to display the energy of a system correctly. If the elements affected by the energy input are not able to correctly represent the introduced energy, the difference is balanced by adjacent elements. Therefore, the selection of the element size depends on the time step size, and no isolated evaluation is possible. According to Hogge & Gerrekens (1983), a discretization of the penetration depth with five elements is necessary to avoid oscillations. This results in the Penetration Depth Condition (PDC):

$$d_\Theta < d_{\mathrm{PD}} := \frac{\delta(\Delta t^0)}{5}. \tag{3.1}$$

Simplified, the phase 'Heating' corresponds to heating a domain with the same initial temperature over a constant temperature boundary condition. Since finite domains are allowed to be considered infinite for very small times, the analytical solution from Equation 2.21 is used to derive requirements for 'Heating'. The time $\Delta t_{\mathrm{heating}} = 0.52\,\mathrm{ms}$ in conjunction with the characteristic factor η is an upper limit for the first time step, and the layer height d_{LC} is an upper limit for the element size d_Θ (it is assumed that the resolution of the heat source in the xy-plane is not smaller than the layer height). A value of $3.48 \times 10^{-6}\,\mathrm{m}^2\,\mathrm{s}^{-1}$ is assumed for the mean thermal diffusivity a of Inconel 718 in the LBM process. Together with the penetration depth from Equation 2.33 and Equation 3.1, the additional requirement

$$d_\Theta < d_{\mathrm{PD}} = 1.55 d_{\mathrm{LC}} \tag{3.2}$$

results for the element size d_Θ. Although the PDC is not violated by the element size d_{LC} resulting from the layer height, it is nevertheless in the order of magnitude for the occurrence of oscillations. In addition, Equation 3.2 assumes that only one time step is used for the temporal discretization. The selection of numerical methods is therefore carried out with the aim of obtaining a procedure that is as robust as possible. For this reason, linear shape functions are used, as well as the backward Euler method as the numerical integrator. Linear shape functions are less sensitive to oscillation and their solutions are more accurate than higher order shape functions in that case (DAMJANIĆ & OWEN 1982). The backward Euler method is chosen in the further considerations of this thesis as more robust solutions are expected from it due to its L-stability (cf. 2.4.1 and (DEUFLHARD & WEISER 2012)).

This choice of shape functions and of the numerical integrator is also made for the two further phases 'Cooling' and 'End Cooling'. High gradients are still present at the beginning of the phase 'Cooling', which is why it is also sensitive to the described oscillations. In addition, the largest possible time steps are used with regard to the computing time if only small gradients are left towards the end of the cycle. However, time steps that are too large lead to temporal oscillations with some integrators (WOOD & LEWIS 1975; ZIENKIEWICZ & TAYLOR 2002). The backward Euler is also robust against these kinds of oscillations due to its stability (DAMJANIĆ & OWEN 1982; DEUFLHARD & WEISER 2012). It is also applied to the phase 'End Cooling' for the same reason. The linear shape functions are used for this phase due to the associated lower DoFs.

The FEM requires fine meshing in the area of steep gradients. To reduce the DoFs at least in the range distant from the heat source, a widespread approach is the use of refinement and coarsening strategies. Since mesh adaptations are not trivial while maintaining conformity, nonconformal meshes (cf. Subsection 2.5.2) are typically used, leading to hanging nodes (KOLOSSOV et al. 2004; MARTUKANITZ et al. 2014; RIEDLBAUER et al. 2014). To maintain conformity, PATIL et al. (2015) and PAL et al. (2016) developed a special refinement strategy, which, in combination with a structured mesh, leads to conformal meshes in arbitrary areas. The mentioned publications have in common the use of structured meshes and hexahedra as elements. This meshing is often not possible for complex geometries and in case of high demands on the geometric accuracy of the mesh, as is the case in aerospace applications.

Y. ZHANG et al. (2018) used a mesh refinement with tetrahedra and a conformal mesh at any time, but, in the example, only a simple cuboid geometry was considered. In addition, two further approaches to reducing the computational effort with a coarser mesh while violating the PDC were considered in the paper. On the one hand, the scaling of the observation period, as well as the time steps until the PDC was fulfilled, were considered. On the other hand, the *Asynchronous Method* from JAOUEN (1998) was applied. The former avoided the occurrence of oscillations, but influenced the physics of the system by extending the observation period, which was confirmed by the changed temperature profiles in the presented results. The fact that there were no temperature differences at the end of the dwell time is attributed to the fact that the problem under consideration generally has no gradients at the end of the heating of a layer. To use the self-similarity of solutions of the heat equation, the characteristic in Equation 2.30 must be fulfilled as it is done for scaling the time with the characteristic factor η (cf. Equation 2.74) concerning the LCs. The Asynchronous Method first uses a larger time step that does not violate the PDC. The results are then linearly interpolated to the actual smaller time step. While the method ensures that the nodal values have no oscillations, it does not avoid the problem that the shape functions of the FEM are not able to represent the gradients. Especially with regard to the use of the temperature field in a subsequent calculation of the residual stresses, however, these gradients are decisive. The method also assumes a linear relationship between temperature and time. This, however, is nonlinear, which is particularly evident from the analytical solutions. This issue has not been addressed so far.

Alternative approaches exist to overcome the shortcomings of the FEM. Suggested by PAL et al. (2014), there is the Spectral Finite Element Method (SFEM), which uses exponential functions instead of polynomial functions. An implementation has not taken place thus far, however. CARRATURO et al. (2019) used more powerful shape functions, more precisely the method Isogeometric Analysis (IGA) with truncated hierarchical B-splines, which achieved good results, but so far only for a simplified two-dimensional geometry. KOLLMANNSBERGER et al. (2018) used the Finite Cell Method (FCM) of PARVIZIAN et al. (2007) to calculate the temperature field in LBM. It took advantage of high-order finite elements and combined them with a fictitious domain approach. The geometry was meshed beyond the boundary with a voxel mesh. The consideration of the real dimensions took place via the integration points. Thus, the meshing of highly complex and filigree geometries is possible (JOULAIAN et al. 2014). Due to the high order of the shape functions, relatively coarse meshes were used. For the application of boundary conditions, however, additional requirements must be fulfilled or additional projection methods are necessary (RUESS et al. 2012). Furthermore, the implementation is challenging (ZANDER et al. 2014). PENG et al. (2018) achieved a faster calculation by a factor of two using a thermal circuit network instead of the FEM. However, this was accompanied by deviations of up to 15 %.

3.2.3 Time steps

Due to the short observation period for 'Heating', the maximal temporal gradient in this phase is assumed to be constant, and equidistant time steps are therefore used, since uniform requirements subsequently apply to them. During the 'Cooling' phase, the input energy is distributed over the part and the thermal gradients decreases with time. Accordingly, the time steps become longer with increasing time. Adaptive methods require, apart from the calculation of the desired values, the additional determination of the current solution quality and therefore initially increase the solution effort. In the case of 'Cooling', the basic behavior of the temperature field is known to be associated with a decreasing, directed heat flux from the build surface to the build plate. To reduce effort, the time steps are therefore determined in advance.

According to DEUFLHARD & BORNEMANN (2002), optimal time steps require an equal distribution of the local truncation error, which results from the deviation of the numerical solution $T^m(\boldsymbol{r})$ compared to the exact solution $T(t^m, \boldsymbol{r})$ for all time steps t^m (cf. Equation 2.40). Based on this, a procedure for optimal time step determination for the simulation of LBM is presented.

The first-order Taylor[21] polynomial in time of the exact solution T at t^m and its remainder term in the Lagrange form are given by

$$T\left(t^{m-1}, \boldsymbol{r}\right) = T(t^m, \boldsymbol{r}) - \Delta t^m \frac{\partial T}{\partial t}(t^m, \boldsymbol{r}) + \frac{(\Delta t^m)^2}{2} \frac{\partial^2 T}{\partial t^2}(\xi, \boldsymbol{r}) \tag{3.3}$$

[21] **Brook Taylor** (* August 18th, 1685 - † December 29th, 1731), English mathematician.

for some real number ξ between t^{m-1} and t^m (LITTLE et al. 2015). Along with the backward Euler method

$$T^m(\boldsymbol{r}) = T^{m-1}(\boldsymbol{r}) + \Delta t^m \frac{\partial T}{\partial t}(t^m, \boldsymbol{r}) \tag{3.4}$$

and the assumption $T\left(t^{m-1}, \boldsymbol{r}\right) = T^{m-1}(\boldsymbol{r})$, the following equations holds for the local truncation error τ^m:

$$\begin{align}
\tau^m &= \|T(t^m, \boldsymbol{r}) - T^m(\boldsymbol{r})\| \tag{3.5}\\
&= \left\| \left[T\left(t^{m-1}, \boldsymbol{r}\right) + \Delta t^m \frac{\partial T}{\partial t}(t^m, \boldsymbol{r}) - \frac{(\Delta t^m)^2}{2} \frac{\partial^2 T}{\partial t^2}(\xi, \boldsymbol{r}) \right] \right. \nonumber\\
&\qquad \left. - \left[T^{m-1}(\boldsymbol{r}) + \Delta t^m \frac{\partial T}{\partial t}(t^m, \boldsymbol{r}) \right] \right\| \tag{3.6}\\
&= \frac{(\Delta t^m)^2}{2} \left\| \frac{\partial^2 T}{\partial t^2}(\xi, \boldsymbol{r}) \right\|. \tag{3.7}
\end{align}$$

This indicates that, in the case of the Euler method, the time step size depends on the second temporal derivation of the solution T.

In simplified form, the model of LBM corresponds to the iterative deposition of energy in a layer and the subsequent propagation in direction of the build plate. Neglecting 'Heating', the analytical solution T from Equation 2.25 approximately represents the phase 'Cooling' in z-direction of the local part coordinate system with $d_{\text{ed}} = d_{\text{LC}}$. Its applicability for LBM has already been demonstrated in KRAUSS (2016). However, since the truncation error τ^m still depends on the parameter ξ, the determination of optimal time steps is difficult despite the assumption of a known solution T. Therefore, the local uncertainty κ^m is introduced by the estimation

$$\begin{align}
\tau^m &= \frac{(\Delta t^m)^2}{2} \left\| \frac{\partial^2 T}{\partial t^2}(\xi, z) \right\| \tag{3.8}\\
&\leq \frac{(\Delta t^m)^2}{2} \sup \left\{ \left| \frac{\partial^2 T}{\partial t^2}(t, z) \right| : t \in \left[t^{m-1}, t^m\right], z \in [0, \infty) \right\} \tag{3.9}\\
&= \frac{(\Delta t^m)^2}{2} \left\| \frac{\partial^2 T}{\partial t^2} \right\|_\infty =: \kappa^m. \tag{3.10}
\end{align}$$

The local uncertainty κ^m is thus an upper bound of the local truncation error τ^m.

The principle of the equally distributed local truncation errors for optimal time steps is now transferred to the local uncertainties. In time ranges with large values in the second temporal derivation, smaller time steps must be selected and accordingly larger time steps must be used for smaller values. For a given number m_{Υ}, corresponding time steps $\Upsilon^{m} = \{t^0, t^1, \ldots, t^{m_{\Upsilon}}\}$ with minimum standard deviation σ_{κ} are determined in an optimization[22]:

$$\sigma_{\kappa} = \sqrt{\frac{1}{m_{\Upsilon} - 2} \sum_{s=2}^{m_{\Upsilon}} \left(\kappa^{s} - \mu_{\kappa}\right)^{2}}, \qquad \mu_{\kappa} = \frac{1}{m_{\Upsilon} - 1} \sum_{s=2}^{m_{\Upsilon}} \kappa^{s}. \tag{3.11}$$

The local uncertainty κ^1 is excluded, as the solution of Equation 2.25 has a singularity at $t = 0$. This would result in an infinitely small first time step. However, for the first time step the requirement from the PDC additionally applies. To obtain universally valid time steps, an element size corresponding to the layer height is assumed. Therefore, the first time step is preset with 5×10^{-4} s, whereby the PDC is fulfilled for a thermal diffusivity of $a = 3.48 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$ under consideration of an additional tolerance.

To transfer the results to the 'Cooling' phase with a different number of time steps, the obtained points in time t^m are normalized to the unit interval. These normalized values are then used to fit a function that maps equidistant time points in the unit interval to normalized optimal time points in the sense of equally distributed local uncertainties. In the case of 64 time steps, and the remaining dimensions of the reference layer height of 20 µm and a dwell time of 14 s, the function[23]

$$f_{t^m}(t) = 10^{\hat{f}_{t^m}(t)} \tag{3.12}$$

$$\begin{aligned} \hat{f}_{t^m}(t) = {} & 661.8t^9 - 3210t^8 + 6716t^7 - 7917t^6 + 5785t^5 \\ & - 2712t^4 + 833.6t^3 - 164.3t^2 + 22.69t - 6.504 \end{aligned} \tag{3.13}$$

results in a standard fit error of less than 2.7×10^{-2}. The model function is based on Newton's[24] law of cooling for fluids, which assumes exponential cooling (INCROPERA et al. 2007). In order to take the circumstance of a solid into account, the exponentiation is carried out with a polynomial of ninth order.

The global uncertainty κ of a time step distribution is defined as the maximum of the local uncertainties:

$$\kappa = \max_{2 \le m \le m_{\Upsilon}} \{\kappa^{m}\}. \tag{3.14}$$

[22] ZELLER & ZAEH (2019), run_time_step_optimization.m

[23] ZELLER & ZAEH (2019), fit_time_steps.m

[24] **Sir Isaac Newton** (* December 25th, 1642 - † March 20th, 1726), English mathematician, physicist and astronomer.

With the global uncertainty of several optimal time step distributions with different values m_{Y}, a correlation of both values κ and m_{Y} is established with regression analysis[25]. The fitted function

$$f_{m_{\mathrm{Y}}}(\kappa) = 64.9 \exp\Big(-1.171 \log \frac{\kappa}{\mathrm{K}}\Big) + 10.51 \exp\Big(-0.199 \log \frac{\kappa}{\mathrm{K}}\Big) \tag{3.15}$$

with a standard fit error of less than 1.05×10^{-2} describes the minimum number of time steps required for a desired global uncertainty. Results for both Equation 3.12 and Equation 3.15 are illustrated in Figure 3.1. In addition, the spatial location of the determining value for the supremum norm of Equation 3.10 obtained in the optimization is given (cf. Figure 3.1b). It follows from this that the requirement for the time step size always results from the first layer.

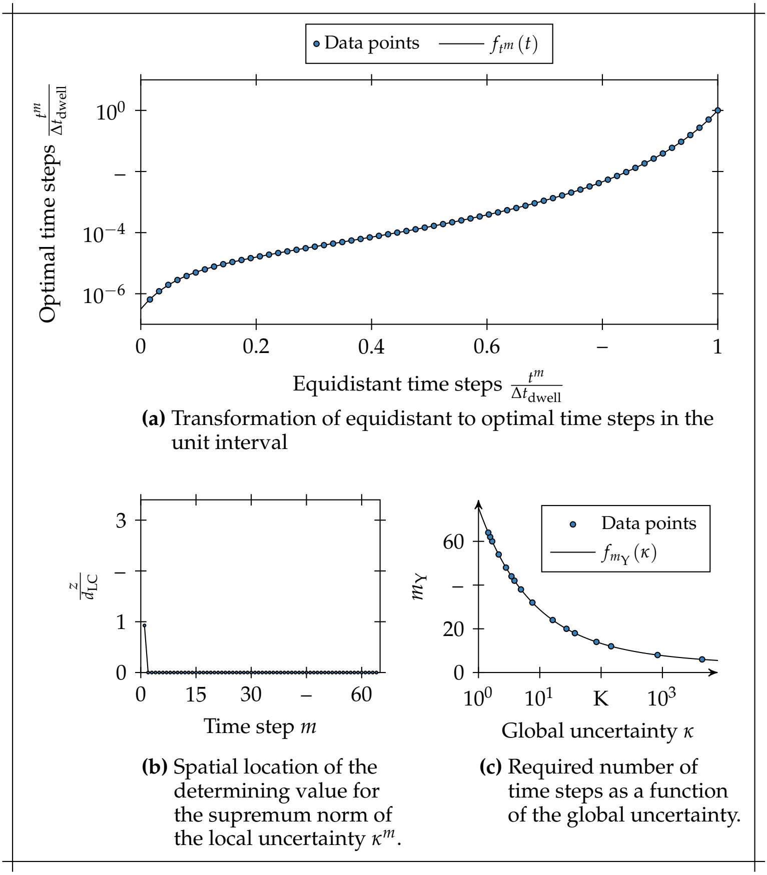
A special adaptive time step control or pre-determination of the time steps is not necessary for the 'End Cooling' phase. With the cooling of an approximately uniformly heated body from preheating temperature to room temperature, this phase has no special features or characteristics compared to general thermal analyses.

3.2.4 Linearization strategy and type of the solver

The procedure presented in Subsection 2.5.4 with iterative updating of the material parameters is used to linearize the system of nonlinear equations. Differences in the material parameters due to temperature dependence are small for metals (READY 1965); therefore, a fast convergence is to be expected. However, iterative methods possess only local convergence. In the context of industrial application, Direct Methods are used to solve the system of linear equations to always obtain a solution. Especially with regard to complex geometries, it is not guaranteed that the solution of the previous time step as a starting value for the next step is sufficiently close to the solution.

The implementation of the buildup process in AM takes place in the simulation through the successive addition of elements. This is accompanied by the assignment of the node numbering and the assembly of the matrices for the system of equations. To decrease the effort of renumbering, MICHALERIS (2014) developed a *hybrid quiet inactive technique*. Elements are added layer by layer into the system of equations, similar to the powder application of a new layer. However, as long as the area of an element is not yet solidified, reduced material parameters are assigned to it to avoid the element influencing the result. Renumbering is therefore only necessary after each layer, but the total number of elements is not reduced.

[25] ZELLER & ZAEH (2019), fit_error.m



(a) Transformation of equidistant to optimal time steps in the unit interval

(b) Spatial location of the determining value for the supremum norm of the local uncertainty κ^m.

(c) Required number of time steps as a function of the global uncertainty.

Figure 3.1: Results for the regression analyses obtained with $a = 3.48 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$, $\Delta t_{\mathrm{dwell}} = 14\,\mathrm{s}$ and $t^0 = 5 \times 10^{-4}\,\mathrm{s}$.

NIKOUKAR et al. (2013) discovered that the layered mesh results in a special structure in the coefficient matrix. As a result of the numbering in layers, a band structure is created. Nodes of a common layer form a block around the main diagonal. The coupling with adjacent layers is carried out on secondary diagonals. They additionally recognized that many elements in the matrix are very small, and their contribution to the solution is negligible. With the introduction of a threshold, below which entries are not considered, the memory requirements and the number of flops required are simultaneously reduced. Neglecting values smaller than 1×10^{-4} results in approximately 100 times fewer flops and 93.7 % fewer non-zero entries, with a simultaneous error of only 1×10^{-5}.

PAL et al. (2014) used a technique that is otherwise mainly used for structural vibrations. The number of eigenmodes is determined using the cosine and sine functions to predict the propagation of the macroscopic thermal response in the direction normal to the build surface. In the immediate vicinity of the heat source, however, the procedure is limited because too many modes are required to define the temperature field due to nonlinear effects.

To avoid complete recalculation of the inverse of the coefficient matrix due to the small changes between two time steps, PAL et al. (2015) used the *Sherman Morrison Woodburry*[26] *algorithm*. This is an explicit representation of the new inverse based on the inverse of the previous time step. If the previous inverse is already known, complete recalculation is avoided.

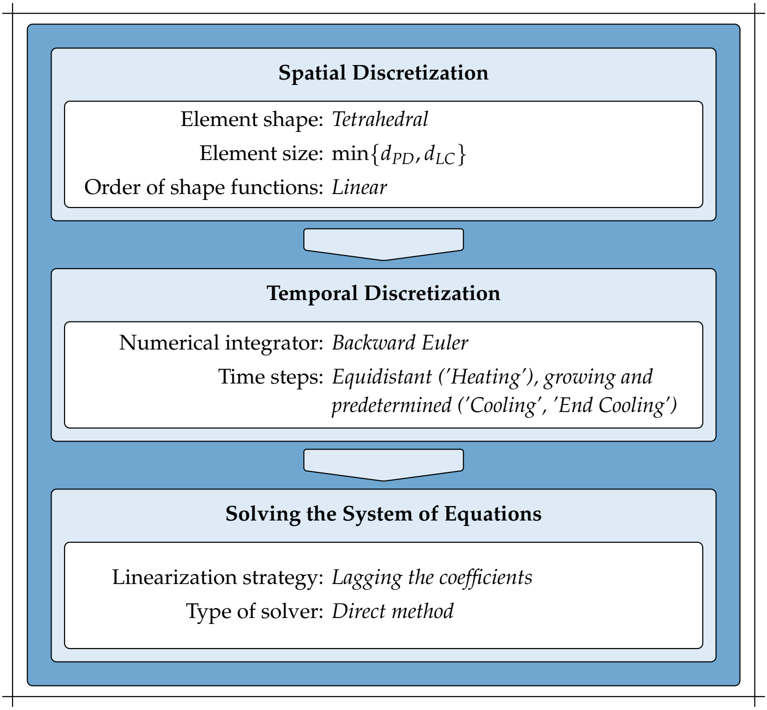
3.3 Interim Conclusion and Need for Action in the Field of Numerical Methods

A summary of the selected standard numerical methods and settings for LBM based on the requirements presented is provided in Figure 3.2. The influence of the degrees of abstraction of the Conceptual Model on the various numerical steps is discussed, and the fields of action are derived accordingly. Better results are achieved through a more detailed representation of the heat source and the combination of fewer layers into LCs (SEIDEL & ZAEH 2018). Both assumptions concern the spatial level of detail. In addition, there is further improvement over the temporal refinement.

The phases 'Heating' and 'Cooling' occur repeatedly in two nested loops. The number of inner loops is influenced by the abstraction of the heat source, and the number of outer loops consisting of 'Heating' and 'Cooling' depends on the LCs (cf. Figure 2.5). Not affected by the modeling assumptions is the phase 'End Cooling', which also occurs only once. It is therefore not considered further due to its weaker influence on the computing time as compared to the other two phases.

The spatial discretization, and thus the number of DoFs, is directly influenced by the combination of layers into an LC (cf. Figure 3.2). Due to the characteristic factor, the layer height is sufficient as element size for the considered material Inconel 718 in the case of one time step for 'Heating' independent of the number of combined layers (cf. Equation 3.2). However, the use of a more precise

[26] **Max Atkin Woodbury** (* April 30th, 1917 - † January 30th, 2010), American mathematician.



Spatial Discretization

Element shape: *Tetrahedral*

Element size: $\min\{d_{PD}, d_{LC}\}$

Order of shape functions: *Linear*

Temporal Discretization

Numerical integrator: *Backward Euler*

Time steps: *Equidistant ('Heating'), growing and predetermined ('Cooling', 'End Cooling')*

Solving the System of Equations

Linearization strategy: *Lagging the coefficients*

Type of solver: *Direct method*

Figure 3.2: Settings in Discretization and Algorithm Selection for M&S of LBM.

temporal discretization also ensures that a finer spatial discretization is required due to the PDC. To avoid increasing the effort despite more precise modeling, a method is therefore required which allows the temporal mapping of high gradients during 'Heating' and, at the same time, allows violation of the PDC.

With the presented procedure to determine the time step size in the 'Cooling' phase, a possibility is already available which, on the one hand, specifies the necessary number of steps for a desired global uncertainty and, on the other hand, provides the optimal distribution of the time step sizes. Thus, the same accuracy is achieved for each time step. However, it has already become apparent that there are significantly different local requirements and that the step size is determined primarily by the requirements of the first layer (cf. Figure 3.1b). Layers further away from the build surface impose fewer requirements on the step size. Therefore, a method is needed which takes into account the local requirements for the time step sizes during 'Cooling'.

Although the abstraction of the heat source is a spatial simplification, since it was assumed that the resolution of the heat source in the xy-plane is not smaller than the layer height, the decreasing degree of abstraction of this modeling assumption only leads to more solver calls from a numerical point of view. By the introduction of the characteristic factor, the combination of fewer layers also manifests itself in more solver calls. While the number of 'Cooling' cycles is only affected by the LCs, the number of 'Heating' phases is influenced by both the heat source and the LCs. Accordingly, due to the detailing of the modeling assumptions, there is a larger increase in the number of solver calls by the 'Heating' phase. This thesis therefore deals with the solution of the system of equations of 'Heating'. Figure 3.2 indicates that Direct Methods are used as solvers. However, the advantage of this class, the reusability of the decomposition of the coefficient matrix, is not exploited due to the non-linearity caused by the temperature-dependent material parameters and the enlargement of the calculation mesh by continuous activation of further scan fields. Although these changes in the material parameters are only small in the case of metal (READY 1965), the coefficient matrix must be decomposed again for each time step. As described in Subsection 2.4.2, decomposition takes much effort in solving the system of equations. Therefore, a procedure is required which enables the reusability of the solution and exploits the full potential of the Direct Methods.

3.4 Computationally Efficient Numerical Methods

Numerical methods from the general field of numerics, which address the fields of action identified in Subsection 3.3, are introduced in this Section. In the case of spatial discretization, this is the mapping of steep gradients despite coarse meshing. In temporal discretization, it is necessary to consider the different local requirements for the time step size. The solving of the system of equations does not yet consider, to the full extent, the advantages of Direct Methods.

3.4.1 Coarse spatial discretization for short-term problems

There are various methods and approaches that result in a physical solution despite the use of a coarse mesh related to the observation period. A distinction is made between methods which merely stabilize the solution and eliminate the oscillations, and methods which aim to map the actual gradient. In what follows, both are presented. However, this Subsection is limited to methods that are strongly based on the FEM, as this has already been identified as the most suitable approach (ZENG et al. 2012). In addition, there is already a large selection of programs, not only for the actual solution of a problem but also for pre-processing and post-processing.

From an algebraic standpoint, the non-physical solutions occur when the coefficient matrix $\boldsymbol{A}^m$ in Equation 2.71, i. e.

$$\boldsymbol{A}^m = \left[\frac{1}{\theta\, \Delta t^m} \boldsymbol{C}\big(T^m_{i-1}\big) + \boldsymbol{K}\big(T^m_{i-1}\big) + \boldsymbol{K}_{\mathrm{C}}\big(T^m_{i-1}\big) \right], \tag{3.16}$$

is not an M-matrix (ORTEGA & RHEINBOLDT 1970). This property requires that $\boldsymbol{A}^m$ is non-singular and that

$$A^m_{kk} > 0, \qquad A^m_{kl} \leq 0, \qquad \forall k \neq l, 1 \leq k, l \leq n_\Xi \tag{3.17}$$

applies. The steady-state heat equation is used to substantiate why this avoids non-physical solutions. In that case, the capacitance matrix $\boldsymbol{C}$ disappears, and only the conductance matrix $\boldsymbol{K}$ is present. The entry K_{kl} on the off-diagonal (i. e. $k \neq l$) results from the integral of the two derivatives of the shape functions corresponding to the nodes $\boldsymbol{r}_k$ and $\boldsymbol{r}_l$ (cf. Equation 2.65b). If this value is positive, both derivatives have the same sign. Accordingly, only an heat flux in one direction is representable between the two nodes. This constellation cannot occur in one dimension but in multidimensional elements with small internal angles. The *Delaunay*[27] *triangulation* provides the best possible mesh in this respect, as it maximizes the smallest internal angle of all elements, but it does not ensure that the conductance matrix is an M-matrix (LETNIOWSKI 1992).

If both the capacitance matrix and the conductance matrix fulfill the property of an M-matrix, it follows for the coefficient matrix as well. The two matrices are therefore converted separately. A consequence is that the procedure is directly independent of the time step used. The capacitance matrix is *lumped*, i. e. the element masses are concentrated in the nodes, which leads to the fact that only entries on the main diagonal arise. For the conductance matrix, PUTTI & CORDES (1998) developed the *Orthogonal Subdomain Collocation*, which generates the M-matrix form under the assumption of a mesh with Delaunay triangulation. This requirement for such a triangulation is not necessarily fulfilled and results in general mesh generators being excluded from use (FACHINOTTI & BELLET 2006).

FACHINOTTI & BELLET (2006) developed the *Diffusion-Split Method*. Similar to the conductance matrix $\boldsymbol{K}$, the matrix $\boldsymbol{K}^*$ is introduced, but with an augmented conductivity λ^*, which is oriented to the step size Δt^m corresponding to

$$\lambda^* = \begin{cases} \lambda & \text{if } \Delta t_{\mathrm{PD}} \leq \Delta t^m, \\ \lambda \frac{\Delta t_{\mathrm{PD}}}{\Delta t^m} & \text{if } \Delta t_{\mathrm{PD}} > \Delta t^m. \end{cases} \tag{3.18}$$

[27] **Boris Nikolaevich Delaunay** (* March 15th, 1890 - † July 17th, 1980), Soviet mathematician.

The value Δt_{PD} represents the increment that satisfies the PDC (Equation 3.1) with equality. If the PDC is violated, the augmented conductivity is scaled accordingly to fulfill it. Since the use of the matrix $\boldsymbol{K}^{*}$ as a conductance matrix does not lead to any oscillations as a consequence, it is utilized in the Diffusion-Split Method for setting up the coefficient matrix. However, to take account of the actual conductivity λ, the difference of the matrices is considered with respect to the temperature from the previous time step $m-1$. In the case of the linear heat equation without a convection boundary condition, the system of equations

$$\left(\frac{1}{\Delta t^{m}}\boldsymbol{C}+\boldsymbol{K}^{*}\right)\cdot\boldsymbol{T}^{m}=\left(\frac{1}{\Delta t^{m}}\boldsymbol{C}+\boldsymbol{K}^{*}-\boldsymbol{K}\right)\cdot\boldsymbol{T}^{m-1}+\boldsymbol{q}(t^{m})\,. \tag{3.19}$$

is solved in each time step m. The PDC is inconclusive for non-uniform grids due to the different element sizes. In the case of monotonous cooling or heating, it is therefore suggested to perform the determination of Δt_{PD} by iterative adjustment until no nodal value is respectively above or below its value of the preceding time step. Although this usually must only be done in the first time step, and the additional computational effort is therefore negligible, the method is associated with a change in the underlying equation. Furthermore, it does not determine the actual gradient but only suppresses the undershoots.

Based on the *Streamline-Upwind* concept proposed by T. J. R. HUGHES & A. BROOKS (1979) and TABATA (1978), a number of stabilization techniques are available for the convection-diffusion equation, i. e.

$$\rho c(T)\frac{\partial T}{\partial t}=\boldsymbol{\nabla}\cdot[\lambda(T)\boldsymbol{\nabla}T]-\boldsymbol{\nabla}\cdot[\boldsymbol{v}T],\qquad\forall\boldsymbol{r}\in\Omega,t\in\Upsilon \tag{3.20}$$

with velocity $\boldsymbol{v}\in\mathbb{R}^{3}$. Among the best known variations are the *Streamline Upwind/Petrov-Galerkin (SUPC)* (A. N. BROOKS & T. J. HUGHES 1982) and the *Galerkin/Least-Squares (GLS)* (T. J. R. HUGHES et al. 1989). However, they are only suitable to a limited extent to pure diffusion, as they take advantage of the information propagation of the convection term. Diffusion describes the mechanism of concentration balancing and is an undirected global information. Convection, on the other hand, is local information describing the heat flux at a point. Rather, the methods address mainly convection-dominated problems in which the undirected information of diffusion is negligible. The application of the GLS method to the heat equation is identical to the use of a larger time step (HACHEM et al. 2010). Therefore, HACHEM et al. (2010) interpolate the solution additionally linear to the actual time step, similar to the mentioned Asynchronous Method, but with the difference that it takes place directly within the formulation. This is accompanied by a stabilization of the nodal values again, but no knowledge about the temperature field is obtained.

The method introduced by HACHEM et al. (2010) is, without interpolation, equivalent to the GLS, but the derivation is actually based on the enrichment of the function space with a *bubble function*, i. e. a shape function that disappears on the edge of the element and whose DoF lies inside. It thus corrects the inadequacy of the linear shape functions to represent the gradient, even with a coarse mesh. To avoid increasing the calculation effort, the additional DoF is eliminated in the developed procedure and only implicitly considered. If the temperature profile is of explicit interest, further methods arise by not eliminating the additionally introduced DoF. A distinction is made between two types of methods. The first type of methods replaces the functional space with a space that better corresponds to the underlying problem, like the previously mentioned methods SFEM and IGA (cf. Subsection 3.2.2). The second type of methods retains the function space and only replaces or extends it in critical areas. This leads to the eXtended Finite Element Method (XFEM). The method was developed by MOS et al. (1999) with the original intention to calculate crack propagation. It is now also widely used in the simulation of fluids (SCHOTT et al. 2015) and fluid-structure interaction (WALL et al. 2010). In principle, however, it is suitable for any problem whose solution has a singularity or discontinuity and therefore has a wide range of applications (KHOEI 2014). If the shape functions are partially replaced, it is referred to as *intrinsic* XFEM; with additional enrichment, it is called *extrinsic* XFEM. ABBAS et al. (2010) used the method to solve the convection-diffusion equation with steep gradients at the boundary. The elements close to the boundary are intrinsically enriched with functions of the form

$$f_{q,c_{\text{scal}}}(x) = \frac{\exp\Bigl(q\left(1 - \frac{x}{c_{\text{scal}} d_\Theta}\right)\Bigr) - 1}{\exp(q) - 1}. \tag{3.21}$$

The parameter q determines the gradient and c_{scal} scales the spatial variable linearly over the enriched elements from 1 to 0. To map the entire spatial extent of the gradient, four enrichment functions were used, some of which were carried by several elements. The functions used by ABBAS et al. (2010) are illustrated in Figure 3.3. This results in correct nodal values as well as an accurate representation of the gradient. Although the results were obtained for convection-dominated problems, the enrichment does not use any properties of the convection, which is why the exact mapping of the spatial gradient despite coarse meshing is also expected in the application for pure diffusion. However, a preliminary optimization was carried out to obtain the appropriate parameters, and the sensitivity of the solution to the parameters is not provided.

A further method that is also suitable for high gradients is the *Discontinuous Galerkin Method* (KANAPADY et al. 2005; PICHELIN & COUPEZ 1999). It uses functions that are discontinuous on the element boundaries. To map high gradients, shape functions are typically used that are constant over the entire element. This means that element values are determined instead of nodal values. Although the results indicate that these are stable against oscillations due to short time steps, an explicit representation of the gradient is not possible.

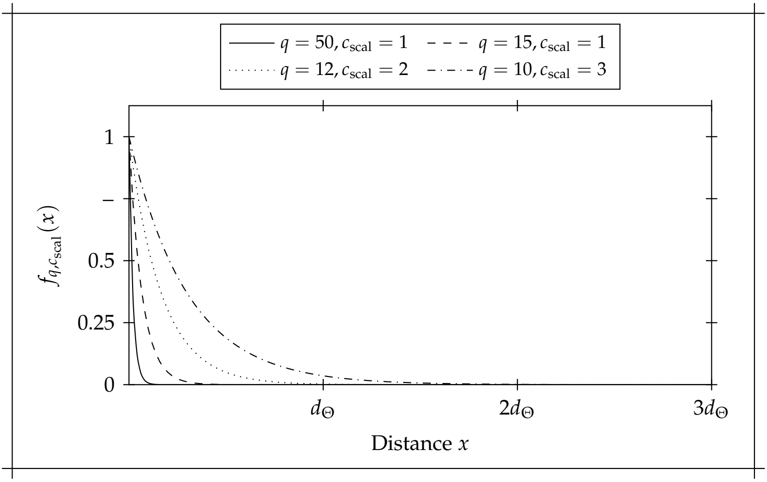


Figure 3.3: Enrichment functions used by ABBAS et al. (2010) for a high gradient at $x = 0$.

3.4.2 Local temporal discretization

The use of numerical integrators with a higher order allows larger time steps with the same quality of the results. A family of methods of arbitrary order are the *Runge*[28]*-Kutta*[29] *methods* (BUTCHER 1986). The temporal integral is determined by additional intermediate stages with different weighting. In the case of explicit higher order methods, stability problems arise. However, implicit higher order methods are not necessarily faster, despite the larger time steps, since the computational effort is increased for the time step itself. Of particular importance in this respect are the *Lobatto*[30] *methods* as a subclass of the Runge-Kutta methods. They have a *First Same As Last (FSAL)* structure which allows the reuse of the last intermediate stage in the next time step and thus makes them efficient solvers (GONZÁLEZ PINTO et al. 1997).

The *Alternating Direction Implicit (ADI)* method for the numerical integration of the diffusion equation also results in reduced effort (PEACEMAN & RACHFORD 1955). The time derivative is split into the various spatial directions. Originally

[28] **Carl David Tolmé Runge** (* August 30th, 1856 - † January 3rd, 1927), German mathematician, physicist, and spectroscopist.

[29] **Martin Wilhelm Kutta** (* November 3rd, 1867 - † December 25th, 1944), German mathematician.

[30] **Rehuel Lobatto** (* June 6th, 1797 - † February 9th, 1899), Dutch mathematician.

developed for the two-dimensional case, extensions exist for several dimensions (DOUGLAS 1962), as well as the use of explicit integrators (FELL & WILLEKE 2010), i. e. the *Alternating Direction Explicit (ADE)* method.

Closely related to the dimensional splitting are *Operator Splitting Methods*. These methods take into account the different terms of the solution. Depending on the requirements, the numerical integrators are combined. ASCHER et al. (1995) used an *implicit-explicit (IMEX) method* for solving the convection-diffusion equation. For the part of convection, an explicit method is applied, and an implicit integrator is used for diffusion. In the case of simulation of the temperature field of LBM at the component level, only diffusion is considered, which is why splitting the operator is not applicable.

All methods described thus far have in common that the effort is lower or a larger time step is used than with the backward Euler, but the time step for the whole domain is still dictated by the domain with the strongest requirement. In any case, the backward Euler is the preferred choice of numerical integrators due to its good stability properties with regard to the avoidance of oscillations. Of interest are therefore methods which partition the system of equations geometrically into subsystems and solve them separately from each other. However, since the different subsystems are dependent on each other, coupling conditions are defined. These are either algebraic constraint equations, such as the Lagrange multiplier, or constitutive equations, such as the same heat flux boundary conditions (MEYER et al. 2018). In an iterative sequence, the subsystems are solved and the coupling conditions are updated. The methods are mainly used in the parallelization of calculations, whereby this aspect is of secondary importance, even if it allows for an acceleration of the calculation. More decisive at this point is that the subsystems are solved separately from each other and thus allow the use of different time steps. The predominant aspect of using local time steps leads to a different partitioning of the calculation domain compared to the general parallelization procedures. For parallelization, the goal is to obtain subsystems with the same solution effort and low coupling at the same time. On the one hand, the time that a subsystem has to wait for the other to update the constraint is minimized, and, on the other hand, the frequency of the necessary exchange is reduced. In LBM, partitioning in the z-direction would therefore be an option, since the heat flux mainly occurs from the build surface to the build plate. With regard to the use of different time steps, a partitioning into the already existing layers would be more reasonable, since the gradient decays towards the build plate. If different time step sizes are selected, in addition, the subsystems are not always evaluated at the same time. The exchange between the subsystems then only takes place in certain time steps. In the remaining time steps, the coupling variables are extrapolated or interpolated (ARNOLD et al. 2014). There are a number of tasks to use local time steps (ALTMANN et al. 2010; BENEŠ & KRUIS 2018; MEYER et al. 2018). To avoid the circumstance of exchange, there are approaches that use the same time steps but distinguish between explicit and implicit integrators (BELYTSCHKO & MULLEN 1978; T. J. R. HUGHES et al. 1979). However, the use of the Lagrange multiplier sometimes also leads to stability problems (NAKSHATRALA et al. 2009).

One approach that avoids this is the *multirate method* of GEAR & WELLS (1984). Although the entire system

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)) \tag{3.22}$$

is partitioned as well, it is not solved individually. Rather, it is only partially reduced in time. One advantage over other methods is that the use of additional coupling parameters is no longer necessary and that coupling errors are reduced as the overall system is calculated in certain time steps. This makes the multirate method additionally suitable for implementation in existing software, as only one system of equations needs to be solved at a time. The multirate method divides the system variable $\boldsymbol{y}$ into a slow $\boldsymbol{y}_1$ and a fast component $\boldsymbol{y}_2$, i. e.

$$\dot{\boldsymbol{y}}_1(t) = \boldsymbol{f}_1(t, \boldsymbol{y}_1(t), \boldsymbol{y}_2(t)), \tag{3.23}$$

$$\dot{\boldsymbol{y}}_2(t) = \boldsymbol{f}_2(t, \boldsymbol{y}_1(t), \boldsymbol{y}_2(t)). \tag{3.24}$$

For the slower component, time steps are used that are an integer multiple of the time steps of the faster component. The overall system is calculated in a common time step, otherwise the values of the slower component serve as a boundary condition for the faster component. These are either extrapolated/interpolated or assumed to be constant between the steps of the overall system. A distinction is made between *fastest-first* and *slowest-first*, which indicates the time step to be performed first. In the fastest-first variant, the calculation of the faster component with the boundary conditions takes place on the basis of the last preceding time step in common. In the slowest-first variant, a step of the overall system is performed first; then, the omitted time steps of the faster component are calculated. For the construction of the boundary conditions, the values of the slow component from a future point in time are used. In the case of fixed time steps, it is recommended to use the fastest-first variant. The method is expandable to an arbitrary number of components. It is often used in the calculation of integrated analogue and digital circuits (GUENTHER & RENTROP 1994; KATO et al. 2009), but also achieved good results for the convection-diffusion equation (SCHLEGEL et al. 2009), as well as for pure diffusion (PROKAJ & CHOUDHURY 2006).

3.4.3 Holistic approaches for solving systems of equations

The numerical solving of the nonlinear heat equation consists of several levels. The observation interval is divided into different time steps (level 1). For each time step, a system of nonlinear equations, as a consequence of the temperature-dependent material parameters, must be solved (level 2). This is iteratively transformed into a system of linear equations whose solution is level 3. The activation of a new layer in the simulation of LBM is even interpretable as an additional level (level 0), since DoFs are added and thus change the system of equations.

In the state of the art, there are many methods for acceleration on level 3. A large number of publications deal with the parallelization of algorithms for the solution of a system of linear equations (HE et al. 2016; MIFUNE et al. 2002; ROSSI & TOIVANEN 1999; SCHENK 2000; VAN HENSON & U. M. YANG 2002). The reason for this is the development of increasingly powerful Graphics Processing Units (GPUs), which, with their ability to execute many processes simultaneously, have also become interesting for calculations beyond their original tasks (General Purpose Computation on Graphics Processing Unit (GPGPU)). The aim of this thesis is to reduce the effort. The aim of parallelization methods is not to reduce the effort required for acceleration as much as to ensure that operations are performed simultaneously and not sequentially. Thus, they are not the focus of this work.

The *Model Order Reduction techniques* are mainly added to level 3. Various methods exist within this class, whereby the general intent of all methods is to reduce the system to its essential DoFs. In the case of the linear two-dimensional heat equation, an acceleration of 44× is achieved (OJO et al. 2015), but there are large differences in the solution quality of the different techniques (ŞAHIN et al. 1995). Starting from the methods for systems of linear equations, there are also approaches for systems of nonlinear equations (PHILLIPS 2003; REWIENSKI & WHITE 2003); thus, level 2 is also addressed by Model Order Reduction. However, as the complexity of the geometry increases, the reduction effect of the DoFs decreases.

STEWART (1998) describes procedures for updating the inverse and Cholesky decomposition as a result of a change in the coefficient matrix, thus avoiding complete recalculation. The *Woodburry matrix identity* specifies the relation between the new inverse and the previous inverse. However, the inverse should usually be avoided, because, even with an existing inverse, the solution of a system of linear equations is more expensive than the forward and backward substitution with triangular matrices. While earlier work on updating the Cholesky decomposition dealt with changes of rank 1 (GILL et al. 1972), methods exist for updating with changes of a higher rank (cf. Section 2.2, DAVIS & HAGER 2001). If the temperature change is initially only in the area close to the heat source, the change is of low rank in the coefficient matrix. As time progresses, large areas are affected by the temperature change, and if all nodes are affected, there is no longer an advantage to updating as compared to recalculating the Cholesky decomposition. Updating the Cholesky decomposition is assigned to level 2 as well as to level 1. Although it is not directly a method for solving systems of nonlinear equations, it refers to the changed matrix as a result of the iterative approach to the solution. This also applies to the changes that occur from one time step to the next. With the extension of updating the Cholesky decomposition to the coefficient matrix with additional columns and rows (STEWART 1998), it is in principle also transferable to level 0.

3.5 Summary and Conclusion

The state of the art in the field of numerical methods for accelerating the computing time in the simulation of LBM has been presented and discussed. It has become clear that numerical methods have not been considered at all, or only as a marginal topic, in many studies on the simulation of LBM. However, the few existing studies have revealed their potential. The interim conclusion on the status quo reveals the increasing need for action in the field of numerical methods in the case of more detailed Conceptual Models.

In the field of spatial discretization, much finer meshes become necessary, and, consequently, larger systems of equations result. To avoid this, methods are needed which are able to represent steep gradients despite coarse meshing. Regarding the temporal discretization, the strong demands on the time step size, which arise in the uppermost layer, are currently being applied for all areas and lead to unnecessary effort. The finer the time discretization is, the stronger this effect becomes. Therefore, numerical methods are needed that take into account the local requirements of the different areas. Overall, this leads to larger systems of equations which need to be solved more often. The Cholesky decomposition from the previous time step and even the previous iteration is not directly reusable due to the non-linearity of the temperature-dependent material parameters. Therefore, one of the greatest advantages of Direct Methods has not been exploited in most research. Approaches exist to reuse the inverse, but the use of triangular matrices for solving systems of equations is preferred. However, it turned out that the matrices have special structures.

The state of the art of the identified need for action in the general field of numerics has been examined. With the XFEM of ABBAS et al. (2010), a method is available that allows the mapping of steep gradients despite coarse meshing. However, on the one hand, several enrichment functions are necessary, and, on the other hand, the determination of the optimal enrichment functions requires a preliminary calculation, which relativizes the reduction of the computing time.

In the area of time discretization, various methods are available that use local time steps. The multirate method of GEAR & WELLS (1984) also allows a rapid implementation in existing FEM programs. However, local time step requirements or partitioning into components are not yet known for use in the LBM simulation. In addition, a growing calculation area has not yet been taken into account in the method.

There is no global strategy that considers solving the system of equations of a transient problem in its entirety. Updating the Cholesky decomposition, however, makes it possible to take into account knowledge derived from solving the previous time step. However, this gain is reduced as the computing domain becomes increasingly affected by the temperature change in the case of the heat equation.

4 Parameterized eXtended Finite Element Method for Steep Spatial Gradients

4.1 Overview

This chapter deals with the use of a coarse spatial discretization of the weak formulation of the heat equation

$$\begin{aligned}\int_{\Omega} \rho c(T) \frac{\partial T}{\partial t} v \,\mathrm{d}\Omega + \int_{\Omega} \lambda(T) \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} T \,\mathrm{d}\Omega + \int_{\Gamma_\mathrm{C}} \alpha v T \,\mathrm{d}\Gamma \\ = \int_{\Gamma_\mathrm{C}} \alpha v T_{\Gamma_\mathrm{C}} \,\mathrm{d}\Gamma - \int_{\Gamma_\mathrm{N}} v \dot{q}_{\Gamma_\mathrm{N}} \,\mathrm{d}\Gamma \qquad \forall t \in \Upsilon \qquad (2.55)\end{aligned}$$

despite the steep local gradients occurring during the energy input through a scan field (cf. Figure 4.1). Due to the calculation time, current models usually use only one element over the layer height for the calculation mesh. To fulfill the PDC and avoid unphysical oscillations, the first time step must be chosen accordingly, so that the temperature gradient spreads directly over at least five layers (HOGGE & GERREKENS 1983). Information about the transient development of the temperature gradient is therefore not obtained by the thermal simulation. However, the gradients in the high temperature range are especially

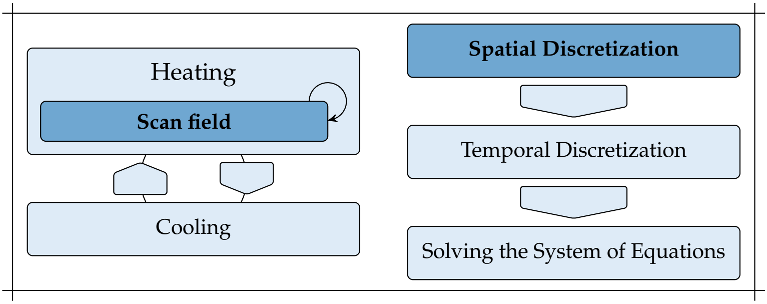


Figure 4.1: Classification of Chapter 4 with regard to the phase of the Mathematical Model of LBM and application area in numerics; field of actions (dark blue).

important for the correct representation of the residual stresses and resulting distortions. More accurate models for a better transient representation of the temperature gradient propagation in the simulation require a smaller local discretization than only one element over the layer height and lead to an increase in calculation time.

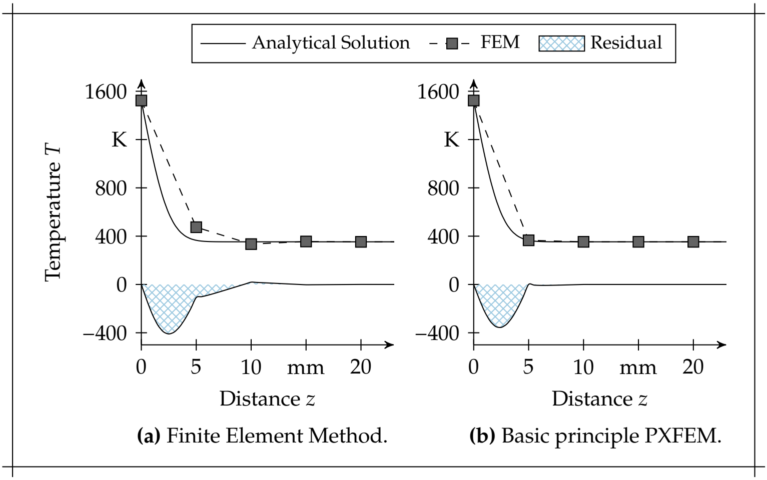
The XFEM enriches the shape functions in the corresponding elements with special functions which are able to represent the steep gradients and thus avoid a fine discretization. For the transient representation of the temperature gradient, however, several enrichment functions are required, which, on the one hand, partially nullify the advantage of fewer DoFs compared to a finer spatial discretization, and, on the other hand, must first be determined via a preceding optimization.

The Parameterized eXtended Finite Element Method (PXFEM) is introduced (Zeller et al. 2018). The PXFEM is an extrinsic XFEM with parameterized enrichment functions. While the shape functions of the FEM determine the correct nodal values, the enrichment functions compensate for the excessive energy in the element and ensure the correct gradient. The parameterization variable is an additional DoF, which enables a single enrichment function to represent a time-varying gradient. It is handed over to the system of equations and determined anew in each iteration. An analytical integration is used to set up the element matrices to cover a large temporal range with the parameterization variable. The PXFEM is verified for the case that only the first layer is affected by a temperature change, as well as that the penetration depth extends over several layers. This is accompanied by the consideration of a single enriched element, as well as the enrichment of several elements and an additional enrichment during the simulation. The investigations were carried out on various enrichment functions, the derivation of which is also presented. The PXFEM is applicable for the three-dimensional case. However, since this is not an established method, it is first examined using the one-dimensional linear heat equation to illustrate the simplified heat flux from the build surface to the build plate. Analytical solutions are available for this and thus allow for an exact evaluation of the method. In the concluding validation, the necessary spatial refinement of the calculation mesh using the FEM is examined in order to obtain at least the same accuracy as the PXFEM with the initial mesh configuration.

4.2 Methodology

The FEM primarily aims at correctly reproducing the energy of the whole system. If the steep gradients within an element exceed the ability of the shape functions to display them, the excess energy is compensated for by the neighboring elements (cf. Figure 4.2a). The basic idea behind the PXFEM is to enable the

FEM to determine the nodal values and, if necessary, to balance the energy by the enrichment functions directly in the affected element (cf. Figure 4.2b). One enrichment function $v_{n,o}$ is used to compensate for the residual between two nodes $\boldsymbol{r}_n$ and $\boldsymbol{r}_o$ of an element. According to Figure 4.2b,



(a) Finite Element Method.

(b) Basic principle PXFEM.

Figure 4.2: Exemplary results of the FEM and PXFEM for the corresponding residual with regard to the analytical solution from Equation 2.21.

$$v_{n,o}(\boldsymbol{r}_s, \zeta_{n,o}) = 0 \tag{4.1}$$

applies to all nodes $\boldsymbol{r}_s$ in Ξ. As is apparent from the solutions of Equation 2.21 for the linear heat equation, the considered temperature range has no direct influence on the temporal behavior, but only scales the height. Therefore, the scaling of the enrichment function is done by the temperature difference of the respective boundary nodes $\boldsymbol{r}_n$ and $\boldsymbol{r}_o$. The parameterization variable $\zeta_{n,o}$ is introduced to map the temporal progression. A solution of the PXFEM with u_ζ enrichment functions thus has the form

$$T(\boldsymbol{r}, t) = \sum_{n=1}^{n_\Xi} T_n(t)\, v_n(\boldsymbol{r}) + \sum_{(n,o)\in\Lambda} [T_n(t) - T_o(t)]\, v_{n,o}(\boldsymbol{r}, \zeta_{n,o}(t)), \tag{4.2}$$

where Λ, $|\Lambda| = u_\zeta$ is a set of the unordered pairs (n, o) with n and o being the indices of the enriched node pairs. Accordingly, this type of enrichment only makes sense if there is a difference between the respective nodal temperatures.

Following the isoparametric concept of the FEM, the test function space for the PXFEM is extended by the set

$$V^{\zeta} = \bigcup_{(n,o)\in\Lambda} v_{n,o} \tag{4.3}$$

of enrichment functions. Thus, a solution T of PXFEM must satisfy the equation

$$\begin{aligned}\int\limits_{\Omega} \rho c(T)\frac{\partial T}{\partial t} v \,\mathrm{d}\Omega + \int\limits_{\Omega} \lambda(T)\boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} T \,\mathrm{d}\Omega + \int\limits_{\Gamma_{\mathrm{C}}} \alpha v T \,\mathrm{d}\Gamma \\ = \int\limits_{\Gamma_{\mathrm{C}}} \alpha v T_{\Gamma_{\mathrm{C}}} \,\mathrm{d}\Gamma - \int\limits_{\Gamma_{\mathrm{N}}} v \dot{q}_{\Gamma_{\mathrm{N}}} \,\mathrm{d}\Gamma, \qquad \forall v \in V^{s}(\Omega) \cup V^{\zeta}\end{aligned} \tag{4.4}$$

for all times t in Y. Unlike the elements of the function space $V^{s}(\Omega)$, the test functions of V^{ζ} depend on another variable besides the spatial variable. It is considered to be identical to the parameterization variable of the corresponding enrichment function and is thus determined instantaneously with the other DoFs.

The dependence of the enrichment function, and thus also of the test function, on the parameterization variable is nonlinear. As a result, in contrast to the FEM (cf. Figure 2.62), additional nonlinear terms

$$\begin{aligned}\boldsymbol{C}(T) \cdot \dot{\boldsymbol{T}}(t) + [\boldsymbol{K}(T) + \boldsymbol{K}_{\mathrm{C}}(T)] \cdot \boldsymbol{T}(t) - \boldsymbol{q}(t) \\ + \boldsymbol{C}^{Z}\big(\boldsymbol{T}, \dot{\boldsymbol{T}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}\big) + \boldsymbol{K}^{Z}(\boldsymbol{T}, \boldsymbol{\zeta}) + \boldsymbol{K}^{Z}_{\mathrm{C}}(\boldsymbol{T}, \boldsymbol{\zeta}) = 0, \qquad \forall t \in \mathrm{Y}\end{aligned} \tag{4.5}$$

result from the fulfillment of Equation 4.4 for each test function of $V^{s}(\Omega)$. The system of equations for the PXFEM is completed by the equations obtained with the test functions V^{ζ}:

$$\boldsymbol{C}^{\zeta}\big(\boldsymbol{T}, \dot{\boldsymbol{T}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}\big) + \boldsymbol{K}^{\zeta}(\boldsymbol{T}, \boldsymbol{\zeta}) + \boldsymbol{K}^{\zeta}_{\mathrm{C}}(\boldsymbol{T}, \boldsymbol{\zeta}) - \boldsymbol{q}(t) = 0, \qquad \forall t \in \mathrm{Y}. \tag{4.6}$$

The vector $\boldsymbol{\zeta}$ represents all parameterization variables $\zeta_{n,o}$ with $(n, o) \in \Lambda$. To specify the new terms, first the set of indices Z_n is introduced, which contains all indices that form a pair with n in the set Λ, i. e.

$$Z_n = \left\{ o \,\middle|\, \exists (n, o) \in \Lambda \right\}. \tag{4.7}$$

For the additional terms in Equation 4.5

$$C_k^Z(\boldsymbol{T},\dot{\boldsymbol{T}},\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}}) = \sum_{o\in Z_n}\left[\int_\Omega \rho c(T)\left(\dot{T}_n - \dot{T}_o\right) v_{n,o}(\boldsymbol{r},\zeta_{n,o})\, v(\boldsymbol{r})\ \mathrm{d}\Omega \right. \tag{4.8a}$$

$$\left. + \int_\Omega \rho c(T)\left(T_n - T_o\right) \dot{\zeta}_{n,o}\dot{v}_{n,o}(\boldsymbol{r},\zeta_{n,o})\, v(\boldsymbol{r})\ \mathrm{d}\Omega \right], \tag{4.8b}$$

$$K_k^Z(\boldsymbol{T},\boldsymbol{\zeta}) = \sum_{o\in Z_n}\left[\int_\Omega \lambda(T)\left(T_n - T_o\right) \boldsymbol{\nabla} v_{n,o}(\boldsymbol{r},\zeta_{n,o})\cdot \boldsymbol{\nabla} v(\boldsymbol{r})\ \mathrm{d}\Omega\right], \tag{4.8c}$$

$$K_{C,k}^Z(\boldsymbol{T},\boldsymbol{\zeta}) = \sum_{o\in Z_n}\left[\int_{\Gamma_C} \alpha\left(T_n - T_o\right) v_{n,o}(\boldsymbol{r},\zeta_{n,o})\, v(\boldsymbol{r})\ \mathrm{d}\Gamma\right] \tag{4.8d}$$

apply. The three terms from Equation 4.6 are not explicitly listed, as they correspond to the respective linear and nonlinear term from Equation 4.5. As a result of the nonlinear test functions V^ζ, however, both terms are nonlinear and are combined into one term.

4.3 Verification

A procedure for obtaining enrichment functions for the PXFEM is presented in the following and carried out using two examples. Subsequently, their ability to map the steep spatial temperature gradient is investigated for the simplified, one-dimensional heat flux in the z-direction from the build surface to the build plate with constant material parameters. Initially, the case is considered that only the first layer is affected by a temperature change due to the applied boundary condition. Since enrichment only makes sense if there is a temperature difference between two nodes, this is accompanied by an enrichment of the boundary element. In the subsequent consideration of the temperature changes over several layers, the investigation of several enriched elements also takes place.

Ten layers with a layer height of 5 mm were modeled, i. e. $\Omega = [0, 50\,\mathrm{mm}]$. All simulations were performed with the maximum element size of one layer height ($d_\Theta = 5\,\mathrm{mm}$) and material parameters of Inconel 718 at a temperature of 748.15 K ($a = 5.18 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$). The following therefore holds for the nodes z_n:

$$z_n = (n-1)d_\Theta, \qquad \forall n \in (1,\dots,11)\,. \tag{4.9}$$

The analytical solution for the semi-infinite body from Equation 2.21 is used as a reference. Although no infinite domain is considered, for short time periods, such as the 'Heating' phase, in which only the boundary is affected by the temperature change, the solution is also valid for finite domains (POLIFKE & KOPITZ 2009).

The PXFEM was implemented with MATLAB AND SYMBOLIC MATH TOOLBOX (2016). The source code is available at the media and publications repository of the Technical University of Munich (ZELLER et al. 2019).

4.3.1 Parameterized enrichment functions

The starting point is a function $v_{\text{init}}(z, \zeta)$ that maps the underlying physical behavior as well as possible over time with the parameter ζ. Thus, the same requirements apply to this function as to the enrichment functions of the XFEM. Following the idea that the shape functions of the FEM determine the nodal values, the initial function v_{init} is correspondingly reduced by these proportions in the domain to be enriched. The remaining part is normalized to the difference of the respective nodal values. In summary, the enrichment functions $v_{n,o}$ for all $(n, o) \in \Lambda$ and $z \in [z_n, z_o]$ result from

$$
\begin{aligned}
v_{n,o}(z, \zeta_{n,o}) = & \frac{v_{\text{init}}(z, \zeta_{n,o})}{v_{\text{init}}(z_n, \zeta_{n,o}) - v_{\text{init}}(z_o, \zeta_{n,o})} \\
& - \frac{\left[v_{\text{init}}(z_n, \zeta_{n,o}) \left(1 - \frac{z - z_n}{d_\Theta}\right) + v_{\text{init}}(z_o, \zeta_{n,o}) \frac{z - z_n}{d_\Theta} \right]}{v_{\text{init}}(z_n, \zeta_{n,o}) - v_{\text{init}}(z_o, \zeta_{n,o})}.
\end{aligned} \tag{4.10}
$$

Outside the enriched element, the respective enrichment function $v_{n,o}$ has the value 0.

In this thesis, enrichment functions are considered based on the function

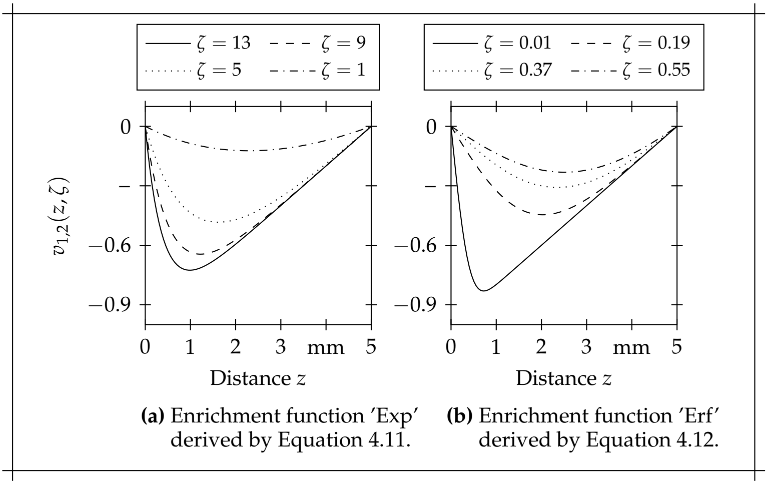
$$
v_{\text{init}}(z, \zeta) = \frac{\exp\left(\zeta - \zeta \frac{z}{d_\Theta}\right) - 1}{\exp(\zeta) - 1} \tag{4.11}
$$

already used in ABBAS et al. (2010) for calculating steep gradients with the XFEM. Enhancement with these functions is referred to as 'Exp' in the following. Furthermore, the function

$$
v_{\text{init}}(z, \zeta) = 1 - \operatorname{erf}\left(\frac{z}{\sqrt{4a\zeta}} \right), \tag{4.12}
$$

derived from the analytical solution, is considered and referred to as 'Erf'. Since the error function occurring in it is a non-elementary function, it is only conditionally suitable for use in application-oriented software. However, the

PXFEM is theoretically able to exactly reproduce the analytical solution with it. Deviations of the results due to inadequacies of the enrichment function to represent the gradients are excluded, and an evaluation of the method itself takes place independently of the enrichment functions. The enrichment function $v_{1,2}$ for the first element derived from both Equation 4.11 and Equation 4.12 is illustrated in Figure 4.3 for exemplary values of the parameterization variable ζ.



(a) Enrichment function 'Exp' derived by Equation 4.11.

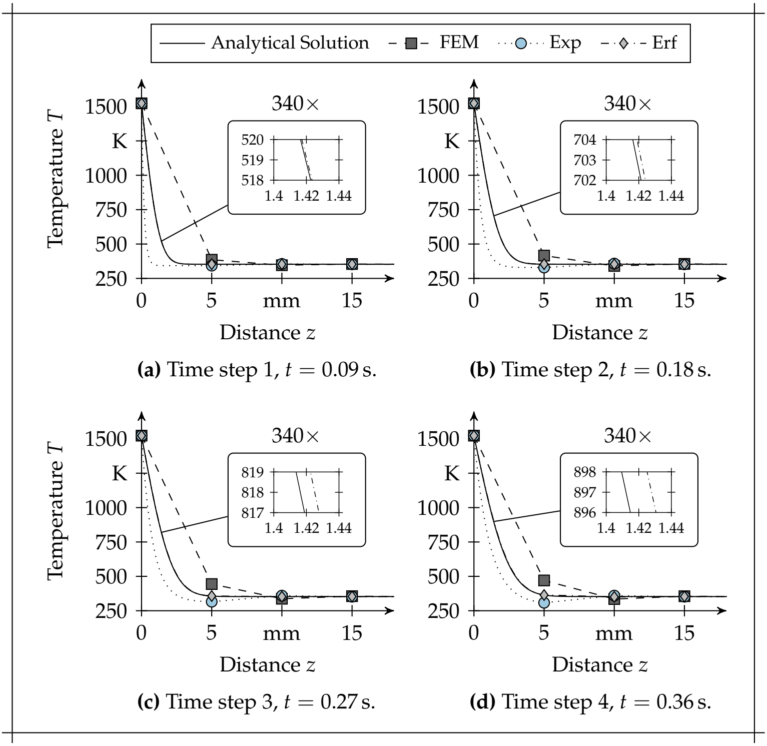
(b) Enrichment function 'Erf' derived by Equation 4.12.

Figure 4.3: Parameterized functions for the enrichment of the first element illustrated for various ζ.

4.3.2 Spatial gradient within one layer

In this subsection, the PXFEM is investigated for the period where only the first element is affected by a temperature change due to the boundary condition. This corresponds to a time interval of up to 0.36 s using the penetration depth from Equation 2.33. Accordingly, only the first element is enriched, since no temperature difference for the other elements is associated with the period under consideration.

For the parameterization variables, initial conditions must be selected, since they are regarded as additional Degrees of Freedom (DoFs). The selected values determine the initial energy in the enriched elements. Therefore, the values are chosen to obtain a steep gradient at the node of the Dirichlet boundary condition, and thus a negligible initial energy input by the enrichment function, i. e. $\zeta_{1,2} = 300$ for 'Exp' and $\zeta_{1,2} = 1 \times 10^{-4}$ for 'Erf'. The results for the PXFEM using four equidistant time steps and the mentioned setup are illustrated in Figure 4.4.



(a) Time step 1, $t = 0.09\,\mathrm{s}$.

(b) Time step 2, $t = 0.18\,\mathrm{s}$.

(c) Time step 3, $t = 0.27\,\mathrm{s}$.

(d) Time step 4, $t = 0.36\,\mathrm{s}$.

Figure 4.4: Results of the FEM and the PXFEM with the enrichment functions 'Erf' and 'Exp', as well as the analytical solution, displayed for the first four nodes; $T_{\Gamma_D} = 1523.15\,\mathrm{K}$, $T^0 = 353.15\,\mathrm{K}$, $a = 5.18 \times 10^{-6}\,\mathrm{m}^2\,\mathrm{s}^{-1}$, $d_\Theta = 5\,\mathrm{mm}$.

The FEM has the largest deviations from the analytical solution. As a result of the shape functions, energy corresponding to the linear gradient from the first to the second node is initially already present. Accordingly, the FEM calculates less the energy flow from the boundary into the first element, but from the first element into the second element. Due to the small time step, only the area at the edge of the second element would be affected by the energy input. Since, again, only a linear gradient is representable by the shape functions, the excess energy is compensated for by the further elements, and oscillations result.

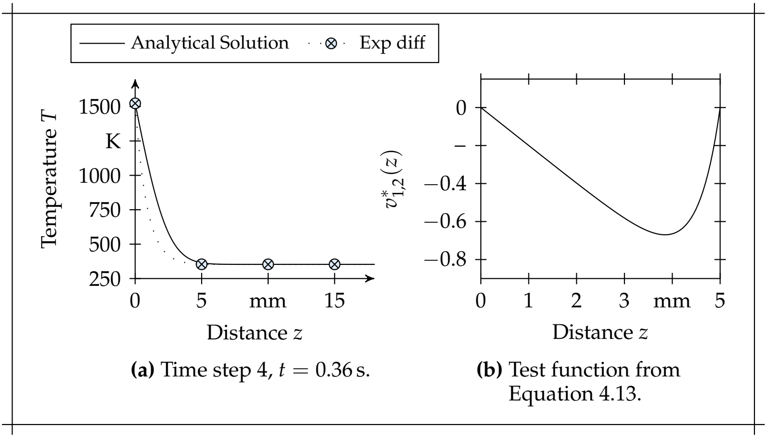
The PXFEM with the enrichment function 'Erf' provides the best result. A graphical differentiation between the analytical solution and the method is only possible with a significantly higher resolution. However, the deviation increases slightly over time.

Not quite as good as with the enrichment function 'Erf', but still considerably better than the FEM, the PXFEM with the enrichment function 'Exp' approximates the analytical solution. The enriched element initially shows stiff behavior and underestimates the temperature curve, but the gradient approaches the analytical solution with increasing time steps. However, an undershoot occurs in the second node, which increases over the time steps. ABBAS et al. (2010) have mentioned the observation of oscillations in the XFEM with only one enrichment function in the form of Equation 4.11. The enrichment function is not able to exactly compensate for the residual from the FEM. With the help of the test functions, however, it is possible to define areas that have a greater influence on the solution (the test function is interpretable as a weighting of the integral in Equation 4.8, and thus as a prioritization of areas). Due to the isoparametric approach, the additional test function corresponds to the enrichment function. With increasing time, and a corresponding decreasing parameter $\zeta_{1,2}$, the weighting shifts to the middle of the first element (cf. Figure 4.3a). The PXFEM with the enrichment function 'Exp' and isoparametric approach thus prioritizes the left side of the interval, whose node is already defined by the Dirichlet boundary condition. The right side, which is in turn connected to the free node, is only subordinate. A remedy is to select a test function that is not equal to the shape function. By weighting the area at the right boundary of the element, the undershoots are reduced, since the respective nodal value is more strongly regarded in the solution. This is confirmed by the result illustrated in Figure 4.5a, obtained using the test function presented in Figure 4.5b. Only the last time step is provided as an example, since Figure 4.4d contains the largest undershoot. With the changed test function, no graphical deviation of the second nodal value is visible (Figure 4.5a). Instead of the enrichment function based on Equation 4.11 with a changing parameterization variable, the function

$$v^*_{1,2}(z) = \frac{\exp\left(10\frac{z}{d_\Theta}\right) - 1}{\exp(10) - 1} - \frac{z}{d_\Theta} \tag{4.13}$$

was used as a test function in each time step (Figure 4.5b). This type of enrichment is further referred to as 'Exp diff'.

The deviation from the analytical solution is not subsequently evaluated, as this is always accompanied by an evaluation of the enrichment function and not just of the PXFEM itself. Therefore, at first, the parameterization variables obtained from the simulation are compared with the optimal values with respect to a minimum error for each enrichment function. Assuming that the nodal values



(a) Time step 4, $t = 0.36\,\text{s}$.

(b) Test function from Equation 4.13.

Figure 4.5: Results for the PXFEM with the enrichment function 'Erf' (left, $T_{\Gamma_\text{D}} = 1523.15\,\text{K}$, $T^0 = 353.15\,\text{K}$, $a = 5.18 \times 10^{-6}\,\text{m}^2\,\text{s}^{-1}$, $d_\Theta = 5\,\text{mm}$) and modified test function (right).

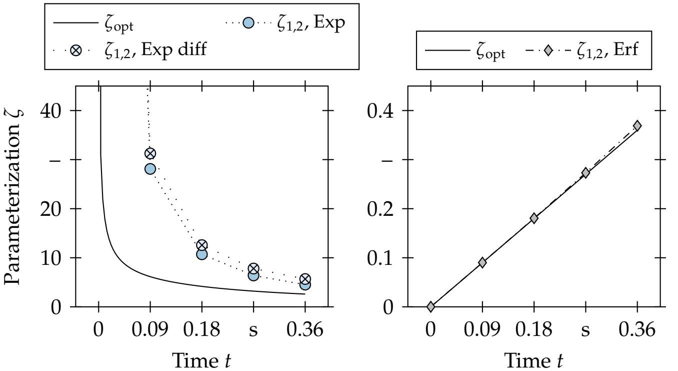
are represented correctly, the optimal parameterization variables ζ_opt minimize the remaining residual, i. e.

$$\left\| \left\{ T_\text{ana}(0)\left(1 - \frac{z}{d_\Theta}\right) + T_\text{ana}(d_\Theta)\,\frac{z}{d_\Theta} \right.\right. \\ \left.\left. + \left[T_\text{ana}(0) - T_\text{ana}(d_\Theta)\right] v_{1,2}(z, \zeta_\text{opt}) \right\} - T_\text{ana}(z) \right\|_2 . \qquad (4.14)$$

The same accuracy requirements for the optimization were applied as for the solution of the PXFEM. The results are displayed in Figure 4.6.

With the enrichment function 'Erf', almost the optimum parameterization variables are obtained. Only in the last time step deviations are graphically recognizable. This is attributed to the fact that the temperature change reaches the second, non-enriched element, although the considered time interval is determined according to the penetration depth from Equation 2.33 to limit the affected area to the first element. However, since the propagation velocity of the heat equation is infinitely high, the penetration depth is based on a tolerated deviation of up to 1 % related to the difference between the boundary condition and the initial condition. Later, looking at the explicit temperature deviations, it becomes apparent that they lie within this tolerance.

In the case of the enrichment functions of type 'Exp', the obtained values of the parameterization variable differ from the optimal, even if they approach it



(a) Enrichment function derived by Equation 4.11.

(b) Enrichment function derived by Equation 4.12.

Figure 4.6: Optimal values of the parameterization variable and values obtained from the PXFEM in Figure 4.4.

better with increasing time steps. If this was initially described as stiff behavior in the description of the graphical results, the reason is apparent from the progression of the optimal value. It is evident that the curve of the optimal parametrization variable is linear for the enrichment function 'Erf' (Figure 4.6b) and highly nonlinear for the enrichment function 'Exp' (Figure 4.6a). In the first case, this is due to the equidistant time steps and the choice of the time variable t from the analytical solution, on which the enrichment function is based, as parameterization variable. Due to the linear progression, no additional errors are caused by the numerical time integration with the backward Euler. This is not the case with the enrichment function 'Exp'. Especially for the first time step, which contains an infinite discontinuity at $t = 0$, a finer temporal discretization would be necessary.

The assumption, that the nonlinear progression of the optimal parameter is responsible for the deviations of the parameterization variable, is verified in two directions. To verify the assumption, the parameterization variables of both enrichment functions 'Erf' and 'Exp diff' are substituted to transfer them at least approximately to the qualitative progression of the respective other enrichment function. The parameterization variable in Equation 4.11 is substituted by

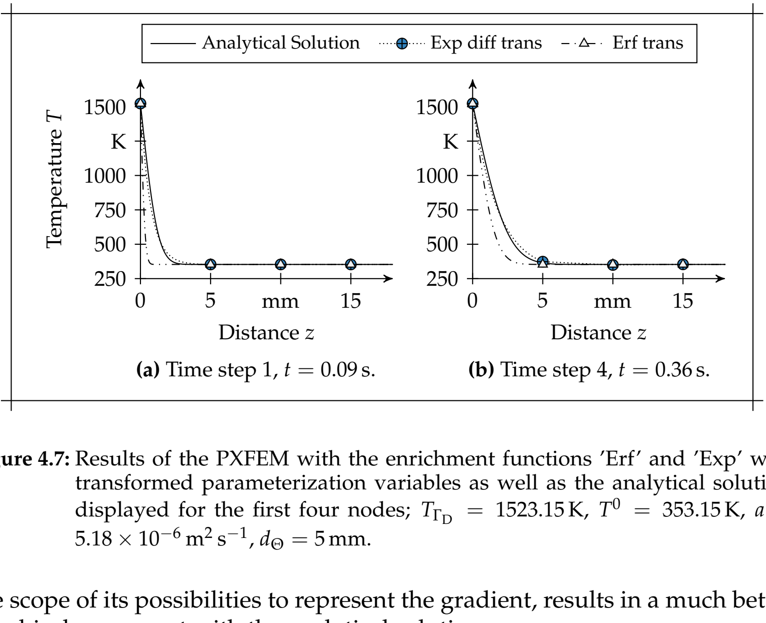
$$\zeta = \frac{1}{\sqrt{\zeta'}}. \tag{4.15}$$

Consequently, the steeper the gradient, the closer the parameterization variable approaches the value 0. Conversely, the parameterization variable in the initial function (cf. Equation 4.12) for the enrichment function 'Erf' is substituted by

$$\zeta = \frac{1}{(\zeta')^2}. \tag{4.16}$$

Accordingly, the progression of the optimal parameterization variable transforms into an infinite discontinuity at $t = 0$. The derived enrichment functions are referred to as 'Exp trans diff' and 'Erf trans', respectively. Thus, it is induced that the modified test function from Equation 4.13 is used in connection with the enrichment function 'Exp trans diff'.

Theoretically, the PXFEM with the enrichment function 'Erf trans' is still able to exactly represent the analytical solution. However, the graphical representation of the results in Figure 4.7 now also illustrates deviations of the obtained values of the parameterization variable from the optimal value for this enrichment function. On the other hand, the enrichment function 'Exp trans diff', within the scope of its possibilities to represent the gradient, results in a much better graphical agreement with the analytical solution.

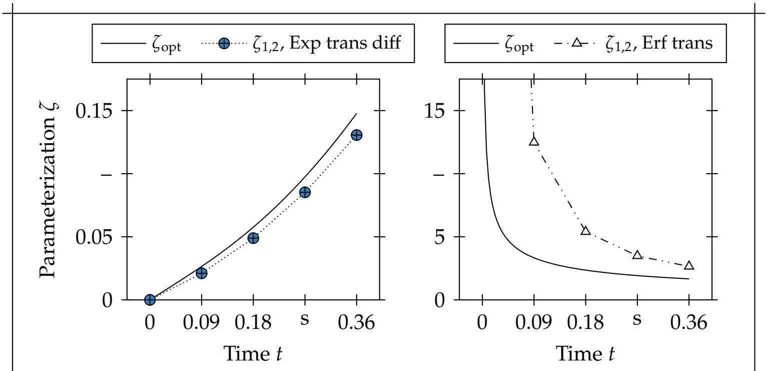


(a) Time step 1, $t = 0.09$ s.

(b) Time step 4, $t = 0.36$ s.

Figure 4.7: Results of the PXFEM with the enrichment functions 'Erf' and 'Exp' with transformed parameterization variables as well as the analytical solution, displayed for the first four nodes; $T_{\Gamma_\mathrm{D}} = 1523.15\,\mathrm{K}$, $T^0 = 353.15\,\mathrm{K}$, $a = 5.18 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$, $d_\Theta = 5\,\mathrm{mm}$.

The impressions from the graphical representation of the solutions are confirmed by comparing the parameterization variables obtained in each time step with the respective optimal value (Figure 4.8). For the enrichment function 'Erf trans', a



(a) Enrichment function 'Exp' and transformation from Equation 4.15.

(b) Enrichment function 'Erf' and transformation from Equation 4.16.

Figure 4.8: Optimal values of the parameterization variable and values obtained from the PXFEM with transformed enrichment functions in Figure 4.7.

strongly non-linear progression of the optimal parameterization variable results (Figure 4.8b). This is accompanied by large deviations in the first time step. The qualitative curve corresponds approximately to that of the enrichment function in Figure 4.6a. Conversely, the curve of the optimal parameterization variable for the enrichment function 'Exp diff trans' is approximately linear, and the results are now closer to the optimal value (Figure 4.8a). Therefore, the assumption of the error due to the nonlinear course of the parameterization variable is considered confirmed.

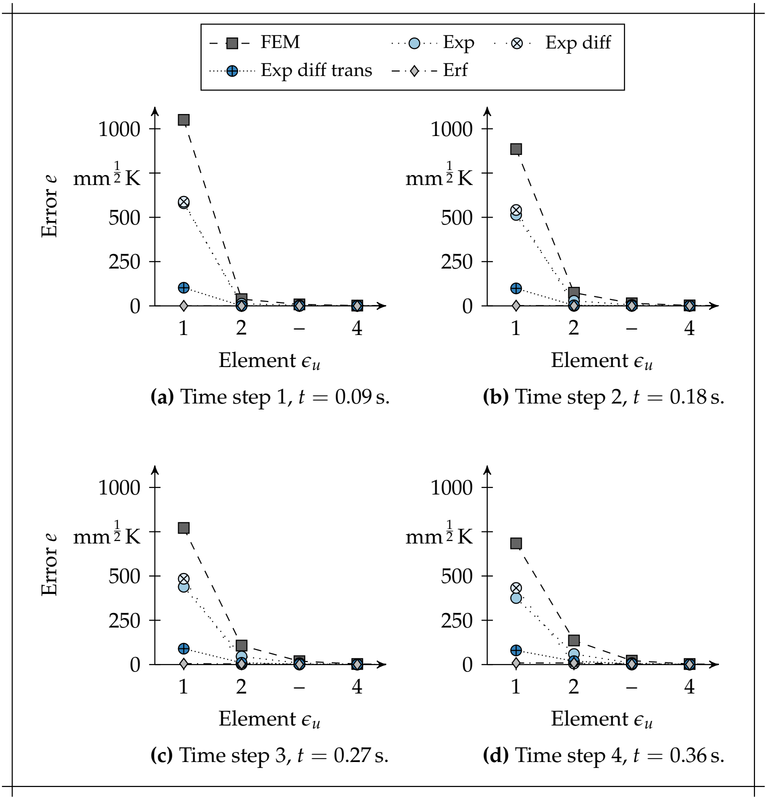
The enrichment function 'Erf trans' is only used to verify the assumption and is not considered further, as it does not provide any added value in terms of accuracy. The enrichment by 'Exp trans diff', however, represents an improvement from 'Exp diff', and is therefore included in further investigations.

For the final quantitative evaluation of the PXFEM, the errors compared to the analytical solution are analyzed according to

$$e = \|T - T_{\mathrm{ana}}\|_2 \,. \tag{4.17}$$

The results by elements for the different time steps are plotted in Figure 4.9.

The largest errors occur in both the FEM and the PXFEM in the first element, with the PXFEM consistently providing more accurate results. This also applies to the PXFEM with the enrichment function 'Exp', despite the larger undershoots. For both the FEM and the PXFEM with the enrichment 'Exp', the errors



(a) Time step 1, $t = 0.09\,\mathrm{s}$.

(b) Time step 2, $t = 0.18\,\mathrm{s}$.

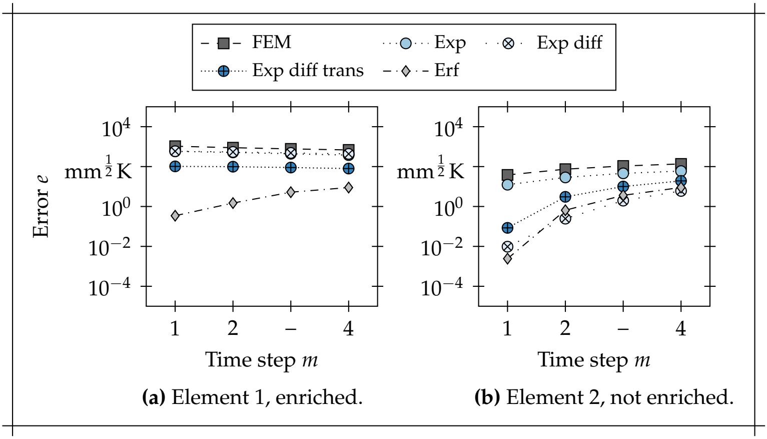
(c) Time step 3, $t = 0.27\,\mathrm{s}$.

(d) Time step 4, $t = 0.36\,\mathrm{s}$.

Figure 4.9: Error per step of the FEM and the PXFEM with various enrichment functions from Figure 4.4 and Figure 4.7, illustrated for the first four elements.

of the individual elements remain nearly the same over the time steps. As illustrated in the graphical comparison of the results, the PXFEM with enrichment function 'Erf' always delivers the most exact results for the first element.

The errors of each method in the first element predominate. Therefore, they are listed again in Figure 4.10 for all time steps. For almost all methods, the errors in the first element are nearly constant or are decreasing minimally over the time steps. An exception is the PXFEM with the enrichment function 'Erf', in which an increase of the error in the range of one order of magnitude is recognizable. However, this is attributed to the fact that the second element is reached by the temperature change. Due to the low error level of the method, this is reflected more strongly in the logarithmic representation than in the other



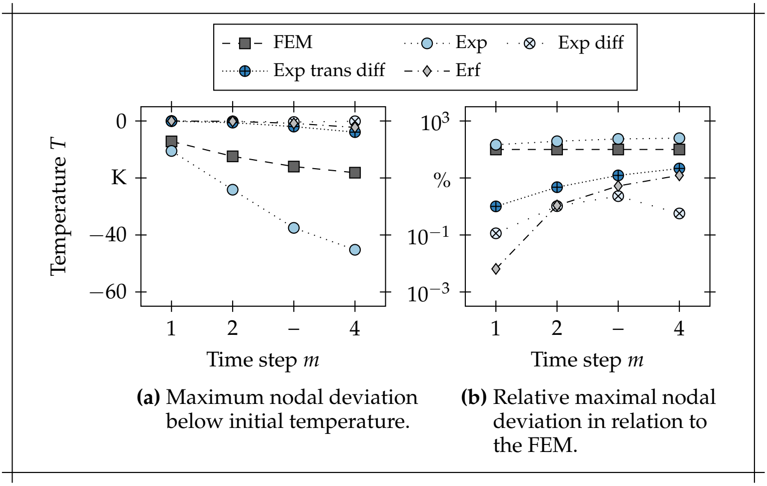
(a) Element 1, enriched.

(b) Element 2, not enriched.

Figure 4.10: Error development of the first and the second element over the time steps for the FEM and the PXFEM with various enrichment functions from Figure 4.4 and Figure 4.7.

methods. The error is still over 32 times smaller than that of the enrichment function 'Exp diff'. The largest error results from the FEM. The smallest error of the remaining methods results from the enrichment function 'Erf trans diff' (10 times smaller compared to the FEM). However, as illustrated in Figure 4.10b, this is not the case for the second element, where the error obtained by the enrichment function 'Exp diff' is smaller. This is due to the overestimation of the parameterization variables, which corresponds to a delayed heat flux. Accordingly, the reaching of the second, unenriched element by the temperature change occurs in a more attenuated way. An overshoot of the solution due to the lack of representability of the gradient in the second element does not occur. This becomes clear when considering the maximal occurring undershoots of the different methods in Figure 4.11.

For the FEM and the PXFEM with enrichment function 'Exp', the values increase with the time steps, whereby the latter are even higher by a factor of two. Due to the changed test function, however, the deviation is reduced to 1 K and thus amounts to a maximum of 4.2 % compared to the FEM. Thus, from time step three, even a smaller maximal deviation is achieved than with the PXFEM and the enrichment function from the analytical solution 'Erf', which is attributed to the previously mentioned delayed heat flux. The deviations of the PXFEM with enrichment functions 'Erf' and 'Exp trans diff' rise in the last time step to 12.4 % and 21.5 %, respectively, compared to the FEM. With explicit values of −2.25 K and −3.91 K, however, the deviation of the PXFEM is in the range of the tolerated temperature change by the penetration depth from Equation 2.33.



(a) Maximum nodal deviation below initial temperature.

(b) Relative maximal nodal deviation in relation to the FEM.

Figure 4.11: Nodal results below initial temperature of the FEM and the PXFEM with various enrichment functions from Figure 4.4 and Figure 4.7.

4.3.3 Spatial gradient over multiple layers

The previous subsection has served to verify the case that only the first layer is affected by a temperature change, and consequently only the first element is enriched. The PXFEM with initial enrichment of several elements, as well as the additional enrichment during the simulation, still needs to be verified. For this purpose, the case is examined that the first two elements are already affected by the temperature change and that the enrichment of the third element takes place during the observation period. This is guaranteed by the choice of six equidistant time steps and a considered period of 3 s. Initially, the first two elements are enriched, as they are already affected by the temperature change in the first step. Following Equation 2.33, the penetration depth reaches the third element in the third step and is thus enriched.

Based on the findings of the previous subsection, only the variants 'Exp trans diff' and 'Erf' are further considered. The substitution

$$\zeta = \frac{1}{\zeta'} \tag{4.18}$$

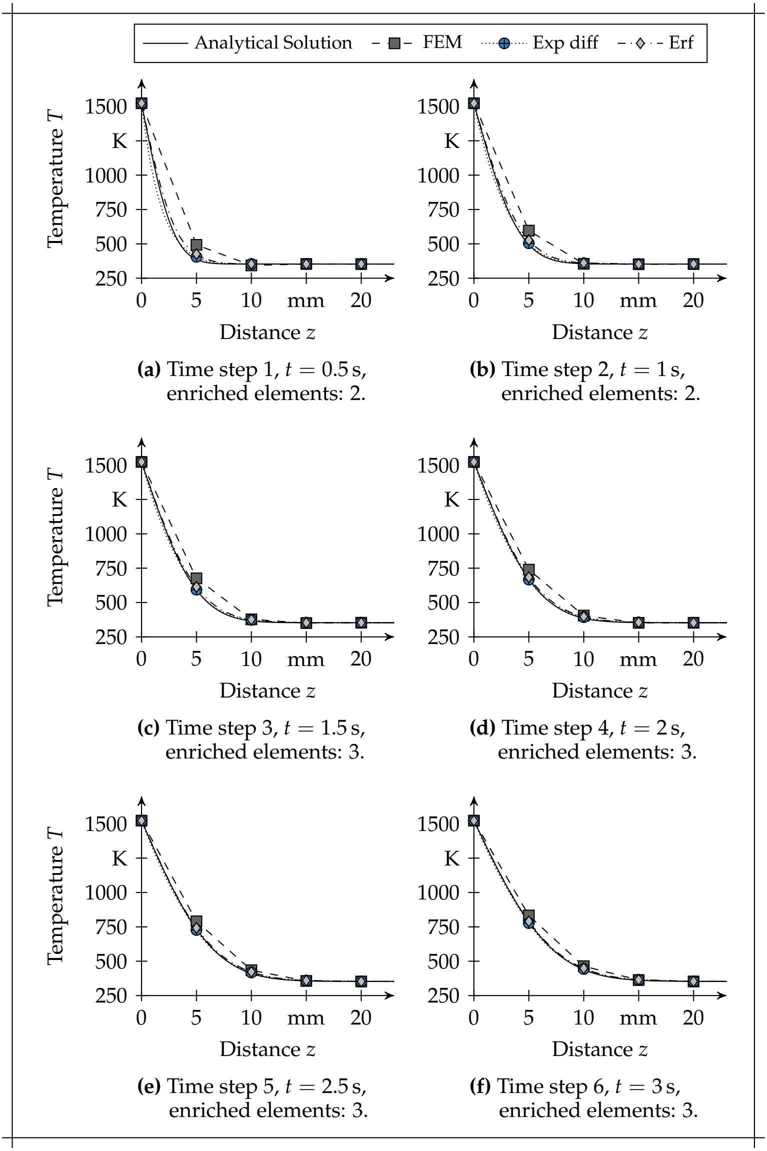
is used, however, as it represents a better linearization of the parameterization variable for the time scale under consideration. The initial values of the initially enriched elements are again aimed at ensuring the lowest possible energy. Therefore, the value 3.2×10^{-3} is used for the variant 'Exp trans diff', and the value 1×10^{-4} for the variant 'Erf'. The initial values for the parameterization variables of the additional enrichments, that take place during the simulation, are based on the theoretical value of the variable in the case of an enrichment already in the previous step. In the case of the enrichment function 'Erf', due to the procedure for deriving the enrichment functions and in conjunction with the fact that it is based on the analytical solution, the optimal parameters in a time step are identical for each enriched element. For this reason, the value of the parameterization variable of the neighboring element from the previous step is used as the initial value. The circumstance of identical optimal values of the parameterization variables is not given for the enrichment functions 'Exp trans diff' based on Equation 4.11. However, the value of the neighboring element is still used, because, as becomes apparent, it still represents a useful approximation. In any case, the significance of the initial energy is much smaller from the second element onwards in the considered case of temperature propagation as a result of the boundary condition. The enrichment starts when the element is reached by the temperature change. Therefore, the temperature difference of the two nodes for scaling of the enrichment function is already small and ensures a low energy in the element.

Similar to the procedure in the previous subsection, the results are first displayed graphically (Figure 4.12). Then, the method itself is examined with regard to the optimal parameterization variable. The results are subsequently evaluated with regard to the analytical solution. At the end, the undershoots due to non-compliance with the PDC are examined.

The largest deviations from the analytical solution result from the use of the FEM (cf. Figure 4.12). The PXFEM with the enrichment functions 'Exp trans diff' and 'Erf' demonstrates similar deviations. With increasing time steps, the results of the two variants are hardly graphically distinguishable from the analytical solution.

Considering the obtained values of the parameterization variables compared to the optimal values in Figure 4.12, there are only minor deviations. The optimal values are determined using the procedure described in Subsection 4.3.2. The PXFEM with the enrichment function 'Erf' has a high correspondence between the parameterization variables of the first and second element and the optimal values (cf. Figure 4.13b). The values of the third element deviate slightly, but still show a qualitative linear progression.

First, for the enrichment function 'Exp trans diff', the linear progression of the optimal values (cf. Figure 4.13a) and the associated subsequent legitimation of the use of substitution in Equation 4.18 is to be pointed out. This applies equally to the optimal values of all enriched elements. The parameterization variables of elements two and three show a clearly better agreement. In contrast, the value of the parameterization variable $\zeta_{1,2}$ deviates, albeit only slightly,



(a) Time step 1, $t = 0.5\,\mathrm{s}$, enriched elements: 2.

(b) Time step 2, $t = 1\,\mathrm{s}$, enriched elements: 2.

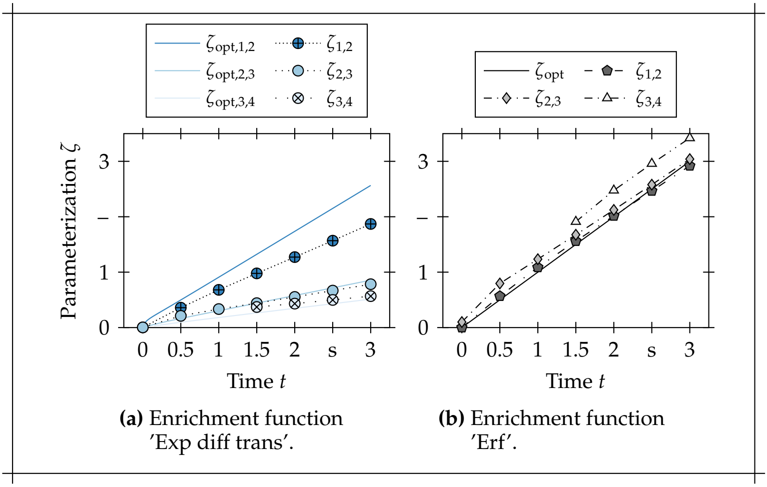
(c) Time step 3, $t = 1.5\,\mathrm{s}$, enriched elements: 3.

(d) Time step 4, $t = 2\,\mathrm{s}$, enriched elements: 3.

(e) Time step 5, $t = 2.5\,\mathrm{s}$, enriched elements: 3.

(f) Time step 6, $t = 3\,\mathrm{s}$, enriched elements: 3.

Figure 4.12: Results of the FEM and the PXFEM with the enrichment functions 'Exp trans diff' and 'Erf', as well as the analytical solution, displayed for the first five nodes; $T_{\Gamma_\mathrm{D}} = 1523.15\,\mathrm{K}$, $T^0 = 353.15\,\mathrm{K}$, $a = 5.18 \times 10^{-6}\,\mathrm{m}^2\,\mathrm{s}^{-1}$, $d_\Theta = 5\,\mathrm{mm}$.



(a) Enrichment function 'Exp diff trans'.

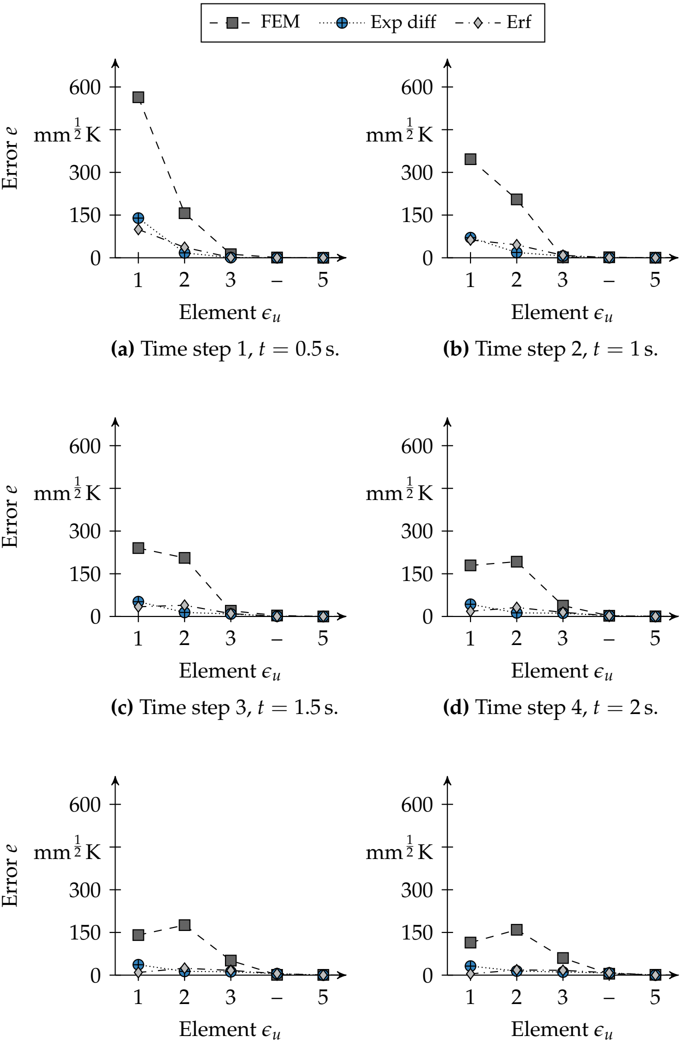
(b) Enrichment function 'Erf'.

Figure 4.13: Optimal values of the parameterization variable and values obtained from the PXFEM in Figure 4.12.

with increasing time steps. The determination of the optimal parameterization values is based on the assumption of correct nodal values. Since the nodal values with the PXFEM are not necessarily correct, the actual optimal values of the parameterization variables vary from the determined ones. In addition, an increase in the parameterization variable is accompanied by an approach of the enrichment function to the zero function. A deviation from later values is therefore less significant and does not mean an increase in error, as the increasing deviation wrongly implies. This is additionally demonstrated by the errors compared to the analytical solution in each element for the different time steps illustrated in Figure 4.14.

In the error diagrams of the FEM, the thermal progression is reflected. First, the largest errors occur in the first element. With increasing time steps, the second element is more strongly affected by the temperature change, and ultimately the error there even prevails. In addition, the temperature increase in the third element shows up from the third step in the increasing error there. The PXFEM with the enrichment functions 'Erf' and 'Exp trans diff' does not feature this. A similar and constantly low error level is apparent for the two variants, which is always at least three times lower than for the FEM.

A better representation of the temperature gradient in the first element is achieved with the PXFEM and the enrichment function 'Erf', as illustrated by the logarithmic representation of the errors per elements in Figure 4.15. In



(a) Time step 1, $t = 0.5\,\mathrm{s}$.

(b) Time step 2, $t = 1\,\mathrm{s}$.

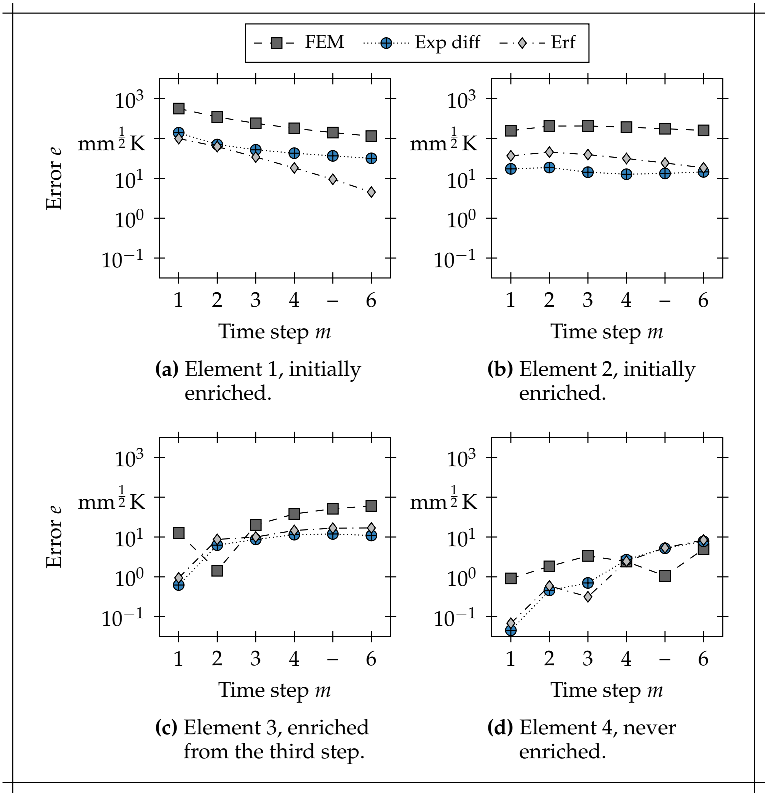
(c) Time step 3, $t = 1.5\,\mathrm{s}$.

(d) Time step 4, $t = 2\,\mathrm{s}$.

(e) Time step 5, $t = 2.5\,\mathrm{s}$.

(f) Time step 6, $t = 3\,\mathrm{s}$.

Figure 4.14: Error per step of the FEM and the PXFEM with various enrichment functions from Figure 4.12, illustrated for the first five elements.



(a) Element 1, initially enriched.

(b) Element 2, initially enriched.

(c) Element 3, enriched from the third step.

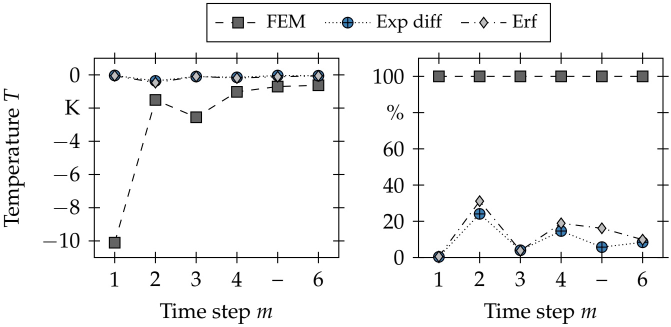
(d) Element 4, never enriched.

Figure 4.15: Error development of the individual elements over the time steps for the FEM and the PXFEM with various enrichment functions from Figure 4.12.

the second element, however, the enrichment function 'Exp trans diff' achieves the smallest errors. The error of the non-initially enriched element 3 first increases (cf. Figure 4.15c). From the moment of enrichment in time step 3, the error in the third element remains constant. At the end of the observation period, the temperature changes shortly before reaching the fourth element. The error reaches a level that would require enrichment in the next step to avoid a larger error than in the previous element (cf. Figure 4.15d).

In general, the errors in each element when using the FEM are higher than those of the PXFEM. However, more precise results are obtained in individual time steps. These only occur at times when the respective elements are not yet affected by the temperature change according to Equation 2.33. The results are therefore traced back to the undershoots that occur, which initially result in an excessively low temperature, but as the heat flux increases, they simply rise to the correct nodal values and lead to a minor error.

The undershoots (Figure 4.16) that occur are smaller than in the case of only one layer affected by the temperature change (cf. Figure 4.11). The reason for this is the lower violation of the PDC (cf. Equation 3.1) as a result of the larger time steps used. Both considered variants of enrichment functions of the PXFEM have a nearly identical progression with a maximum undershoot of −0.47 K. Thus, the undershoots are clearly smaller than those of the FEM, which in the first step even have a value of −10.11 K at most.



(a) Maximum nodal deviation below initial temperature.

(b) Relative maximal nodal deviation in relation to the FEM.

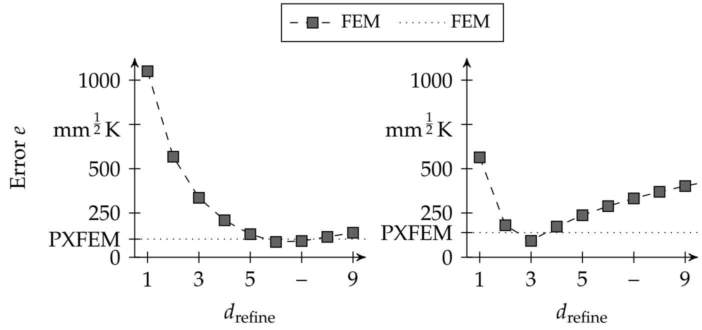
Figure 4.16: Nodal results below initial temperature from Figure 4.12.

4.4 Validation

The previous section has illustrated that the PXFEM provides correct results. In this section, the PXFEM is now validated with regard to the reduced effort. A comparison of the calculation times of PXFEM and FEM is not carried out, because the meaningfulness is low due to the small number of DoFs of the considered simplified one-dimensional heat equation. Since the method achieves even higher accuracies than the FEM, the number of DoFs necessary for the FEM is compared to obtain a similar accuracy.

Starting from the spatial discretization used in Section 4.3, the element length of the FEM is uniformly reduced by the increasing natural number d_{refine} in the entire area, until, relative to the first element of the initial configuration, a smaller maximal error e for all time steps is obtained, as with the PXFEM. The maximum error in the first element is, considering that only the first layer is affected by a temperature change, $102.4\,\sqrt{\text{m}}\text{K}$ for the PXFEM with the enrichment function 'Exp trans diff' and $8.9\,\sqrt{\text{m}}\text{K}$ with the enrichment function 'Erf'. To achieve a smaller error with the FEM than with the variant 'Exp trans diff', the FEM needs an element length smaller by a factor of $d_{\text{refine}} = 6$. That means it requires 5 times more DoFs than the PXFEM and the enrichment function to obtain the same low error.

The error of the enrichment function 'Erf' is not achievable with the FEM by simply reducing the element length. The error consists of a spatial and a temporal portion. With the consideration of the gradient within the first element, only the parameterization variable changes in the observation period when using the PXFEM. The nodal temperatures are not affected. Due to the linear progression of the parameterization variable, the numerical integration in time with the backward Euler method hardly produces any temporal error components. In the FEM, the finer local discretization increases the number of DoFs affected by the temperature change. Their temporal progression is nonlinear, and additional errors occur due to the numerical integrator. Therefore, the error is not to be arbitrarily reduced by local refinement alone, as illustrated by the representation of the error e as a function of the factor d_{refine} in Figure 4.17a. To reach the level of PXFEM with the enrichment function 'Erf' by the FEM, an additional refinement of the temporal discretization would be necessary.



(a) $t^{m_{\text{Y}}} = 0.36\,\text{s}$, $m_{\text{Y}} = 4$.

(b) $t^{m_{\text{Y}}} = 3\,\text{s}$, $m_{\text{Y}} = 6$.

Figure 4.17: Maximal appearing error of the FEM in relation to the spatial refinement; $T_{\Gamma_{\text{D}}} = 1523.15\,\text{K}$, $T^0 = 353.15\,\text{K}$, $a = 5.18 \times 10^{-6}\,\text{m}^2\,\text{s}^{-1}$, $d_\Theta = d_{\text{refine}}5\,\text{mm}$.

For the same reason, no arbitrarily small errors are obtained by a finer discretization (cf. Figure 4.17b) in the case of several layers affected by the temperature change. However, the errors of the PXFEM with 139.4 $\sqrt{\mathrm{m}}$K for the enrichment 'Exp trans diff' and 99.6 $\sqrt{\mathrm{m}}$K for the enrichment 'Erf' are higher and achievable by a finer spatial discretization in the FEM. Both values are reached at a refinement with a factor of 3; i. e., in this case, twice the number of DoFs is required for a similar accuracy.

4.5 Summary

If the PDC is violated, the FEM leads to oscillation, since the spatial discretization is too coarse to represent the gradient with the shape functions. The PXFEM enriches element by element and compensates for the inadequacies of the shape functions. The parameterization variable allows the calculation of the transient course with one additional DoF per element. If only one layer is affected by the temperature change, the method reduces the unphysical undershoots to 4.2 % compared to the FEM. The error for the entire element is decreased by a factor of 10.

A procedure for determining enrichment functions has been presented. However, the investigations revealed that the temporal progression of the parameterization variable has a large influence on the accuracy of the solution. The method has yet to be investigated for nonlinear material parameters and in the three-dimensional case. In practice, efficient strategies for implementation must be found. Due to the non-linearity of the parameterization variable, a linearization with the previous procedure is no longer possible.

Not yet considered is also the theory of convergence. By choosing the test function identical to the enrichment function including the parameterization variable, the solution space becomes infinite in principle. The convergence theory of the FEM is based on the finite dimension of the solution space.

5 LBM Multirate Method for Layer-Specific Time Steps

5.1 Overview

This chapter addresses a procedure for the temporal discretization of the already spatially discretized heat equation

$$\boldsymbol{C}(T) \cdot \dot{\boldsymbol{T}}(t) + [\boldsymbol{K}(T) + \boldsymbol{K}_C(T)] \cdot \boldsymbol{T}(t) - \boldsymbol{q}(t) = 0, \qquad \forall t \in \Upsilon \tag{2.62}$$

as used for the ’Cooling’ phases in the simulation of the LBM buildup process (cf. Figure 5.1). As a result of high temperature changes during the process, strong requirements apply to the time step size and consequently lead to long calculation times. Although these requirements only apply locally, the required small time steps are conventionally used for the complete computational domain. The multirate method of GEAR & WELLS (1984) divides the calculation domain into several components in which different local time steps are used. The solution effort is reduced accordingly, as larger time steps are used for areas with lower requirements. However, the method as derived by GEAR & WELLS (1984) is only applicable for constant calculation areas. In addition, the local requirements in the simulation of LBM are generally unknown.

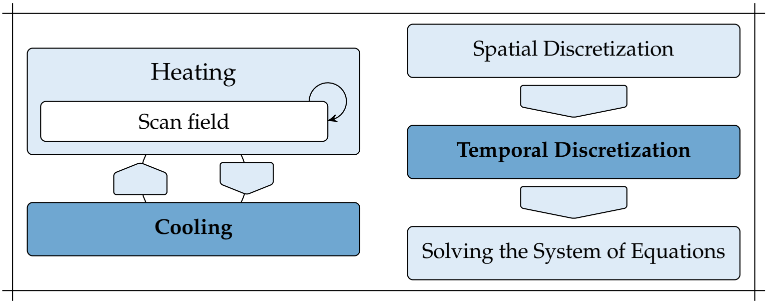


Figure 5.1: Classification of Chapter 5 with regard to the phase of the Mathematical Model of LBM and application area in numerics; field of actions (dark blue).

An LBM Multirate Method (LBM-MM) is proposed, which extends the method from GEAR & WELLS (1984) to a growing calculation domain. Both the directed, decreasing heat flux towards the heating plate and the layered structure of meshes used in the simulation of LBM are taken advantage of in this method. The set of layers is dynamically partitioned into components, taking into account the addition of layers during the buildup process.

First, the LBM-MM is presented. It is then verified by the solution of the simulation without the method, and a permissible parameter window is determined for the additionally introduced parameters. This is done both empirically, by evaluating various simulation runs with the LBM-MM implemented in ccx (BANTE et al. 2019) and the input script program *ascent*[31], and theoretically, to increase understanding of the process. Afterwards, the permissible parameter combinations are validated with regard to the calculation time required in ccx, i. e. the simulation is faster than without the developed method. For both verification and validation, the initial simulation without the LBM-MM is used as a reference.

5.2 Methodology

The energy input takes place in the uppermost layer. Accordingly, the strictest requirements for the time steps apply in the layers near the build surface. In deeper layers, lower temperature changes occur and longer time steps would be applicable to meet the same accuracy requirements as in the upper layers as illustrated in Figure 5.2. The LBM-MM clusters the layers with similar requirements

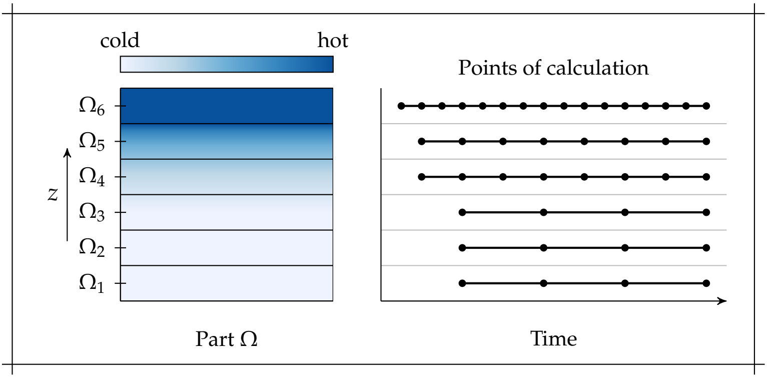


Figure 5.2: Exemplary illustration for sufficient time steps of the respective layers to meet the same accuracy requirements.

[31] https://gitlab.lrz.de/flayerbein/ascent.git

into coupled components and uses component-specific time steps.

In accordance with the layers, the discrete set of nodes Ξ is gradually partitioned into the number of h_{mr} coupled subsystems Φ_h with $1 \leq h \leq h_{\mathrm{mr}}$ during the simulation. Starting from the current top layer, the nodes of adjacent activated layers are dynamically combined into components. The build plate is assigned to the lowest layer. The component Φ_h consists of up to the number of w_h layers, except for the last component $\Phi_{h_{\mathrm{mr}}}$, which includes all remaining active layers. The allocation for a component Φ_h starts when the previous subsystem Φ_{h-1} has reached its designated number of w_{h-1} layers, or, in the case of Φ_1, from the beginning of the simulation.

Alternatively to the number of layers per component w_h, the number of layers c_h from the current top layer prior to the first layer of the component Φ_h is specified. The connection

$$c_h = \begin{cases} 0, & h = 0, \\ \sum_{s=1}^{h-1} w_s, & \text{else} \end{cases} \tag{5.1}$$

applies.

Instead of calculating the complete system, the components are only calculated after a certain number of time steps with the LBM-MM. The lower declaration h represents faster subsystems as the temporal temperature change decreases towards the heating plate. Accordingly, the interval of skipped time steps increases from one component to the next higher one. To avoid unnecessary interpolations, all components from Φ_1 to Φ_{h-1} are also calculated if the component Φ_h is calculated. This means that the calculation interval for a slower component Φ_h only takes a whole-numbered multiple value k_h of the calculation interval k_{h-1} of the previous component Φ_{h-1}, similar to the original method proposed by GEAR & WELLS (1984). The index set I^{t^m} contains the indexes of the calculated components at time t^m. The initial calculation of the component Φ_h takes place at the k_h-th time step after the first layer is allocated to Φ_h. In consequence, the local time step Δt_h^m for the component Φ_h at time t^m is determined by the sum of all skipped time steps, i. e.

$$\Delta t_h^m = \sum_{s=1}^{k_h} \Delta t_{m-s+1}. \tag{5.2}$$

The component Φ_1 is calculated for every time step, i. e. $k_1 = 1$, since it includes the top layer with the highest existing gradient, and the time step Δt^m is determined accordingly. An illustration of the calculation procedure and the associated dynamic partitioning is provided in Figure 5.3. As the figure implies, a fastest-first approach is applied, because, comparable to a propagating temperature wave, the faster components must first transport the change in

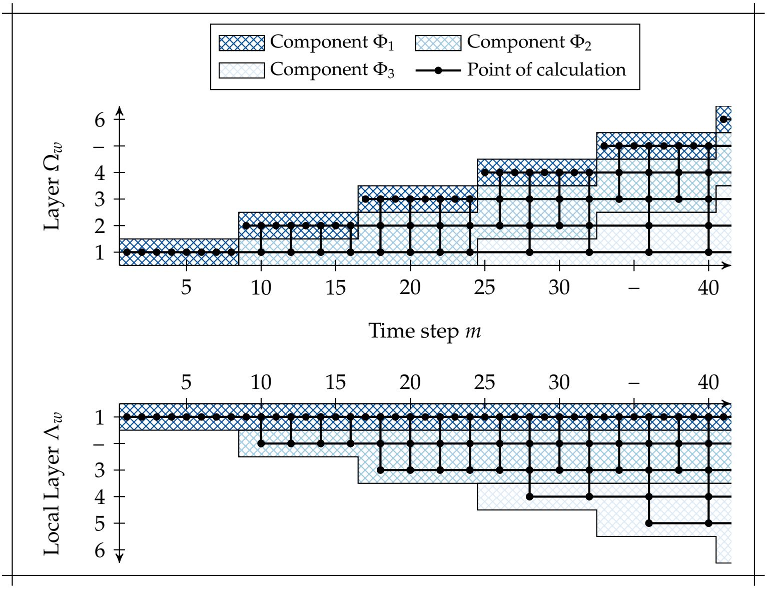


Figure 5.3: Dynamic subdivision of the layers into components during the buildup process according to the parameters $h_{\text{mr}} = 3$, $w_1 = 1$, $w_2 = 2$, $k_2 = 2$, $k_3 = 4$; a knot represents the calculation of a layer in a time step.

temperature to the slower, subjacent components before a calculation of these is even necessary. Additionally, the fastest-first method is most appropriate for known time step sizes (GEAR & WELLS 1984).

In terms of equations, the procedure described above leads, at time t^m, to the following system of equations for all components Φ_h with $h \in I^{t^m}$:

$$\left[\frac{1}{\Delta t_h^m}\boldsymbol{C}_h(\boldsymbol{T}^m) + \boldsymbol{K}_h(\boldsymbol{T}^m)\right] \cdot \boldsymbol{T}^m = \frac{1}{\Delta t_h^m}\boldsymbol{C}_h(\boldsymbol{T}^m) \cdot \boldsymbol{T}^{m-1}, \tag{5.3}$$

where $\boldsymbol{C}_h(\boldsymbol{T}^m)$ and $\boldsymbol{K}_h(\boldsymbol{T}^m)$ represent the rows of the matrices $\boldsymbol{C}(\boldsymbol{T}^m)$ and $\boldsymbol{K}(\boldsymbol{T}^m)$ corresponding to the nodes of Φ_h. The domain of calculation is thus reduced in every time step m where $\left|I^{t^m}\right| =: h_m \neq h_{\text{mr}}$. This results in a new boundary $\Gamma_\Phi(t^m)$ between Φ_{h_m} and Φ_{h_m+1} for those time steps. The boundary $\Gamma_\Phi(t^m)$ represents the border to components with lower temperature changes with respect to the considered time interval. Therefore, the state of the nodal temperature values for the components Φ_h with $h \notin I^{t^m}$ is regarded as constant. In addition to the resulting Dirichlet boundary conditions

$$\boldsymbol{T}(\boldsymbol{r}_n, t^m) = \boldsymbol{T}\left(\boldsymbol{r}_n, t^{m-1}\right), \qquad \forall \boldsymbol{r}_n \in \Gamma_\Phi(t^m) \tag{5.4}$$

such as in GEAR & WELLS (1984), the LBM-MM also allows for the use of adiabatic boundary conditions

$$-\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{n}}(\boldsymbol{r}_n, t^m) = 0, \qquad \forall \boldsymbol{r}_n \in \Gamma_\Phi(t^m)\,. \tag{5.5}$$

This offers the nodes of the boundary some amount of flexibility, and they are able to react to the temperature changes. The following sections provide information as to what extent this offers added value.

5.3 Verification

The proposed LBM-MM periodically reduces the system of equations to decrease the calculation effort and, in consequence, speeds up the simulation. However, the additional parameters introduced by the method influence the simulation's solution. These include the number of components h_{mr}, the accumulated layers per component w_h, the component-specific calculation interval k_h and the type of used boundary condition. Therefore, permissible parameter combinations are determined in this subsection which affect the simulation results only within a tolerated threshold.

Starting with its activation, each layer experiences a similar thermal history of attenuated reheating due to the activation of the following layers. Accordingly, every layer passes through similar decreasing requirements for the time step sizes over the buildup process. Therefore, only component-specific calculation intervals k_h are used, which are equal to a divisor of the total number of time steps per cooling cycle m_{Υ}. This ensures that, starting from its activation, each layer is calculated at the same corresponding time steps. Otherwise, a shift in the used time steps beyond the cooling cycle occurs for all layers of the component Φ_h. In this case, there are several shifts for the different components corresponding to the lowest common multiple of k_h, and the total number of time steps per cooling cycle m_{Υ} appears. The overall valuation of the parameter combination corresponds to the worst appearing selection of time steps. For this reason, $64 = 2^6$ cooling steps are used to allow a maximum flexibility for the LBM-MM, i. e. the values 2, 4, 8, 16, 32 and 64 are allowed for k_h.

The time steps were determined according to the procedure presented in Section 3.2.3 with the material Inconel 718, a dwell time $\Delta t_{\mathrm{dwell}}$ of 14 s and a reference layer height d_{l} of 20 µm. To ensure that the gradients in the upper temperature range are well discretized, a slightly higher constant, efficient thermal diffusivity a of $3.48 \times 10^{-6}\,\mathrm{m}^2\,\mathrm{s}^{-1}$ as compared to that of KRAUSS (2016) was applied. These parameters correspond to an EOS M270 manufacturing system and result in a global uncertainty κ of 1.45 K. However, without loss of generality, the following verification procedure can be repeated for other materials and manufacturing systems.

The LBM-MM addresses the cooling cycle of each layer, and more precisely the heat flux towards the base plate and the associated temperature changes in the z-direction. For verification and subsequent validation, the heating is neglected, and only one scan field is used for each layer. All layers are initially set to solidus temperature. Upon activation of each layer, the heat flux occurs as a result of the deposited energy. This corresponds to the config option *ThermalMode=cooling* in the ascent program[32]. Heating is thus neglected; however, it takes place on a significantly shorter time scale than the cooling behavior (KRAUSS 2016). Nevertheless, the verification and validation procedure is also valid for several scan fields. In that case, the highest gradients during 'Cooling' result from the last scan field. The gradients of the simultaneous heat input on the whole layer in the z-direction approximately correspond to these, and even represent an upper limit. Since there is also a heat flux in the xy-plane compared to the simultaneous energy input into the entire layer, the relevant temperature change in the z-direction decreases even faster.

5.3.1 Local uncertainty

Stability and consistency are sufficient for convergence (cf. Section 2.5.3). These two properties mainly depend on the used numerical integrator for the LBM-MM. In the case of the backward Euler, the LBM-MM is convergent, as both properties are inherited by the numerical integrator. The stability region is the entire left half of the complex plane, as the backward Euler is A-stable/L-stable (and unconditionally stable for the heat equation) (DEUFLHARD & BORNEMANN 2002). Therefore, it is stable for every step length and thus in particular for the LBM-MM, which is stable for arbitrarily accumulated time steps. For other numerical integrators, the time steps for the components may exceed the allowed increment. The backward Euler is consistent of order 1 (DEUFLHARD & BORNEMANN 2002). This also holds for the LBM-MM with locally extended time steps, due to the definition of consistency. However, the local uncertainty κ^m for certain time steps m might change through the application of the LBM-MM and influence the global uncertainty κ. For a detailed discussion, the uncertainty κ_w^m

[32] `https://gitlab.lrz.de/flayerbein/ascent.git`

is introduced, which, in addition to the time step m, also distinguishes the layer Λ_w, i. e.

$$\kappa_w^m = \frac{(\Delta t^m)^2}{2} \max_{\substack{t \in \left(t^m, t^{m+1}\right) \\ \boldsymbol{r} \in \Lambda_w}} \left| \frac{\partial^2 T}{\partial t^2} (t, \boldsymbol{r}) \right| . \tag{5.6}$$

The following correlation between the local uncertainties κ^m and κ_w^m applies:

$$\kappa^m = \max_w \kappa_w^m . \tag{5.7}$$

As with the determination of the time steps by means of the local uncertainty κ^m, the one-dimensional analytical solution from Equation 2.25 is used as an approximate representation of the cooling behavior in LBM for the investigation of the local uncertainty κ_w^m. The distribution of κ_w^m for the 64 time steps used is illustrated in Figure 5.4. The highest uncertainty is always located in layer Λ_1 for each time step m. This makes it clear once again that the top layer specifies the time step size throughout the entire 'Cooling' phase. Towards the end of the cooling cycle, the uncertainties in all layers approach the value of the top layer. During the cooling cycle, all uncertainties undergo two local maxima with a global maximum at the end. These result from the reheating of deeper layers due to the energy input in the first layer. The first maximum arises from the initial increase in temperature. Once warming finally decreases, the second

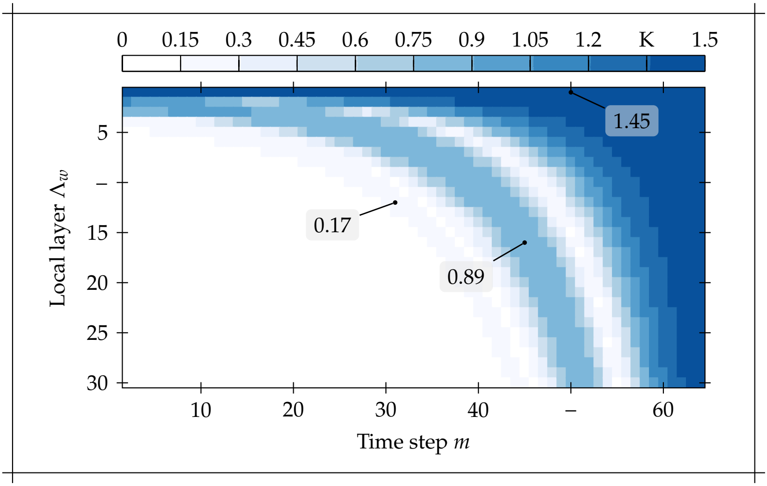


Figure 5.4: Distribution of the local uncertainty κ_w^m for equally distributed κ^m during a cooling cycle; $m_\Upsilon = 64$ and $w_\Psi = 30$.

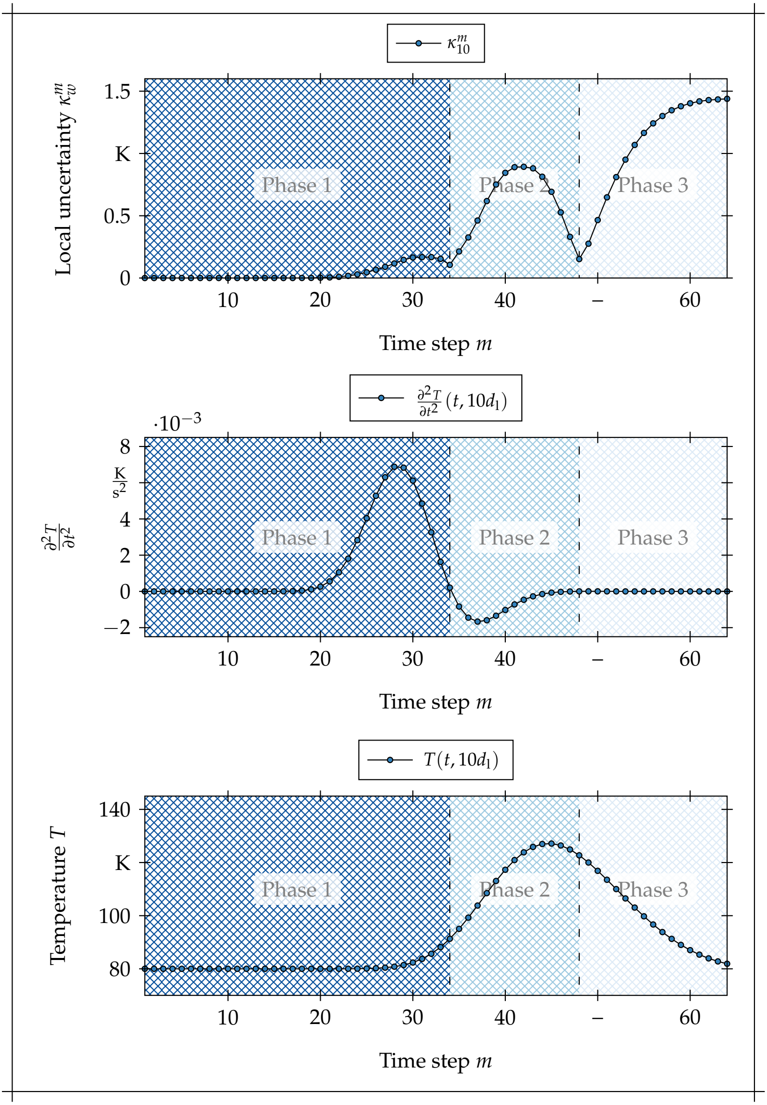


Figure 5.5: Relationship between maxima of the local uncertainty κ_w^m and the temperature field evolution for $z = 10d_1$.

maximum occurs. Re-approaching the preheating temperature again at the end leads to the last maximum. The three phases are illustrated in Figure 5.5.

Exceptions to the behavior previously described are the layers which have already experienced a temperature change with the first time step. The first maximum is omitted here. Due to the stability conditions described in Section 3.2.2, minimum length requirements apply for the first time step by the PDC from Equation 3.1. A temporal discretization of the first maximum for these layers in the simulation is therefore not possible.

Quantitatively, the respective maxima in Figure 5.5a are determined with the same uncertainty in each layer. Thus, the different phases are always calculated with the same accuracy, regardless of the position of the layer. The reason for this is the determination of the time step size by minimizing the uncertainty κ. The procedure regulates the different maximum values in the second derivative over the time step size to obtain the same local uncertainty κ^m for all time steps. All uncertainties κ^m are thus normalized to a common value by the time step sizes Δt^m. An optimization is necessary to determine this value. At this point, however, it is only relevant that the procedure can be simplified as the normalization of the maximum value that occurs in the second derivation. As noted, the highest value always occurs in the first layer. Moreover, Figure 3.1b has revealed that this is always on top of the first layer, with the exception of the first time step. The determination of the local uncertainty is thus formally specified as the maximum norm of an auxiliary function

$$f_{\text{aux}}(t^m, z) = \kappa^m \frac{\dfrac{\partial^2 T}{\partial t^2}(t^m, z)}{\dfrac{\partial^2 T}{\partial t^2}(t^m, 0)}. \tag{5.8}$$

The solutions of the linear heat equation at different times are transferable to each other due to self-similarity (cf. Equation 2.30). Through the normalization with the value in the origin, the time step scales only the function argument z without directly affecting the function value; i. e., it is equivalent to

$$f_{\text{aux}}(t^m, z') = \kappa^m \hat{f}_{\text{aux}}\left(\frac{z'}{t^m}\right) \tag{5.9}$$

with a function $\hat{f}_{\text{aux}}(z)$ normalized to one at $z = 0$. As a consequence, the values of the maxima, i. e. ≈0.17 K, ≈0.89 K and ≈1.45 K, remain identical and only shift locally. These values also reveal that the initial increase in the temperature (cf. Figure 5.5c), which leads to the first maximum, is best temporally discretized (cf. Figure 5.5a), although it corresponds to the largest values in the second derivation (cf. Figure 5.5b). The procedure for determining the time steps therefore leads to the most accurate display of the major changes at the beginning.

The LBM-MM combines several time steps for certain components Φ_h. In the layers Λ_w of these components, for a time step m in which the component is calculated, the deviating local uncertainty $\hat{\kappa}_w^m$ compared to Equation 5.6 is as follows:

$$\hat{\kappa}_w^m = \frac{\left(\Delta t_h^m\right)^2}{2} \max_{\substack{t \in \left(t^{m-k_h+1}, t^{m+1}\right) \\ \boldsymbol{r} \in \Lambda_w}} \left| \frac{\partial^2 T}{\partial t^2} (t, \boldsymbol{r}) \right| . \tag{5.10}$$

The time step size increases and the domain for the maximal value extends by the application of the LBM-MM. This leads to higher uncertainties in the corresponding components.

The local time step Δt_h^m results from the sum of the preceding k_h time steps (cf. Equation 5.2). Assuming an upper growth rate c_Υ for the time step sizes Δt^m of Υ, i. e.

$$\Delta t^m \leq c_\Upsilon \Delta t^{m+1}, \tag{5.11}$$

the estimate

$$\Delta t_h^m < \Delta t^m \sum_{s=1}^{k_h} {c_\Upsilon}^{s-1} \tag{5.12}$$

holds for the local time step size Δt_h^m. The observed constant maxima across the layers in Figure 5.4 demonstrate that, throughout the procedure to determine the time step size, the level of uncertainty κ_w^m is independent of the layer Λ_w. Rather, an evaluation with regard to the different phases is advisable. Assuming a maximum in κ_w^m and combining Equation 5.12 with Equation 5.10, the estimate

$$\hat{\kappa}_w^m < \left(\sum_{s=1}^{k_2} {c_\Upsilon}^{s-1} \right)^2 \kappa_w^m = c_\kappa \kappa_w^m \tag{5.13}$$

is obtained. The introduced parameter c_κ thus indicates the change in uncertainty due to the application of the LBM-MM. Figure 5.6 shows the growth rates for each time step. The last occurrences of the first and second maxima is in the time steps 44 and 51, respectively (cf. Figure 5.4). Accordingly, the maximum growth rates of 1.25 and 1.36 apply for phase 1 and 2, respectively. Correspondingly, the application of the LBM-MM results in a change in uncertainty by the factors listed in Table 5.1. With a global uncertainty κ of ≈1.45 K and a local uncertainty of ≈0.17 K for the first maximum, there is therefore no increase in global uncertainty by the parameters $k_2 = 2$ and $k_2 = 4$. At the second maximum, the LBM-MM directly leads to an increase in global uncertainty, regardless of the choice of parameters k_2 and w_1. Under tolerance of a slightly increasing global uncertainty (5.57 K), the LBM-MM with $k_2 = 2$ is also applicable to the appearance of the second maximum. Figure 5.4 already illustrates that the last maximum corresponds to the global uncertainty since there are almost no differences in the local uncertainties between the individual

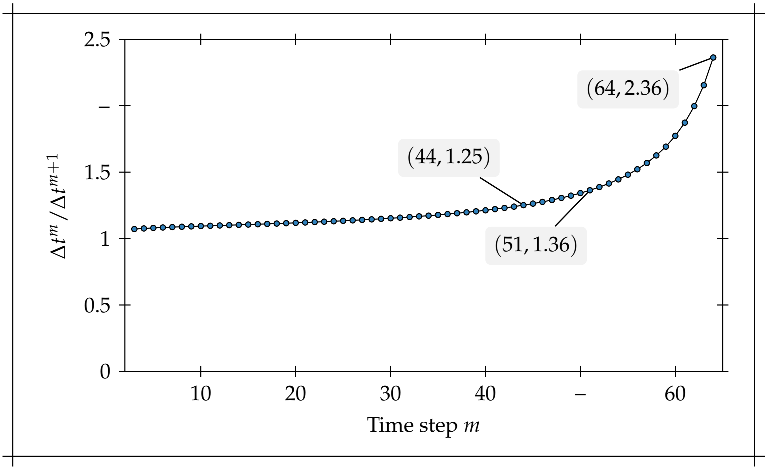


Figure 5.6: Growth rates of the time step sizes Δt^m.

layers. Thus, even small changes in the local uncertainty lead to an increase in the global uncertainty.

Since the different maxima are not decoupled in time, they spread simultaneously over the layers, albeit with a spatial offset. The layers required in the first component for the isolated observation of the individual maxima are therefore dependent on this time-varying offset and are independent of the value k_2. In addition, the procedure for determining the time steps results in the individual maxima not being damped across the layers. With regard to uncertainty, it is therefore not possible, for the reasons given, to specify a generally valid number w_1 of layers for the first component throughout an entire cooling cycle. In the case of the first maximum, up to 23 layers as part of the first component (i. e. $w_1 = 23$) are required to allow for an isolated consideration during its entire occurrence by the LBM-MM (cf. Figure 5.4). For the second maximum, the condition of up to 9 layers for the first component (i. e. $w_1 = 9$) applies to avoid the additional influence of the last maximum on the uncertainty.

Valid parameters for the LBM-MM were found under theoretical considerations. However, restrictions must be made on their area of validity, especially with regard to later time steps of the cooling cycle. The underlying analytical solution represents the energy deposition in a semi-infinite beam. The incipient influence of the preheating that occurs during LBM in the propagation of heat is thus not taken into account. Furthermore, the uncertainty is only an upper limit of the actual error, in the derivation of which the supremum norm is used as an estimate (cf. Equation 3.10). For functions with small changes in the interval

Table 5.1: The factor c_κ and the changed local uncertainty $\hat{\kappa}_w^m$ for different values k_2 in Phase 1 and 2.

Phase 1	k_2	2	4	8
	c_κ	5.06	33.24	393.70
	$\hat{\kappa}_w^m$	0.86 K	5.65 K	66.93 K
	k_2	16	32	64
	c_κ	1.91×10^4	2.54×10^7	4.06×10^{13}
	$\hat{\kappa}_w^m$	3.24×10^3 K	4.32×10^6 K	6.90×10^{12} K
Phase 2	k_2	2	4	8
	c_κ	5.57	45.23	883.97
	$\hat{\kappa}_w^m$	4.96 K	40.25 K	786.73 K
	k_2	16	32	64
	c_κ	1.43×10^5	2.72×10^9	9.56×10^{17}
	$\hat{\kappa}_w^m$	1.27×10^5 K	2.42×10^9 K	8.51×10^{17} K

under consideration, this estimate provides reliable values. The changes in the second derivation weaken with increasing time. However, with regard to the actual points in time t^m, even larger changes occur in the second derivative towards the end of the cooling cycle. Figure 5.7 illustrates the change of the second derivative in the area of the first layer with regard to the time steps. The local uncertainty κ_w^m thus represents an increasingly rough estimate of the error with rising m. By quadratic consideration of the time step size in the local uncertainty κ_w^m (cf. Equation 5.6), this effect is even intensified when using the LBM-MM and the associated combination of time steps (cf. Equation 5.10). This also explains the extremely high values of the local uncertainty $\hat{\kappa}_w^m$ in Table 5.1. However, Figure 5.7 reveals that the use of the maximum norm is a good estimate for the initial and middle areas of the cooling cycle. Thus, the use of larger time steps is permissible, especially for the first and second phase.

The uncertainty analysis only verifies the time step size, but not the temperature values, since it does not consider the additional boundary condition when skipping the time steps. Only the second temporal derivation is used. Accordingly, the results do not differentiate the number of components, since the mutual influence is not mapped in the theoretical consideration. Therefore, for a complete verification, it is still necessary to consider the occurring temperature deviation by application of the LBM-MM.

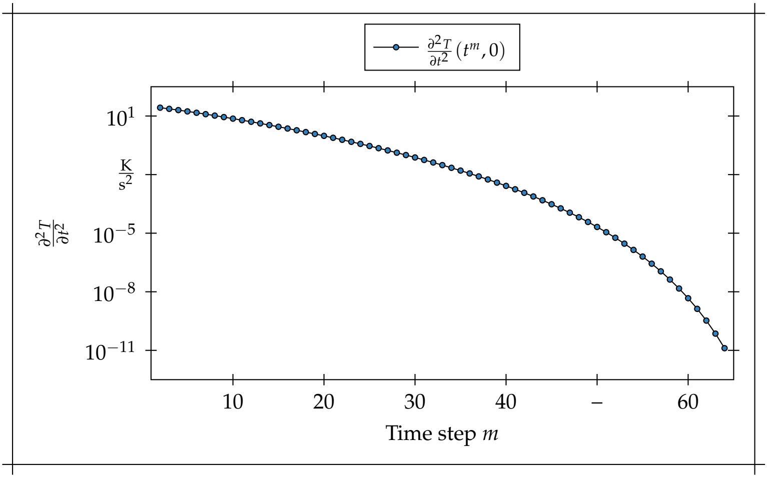
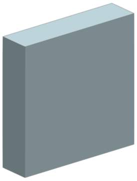


Figure 5.7: Second derivative of the temperature T with respect to t.

5.3.2 Two components

The source code of the open source FEM software ccx release 2.14 was extended by the LBM-MM within in this thesis project. The modified source code is available at the media and publications repository of the Technical University of Munich (BANTE et al. 2019). Internal routines were used to a large extent. Among other things, ccx already has an internal function for deactivating nodes. Before each increment, a loop over all components determines the nodes for the system of equations to be solved according to the input parameters of the LBM-MM. In addition, the new boundary conditions and the local time steps for the respective components are set. The conventional routines and solvers of ccx are used for solving the system of equations. With regard to the validation of the calculation time in Section 5.4, it should be explicitly mentioned here that the new routines are only called on when the corresponding parameters for the method are passed. Simulation runs, in which the LBM-MM is not used, are not affected by the extensions, with the exception of a negligible query at the beginning.

First, only a cuboid structure and two components $h_{\mathrm{mr}} = 2$ are considered: i. e., the cumulated layers of the first component w_1 (equivalent to the number of layers c_2 before the second component starts), the component-specific calculation interval of the second component k_2 and the kind of used boundary condition are varied. The results are later transferred to the usage of further components. The cuboid structure is placed centrally on the base plate. Further information on the dimensions of the geometry and the finite element mesh are listed in Figure 5.8.



(a) Part

Geometry $w \times d \times h$ in mm	
part	$40 \times 10 \times 40$
build plate	$250 \times 250 \times 40$
Mesh	
element size, part	2 mm
element size, base plate	30 mm
height of the LCs	2 mm
elements	35 683
nodes	8069
layer compounds	20

(b) Geometry and mesh information

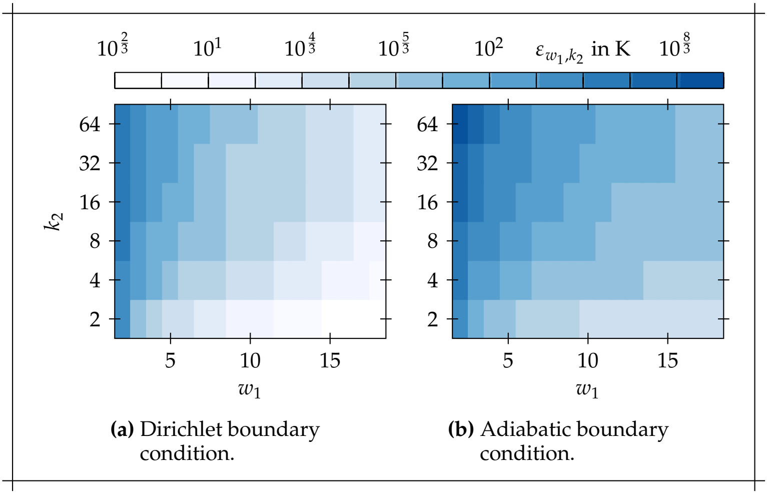
Figure 5.8: Configuration of the benchmark geometry: Cuboid.

The simulation result T_0 obtained without the LBM-MM serves as a reference for the results T_{w_1,k_2} of the parameters w_1 and k_2. Each parameter combination is evaluated with regard to the maximal absolute appearing nodal temperature deviations per cooling cycle ε_{w_1,k_2} averaged over the number of layers w_Ψ, i. e.

$$\varepsilon_{w_1,k_2} = \sum_{s=1}^{w_\Psi} \frac{\max_{t \in \Upsilon, \boldsymbol{r} \in \Omega} \left(\left\| T_0 - T_{w_1,k_2} \right\| \right)}{w_\Psi}. \tag{5.14}$$

This ensures that the maximum temperature deviations are taken into account, while considering the changing thermodynamics of the LBM processes throughout the buildup process. The different values ε_{w_1,k_2} for all possible parameter combinations for the two component cases are illustrated and color-coded in Figure 5.9.

The results indicate a similar qualitative progression for both types of boundary conditions. The deviation ε_{w_1,k_2} corresponds indirectly proportional to the number of cumulated layers w_1 and directly proportional to the component-specific calculation interval k_2. An increasing number of layers w_1 results in lower temperature changes reaching the second component. Accordingly, the deviation ε_{w_1,k_2} decreases. However, the temperature deviations rise again if the calculation interval k_2 is increased, since the considered period between two steps in the second component is enlarged and the deviation consequently



(a) Dirichlet boundary condition.

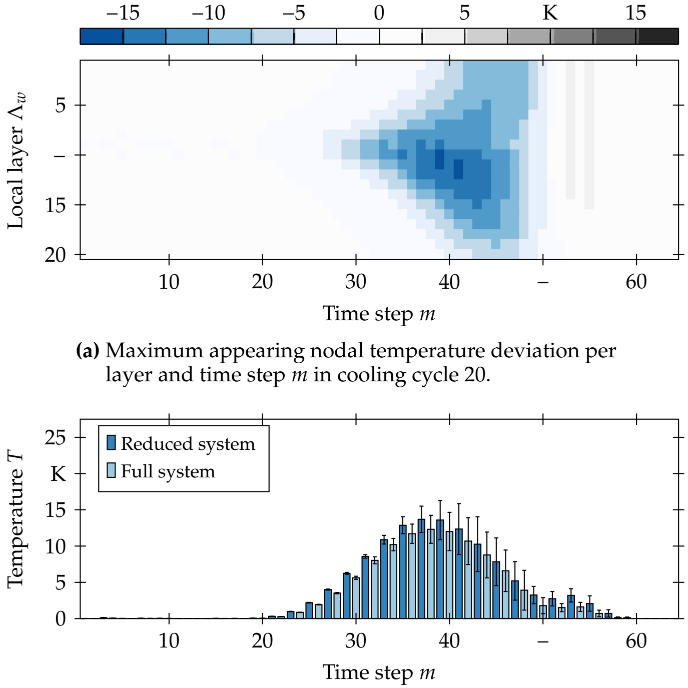
(b) Adiabatic boundary condition.

Figure 5.9: Averaged value of the maximal absolute appearing nodal temperature deviation per cooling cycle ε_{w_1,k_2} in logarithmic scale for several parameters w_1 and k_2 of the LBM-MM with Dirichlet boundary condition (left) and adiabatic boundary condition (right); material: Inconel 718; part: Cuboid.

increases. In quantitative terms, the Dirichlet boundary condition provides better results for all parameter combinations of w_1 and k_2.

The similar qualitative progression only holds for the overall valuation of the parameter combinations by ε_{w_1,k_2}. The more detailed spatial and temporal analyses reveal significant differences for the types of boundary conditions (Figures 5.10 and 5.11). However, the results for a boundary condition at different w_1 and k_2 are qualitatively similar again. For this reason, the parameter combination $w_1 = 8, k_2 = 2$ is representatively discussed. Figures 5.10a and 5.11a illustrate the temporal and spatial temperature deviations that occur in cooling cycle 20. The color-coded values represent the maximum appearing nodal temperature deviation in the layer Λ_w at time step m. The maximum deviations in the respective time steps averaged over all cooling cycles, in which the method is active, are provided in Figures 5.10b and 5.11b. The LBM-MM starts with the presence of the second component. For this reason, the cooling cycles with only one component are not included in the evaluations to avoid falsifying the results.

There is no difference between the reference simulation and the LBM-MM in time steps at the beginning, as the temperature changes have not yet reached the second component. Accordingly, no temperature deviations occur until this

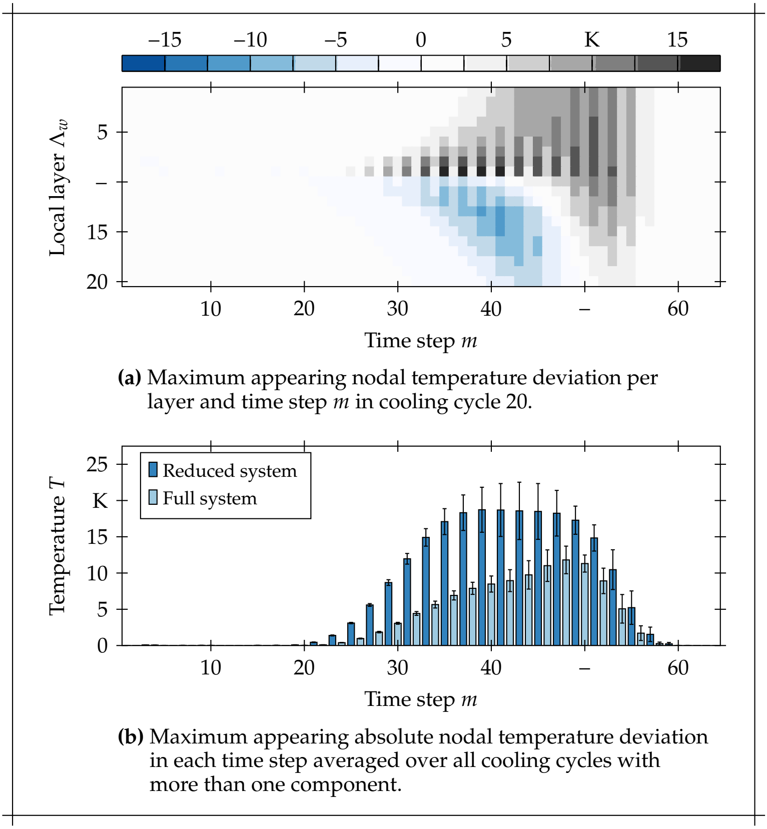


(a) Maximum appearing nodal temperature deviation per layer and time step m in cooling cycle 20.

(b) Maximum appearing absolute nodal temperature deviation in each time step averaged over all cooling cycles with more than one component.

Figure 5.10: Spatial and temporal temperature deviations of the LBM-MM with the parameter combination $w_1 = 8, k_2 = 2$ and Dirichlet boundary condition; material: Inconel 718; part: Cuboid.

point in time (Figure 5.10 and 5.11). For cooling cycles with a present second component, the solution begins to differ from the reference when the penetration depth reaches the first layer of the second component. The deviations initially occur at the interface of the components and then spread out over the layers (Figures 5.10a and 5.11a). The propagation applies equally to both components and is independent of the used boundary variant. The calculation of the full system always improves the results for both types of boundary conditions (Figures 5.10b and 5.11b). In addition, at the end of a cooling cycle, the solution of the LBM-MM always corresponds to the reference solution. Therefore, no accumulation of deviations takes place over the cooling cycles. However, this is



(a) Maximum appearing nodal temperature deviation per layer and time step m in cooling cycle 20.

(b) Maximum appearing absolute nodal temperature deviation in each time step averaged over all cooling cycles with more than one component.

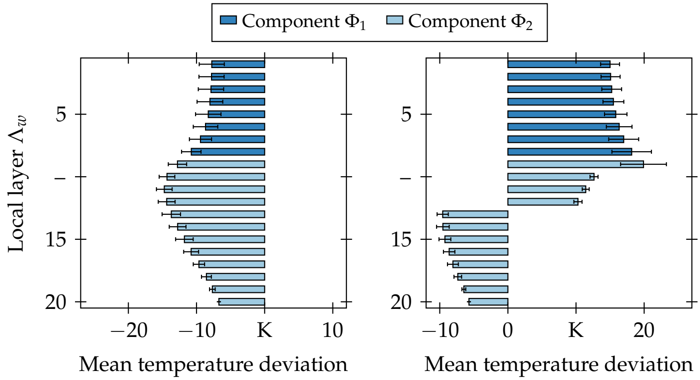
Figure 5.11: Spatial and temporal temperature deviations of the LBM-MM with the parameter combination $w_1 = 8, k_2 = 2$ and adiabatic boundary condition; material: Inconel 718; part: Cuboid.

a characteristic of the reference solution, which, in the case under consideration, has no gradients at the end of a cooling cycle. It depends on the geometry, the material and the machine parameter and is the consequence of the relation between the cross-sectional change in the z-direction, the thermal diffusivity and the dwell time.

The LBM-MM with the Dirichlet boundary condition generally first underestimates and then overestimates the temperature (Figure 5.10). As a result of the fastest-first approach, the component Φ_2 is calculated after k_2 steps of the component Φ_1. This leads to a delayed energy transport to the second component, and the nodal values are too cool when the temperature increases and too warm when it decreases. The same applies to the nodes of the first component. The boundary nodes Γ_Φ follow the temperature change only every k_2-th step and thereby slow down the temperature change; i. e., nodal values are colder when the temperature increases and warmer when it decreases. Overall, the maximal value of the underestimation is higher than that of the overestimation.

Using the LBM-MM with the adiabatic boundary variant, the deviations of the second component behave similarly to the usage of the Dirichlet boundary condition, but they differ for the first component (Figure 5.11a). The boundary nodes Γ_Φ of the first component are able to absorb the energy coming from the top layer. However, this results in nodes that are too warm, as well as heat accumulation, as they are not able to forward it. The heat accumulation at the interface reduces the heat flux from the top layer, and a shift in the temporal behavior occurs with similar consequences for the second component as previously described for the LBM-MM with the Dirichlet boundary. In general, the deviations are higher for the LBM-MM with the adiabatic boundary in comparison to the Dirichlet variant. In relative terms, however, the calculation of the total system leads to greater improvements (Figures 5.11b). The adiabatic variant affects the thermal energy of the overall system less, and therefore results in solutions closer to the reference when the total domain is taken into account. The calculations of the first component are energy conserving due to the adiabatic boundary. At the same time, however, this causes the violation of the energy balance in the second component. The changing nodal values at the interface affect the spatial gradient in the neighboring, deactivated elements in the second component, and therefore implicitly influence the energy balance for the next calculation of the full system of equations.

A consideration of the maximum nodal temperature deviations in the layers, averaged over all cooling cycles in which the method is active, shows that the Dirichlet boundary condition generally leads to colder temperatures compared to the reference solution as the negative deviations in Figure 5.12a illustrate. The interface nodes fixed by the Dirichlet boundary condition operate according to a cooler or a heater when they slow down the temperature change. As the gradients for the temperature increase are higher, the cooling effect predominates and leads to an overall energy deficit. Spatially, the maximal values of deviation per cooling cycle increase towards the first layers of the second component, as the additionally introduced boundary condition causes the deviation. In comparison, the deviations of the layers in the first component are lower. The maximal temperature changes in the first component have already elapsed when the penetration depth reaches the second component. Thus the deviations are smaller, since the consequences of the additionally introduced boundary condition solely affect the cooling. The overheating prevails in most layers when using the adiabatic boundary condition, with larger values in the



(a) Dirichlet boundary condition.

(b) Adiabatic boundary condition.

Figure 5.12: Maximum nodal temperature deviations in each layer averaged over all cooling cycles with more than one component; material: Inconel 718; part: Cuboid.

first component and a maximum mean of the deviation at the interface layer (Figure 5.12b). Due to the adiabatic boundary condition, the thermal energy remains constant for calculations of the first component. This counteracts the energy release of the nodal values and leads to high deviations. For both kinds of boundary conditions, the standard deviation decreases with increasing layers as the number of cooling cycles, the layer has gone through, decreases.

As a result of the subdivision into components and the introduction of an additional boundary, the described deviations occur when using the LBM-MM. Nevertheless, parameter combinations, whose deviations do not exceed a specified threshold ε_{tol}, are regarded as permissible. This threshold is chosen as 2 % of the considered temperature range during the buildup process, since it corresponds to the measurement uncertainty of the material parameters used from POTTLACHER et al. (2002). The deviations to be expected are thus smaller than the uncertainties introduced by the material parameters. In the case of Inconel 718 with a preheating temperature of 353 K and a solidus temperature of 1335 K, this corresponds to a tolerated deviation ε_{tol} of 19.64 K.

Table 5.2 lists the parameter combinations with the lowest permissible number of layers w_1 with regard to ε_{tol} for the different values of k_2. It is evident that the same number of layers $w_1 = 13$ is required for the k_2 values of 16, 32 and 64. The different values of k_2 correspond to four, two and one calculation, respectively,

Table 5.2: Permissible parameter combinations for two components with the lowest number of layers w_1 for each value k_2; material: Inconel 718; part: Cuboid.

Boundary condition	w_1	k_2
Dirichlet	7	2
	10	4
	11	8
	13	16
	13	32
	13	64
Adiabatic	8	2

of the second component during a cooling cycle. Until the second ($k_2 = 16$) and first ($k_2 = 32$) calculations, the second component hardly experiences any temperature changes. The third calculation ($k_2 = 16$) takes place around the point when the maximal deviation has already been reached. In summary, the calculations do not contribute to a significant reduction of the deviation and therefore offer no added value compared to $k_2 = 64$.

5.3.3 Multiple components

Permissible parameters c_2 and k_2 for two components have been determined as described in Subsection 5.3.2 using a full factorial experiment design. This is no longer practical for determining parameters c_h and k_h for additional components $h > 2$. With up to seven components for the considered geometry with 20 layers, a total of 460 458 parameter combinations would have to be investigated (cf. Section A.2). Therefore, a different procedure has to be chosen, which corresponds to a combination of the Breadth-First Search (BFS) and the Depth-First Search (DFS) algorithms from graph theory, to directly identify permissible parameters with a low expected computing time.

A *graph* is a pair consisting of a set of *vertices* and a set of *edges*, each edge connecting two vertices. The assembled connection of two vertices over multiple edges and further vertices, where any vertex is visited at most once, is called a *path*. The number of visited edges on the path is the *distance* between two vertices. If a unique path exists for each pair of vertices, the graph is also referred to as a *tree*. In this case, a vertex can be designated as a *root* in the tree. The *parent* of a vertex is then the vertex within distance 1 on a path to the root. Conversely, the vertex is the *child* of the corresponding parent. All vertices with the same parent are *siblings*. A node that has no children is called a *leaf*. There are two common methods for finding all nodes in a tree: the BFS and the DFS algorithm. Starting from a vertex, the BFS travels to all vertices with the same distance to the root before it visits the next distant vertices. Meanwhile, the DFS goes directly to a child of the vertex, and when there are no children, it returns to the first vertex with a sibling. (CORMEN 2007)

The parameter combinations of the LBM-MM are interpretable as a graph, and more precisely as a tree (cf. Figure 5.13). The vertices are the tupels (c_h, k_h), and an edge leads from one tuple (c_h, k_h) to possible parameters of an additional component (c_{h+1}, k_{h+1}). To obtain a tree structure, several nodes of a specific tuple, e. g. $(c_3 = 3, k_3 = 4)$, exist according to the number of occurrences in the possible parameter combinations. The vertices of each path, starting from the root to a vertex, thus represent a parameter combination. For the sake of simplicity, the tuple brackets are omitted when specifying a parameter combination P, i. e. $P = [(c_1, k_1), \dots, (c_h, k_h)]$.

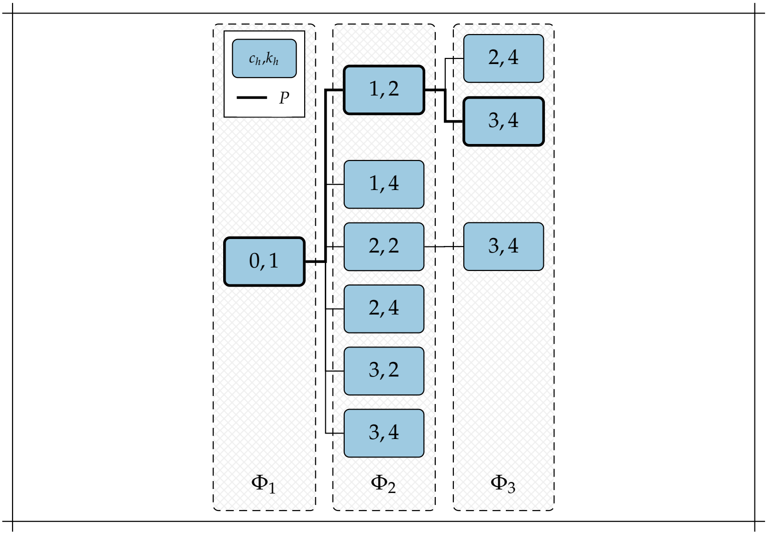


Figure 5.13: Minimal example of the LBM-MM as a tree for a geometry with 3 layers and 4 steps per cooling cycle; exemplary highlighting of the parameter combination $P = [(0,1), (1,2), (3,4)]$.

According to the DFS, first, a parameter combination with a maximum number of components is searched for. The BFS, limited to the children of the current vertex, is used to find the next vertex with the minimum expected computing time. The procedure is as follows: The tuple (c_h, k_h) is the last component from the currently permissible parameter combination, then the BFS starts with $(c_{h+1} = c_h + 1, k_{h+1} = 2 \cdot k_h)$. If the deviations of the parameter combination extended by (c_{h+1}, k_{h+1}) are outside the tolerance $\varepsilon_{\mathrm{tol}}$, the value c_{h+1} of the additional component is iteratively increased by one layer until a permissible parameter combination is reached. When no deviation within the tolerance is reachable for the remaining layers, the path from the root to (c_h, k_h) represents a non-extendable parameter combination, and the DFS has arrived at a leaf of the tree of permissible parameter combinations with the minimum expected computing time. Otherwise, the maximum k_{h+1} allowed for c_{h+1} just received is determined next. In case $k_{h+1} = 64$ is permissible, the parameter combination extended by (c_{h+1}, k_{h+1}) is the end of a path. For a maximum k_{h+1} with $k_{h+1} < 64$ the next vertex for the DFS is found.

Based on the tree of all possible parameter combinations, a tree of permissible parameter combinations with the minimum expected computing time is successively extracted by the algorithm described above. Minimum expected computing time in this context means that it is assumed that increasing the number of layers w_h in a component also leads to an increase in computing time. Furthermore, it is assumed that a higher value k_h corresponds to less computing time. Illustratively described, the presented algorithm searches for the limits between permissible and non-permissible parameter combinations of the LBM-MM. In the matrix representation of the LBM-MM in Figure 5.9, the algorithm first searches column by column, starting in the first column and the lowest row, for a valid parameter c_{h+1}. Starting from the parameter c_{h+1} found, the maximum permissible parameter k_{h+1} is then searched for line by line.

The tree of permissible parameters for the LBM-MM with Dirichlet boundary condition shown in Figure 5.14 is extracted with the described algorithm. Not listed in the tree are the siblings of a leaf if the algorithm has identified a valid parameter combination for the last component $\Phi_{h_{\mathrm{mr}}}$ with a greater number of layers $c_{h_{\mathrm{mr}}}$ but without increasing $k_{h_{\mathrm{mr}}}$. An acceleration of the calculation is thereby not to be expected.

In Subsection 5.3.2, the minimum number of layers c_2 required for permissible application of the different component-specific calculation intervals k_2 was specified. It is also included in the tree in Figure 5.14. However, the result of the algorithm also shows that a naive generalization of the results for two components to several components is not successful. It is clear that deriving additional components with the different values of the component-specific calculation interval by adopting the corresponding values c_2 from Table 5.2 does not lead to permissible parameter combinations, since these are not contained in the tree in Figure 5.14. When using the LBM-MM with the Dirichlet boundary condition, an energy deficit occurs first, in both the first and the second component (cf. Figure 5.10). If the temperature approaches the initial temperature again towards the end, it is overestimated. However, in terms of absolute values, the energy deficit prevails. This effect is increased accordingly when several components are used, since the further components contribute an additional energy deficit.

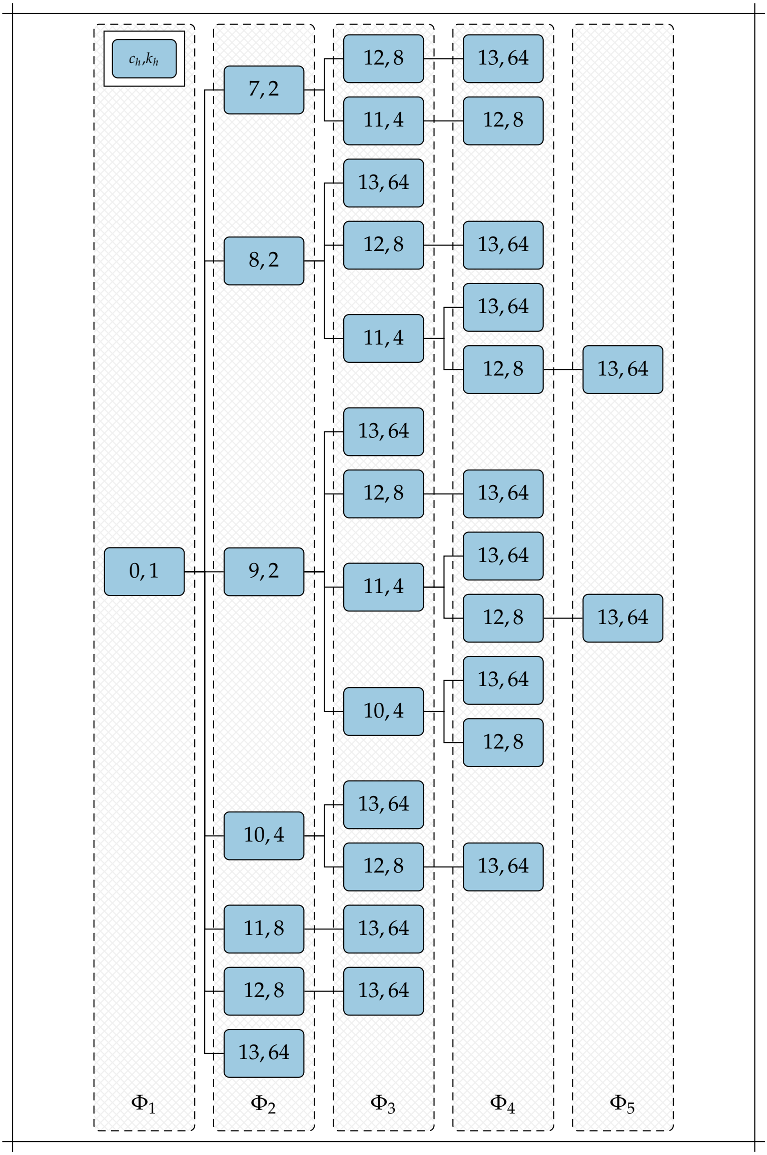


Figure 5.14: Tree of permissible parameter combinations with the minimum expected computing time of the LBM-MM with the Dirichlet boundary condition; material: Inconel 718; part: Cuboid.

Therefore, the separate components must meet higher accuracy requirements. This is clearly expressed in the combination of components with $k_2 = 2$, $k_3 = 4$ and $k_4 = 8$. The corresponding minimal values from the LBM-MM with solely two components are $c_2 = 7$, $c_3 = 10$ and $c_4 = 11$. For each component, the number of layers must be extended by at least one layer to allow further components. A permissible parameter combination is also achieved with an attenuated form of this requirement. However, this results in a deviation that allows no further component, even if the maximum possible number of the component-specific interval 64 is not reached, as indicated by the respective parameter combinations $[(0,1),(7,2),(11,4),(12,8)]$ and $[(0,1),(9,2),(10,4),(12,8)]$ in Figure 5.14. In the case of $k_h = 8$, the requirement of an additional layer for c_h even applies to $h > 2$. If the above mentioned conditions are fulfilled, no further requirements apply for an additional component Φ_h with $k_h = 64$ and the minimum number of layers $c_h = 13$. The results also imply that the values 16, 32 and 64 for k_h are again permissible from the same value c_h. For computational reasons, the maximum number $k_h = 64$ is used directly. Therefore, parameter combinations with a maximum of five components were determined by the algorithm.

Based on the results for two components, no permissible parameters for further components are expected for the LBM-MM with the adiabatic boundary condition. However, the tree in Figure 5.15 contains permissible parameter combinations with three components. With the adiabatic boundary condition, the first component is too warm, while the second is too cold (cf. Figure 5.11). The same applies to the second component as opposed to a third one. The overheating prevails and is accordingly decisive for whether a parameter combination is permissible or not. Since the second component has an energy deficit due to the first component, the overheating is compensated for to a certain extent. With increasing size of the first component w_1, the requirement for the size of the second component w_2 is reduced, however, not to the same extent as the increase of the first component is required. This means that the number of layers c_3 still increases.

5.4 Validation

The permissible parameter combinations from Figure 5.14 and Figure 5.15 are evaluated with regard to their calculation time compared to the time for the simulation without using the LBM-MM. For this purpose, the set of permissible parameter combinations was initially reduced for each kind of boundary condition. Parameter combinations that are a subset of another parameter combination are discarded. The same applies to parameter combinations whose component-specific intervals are contained in other combinations, but with the same or a smaller number of corresponding layers. Thereby, it is assumed that

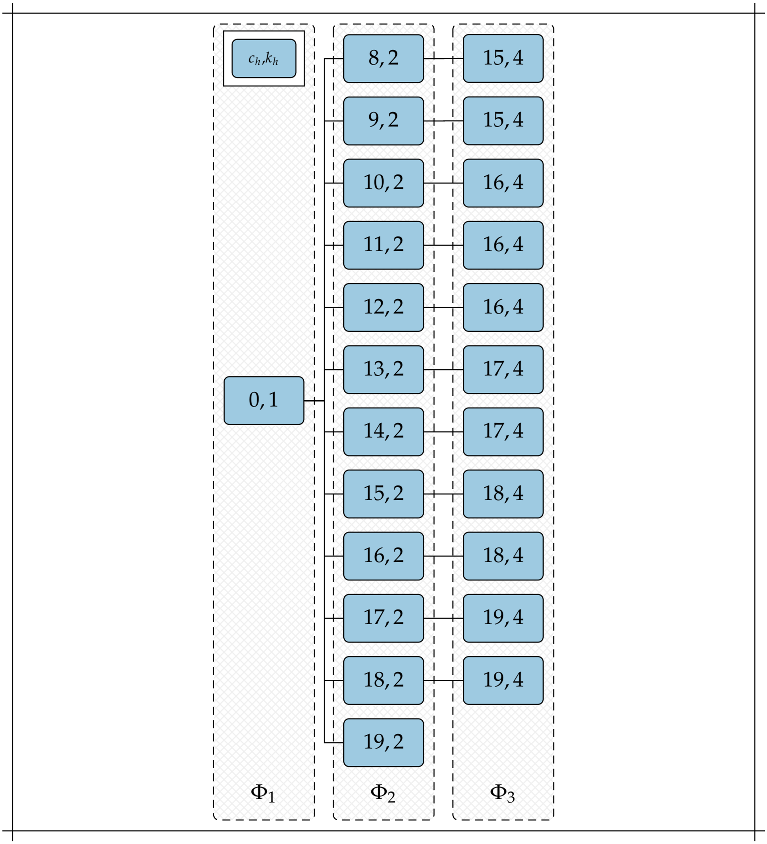


Figure 5.15: Tree of permissible parameter combinations with the minimum expected computing time of the LBM-MM with the adiabatic boundary condition; material: Inconel 718; part: Cuboid.

starting a component at the earliest possible layer, as well as adding another component, always leads to a greater reduction in computing time.

5.4.1 Part: Cuboid

Five simulations using ccx (BANTE et al. 2019) with the PARDISO solver from the Intel MKL were carried out with each of the remaining parameter combinations for the cuboid. The same applies to the reference solution and all sub-parameter combinations, i. e. the resulting parameter combinations on the path between the root and all leaves. At this point it is to be mentioned explicitly that the reference is equivalent to the parameter combination $P = [(0,1)]$. All simulation runs were performed on a virtual machine with SUSE Linux, 16 cores and 32 GB RAM on a Windows workstation. To isolate the influence of the LBM-MM on the calculation time, the shortest obtained calculation time t_P of each parameter combination P is used for the evaluations of the Dirichlet boundary condition in Figure 5.16 and in Figure 5.17 for the adiabatic boundary condition. Assuming that ccx always performs the same operations in the same order for the same input, these runs represent the calculations with the least secondary work on the computer.

Figure 5.16 and Figure 5.17 illustrate the relative calculation time $\bar{t}_P$ for a parameter combination P in relation to the calculation time of the reference solution t_0,

$$\bar{t}_P = \frac{t_P}{t_0}, \tag{5.15}$$

as well as the relative reduction of the calculation time $\Delta\bar{t}_P$ due to the additional component compared to the previous parameter combination P',

$$\Delta\bar{t}_P = \bar{t}_P - \bar{t}_{P'}. \tag{5.16}$$

The minimal duration for the simulation of the reference solution was 499.22 s. The LBM-MM with the parameter combination $P^* = [(0,1),(7,2),(12,8),(13,64)]$ and the Dirichlet boundary condition led to the greatest reduction, with a relative computing time of $\bar{t}_{P^*} = 58.22\,\%$. A difference in the calculation time when using the two types of boundary conditions of the LBM-MM was not observed on the basis of the parameter $P = [(8,2)]$ used in each case. In general, each subsequent component represents an additional acceleration of the calculation where the first components always have the largest portion with 20.78 % to 33.69 %. Thus, the assumptions made at the beginning to reduce the set of parameter combinations are confirmed. This also results from the lower calculation savings of the fastest permissible parameter combination with the adiabatic boundary conditions compared to the corresponding results of the Dirichlet boundary condition.

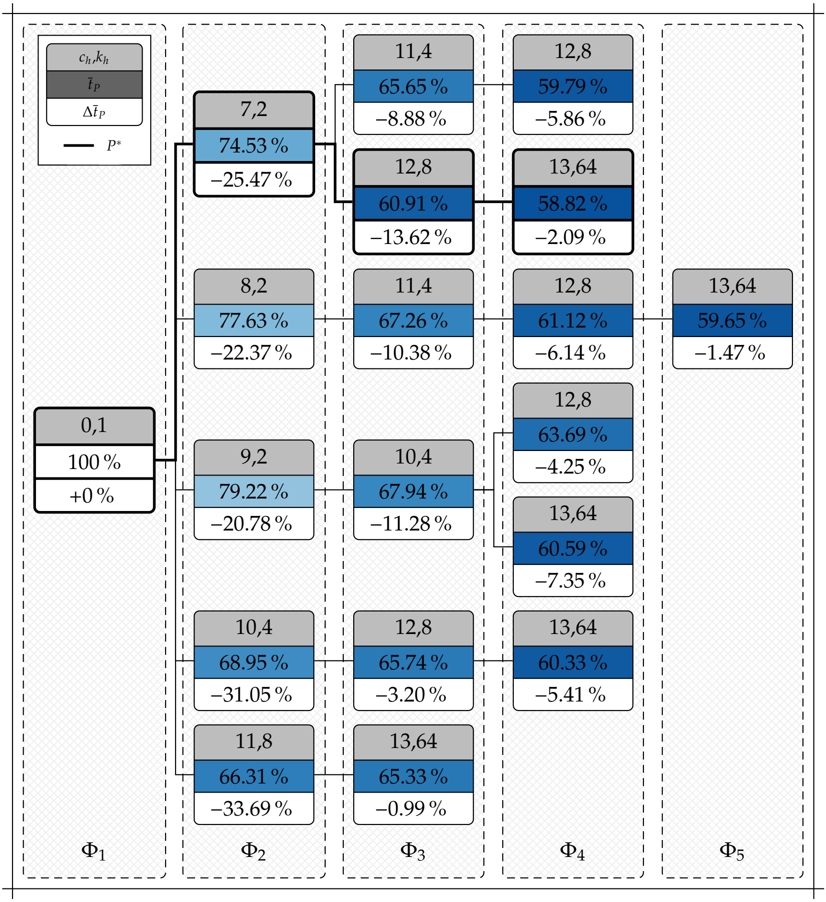


Figure 5.16: Reduced tree of permissible parameter combinations for the Dirichlet boundary condition with the corresponding relative calculation time $\bar{t}_P$ and the relative reduction of the calculation time $\Delta\bar{t}_P$ rounded to two digits; bold: path of the parameter combination with maximal reduction P^*; material: Inconel 718; part: Cuboid.

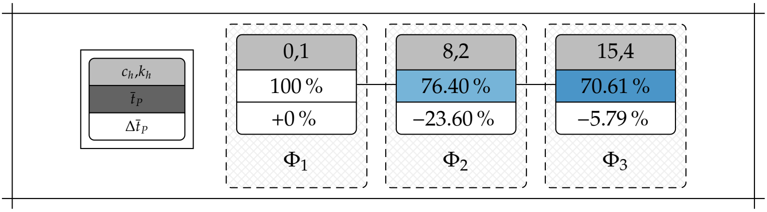


Figure 5.17: Reduced tree of permissible parameter combinations for the adiabatic boundary condition with the corresponding relative calculation time $\bar{t}_P$ and the relative reduction of the calculation time $\Delta\bar{t}_P$ rounded to two digits; material: Inconel 718; part: Cuboid.

For the cuboid, a total reduction in calculation time of up to 41.78 % was achieved. However, an even higher reduction is possible for parts with more layers. Only with the presence of a second component the system of equations can be reduced, which merely leads to a faster calculation of the remaining layers. With a larger number of remaining layers, this advantage is exploited to a correspondingly greater extent. The consideration of the relative size of the system of equations to be solved for the considered cuboid structure in each cooling cycle with various parameter combinations supports this assumption of the higher reduction (Figure 5.18).

In addition to the 20 layers of the cuboid structure, Figure 5.18 also illustrates the size of the system of equations when assuming further layers for the part. The average sizes of the first 20 layers were used for extrapolation of the additional layers. The system of equations is reduced to about 20.41 % in cooling cycle 20. In an assumed cooling cycle 90, there would even be an average reduction of up to 9.41 % of the size of the full system of equations. With more layers, there are not only more steps in which the advantage of a small system of equations is exploited, but the average size of the system of equations also continues to decrease, although no further components are added. The reduction is indirect, since the calculation domain is further increased by new layers, but this only applies to the LBM-MM for time steps in which the entire domain is calculated. Due to the fixed number of layers per component, the absolute size of the system of equations remains approximately constant for the remaining steps. However, the size of the system of equations generally does not allow conclusions to be drawn about the expected computing time, since this depends on the solver used and its computational complexity, i. e. the required number of floating point operations to solve a system of equations with n_Ξ unknowns. This ratio is nonlinear, i. e. the required number of floating points increases disproportionately with increasing unknowns n_Ξ. Therefore, it can be assumed that the relative calculation time decreases further when using the LBM-MM

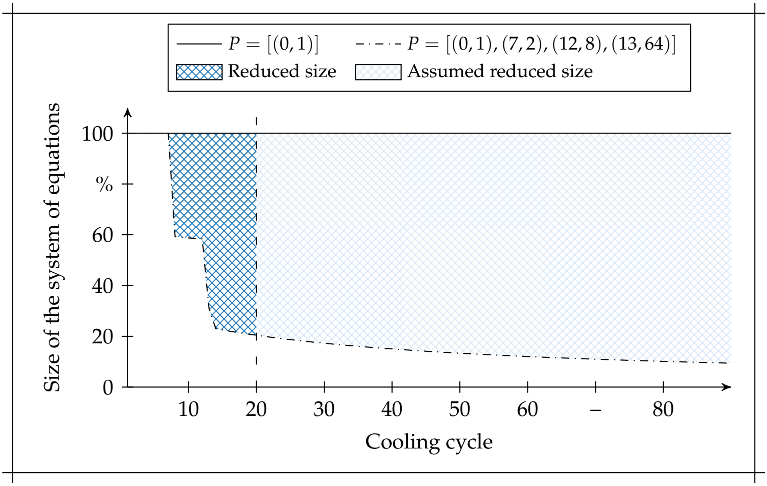


Figure 5.18: Average size of the system of equations to be solved for the considered cuboid structure in each cooling step using the LBM-MM with $P = [(0,1),(7,2),(12,8),(13,64)]$ in comparison to the reference configuration; material: Inconel 718; part: Cuboid.

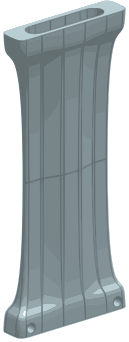
not only with increasing number of layers, but also with increasing number of nodes n_{Ξ}.

5.4.2 Part: Strut

The uniform heat flux of the cuboid in the z-direction due to the corresponding constant cross-section provided a fundamental understanding of the effects and interactions of the various parameters of the method. However, this geometry is hardly relevant for AM. Therefore, the LBM-MM was finally applied to an actual part based on the results of the cuboid. The results indicate that the method is particularly suitable for components with many layers and a large number of nodes. A strut therefore serves as the application case. This is a target component of *MTU Aero Engines AG*[33] for the next generation of additive series production (MTU Aero Engines AG 2018). For confidentiality reasons, the strut is not used in its original geometry, but in a slightly modified form that still has all relevant geometry features. The component was placed centrally on

[33] MTU Aero Engines AG, Dachauer Strasse 665, 80995 Munich, Germany

the base plate and rotated with 5° against the coating direction. Further data on the geometry and its meshing for simulation are listed in Figure 5.19.



(a) Part

Geometry $w \times d \times h$ in mm	
part	$65 \times 15 \times 150$
build plate	$250 \times 250 \times 40$
Mesh	
element size, part	5/3 mm
element size, base plate	30 mm
height of the LCs	5/3 mm
elements	147 094
nodes	37 953
layer compounds	90

(b) Geometry and mesh information

Figure 5.19: Configuration of the benchmark geometry: Strut.

For parts with a tapered cross-section in the direction of the base plate, a delayed heat transport into the lower layers occurs due to heat accumulation. As a result of the smaller volume in deeper layers, which absorb the thermal energy, these areas are also reheated to a greater extent by activation of subsequent layers. Concerning the number of layers w_{h-1} for a fixed value k_h of each component Φ_h, the determined permissible parameter combinations for the cuboid thus represent a lower limit for these parts. Therefore, an adjustment of the permissible parameter combinations is necessary. Within the scope of this work, these were determined for the strut and two components using Dirichlet boundary conditions. Since the adiabatic boundary condition led to a significantly lower reduction in computing time, it is no longer taken into account here. The determined permissible parameter combinations for the strut and the resulting reduction of the calculation time are displayed in Figure 5.20. Again, five calculation runs were performed for each parameter combination under the same conditions as for the cuboid structure (cf. Subsection 5.4).

A relative calculation time of 39.45 % with respect to the reference was achieved for the strut with the parameter combination $P^* = [(0,1),(18,32)]$. This means, despite larger values c_2, a relative calculation time saving by a factor of 1.5 compared to the cuboid. In contrast to the cuboid, the parameter combinations of the lowest computing time do not correlate with the lowest values c_2. More important is k_2. Due to a higher value of k_2, the method starts later, but leads, on

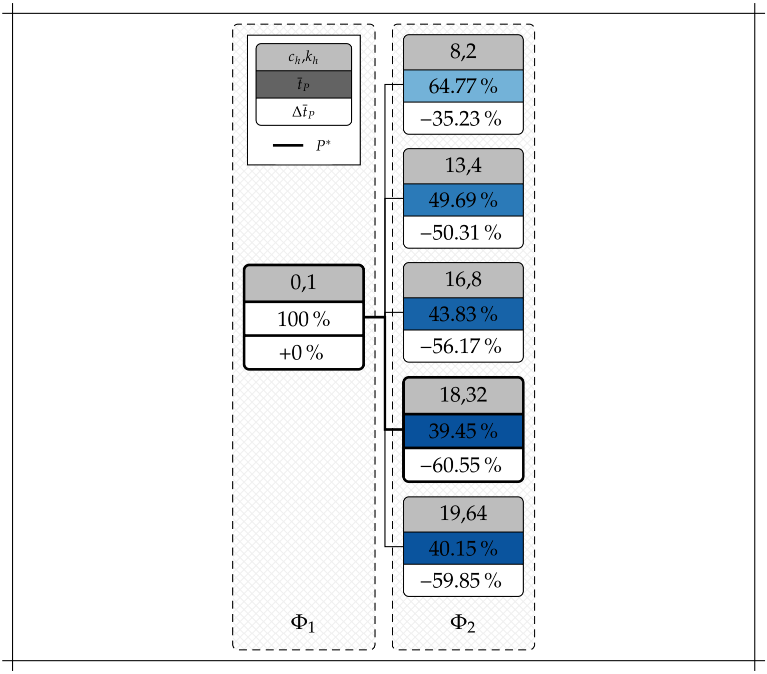


Figure 5.20: Reduced tree of permissible parameter combinations for the Dirichlet boundary condition with the corresponding relative calculation time $\bar{t}_P$ and the relative reduction of the calculation time $\Delta\bar{t}_P$ rounded to two digits; bold: path of the parameter combination with maximal reduction P^*; material: Inconel 718; part: Strut.

average, to significantly smaller systems of equations. This is also apparent in Figure 5.18 when deploying components Φ_3 and Φ_4. Both lead to a significantly lower level in terms of reducing the relative size of the system of equations. For many layers, as in the case of the strut, the effect of the skipped time steps k_h predominates in the reduction of the computing time compared to an early start of further components, i. e. a small value of c_h.

5.5 Summary

The theoretical consideration shows that the order of the numerical integrator is decisive for the uncertainty. In the case of the backward Euler with order 1, it is accordingly the second derivative of the solution. Using an analytical solution, it was revealed that the selected time step size determination allows the large change at the beginning of the simulation to be mapped with a low uncertainty and independently of the layer. In the initial and middle temporal range of a cooling cycle, the application of larger time steps by the LBM-MM for deeper layers is also theoretically permissible without an increase in global uncertainty. Towards the end of the cooling cycle, the statement is only conditionally meaningful due to the derivation used.

By tolerating temperature deviations of up to 2 % of the total considered temperature range, the use of the LBM-MM with two components already yielded a calculation time saving of 25.47 % for a simple cuboid geometry. A combination of DFS and BFS algorithms identified parameter combinations with up to five components and a low expected computing time. However, the permissible parameter combinations for more than two components showed that components are not generated directly from the results of two components and the corresponding different time step intervals. For additional components, the previous components must meet higher precision requirements. Nevertheless, time savings of up to 41.18 % were achieved with additional components.

For the case of an actual AM part, it was revealed that with more layers and nodes, even greater reductions in computing time are possible (60.55 %). However, the application example has also shown that a relationship exists between permissible parameter combinations and geometric complexity.

6 LBM-specific Direct Solving Strategy for Layer-Wise Growing Domains

6.1 Overview

This chapter presents a strategy for solving the linearized system of equations

$$\left[\frac{1}{\Delta t^m}\boldsymbol{C}\left(T^m_{i-1}\right)+\boldsymbol{K}\left(T^m_{i-1}\right)+\boldsymbol{K}_C\left(T^m_{i-1}\right)\right]\cdot\boldsymbol{T}^m_i = \frac{1}{\Delta t^m}\boldsymbol{C}\left(T^m_{i-1}\right)\cdot\boldsymbol{T}^{m-1}+\boldsymbol{q}\left(t^m\right) \quad (2.71)$$

resulting after the spatial and temporal discretization of the heat equation. Instead of the isolated consideration of a single system of equations, the focus is on a holistic approach for the efficient solution of all time steps within the 'Heating' phase. The fields of action of this chapter are illustrated in Figure 6.1.

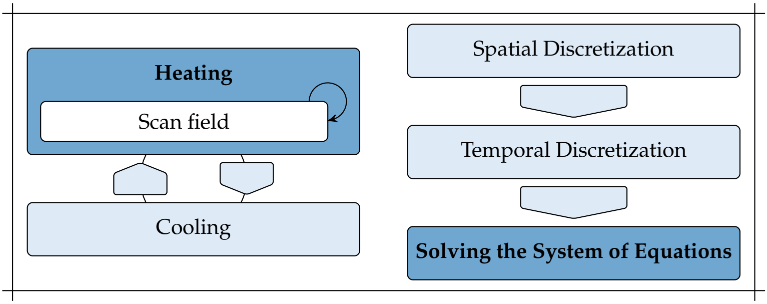


Figure 6.1: Classification of Chapter 6 with regard to the phase of the Mathematical Model of LBM and application area in numerics; fields of action (dark blue).

As a consequence of the use of Conceptual Models with a lower degree of abstraction, larger systems of equations occur, especially in the 'Heating' phase, which also have to be solved more often. However, the advantage of Direct Methods, the reusability of the decomposition for arbitrary right-hand sides of the system of equations, is not exploited due to the non-linearity of the temperature-dependent material parameters and the growing system of equations by continuous activation of further scan fields. As is apparent from the current research, a special structure is obtained for all matrices in LBM by rearranging the node numbers (NIKOUKAR et al. 2013). However, only the solution of a single time step was considered.

An LBM-specific Direct Solving Strategy (LBM-DSS) is presented which divides the coefficient matrix into two parts: A constant and a temperature-dependent part. Using a splitting approach, the system of equations is reduced to the decomposition of the linear part of the coefficient matrix, and the small deviation of material parameters of metals due to temperature changes is taken advantage of. When enlarging the calculation mesh by the iterative activation of the scan fields, the structure of the matrix and the properties of the Cholesky decomposition are exploited. Based on NIKOUKAR et al. (2013), an extended rearrangement of the node numbering is presented. Thus, it is only necessary to partially decompose the new coefficient matrix resulting from the activation of an additional scan field. This applies to scan fields of a single layer, but also to the transition from one layer to the next. The procedure is presented with the standard Cholesky decomposition. However, the LBM-DSS is applicable with arbitrary solvers which provide a decomposition of the coefficient matrix in the form $\boldsymbol{A}^m = \boldsymbol{L}^m \cdot \boldsymbol{L}^{m\top}$. The Cholesky decomposition is used since it considers the symmetry of the coefficient matrix and is, among the Direct Methods, the one with the least effort.

First, the LBM-DSS is presented. This includes the splitting approach, as well as the permutation of the coefficient matrix and the resulting reduced decomposition for the extension of the calculation mesh. By using the splitting approach, the solving of the system of equations is partially converted into an iterative procedure. To further ensure convergence, the necessary requirements are defined in the verification and the corresponding settings are derived. The method is then validated with regard to the number of iterations, to the solution effort and to the computing time. The chapter concludes with a summary.

Due to the equidistant time steps in the 'Heating' phase,

$$\Delta t^m = \Delta t \tag{6.1}$$

applies for $m \in (1, \dots, m_Y)$. The notation of Equation 2.71 is simplified for this chapter to

$$\boldsymbol{A}^m\left(T_{i-1}^m\right) \cdot \boldsymbol{T}_i^m = \boldsymbol{b}^m\left(T_{i-1}^m\right) \tag{6.2}$$

with

$$\boldsymbol{A}^m\left(T_{i-1}^m\right) = \frac{1}{\Delta t}\boldsymbol{C}\left(T_{i-1}^m\right) + \boldsymbol{K}\left(T_{i-1}^m\right) + \boldsymbol{K}_C\left(T_{i-1}^m\right), \tag{6.3}$$

$$\boldsymbol{b}^m\left(T_{i-1}^m\right) = \frac{1}{\Delta t}\boldsymbol{C}\left(T_{i-1}^m\right) \cdot \boldsymbol{T}^{m-1} + \boldsymbol{q}(t^m). \tag{6.4}$$

6.2 Methodology

Assuming a proper node numbering, the entries of the coefficient matrix $\boldsymbol{A}^m$ as well as of the Cholesky decomposition $\boldsymbol{L}^m$ are assignable to the different layers of the meshed geometry Θ. The Figure 6.2 shows that adding another layer is an extension of the Cholesky decomposition. The calculation of a matrix entry of $\boldsymbol{L}^m$ only requires the values to the left and above the entry (cf. Subsection 2.4.2). Therefore, if the decomposition from the previous step is known, only the corresponding entries have to be decomposed for an additional layer. However, since temperature-dependent material parameters are used, all entries, whose associated nodes have undergone a temperature change, differ from the previous time step. The presented LBM-DSS transforms the system of nonlinear equations accordingly, thus only a partial Cholesky decomposition is required. This is equivalent to the fact, that, in a time step with no further layer, the matrix $\boldsymbol{L}^m$ remains valid and no additional decomposition of the matrix is necessary.

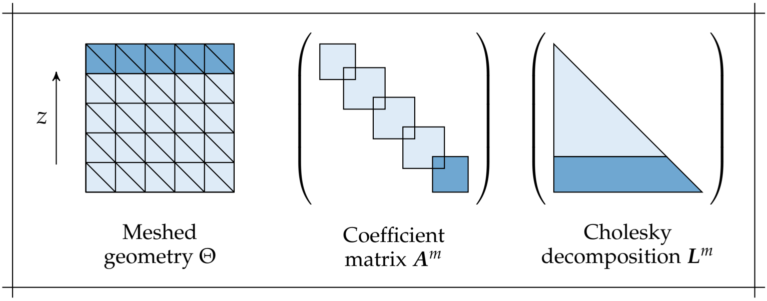

z
Meshed geometry Θ
Coefficient matrix $\boldsymbol{A}^m$
Cholesky decomposition $\boldsymbol{L}^m$

Figure 6.2: Highlighting of the top layer elements (left) and the corresponding entries of the coefficient matrix (middle) as well as of the Cholesky decomposition (right).

The LBM-DSS consists of three parts, which are presented in the following subsections. First, the system of nonlinear equations is transformed by a splitting method into an iteration of systems of linear equations with an identical, temperature-independent coefficient matrix. The presented node numbering for layered meshes then only requires the partial decomposition of the coefficient matrix when adding new domains.

6.2.1 Linearizing matrix splitting

The coefficient matrix $\boldsymbol{A}^m(T_{i-1}^m)$ is split into a constant part corresponding to $\boldsymbol{A}^m$ at a prescribed temperature T_{fix} and a remaining temperature-dependent part $\Delta\boldsymbol{A}^m$, i. e.

$$\boldsymbol{A}^m\left(T_{i-1}^m\right) = \boldsymbol{A}^m(T_{\text{fix}}) - \Delta\boldsymbol{A}^m\left(T_{i-1}^m\right) \tag{6.5}$$

For simplification, the notation

$$\boldsymbol{A}_{\text{const}}^m := \boldsymbol{A}^m(T_{\text{fix}}) \tag{6.6}$$

is introduced. The procedure for Stationary Iterative Methods according to Equation 2.47 with the introduced splitting of the coefficient matrix is combined with the linearization strategy from Equation 2.71 and results in the iteratively solvable system of equations

$$\boldsymbol{A}_{\text{const}}^m \cdot \boldsymbol{T}_i^m = \Delta\boldsymbol{A}^m\left(T_{i-1}^m\right) \cdot \boldsymbol{T}_{i-1}^m + \boldsymbol{b}^m\left(T_{i-1}^m\right) \tag{6.7}$$

or, in residual representation,

$$\begin{aligned}\boldsymbol{A}_{\text{const}}^m \cdot \Delta\boldsymbol{T}_i^m &= \boldsymbol{b}^m\left(T_{i-1}^m\right) - \boldsymbol{A}^m\left(T_{i-1}^m\right) \cdot \boldsymbol{T}_{i-1}^m && \text{(6.8a)}\\ &= \boldsymbol{r}^m\left(T_{i-1}^m\right) && \text{(6.8b)}\end{aligned}$$

with $\boldsymbol{T}_i^m = \boldsymbol{T}_{i-1}^m + \Delta\boldsymbol{T}_i^m$ and the *residual flux* $\boldsymbol{r}^m(T_{i-1}^m)$. Since the combination with the linearization strategy changes the splitting of the matrix in each iteration, the procedure is a Nonstationary Iterative Method. It is comparable to the principle of a *Quasi-Newton method*, in which the coefficient matrix, respectively the Jacobian matrix, is not updated.

6.2.2 Node numbering for layered meshes

The ascending node numbering is carried out for each layer, starting from the lowest layer, using the following two-step permutation procedure:

1. Numbering of nodes that are not shared with other layers.
2. Numbering of nodes that are shared with the subsequent layer.

The build plate is interpreted in this case as an additional layer and thus also as the lowest layer. The resulting structure of the coefficient matrix $\boldsymbol{A}^m$ is illustrated in Figure 6.3. Explicit numbering of nodes within a set is not considered in this thesis because it is not relevant for the applicability of the solving strategy. However, an additional condition for the node numbering to reduce the bandwidth is introduced in Subsection 6.4.3.

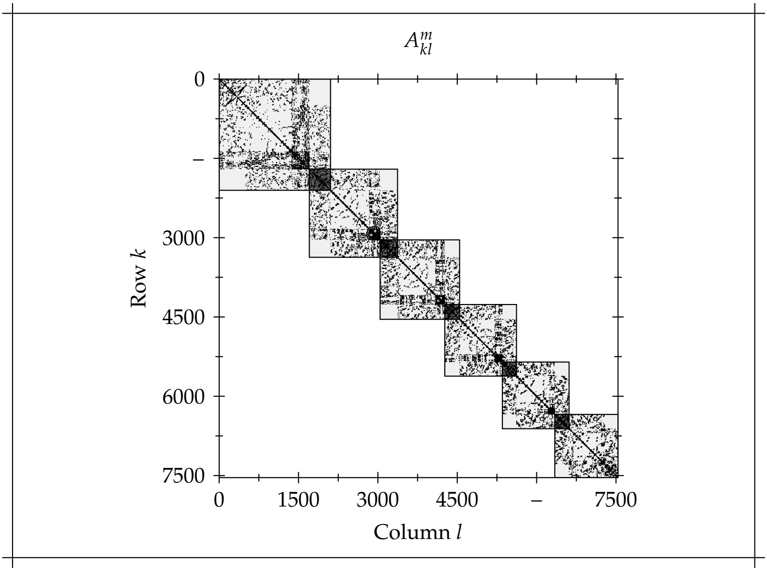


Figure 6.3: Sparsity pattern of A^m for an exemplary geometry with six layers and 7538 nodes, resulting from the presented permutation procedure; nodes numbered according to step one (light gray background), nodes numbered due to step two (dark gray background).

6.2.3 Partial Cholesky decomposition

For the sake of clarity, the introduction of the LBM-DSS initially distinguishes between a further heating step for the current scan field and a time step in which a new scan field is added. In the first case, $\boldsymbol{A}^{m-1}_{\text{const}} = \boldsymbol{A}^{m}_{\text{const}}$ holds true for the time step m to be solved by Equation 6.8, and thus $\boldsymbol{L}^{m-1} = \boldsymbol{L}^{m}$ also applies. As a consequence, only a forward and a backward substitution corresponding to

$$\boldsymbol{L}^m \cdot \Delta \boldsymbol{y}^m_i = \boldsymbol{r}^m \left(T^m_{i-1} \right), \tag{6.9a}$$

$$\boldsymbol{L}^{m\top} \cdot \Delta \boldsymbol{T}^m_i = \Delta \boldsymbol{y}^m_i \tag{6.9b}$$

is necessary for solving, since the Cholesky decomposition is already available. Activating a new scan field affects the entries of the coefficient matrix that belong to the elements of the newly added scan field. In generalized block matrix representation, this means the coherence

$$\boldsymbol{A}^{m-1}_{\text{const}} = \begin{pmatrix} \boldsymbol{A}^{m-1}_{1} & {\boldsymbol{A}^{m-1}_{21}}^{\top} & \boldsymbol{0} \\ \boldsymbol{A}^{m-1}_{21} & \boldsymbol{A}^{m-1}_{2} & {\boldsymbol{A}^{m-1}_{32}}^{\top} \\ \boldsymbol{0} & \boldsymbol{A}^{m-1}_{21} & \boldsymbol{A}^{m-1}_{3} \end{pmatrix}, \tag{6.10a}$$

$$\boldsymbol{A}^{m}_{\text{const}} = \begin{pmatrix} \boldsymbol{A}^{m-1}_{1} & {\boldsymbol{A}^{m-1}_{21}}^{\top} & \boldsymbol{0} \\ \boldsymbol{A}^{m-1}_{21} & \boldsymbol{A}^{m}_{2} & {\boldsymbol{A}^{m}_{32}}^{\top} \\ \boldsymbol{0} & \boldsymbol{A}^{m}_{32} & \boldsymbol{A}^{m}_{3} \end{pmatrix} \tag{6.10b}$$

for the coefficient matrices $\boldsymbol{A}^{m-1}_{\text{const}} \in \mathbb{R}^{n_\Xi \times n_\Xi}$ and $\boldsymbol{A}^{m}_{\text{const}} \in \mathbb{R}^{n'_\Xi \times n'_\Xi}$. For the block matrices specified in Equations 6.10

$$\boldsymbol{A}^{m-1}_{1} \in \mathbb{R}^{n_1 \times n_1}, \qquad \boldsymbol{A}^{m-1}_{21} \in \mathbb{R}^{n_1 \times n_2}, \tag{6.11a}$$

$$\boldsymbol{A}^{m-1}_{3} \in \mathbb{R}^{n_3 \times n_3}, \qquad \boldsymbol{A}^{m-1}_{21} \in \mathbb{R}^{n_2 \times n_3}, \tag{6.11b}$$

$$\boldsymbol{A}^{m}_{3} \in \mathbb{R}^{n'_3 \times n'_3}, \qquad \boldsymbol{A}^{m}_{32} \in \mathbb{R}^{n_2 \times n'_3}, \tag{6.11c}$$

$$\boldsymbol{A}^{m-1}_{2}, \boldsymbol{A}^{m}_{2} \in \mathbb{R}^{n_2 \times n_2} \tag{6.11d}$$

apply with the relations

$$n_1 + n_2 + n_3 = n_\Xi, \qquad n_1 + n_2 + n'_3 = n'_\Xi \tag{6.12}$$

and $n_3 \leq n'_3$. A matrix entry $A^{m}_{\text{const},kl}$ outside the diagonal (i. e. $k \neq l$) represents an edge between two nodes $\boldsymbol{r}_k$ and $\boldsymbol{r}_l$. The diagonal $A^{m}_{\text{const},kk}$ indicates the node $\boldsymbol{r}_k$ itself. The matrix $\boldsymbol{A}^{m-1}_{1}$ contains the entries resulting from the nodes and the edges between nodes of all previous layers, except the shared nodes with the subsequent layer and the edges to these. Corresponding entries of the current top layer are represented by the matrix $\boldsymbol{A}^{m}_{3}$. The shared nodes and edges of the previous layer with the current one are represented by the matrix $\boldsymbol{A}^{m}_{2}$. The edges to the shared nodes in the previous layer and in the current layer form the matrices $\boldsymbol{A}^{m-1}_{21}$ and $\boldsymbol{A}^{m}_{32}$ respectively. An illustration of the block matrix representation of $\boldsymbol{A}^{m}_{\text{const}}$ in Equation 6.10b is provided in Figure 6.4.

Based on the known Cholesky decomposition $\boldsymbol{L}^{m-1}$ of the coefficient matrix $\boldsymbol{A}^{m-1}_{\text{const}}$ in the block matrix representation corresponding to Equation 6.10, i. e.

$$\boldsymbol{L}^{m-1} = \begin{pmatrix} \boldsymbol{L}^{m-1}_{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}^{m-1}_{21} & \boldsymbol{L}^{m-1}_{2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}^{m-1}_{32} & \boldsymbol{L}^{m-1}_{3} \end{pmatrix}, \tag{6.13}$$

the Cholesky decomposition for $\boldsymbol{A}^{m}_{\text{const}}$ yields

$$\boldsymbol{L}^{m} = \begin{pmatrix} \boldsymbol{L}^{m-1}_{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}^{m-1}_{21} & \boldsymbol{L}^{m}_{2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}^{m}_{32} & \boldsymbol{L}^{m}_{3}, \end{pmatrix} \tag{6.14}$$

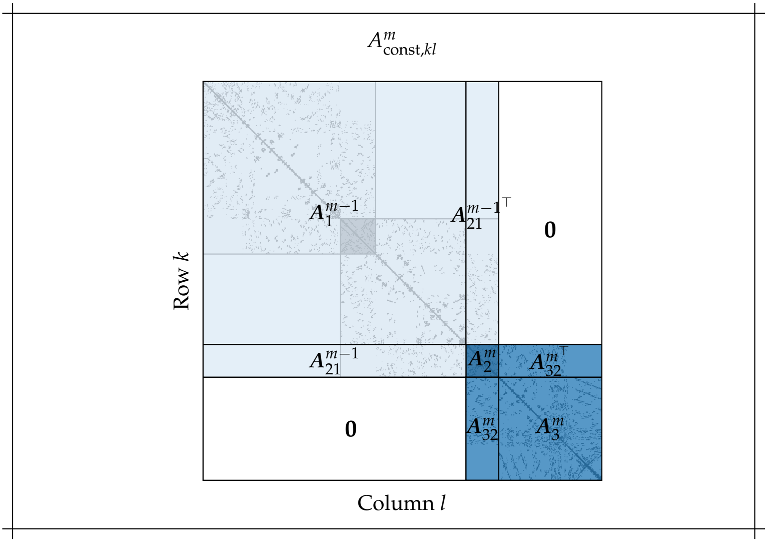


Figure 6.4: Composition of the block matrix representation of $\boldsymbol{A}^m_{\text{const}}$ in the case of three layers; constant matrices (light blue) and changing matrices (dark blue) in comparison to the coefficient matrix $\boldsymbol{A}^{m-1}_{\text{const}}$ of the previous time step $m-1$.

where the matrices $\boldsymbol{L}_1^{m-1} \in \mathbb{R}^{n_1 \times n_1}$, $\boldsymbol{L}_2^{m-1}, \boldsymbol{L}_2^m \in \mathbb{R}^{n_2 \times n_2}$ and $\boldsymbol{L}_3^{m-1} \in \mathbb{R}^{n_3' \times n_3'}$ are lower triangular matrices. Accordingly, the shares $\boldsymbol{L}_1^{m-1}$ and $\boldsymbol{L}_{21}^{m-1}$ are known. The Cholesky decomposition uses previously calculated values to the left and above to determine the current entry of the decomposition (cf. Equations 2.45). As a consequence, the determination of the lower triangular matrix $\boldsymbol{L}^m$ requires only a partial Cholesky decomposition of the coefficient matrix $\boldsymbol{A}^m_{\text{const}}$. The entries of the matrix $\boldsymbol{L}_2^m$, $\boldsymbol{L}_3^m$ and $\boldsymbol{L}_{32}^m$ are obtained by

$$L^m_{ll} = \sqrt{A^m_{\text{const},ll} - \sum_{s=1}^{l-1} L^{m2}_{ls}}, \tag{6.15a}$$

$$L^m_{kl} = \frac{1}{L^m_{ll}} \left(A^m_{\text{const},kl} - \sum_{s=1}^{l-1} L^m_{ks} L^m_{ls} \right), \qquad k > l \tag{6.15b}$$

restricted to $n_1 < k, l \leq n'_{\Xi}$. With the obtained Cholesky decomposition $\boldsymbol{L}^m$, Equation 6.7 is solved again according to Equations 6.9. For the decomposition in the area of matrices $\boldsymbol{L}_3^m$ and $\boldsymbol{L}_{32}^m$ it is sufficient for the run variable s in Equations 6.15 to start with $s = n_1 + 1$, since the lower left block matrix of L^m_{kl} is the zero matrix. However, this is implicitly taken into account later by a sparse implementation that only uses the non-zero entries.

The relation of Equations 6.10 applies both to the activation of another scan field within a layer and to the activation of the first scan field in a new layer. The second corresponds to the special case $n_3 = 0$. However, the previous time step $m-1$ refers to the last step in the last occurring 'Heating' phase. The time step size is included in the definition of the coefficient matrix (cf. Equation 6.3). The coherence from Equations 6.10 is only guaranteed for time steps that fulfill the requirement of the same time step size from Equation 6.1. The distinction for a further heating step introduced at the beginning for reasons of understanding is not necessary with the representation in Equations 6.10, because it corresponds to the special case $n_3' = 0$.

6.3 Verification

The described procedure of the LBM-DSS allows the reuse of the already decomposed coefficient matrix from the previous time step and thus exploits the advantage of Direct Methods. Global convergence is not guaranteed because of the necessary linearization due to the temperature-dependent material parameters (cf. Section 3.2). However, differences in material parameters due to temperature dependence of metals are small; therefore, convergence is to be expected in general. This section thus deals with the choice of a reference temperature T_{fix} to ensure that this continues to be the case with the introduced splitting method. Furthermore, the legitimacies of the assumed matrix structures in Equation 6.13 and Equation 6.14 are demonstrated. The presented numbering of the nodes is not verified, since the resulting structure of the coefficient matrix is directly apparent.

6.3.1 Convergence of the linearizing matrix splitting

The goal of this subsection is to gain knowledge about the convergence behavior and to identify a range of permissible reference temperatures, to maintain convergence by the additionally introduced splitting method. A splitting method converges if and only if the spectral radius of the iteration matrix is smaller than 1 (cf. Subsection 2.4.2). As this criterion only holds for Stationary Iterative Methods, it also applies to the presented splitting approach, since it is interpretable as stationary for each individual iteration i. The splitting approach corresponds to an additional sub-iteration j with only one step for each iteration i according to

$$\boldsymbol{A}^m_{\text{const}} \cdot \boldsymbol{T}^m_{i-1,j+1} = \Delta \boldsymbol{A}^m \left(T^m_{i-1}\right) \cdot \boldsymbol{T}^m_{i-1,j} + \boldsymbol{b}^m \left(T^m_{i-1}\right) \tag{6.16}$$

with

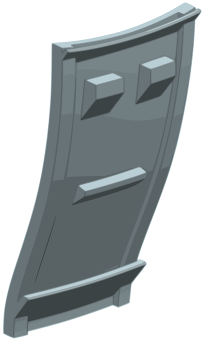
$$\begin{aligned} \boldsymbol{T}^m_{i-1,0} &:= \boldsymbol{T}^m_{i-1}, && (6.17)\\ \boldsymbol{T}^m_{i} &:= \boldsymbol{T}^m_{i-1,1}. && (6.18) \end{aligned}$$

If the spectral radius of the iteration matrix

$$\begin{aligned} \boldsymbol{M}^m_i &= \left(\boldsymbol{A}^m_{\text{const}}\right)^{-1} \cdot \Delta\boldsymbol{A}^m\left(T^m_{i-1}\right) && (6.19a)\\ &= \boldsymbol{I}^m - \left(\boldsymbol{A}^m_{\text{const}}\right)^{-1} \cdot \boldsymbol{A}^m\left(T^m_{i-1}\right), && (6.19b) \end{aligned}$$

with the unit matrix $\boldsymbol{I}^m$ corresponding to the size of the matrix $\boldsymbol{A}^m_{\text{const}}$, is smaller than 1 in each iteration i, then it is guaranteed that $\boldsymbol{T}^m_{i-1,1}$ represents an improvement of $\boldsymbol{T}^m_{i-1}$ in relation to $\boldsymbol{T}^m_{i}$. To ensure this, the parameter T_{fix} introduced by the method is used.

In addition to the cuboid (cf. Figure 5.8) and strut (cf. Figure 5.19) geometries used in Chapter 5, the spectral radius for different reference temperatures is examined on a panel geometry (cf. Figure 6.5). For confidentially reasons, the panel is not used in its original geometry. A modified form is used that still has all relevant geometry features. As with the strut, this is a target component of the MTU Aero Engines AG for the next generation of additive series production (MTU Aero Engines AG 2018). Deviating from Chapter 5, however, the strut is used with an LC height of 5 mm and corresponding 30 layers to demonstrate the LBM-DSS on components with more than one element over the layer height. The other values remain identical.



(a) Part

Geometry $w \times d \times h$ in mm	
part	75 × 50 × 150
build plate	250 × 250 × 40
Mesh	
element size, part	5 mm
element size, base plate	30 mm
height of the LCs	5 mm
elements	41 189
nodes	8593
layer compounds	30

(b) Geometry and mesh information

Figure 6.5: Configuration of the benchmark geometry: Panel.

For all three mentioned geometries, the spectral radius of $\boldsymbol{M}_i^m$ was determined for all iterations i of the time steps m from the 'Heating' phase and different reference temperatures T_{fix}. The averaged spectral radii over all i and m are plotted in Figure 6.6, as well as the corresponding standard deviations. A calculation with one reference temperature was performed for each geometry, and the resulting matrices $\boldsymbol{A}^m\left(T_i^m\right)$ were used to determine the spectral radii of $\boldsymbol{M}_i^m$ for further reference temperatures. Thus, the strictly theoretical results of reference temperatures are obtained that lead to spectral radii greater than one and would not allow further evaluations due to the consequent divergence. Furthermore, the number of iterations i, and thus also the resulting matrices $\boldsymbol{A}^m\left(T_i^m\right)$, depend on the selected reference temperature T_{fix}. However, the choice of other reference temperatures to determine the matrices $\boldsymbol{A}^m\left(T_i^m\right)$ led to negligible differences in the resulting spectral radii of $\boldsymbol{M}_i^m$, as it only influences the number of iterations for convergence but not the limit itself. Therefore, the results of all reference temperatures provided are based on the same matrices for each geometry, even if the reference temperature itself produces a converging method. Two equidistant heating steps, as well as one scan field for an entire layer, together with the material Inconel 718 (Pottlacher et al. 2002; Special Metals Corporation 2007), were used. The number of heating steps ensured that both the sudden energy input and the further propagation of the thermal energy within the component are represented. The configuration with multiple scan fields produces similar results and is therefore not presented for the sake of clarity. Since the 'Cooling' phase is not targeted by the LBM-DSS, it is not considered in any evaluations that follow in this chapter.

All geometries indicate a similar dependence of the spectral radius on the reference temperature (cf. Figure 6.6). The convergence of the introduced splitting approach is therefore considered to be independent of the underlying geometry. This also explains why no further findings were obtained when several scan fields were used, since these correspond to a deviation in the geometry. The smallest mean spectral radius results for values around the 900 K range. At the boundary areas, i. e. preheating temperature and solidus temperature, the spectral radius increases. In the preheating temperature range, the mean spectral radius exceeds 1 and therefore causes divergence. The standard deviation is negligibly small for reference temperatures between 850 K and the solidus temperature. For the selection of the reference temperatures close to the preheating temperature, on the other hand, increasingly larger standard deviations result. The reason for this is the iteration for the first time step in a new layer (cf. Figure 6.7). The start vector $\boldsymbol{T}_0^m$ is the temperature of the previous step $\boldsymbol{T}^{m-1}$. Due to the previous cooling of the layer, this temperature almost corresponds to the preheating temperature. Accordingly, the matrix $\boldsymbol{A}_{\text{const}}^m$ with reference temperatures T_{fix} in this range initially approximates the matrix $\boldsymbol{A}^m\left(\boldsymbol{T}_1^m\right)$ very well, resulting in a small spectral radius for $\boldsymbol{M}_1^m$. However, the correct temperature gradient is formed over i iterations, which is why the approximation worsens

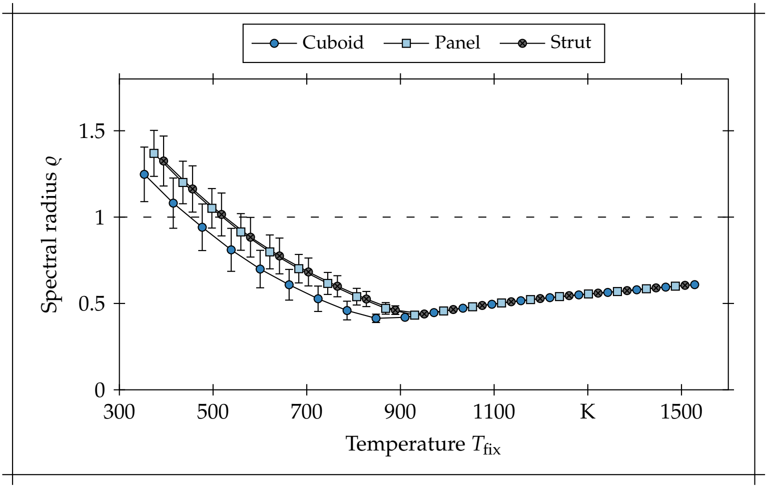


Figure 6.6: Spectral radius of $\boldsymbol{M}_i^m$ averaged over all iterations i as a function of the reference temperature T_{fix} and the associated standard deviation; 20 values of T_{fix} were evaluated, equidistantly distributed between preheating temperature and solidus temperature; material: Inconel 718.

and the spectral radius also increases.

If convergence occurs, the solution of Equation 6.7 converges against the solution of Equation 2.71 with deviations in the range of the selected termination criterion. However, the convergence of the initial Equation 2.71 and a selection of the reference temperature with a spectral radius smaller than 1 for the iteration matrix does not directly result in the convergence of the LBM-DSS. For the two iterations i and j, the limits

$$\lim_{j\to\infty} \boldsymbol{T}_{i-1,j}^m = \boldsymbol{T}_i^m, \tag{6.20a}$$

$$\lim_{i\to\infty} \boldsymbol{T}_i^m = \boldsymbol{T}^m \tag{6.20b}$$

apply. Since only one step is performed for iteration j, there is a possibility that $\boldsymbol{T}_{i-1,1}^m$ approaches $\boldsymbol{T}_i^m$, but the distance to $\boldsymbol{T}^m$ increases and is outside the convergence radius for iteration i. This would be avoided by multiple steps of sub-iteration j. In this case, assuming that iteration i converges, a spectral radius smaller than 1 would be directly sufficient for the convergence of the LBM-DSS. Nevertheless, the LBM-DSS was chosen with only one step for iteration j. CalculiX CrunchiX provides extensive methods for checking convergence, which are directly accessible by merging iterations i and j, but not for solely j. Since, in this thesis, the Discretization and Algorithm Selection is in

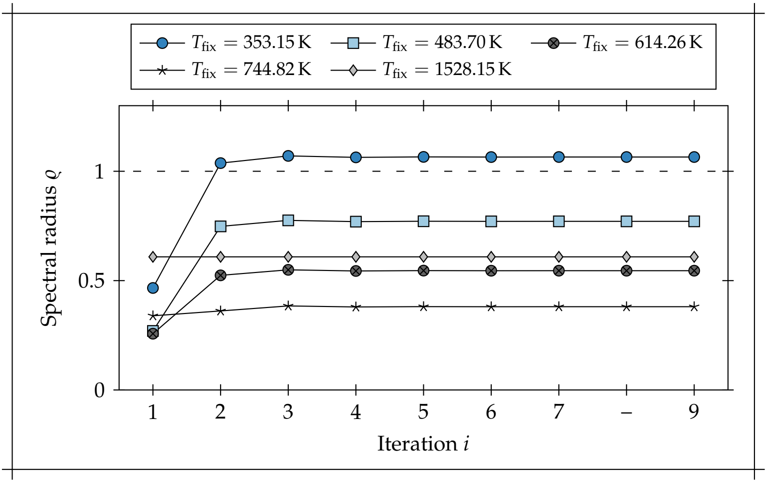
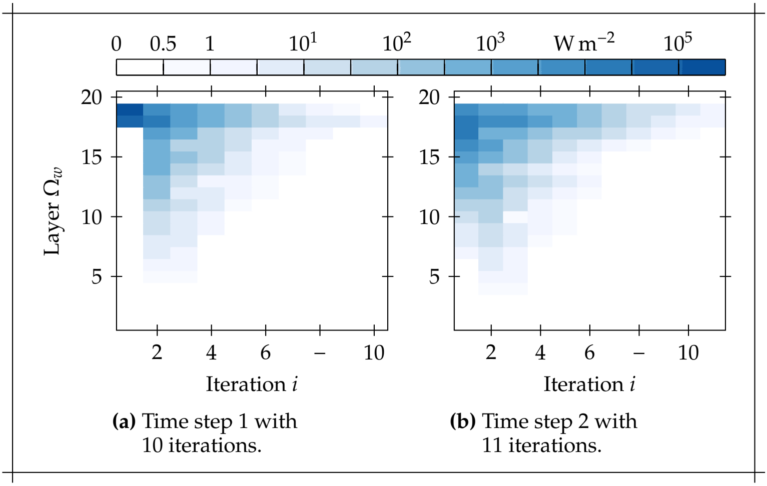


Figure 6.7: Spectral radius of $\boldsymbol{M}_i^1$ for i iterations of the first time step in the first layer for various reference temperatures T_{fix}; material: Inconel 718; part: Cuboid.

the foreground, rather than the implementation, existing routines were reused as much as possible. These are already verified and optimized with regard to computing time. Furthermore, the results in the next subsection demonstrate that a single step is sufficient for iteration j. To check the convergence of the sequence $\boldsymbol{T}_{i-1,j}^m$, two to three iterations would be necessary in addition to each iteration i and, in sum, more than the number resulting from using only one iteration. With the aim of reducing computing time in this thesis, one iteration for j thus represents the preferred method.

Divergence despite a spectral radius of the iteration matrix $\boldsymbol{M}_i^m$ smaller than 1 was only observed in the first time step when adding a new layer. However, it is also possible to counteract this by the choice of the reference temperature. Due to the use of the temperature vector from the previous time step of the 'Cooling' phase as the start vector $\boldsymbol{T}_0^m$, it is farthest away from $\boldsymbol{T}^m$. Figure 6.8 illustrates this with the representation of the maximum residual flux in each layer over the iterations for each of the first two time steps. The uppermost layer does not provide any changes, because the temperature there is given by the Dirichlet boundary condition. As mentioned, the highest value of the residual flux is obtained in the first iteration of the first time step. In addition, the first time step shows that the residual flux initially occurs only in the two layers below the newly activated one (Figure 6.8a). The divergence can be avoided by selecting a reference temperature which better corresponds to the target temperature in this range. With the choice $T_{\text{fix}} > 1073.15\,\text{K}$, convergence is always obtained. In addition, it becomes apparent again why the preheating temperature as reference temperature is initially a good choice for the matrix A^{m}_{const}.



(a) Time step 1 with 10 iterations.

(b) Time step 2 with 11 iterations.

Figure 6.8: Maximal absolute residual flux per iteration and layer for the heating steps 1 and 2 in layer 20; the values are color coded with a linearized logscale between 0 and 1, i. e. a logarithmic scale is used except for the values between 0 and 1, for which a linear scale is used; $T_{\text{fix}} = 873.15\,\text{K}$; material: Inconel 718; part: Cuboid.

6.3.2 Partial Cholesky decomposition

Yet to be shown are the structure of the Cholesky decomposition L^{m-1} from A^{m-1}_{const} in Equation 6.13 and the coherence between L^{m-1} and L^{m} in Equation 6.14. Assuming a general Cholesky decomposition L of a matrix A with the corresponding structure from Equations 6.10, i. e.

$$A = \begin{pmatrix} A_1 & {A_{21}}^{\top} & \mathbf{0} \\ A_{21} & A_2 & {A_{32}}^{\top} \\ \mathbf{0} & A_{32} & A_3 \end{pmatrix}, \qquad L = \begin{pmatrix} L_1 & \mathbf{0} & \mathbf{0} \\ L_{21} & L_2 & \mathbf{0} \\ L_{31} & L_{32} & L_3 \end{pmatrix}, \tag{6.21}$$

the coherence

$$A_1 = L_1 \cdot {L_1}^{\top} \tag{6.22}$$

results. Thus, $\boldsymbol{L}_1$ is the Cholesky decomposition of matrix $\boldsymbol{A}_1$; moreover, $\boldsymbol{L}_1$ is a regular matrix. From this,

$$\mathbf{0} = \boldsymbol{L}_{31} \cdot \boldsymbol{L}_1{}^{\top} \qquad \Rightarrow \qquad \mathbf{0} = \boldsymbol{L}_{31} \tag{6.23}$$

follows, and therefore the proof for the structure in Equation 6.13. Equation 6.22 demonstrates that the Cholesky decomposition of $\boldsymbol{A}_1$ is independent of the other matrices in the block representation. Accordingly, it follows from

$$\boldsymbol{A}_{21} = \boldsymbol{L}_{21} \cdot \boldsymbol{L}_1{}^{\top} \quad \Rightarrow \quad \boldsymbol{L}_{21} = \boldsymbol{A}_{21} \cdot \left(\boldsymbol{L}_1{}^{\top}\right)^{-1} \tag{6.24}$$

that the share $\boldsymbol{L}_{21}$ of the Choleksy decomposition is only dependent on $\boldsymbol{A}_1$ and $\boldsymbol{A}_{21}$. Transferred to $\boldsymbol{L}^m$ and the coherence in Equations 6.10, matrices $\boldsymbol{L}_1^{m-1}$ and $\boldsymbol{L}_{21}^{m-1}$ apply accordingly to the decomposition in Equation 6.13.

6.4 Validation

The range from 1073.15 K to solidus temperature has been verified in the previous section as permissible for the parameter T_{fix} of the LBM-DSS and is now validated in this section with regard to computing time. The LBM-DSS is examined concerning the rate of convergence, computing time and a further acceleration of the computing time by additional requirements to the sorting of the node numbering. The validation of the LBM-DSS takes place with the configurations already presented in the verification. This means that two heating steps and a scan field were used for the geometries presented (cuboid, panel and strut). In each case, the conventional procedure of Subsection 2.5.4 with the standard Cholesky method is used as a reference. This entire procedure is referred to below as 'Standard Cholesky'.

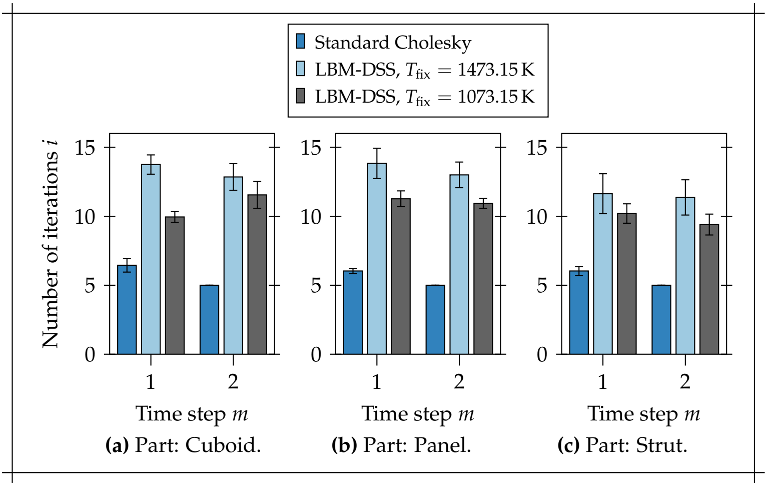
The LBM-DSS was implemented in ccx. The modified source code is available at the media and publications repository of the Technical University of Munich (BANTE et al. 2019). For pre- and post-processing, the routines of ccx are used. The solving of the actual system of equations, i. e. the decomposition of the matrix, as well as forward and backward substitution, is performed with custom routines. This applies both to the calculations with the method and to the determination of the reference solution without the method. Existing routings in ccx were deliberately not used for this purpose, as the focus is on validating the LBM-DSS. Falsification of the results, e. g. by a special decomposition or parallelization, is thus prevented.

6.4.1 Rate of convergence

The literature states that as the spectral radius of the coefficient matrix becomes smaller, the splitting methods converge faster, as the spectral radius represents an upper limit for the convergence rate (SAAD 2003). Based on the results from Figure 6.6, it is therefore assumed that the LBM-DSS with $T_{\text{fix}} = 1073.15\,\text{K}$ leads to the fewest iterations and thus to the shortest calculation time. As mentioned in the previous section, the LBM-DSS only executes one iteration of the splitting procedure. It is therefore important to check whether this statement is valid. The results reveal a linear curve of the spectral radius (cf. Figure 6.6) in the range of the permissible values for the reference temperature T_{fix}. Therefore, the values $T_{\text{fix}} = 1073.15\,\text{K}$ and $T_{\text{fix}} = 1473.15\,\text{K}$ at the margins are used for the validation.

The rate of convergence describes the relative decrease of the deviation of a sequence from the limit in an iteration and thus also the required number of iterations. First, the number of iterations for different geometries is evaluated. Figure 6.9 provides the average number of iterations required for each heating step averaged over all layers. For the respective solvers, only small deviations occur in the necessary iterations for the different geometries. The Standard Cholesky requires the least number of iterations. The highest number of iterations results for the LBM-DSS with $T_{\text{fix}} = 1473.15\,\text{K}$. The assumption of faster convergence due to the smaller spectral radius is therefore still valid. Compared to the Standard Cholesky, the LBM-DSS requires approximately 2 to 2.5 times more iterations. This also shows the result, presented in Subsection 6.3.1, that a single iteration for the splitting procedure leads to a similar total number of iterations as a full iteration. The use of a single iteration is therefore sufficient for the splitting approach.

Similar to the number of iterations, the development of the residual is almost independent of the geometry and the heating step. The maximum residual in each iteration for the second heating step in the second layer of the panel geometry in Figure 6.10 is therefore representative for the remaining geometries and time steps. All approaches to solving the system of nonlinear equations show linear convergence, but with a different rate of convergence. The average convergence rate for the Standard Cholesky is 0.11. A convergence rate of 0.41 ($T_{\text{fix}} = 1073.15\,\text{K}$) or 0.51 ($T_{\text{fix}} = 1473.15\,\text{K}$) applies to the LBM-DSS. Taking into account the corresponding spectral radii of 0.48 ($T_{\text{fix}} = 1073.15\,\text{K}$) and 0.59 ($T_{\text{fix}} = 1473.15\,\text{K}$), these therefore still represent an upper limit for the convergence rate (cf. Equation 2.50).



(a) Part: Cuboid. **(b)** Part: Panel. **(c)** Part: Strut.

Figure 6.9: Required iterations for the Standard Cholesky and the LBM-DSS with different reference temperatures and various geometries; material: Inconel 718.

6.4.2 Calculation time

The previous section demonstrated that, although the LBM-DSS has linear convergence like the Standard Cholesky, it requires 2 to 2.5 times as many iterations. However, considering the calculation time required, it is apparent that the LBM-DSS is nevertheless significantly faster than the Standard Cholesky (cf. Figure 6.11). Even for geometries with a small number of nodes, such as the cuboid, the LBM-DSS results in a reduction of computing time by more than 93 %. For meshes with a significantly higher number of DoFs, such as the strut, computing times are achieved which correspond to approximately 2 % of those of the standard procedure.

As for the LBM-MM, all simulation runs were performed on a virtual machine with SUSE Linux, 16 cores and 32 GB RAM on a Windows workstation. Again, the results with the lowest computing time are used, since these are the calculations with the least secondary activities of the workstation and thus best represent the actual computing time for the calculation itself. The node numbering from Subsection 6.2.2 was used both for the simulations with the Standard Cholesky and with the LBM-DSS.

Splitting methods have a slow convergence compared to other methods. The presented LBM-DSS, however, allows the avoidance, or reduction for new areas, of the costly Cholesky decomposition. The effort for solving an iteration with

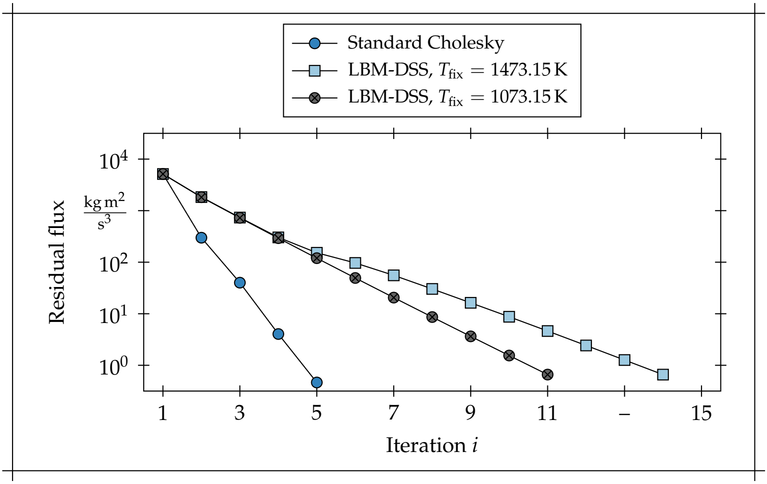


Figure 6.10: Maximal absolute residual flux per iteration in heating step 2 of layer 2; material: Inconel 718; part: Panel.

the Standard Cholesky consists of the decomposition of the coefficient matrix $\boldsymbol{A}^m \in \mathbb{R}^{n_\Xi \times n_\Xi}$ and the subsequent forward and backward substitution. Due to the sparse implementation, only non-zero entries of vectors and matrices are considered in the calculation. Therefore, it is not possible to deduce the effort directly from the matrix size. However, the Cholesky decomposition partially fills the lower triangular matrix within diagonals and the farthest point of the respective row, although the corresponding entries of the coefficient matrix are zero. For this reason, the bandwidth k_d is used to approximate the effort. The decomposition of the coefficient matrix requires $\frac{n_\Xi {k_d}^2}{6}$ operations. In addition, $\frac{n_\Xi k_d}{2}$ operations are necessary for forward and backward substitution. With an assumed iteration number of $i_{\text{total}} \in \mathbb{N}$, the total effort of the Standard Cholesky is

$$i_{\text{total}} \left(\frac{n_\Xi {k_d}^2}{6} + \frac{n_\Xi k_d}{2} \right). \tag{6.25}$$

For the LBM-DSS, the decomposition of the coefficient matrix is only necessary in the first iteration. Assuming a greater number of iterations required by the factor $c_{\text{fac}} \in \mathbb{R}$ compared with the standard method, the total effort is

$$\frac{n_\Xi {k_d}^2}{6} + c_{\text{fac}} i_{\text{total}} \left(\frac{n_\Xi k_d}{2} \right). \tag{6.26}$$

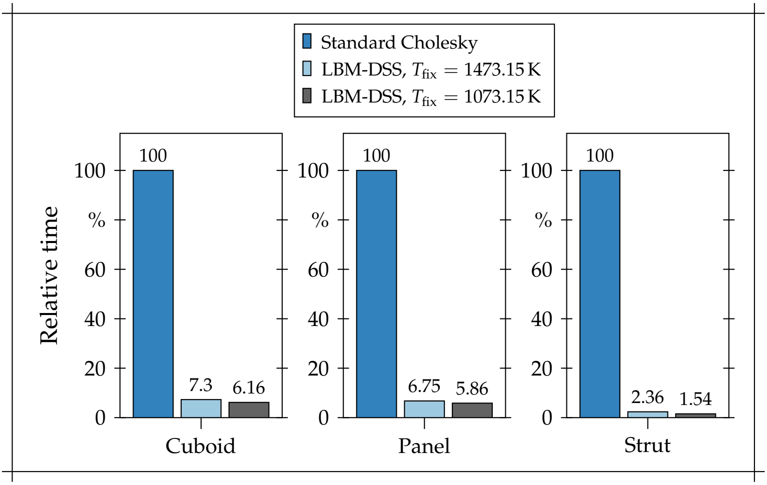


Figure 6.11: Relative calculation times for the Standard Cholesky and the LBM-DSS with different reference temperatures and various geometries; 2 heating steps per layer; material: Inconel 718.

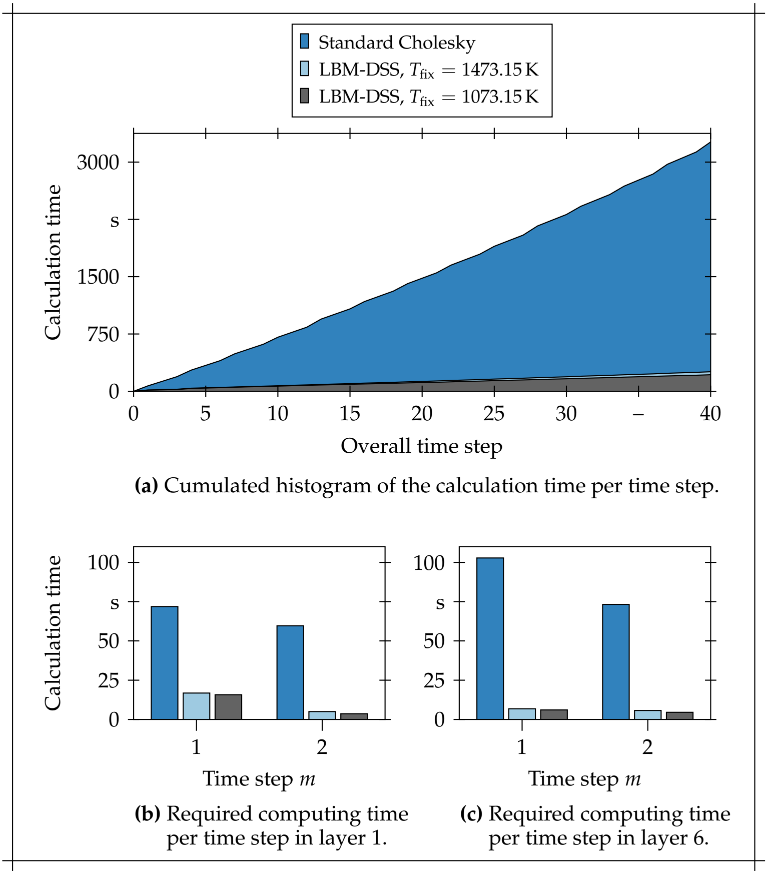
This results in a lower effort for the LBM-DSS compared to the Standard Cholesky as long as the following applies:

$$c_{\text{fac}} < \frac{(i_{\text{total}} - 1)\,\frac{k_d}{3} + 1}{i_{\text{total}}}. \tag{6.27}$$

With bandwidths that are usually significantly larger than the number of iterations i_{total}, this clearly exceeds the value of 2 to 2.5 received for c_{fac} (cf. Figure 6.9).

The previous statement considers only the case of the very first time step, in which the decomposition of the coefficient matrix for the Standard Cholesky and the LBM-DSS are identical. For time steps, in which only a partial Cholesky decomposition takes place or no decomposition at all is necessary, the effort is once again lower compared to the Standard Cholesky. This becomes apparent when considering the required computing times for the individual steps, which are illustrated in Figure 6.12.

The increase in computing time is steeper for the Standard Cholesky than for the LBM-DSS (Figure 6.12a). Already in the first time step, where the same decomposition takes place, the LBM-DSS is significantly faster (Figure 6.12b). If only partial decomposition is carried out, a further reduction is evident (Figure 6.12c). While the effort for the decomposition due to the growing area



(a) Cumulated histogram of the calculation time per time step.

(b) Required computing time per time step in layer 1.

(c) Required computing time per time step in layer 6.

Figure 6.12: Distribution of the calculation time among the time steps; 2 heating steps per layer; material: Inconel 718; part: Cuboid.

also increases with the Standard Cholesky, it remains roughly the same with the LBM-DSS.

6.4.3 Improved node numbering

With regard to the effort in Equation 6.26 to solve a system of equations, it becomes apparent that the matrix structure, or more precisely the bandwidth, has a significant influence on it. This applies to both decomposition and substitution. In addition to the node numbering introduced in Subsection 6.2.2, which is necessary for the application of the method, an enhancement is presented to further reduce the bandwidth of the coefficient matrix. In comparison to the already introduced node numbering, the set of nodes not shared by two layers is further differentiated. Starting from the lowest layer, all nodes of a layer are numbered in ascending order according to the following procedure:

1. Numbering of nodes that are not shared with other layers.
    a) Numbering of nodes that are connected to nodes shared with the previous layer.
    b) Numbering of nodes that are not connected to nodes shared with the previous layer or to nodes shared with the subsequent layer.
    c) Numbering of nodes that are connected to nodes shared with the subsequent layer.
2. Numbering of nodes that are shared with the subsequent layer.

A comparison of the different ordering strategies is illustrated in Figure 6.13. Besides the node numberings presented in this thesis, the sorting of NIKOUKAR et al. (2013) is provided.

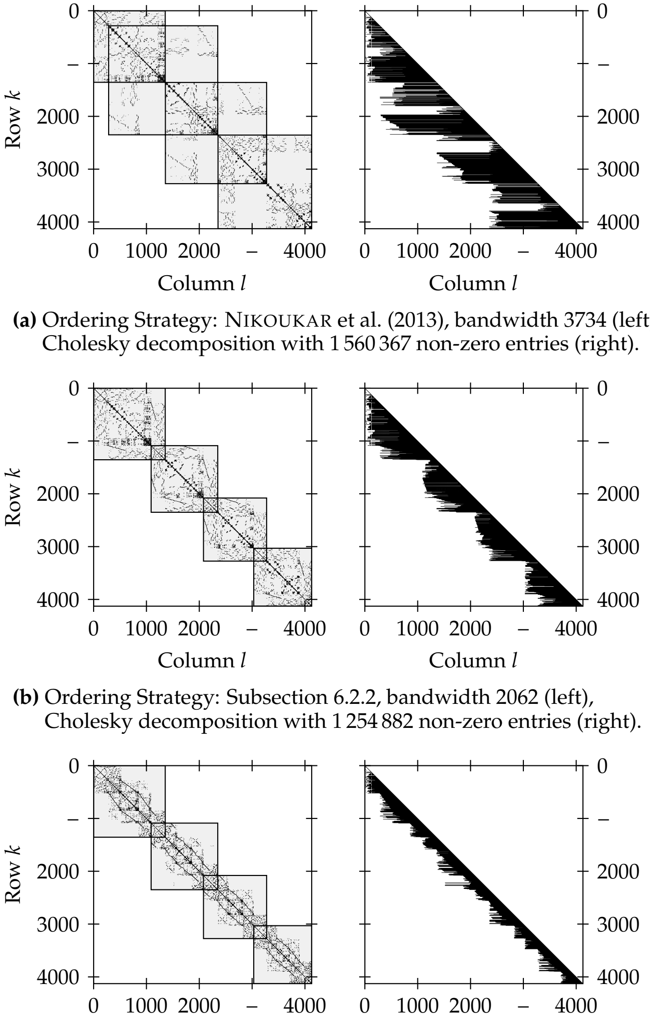
Figure 6.13 illustrates how the ordering strategy from Subsection 6.2.2 first reduces the overlap of two subsequent layers, and, as a consequence, the numbering of each layer becomes more compact. The strategy presented in this subsection then affects the bandwidth of each layer and consequently the bandwidth of the entire coefficient matrix. The sorting is accompanied by a bandwidth reduction by almost 60 % and more than 35 % fewer non-zero entries in the lower triangular matrix of the Cholesky decomposition.

Figure 6.14 contains the comparison of the computing time for the different ordering strategies. Apart from that, the results were determined under the same configurations for the simulation runs as in the validation in Subsection 6.4.2. With the sorting of the node numbering presented in this subsection, a further reduction of the computing time by almost 10 % was achieved. The Standard Cholesky even achieves a saving of 12.1 % of computing time.

6.5 Summary

In this chapter, the LBM-DSS for solving of the system of nonlinear equations obtained in the 'Heating' phase of the simulation of LBM has been presented. The LBM-DSS exploits the potential of Direct Methods by reusing the decomposition of the coefficient matrix and considering the layered mesh structure.

A splitting approach initially ensures that the coefficient matrix to be solved no longer depends on the temperature of the current time step. Due to the



(a) Ordering Strategy: NIKOUKAR et al. (2013), bandwidth 3734 (left), Cholesky decomposition with 1 560 367 non-zero entries (right).

(b) Ordering Strategy: Subsection 6.2.2, bandwidth 2062 (left), Cholesky decomposition with 1 254 882 non-zero entries (right).

(c) Ordering Strategy: Subsection 6.4.3, bandwidth 1506 (left), Cholesky decomposition with 1 003 836 non-zero entries (right).

Figure 6.13: Sparsity pattern of $\boldsymbol{A}$ (right) and corresponding Cholesky decompositions $\boldsymbol{L}$ (left) for different ordering strategies, exemplarily illustrated for layers 5 to 8; left: nodes of a layer are highlighted (gray); part: Strut.

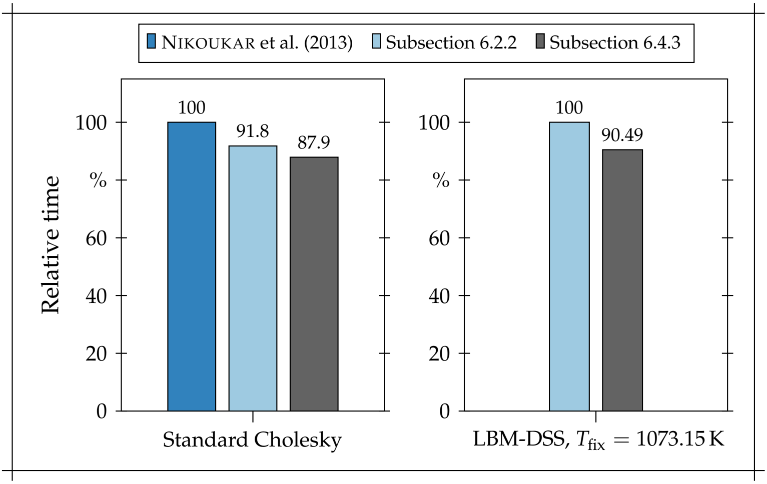


Figure 6.14: Relative calculation times obtained for different ordering strategies; 2 heating steps per layer; material: Inconel 718; part: Strut.

additional consideration of the layered mesh structure in the presented node numbering, the decomposition of the previous time step remains valid in the already present layers if an additional layer is activated. As a consequence, only a partial decomposition of the coefficient matrix is necessary in this case. Although this increases the number of iterations, the overall effort is significantly lower than the usual procedure with a full Cholesky decomposition and fewer iterations. However, whether the LBM-DSS converges, depends on the introduced parameter T_{fix}. With $T_{\text{fix}} > 1073.15\,\text{K}$, a range was determined which always ensured convergence. In the case of the strut, the calculation time was reduced by more than 98 % by using the LBM-DSS.

At this point it should be mentioned that, though very high reductions of the computation time were achieved using the LBM-DSS compared to the Standard Cholesky, solvers are already available which are clearly faster than the Standard Cholesky. The solver PARDISO from the Intel MKL, usable with ccx, for example, provides calculation times below the presented LBM-DSS. Among other things, it exploits the memory hierarchy of the processor architecture, as well as algorithms for parallelization. However, the presented LBM-DSS constitutes both a solver on its own and a solving strategy. PARDISO can also be used for the subtasks of the LBM-DSS, i. e. for the partial matrix decomposition, as well as for the forward and backward substitution. The computing time advantages of PARDISO are directly transferable to the individual subtasks. The achievable explicit calculation time savings, however, depend on the ratio of decomposition to forward and backward substitution of PARDISO.

7 Conclusion

7.1 Summary

As a result of selective solidification by a laser, thermally induced residual stresses and distortions outside the tolerance occur in Laser Beam Melting (LBM). One measure to counteract is to compensate for the distortions by pre-deformation. For this purpose, the component is manufactured and measured and the distortions are applied inverted to the geometry. Due to non-linearity, several of these iterations are necessary. This procedure is not only cost-intensive due to the personnel, material and system demand, but also is not feasible for areas of the part that are not accessible for measuring instruments. For this reason, simulation programs are currently being used to simulate the buildup process and predict warpage. However, due to the different temporal and spatial scales, as well as the physical phenomena involved in LBM, the computation times of detailed models are very long. To accelerate the simulation, different modeling assumptions are made, which in turn reduce the quality of the results. Initial approaches have already shown that Discretization and Algorithm Selection according to Figure 1.2a is capable of drastically reducing computing time without simplifying the model.

This thesis has therefore explored the adaptation of numerical methods for the simulation of the temperature field during the buildup process in LBM. The aim was to enable the use of detailed Conceptual Models and thus to further increase the significance of the simulation. These models require both a finer temporal and spatial discretization. This results in larger systems of equations, which also have to be solved more frequently.

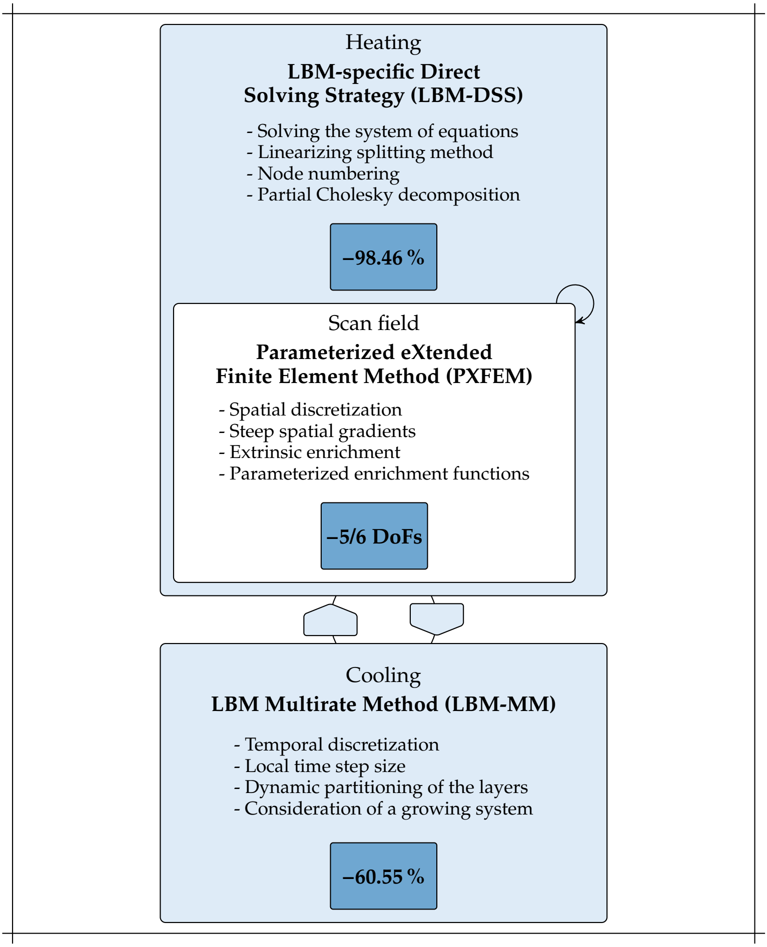
The simulation of LBM consists of the two alternating phases 'Heating' and 'Cooling', in which the thermal load is successively applied to the various scan fields of a layer. The layer then cools down. In addition, the entire part is cooled from preheating temperature to room temperature after the manufacturing process of all layers. Compared to the other two phases, however, this occurs only once at the end, and its computing time is therefore negligible. The numerical consideration of the unsteady heat equation includes the spatial and temporal discretization, as well as the solution of the resulting system of equations. For this thesis, independent numerical methods for all three steps were developed for the special requirements of LBM. These address the greatest challenges resulting from the use of more detailed models. The developed methods also target the different phases of modeling.

The Parameterized eXtended Finite Element Method (PXFEM) was developed for the energy input into a scan field, which avoids a fine spatial discretization. This is the use of the eXtended Finite Element Method (XFEM) with parameterized enrichment functions that disappear on the boundaries of the elements ($v|_{\partial\epsilon_u} = 0$). The shape functions of the Finite Element Method (FEM) therefore continue to determine the nodal values, while the enrichment functions compensate for the inadequacies of the linear elements in the representation of steep gradients. The parameterization variable was transferred to the system of equations as an additional Degree of Freedom (DoF) and solved in conjunction with the values of the nodes. The enrichment with several approach functions for the representation of a temporal progression was therefore not necessary, since this was done via parameterization. In the considered cases a mesh refinement by a factor of 6 was necessary for the FEM to achieve a smaller error than the presented PXFEM.

The LBM Multirate Method (LBM-MM) addressed the temporal discretization during the cooling of each layer. Layers were dynamically partitioned into components with different time steps, taking into account the additional layers throughout the buildup process. The method achieved a reduction of the computing time by up to 60.55 %.

The LBM-specific Direct Solving Strategy (LBM-DSS) allowed the reuse of the Cholesky decomposition despite a system of nonlinear equations, and thus exploited one of the greatest advantages of Direct Methods. Using a splitting approach, the temperature-dependent part of the coefficient matrix was transferred to the right side of the system of equations. Due to the consideration of the layer affiliation of nodes in the numbering, only partial decomposition was necessary when adding new areas. Thus, the LBM-DSS addresses not only the heating of individual layers, but the method is a holistic approach that takes into account the 'Heating' phase of all layers. The overall calculation time of the 'Heating' phase was reduced by almost 99 % in the application example.

A summary of the methods and results is provided in Figure 7.1. These results are based on the consideration of the material parameters for Inconel 718, as well as the machine data of the manufacturing system EOS M270, and are therefore not generally valid. However, the procedure used to determine the results is generally valid, and the methods developed are applicable with arbitrary material parameters and machine data.



Heating

LBM-specific Direct Solving Strategy (LBM-DSS)

- Solving the system of equations
- Linearizing splitting method
- Node numbering
- Partial Cholesky decomposition

−98.46 %

Scan field

Parameterized eXtended Finite Element Method (PXFEM)

- Spatial discretization
- Steep spatial gradients
- Extrinsic enrichment
- Parameterized enrichment functions

−5/6 DoFs

Cooling

LBM Multirate Method (LBM-MM)

- Temporal discretization
- Local time step size
- Dynamic partitioning of the layers
- Consideration of a growing system

−60.55 %

Figure 7.1: Summary of the results and achievements of the developed methods for Inconel 718 and an EOS M270 manufacturing system.

7.2 Economic Evaluation

While the technical evaluation of the methods took place in the respective verification and validation sections, a final economic evaluation is carried out. This includes the LBM-MM and LBM-DSS. The PXFEM is not considered in this context. Although it was shown that a reduction in the DoFs was achieved, no statement could be made about the associated reduction in the computing time due to the low degree of maturity. An economic classification is therefore currently not possible. Furthermore, the term 'CalculiX CrunchiX (ccx) Additive Manufacturing (AM)' includes the adapted source code of ccx (BANTE et al. 2019), which contains the implementation of both methods, as well as the program[34] for creating an input script for the simulation of LBM.

The distortion simulation of LBM is used in process design and partially replaces the necessary iterations of test production to determine the correct adaptation of the geometry and to compensate for the process-inherent distortions. Although the developed numerical methods allow the use of more detailed Conceptual Models and thus the saving of further iterations, an economic evaluation is carried out only with regard to the saving of computing time, and not with regard to the avoidance of test productions. The economic evaluation thus follows the primary goal of this thesis, which was to accelerate the simulation.

A cost comparison is made on the basis of the hourly rates. In addition to accelerating the computing procedure, the aim of this thesis was the freely accessible dissemination of the findings and the methods developed. Therefore, the use of *Free Software*[35] was a central component, since existing simulation models are currently mainly based on commercial programs. Proprietary software therefore serves as a comparison.

The values used for the cost comparison are listed in Table 7.1. In addition to the data for the software, the data for the required hardware is also listed. A workstation is mentioned, whereby simulations are also executable on a personal computer. The price already includes the necessary peripherals such as a screen, a mouse, a keyboard and sufficient memory. All information refers to customary values. Some are nevertheless explained in detail.

The indicated depreciation periods refer to the depreciation table for generally usable fixed assets of the German Federal Ministry of Finance (BUNDESMINISTERIUM FUER FINANZEN 2000). For the two software programs, an availability of 100 % is indicated, but maintenance work in the form of e. g. updates is

[34] `https://gitlab.lrz.de/flayerbein/ascent.git`

[35] "users have the four essential freedoms: (0) to run the program, (1) to study and change the program in source code form, (2) to redistribute exact copies, and (3) to distribute modified versions" (FREE SOFTWARE FOUNDATION 2018; STALLMAN & GAY 2002)

Table 7.1: Data used for cost comparison of the software solutions and hardware.

Designation	Unit	Proprietary software	ccx AM	Work-station
Purchase price	EUR	100 000	0	7000
Depreciation period	a	5	5	3
Replacement value	%	0	0	8
Availability	%	100	100	95
Imputed interest rate	%	2.75	2.75	2.75
Floor space required	m^2	-	-	2.5
Monthly rent	EUR/m^2	-	-	16.9
Nominal capacity	kW	-	-	0.7
Electricity price	EUR/kW	-	-	0.3
Power exploitation	%	-	-	90
Relative maintenance costs	%	10	0	10
Fixed maintenance costs	EUR/a	0	6000	0

also necessary, and they are not available at all times. Usually, however, these processes go hand in hand with the maintenance of the workstation and are therefore taken into account via its availability. The imputed interest rate refers to the value set by the Munich City Council for 2019 (CITY COUNCIL MUNICH 2018). The area of the required workplace is composed of the minimum size of the movement area while sitting and standing (1.5 m^2) and the rounded minimum size of the usable work area (0.96 m^2) (FEDERAL INSTITUTE FOR OCCUPATIONAL SAFETY AND HEALTH 2013). If the workstation is only used via remote access, the space requirement must be reduced accordingly. In this case, labor law regulations are no longer to be observed. The stated monthly rental costs per square meter correspond to the average value for office space in Munich (STATISTA GMBH 2018). Usually, the maintenance costs for hardware and software refer to the purchase price. In the case of free software, a common business model is to provide it free of charge. Consulting, training and customization are then provided for a fee. An amount of 6000 EUR a^{-1} is applied here. However, due to the open source code, free software also offers the potential to build up personal know-how and to perform services by oneself.

Since the simulation is usually started automatically, its use is not limited by weekends or holidays. Thus, the availability of 95 % results in 8322 h/a for the service life of the software. Together with the values in Table 7.1, this results in the annual costs listed in Table 7.2. Considering the costs of the workstation required, the implementation in a freely available software with open source code results in a reduction of the annual costs up to 38.96 % compared to the

Table 7.2: Annual breakdown of the costs for a proprietary software, the free open source software ccx AM and a workstation.

Designation	**Unit**	**Proprietary software**	**ccx AM**	**Work-station**
Imputed depreciation	EUR/a	20 000	0	2652.63
Imputed interest	EUR/a	1375	0	103.95
Space costs	EUR/a	0	0	507
Maintenance costs	EUR/a	2000	6000	252
Energy costs	EUR/a	0	0	1572.86
Total costs	**EUR/a**	**23 375**	**6000**	**5088.44**

proprietary software.

The efficiency of the simulation is now considered for further evaluation of the developed methods. In addition to the calculation time for the thermal analysis, this also includes the calculation time for the mechanical analysis. A general statement is not possible, since the distribution of the individual times depends strongly on the considered part. Under certain assumptions, however, relative statements are possible. It is assumed that the phases 'Heating' and 'Cooling' are equally distributed and each comprises half of the calculation time of the thermal calculation. The 'End Cooling' phase is neglected. The share of the mechanical analysis in the total computing time is essentially larger, since each node has three DoFs compared to one in the thermal analysis. Due to the known temperature, there are no nonlinearities as a result of the temperature-dependent material parameters. However, nonlinearities arise due to plasticity. The different nonlinearities of the thermal and mechanical analysis lead to varying proportions of the total computing time. In most cases, however, the proportion of the thermal calculation lies between 15 % and 35 %. Based on the assumptions made and within this interval, the total computing time is reduced by 4.54 % to 10.60 % when using the LBM-MM and by 7.43 % to 17.33 % when using the LBM-DSS (cf. Figure 7.2). Including the difference of the annual costs, this means that the costs for the calculation of one part with the LBM-MM are 65.17 %, and with the LBM-DSS 67.79 %, cheaper compared to the use of proprietary software.

7.3 Outlook

Although both the LBM-MM and the LBM-DSS are already integrated into the FEM program ccx, they represent independent methods in the implementation

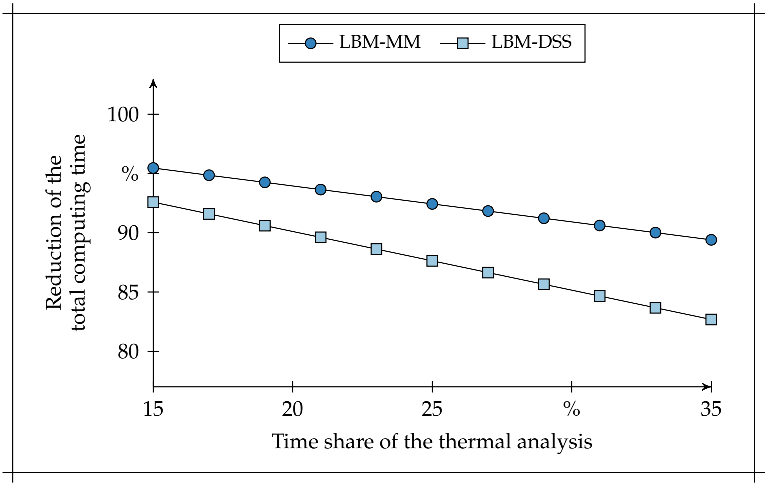


Figure 7.2: Reduction of the total computing time by the LBM-MM and the LBM-DSS.

thus far. Here it is necessary to find approaches to unite the two methods. Since the LBM-DSS uses the Cholesky decomposition from the previous heating step, it is necessary to efficiently provide this despite of an interim 'Cooling' phase. This especially concerns phase 4, 'Computer Programming of the Discrete Model', in the classification of Modeling and Simulation (M&S) by OBERKAMPF et al. (2002) (cf. Figure 1.2). In addition, the integration of the solver PARDISO available in ccx into the LBM-DSS promises further savings in computing time.

The PXFEM revealed a significant reduction in the DoFs in the one-dimensional case. Even if the refinement of elements in the multi-dimensional case is accompanied by a proportionally higher increase in DoFs, and the application of the PXFEM thus holds even greater potential, the applicability has yet to be proven. For this purpose, the degree of maturity must be increased and suitable enrichment functions have to be identified.

The developed methods are a further step towards a simulation of LBM that takes into account all effects that contribute to dimensional deviations (cf. Figure 1.1). The next step must be to reduce the calculation time of the mechanical analysis following the temperature field calculation. On the one hand, adapted numerical methods must be developed. However, with increasingly temporally and locally detailed temperature fields, the naive application of temperatures as loads no longer seems efficient, especially in view of the fact that, in mechanics with the displacements in all three spatial directions, a threefold number of

DoFs results as compared to the temperature field calculation. Therefore, new Conceptual Models for the mechanical part of the simulation are required. As stated in the introduction, the thermomechanical model and the model based on the inherent strain method jointly cover all effects. One possible future approach is to combine both models. The distortions resulting from the melting in the upper part of the component are covered by the inherent strain method. The calculated transient temperature field is only used for deeper layers. Thus, it is only necessary to use the time step, at which the penetration depth has reached the respective uppermost layer. This goes hand in hand with a significantly reduced number of time steps used, since the fine temporal resolution is necessary for the top layers in particular and is therefore not required for the lower portion of the part.

Also conceivable is an additional step preceding the mechanics which analyzes the entire transient temperature field of each heating-cooling cycle and determines a local strain tensor for each area of the component on the basis of the temperature gradients. In addition to the initial application of the inherent strain when activating the layer, the update is carried out when adding each new layer, and the heat flux due to the solidification of the new layer is taken into account. This model would mean a significant saving of computing time for the mechanical analysis, since only one calculation step is necessary for each layer and it would be, at the same time, complete with regard to the occurring effects that contribute to the dimensional deviations in LBM (cf. Figure 1.1).

Bibliography

ABBAS et al. 2010
ABBAS, S.; ALIZADA, A.; FRIES, T.-P. (2010): The XFEM for high-gradient solutions in convection-dominated problems. In: International Journal for Numerical Methods in Engineering 82.8, pp. 1044–1072. ISSN: 0029-5981. DOI: 10.1002/nme.2815.

ALI & L. C. ZHANG 2005
ALI, Y. M.; ZHANG, L. C. (2005): Relativistic heat conduction. In: International Journal of Heat and Mass Transfer 48.12, pp. 2397–2406. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2005.02.003.

ALKAHARI et al. 2012
ALKAHARI, M. R.; FURUMOTO, T.; UEDA, T.; HOSOKAWA, A.; TANAKA, R.; ABDUL AZIZ, M. S. (2012): Thermal Conductivity of Metal Powder and Consolidated Material Fabricated via Selective Laser Melting. In: Key Engineering Materials 523-524, pp. 244–249. ISSN: 1662-9795. DOI: 10.4028/www.scientific.net/KEM.523-524.244.

ALTMANN et al. 2010
ALTMANN, C.; GASSNER, G.; LÖRCHER, F.; TAUBE, A.; UTZMANN, J.; MUNZ, C.-D. (2010): An Explicit Space-Time Discontinuous Galerkin Scheme with Local Time-Stepping for Unsteady Flows. In: New Results in Numerical and Experimental Fluid Mechanics VII. Contributions to the 16th STAB/DGLR Symposium. Aachen / Germany, November 3-4, 2008. Ed. by E. H. HIRSCHEL; W. SCHRÖDER; K. FUJII; W. HAASE; B. LEER; M. A. LESCHZINER; M. PANDOLFI; J. PERIAUX; A. RIZZI; B. ROUX; Y. I. SHOKIN; A. DILLMANN; G. HELLER; M. KLAAS; H.-P. KREPLIN; W. NITSCHE. Notes on Numerical Fluid Mechanics and Multidisciplinary Design 112. Springer, pp. 151–158. ISBN: 978-3-642-14242-0. DOI: 10.1007/978-3-642-14243-7_19.

ARNOLD et al. 2014
ARNOLD, M.; CLAUSS, C.; SCHIERZ, T. (2014): Error Analysis and Error Estimates for Co-simulation in FMI for Model Exchange and Co-Simulation v2.0. In: Progress in Differential-Algebraic Equations. Ed. by S. SCHOEPS; A. BARTEL; M. GUENTHER; E. J. W. TER MATEN; P. C. MÜLLER. Differential-Algebraic Equations Forum. Berlin, Heidelberg: Springer, pp. 107–125. ISBN: 978-3-662-44925-7. DOI: 10.1007/978-3-662-44926-4_6.

ASCHER et al. 1995
ASCHER, U. M.; RUUTH, S. J.; WETTON, B. T. R. (1995): Implicit-Explicit Methods for Time-Dependent Partial Differential Equations. In: SIAM Journal on Numerical Analysis 32.3, pp. 797–823. ISSN: 0036-1429. DOI: 10.1137/0732037.

BAEHR & STEPHAN 2009
BAEHR, H. D.; STEPHAN, K. (2009): Wärme- und Stoffübertragung. 6th ed. Berlin: Springer. ISBN: 978-3-540-87688-5. DOI: 10.1007/978-3-540-87689-2.

BANTE et al. 2019
BANTE, L.; ZELLER, C.; BAYERLEIN, F.; ROBL, C.; DHONDT, G.; ZAEH, M. F. (2019): AscentAM Additions to CalculiX CrunchiX 2.14 [Code book]. In: mediaTUM - Media and Publications Repository of the Technical University of Munich. DOI: 10.14459/2018mp1515158.

BARRETT et al. 1994
BARRETT, R.; BERRY, M.; CHAN, T. F.; DEMMEL, J.; DONATO, J.; DONGARRA, J.; EIJKHOUT, V.; POZO, R.; ROMINE, C.; VAN DER VORST, H. (1994): Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. Philadelphia: Society for Industrial and Applied Mathematics. 112 pp. ISBN: 978-0-89871-328-2. DOI: 10.1137/1.9781611971538.

BATHE 2006
BATHE, K.-J. (2006): Finite element procedures. 2nd ed. Boston, Massachusetts: Bathe. 1037 pp. ISBN: 978-0-9790049-0-2.

BAYERLEIN & ZAEH 2016
BAYERLEIN, F.; ZAEH, M. F. (2016): Reduction of Manufacturing-Induced Dimensional Deviations in Laser Beam Melting by Pre-Deformation. In: Proceedings of the 6th International Conference on Additive Technologies - iCAT 2016. Nuremberg / Germany, November 29-30, 2016. Ed. by I. DRSTVENŠEK; D. DRUMMER; M. SCHMIDT, pp. 208–214. ISBN: 978-961-285-537-6.

BAYERLEIN et al. 2015
BAYERLEIN, F.; ZELLER, C.; ZAEH, M. F.; WEIRATHER, J.; WUNDERER, M.; SEIDEL, C. (2015): Improving cost effectiveness in additive manufacturing – Increasing dimensional accuracy in laser beam melting by means of a simulation-supported process chain. In: Die Fachkonferenz zur Numerischen Simulation. Bremen / Germany, June 24-26, 2015. CADFEM GmbH, pp. 1–9. ISBN: 393-752-312-X.

BAYERLEIN et al. 2016
BAYERLEIN, F.; ZELLER, C.; WUNDERER, M.; WEIRATHER, J.; SCHMID, M.; SEIDEL, C.; ZAEH, M. F.; HESSERT, R.; SCHLICK, G.; UIHLEIN, T.; HOFMANN, M. (2016): Validation of modeling assumptions for the buildup simulation of laser beam melting on the basis of the residual stress distribution. In: ECCOMAS Congress 2016. VII European Congress on Computational Methods in Applied Sciences and Engineering: proceedings. Crete / Greece, June 5-10, 2016. Ed. by M. PAPADRAKAKIS; V. PAPADOPOULOS; G. STEFANOU; V. PLEVRIS. Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA), pp. 469–479. ISBN: 978-6-18828-440-1.

BELYTSCHKO & MULLEN 1978
BELYTSCHKO, T.; MULLEN, R. (1978): Stability of explicit-implicit mesh partitions in time integration. In: International Journal for Numerical Methods in Engineering 12.10, pp. 1575–1586. ISSN: 1097-0207. DOI: 10.1002/nme.1620121008.

BENEŠ & KRUIS 2018
BENEŠ, M.; KRUIS, J. (2018): Multi-time-step domain decomposition and coupling methods for nonlinear parabolic problems. In: Applied Mathematics and Computation 319, pp. 444–460. ISSN: 0096-3003. DOI: 10.1016/j.amc.2017.04.026.

BERGHEAU & FORTUNIER 2008
BERGHEAU, J.-M.; FORTUNIER, R. (2008): Finite Element Simulation of Heat Transfer. London, UK. 279 pp. ISBN: 978-184821053-0. DOI: 10.1002/9780470611418.

BIKAS et al. 2016
BIKAS, H.; STAVROPOULOS, P.; CHRYSSOLOURIS, G. (2016): Additive manufacturing methods and modelling approaches: A critical review. In: The International Journal of Advanced Manufacturing Technology 83.1-4, pp. 389–405. ISSN: 0268-3768. DOI: 10.1007/s00170-015-7576-2.

BRANNER 2011
BRANNER, G. (2011): Modellierung transienter Effekte in der Struktursimulation von Schichtbauverfahren. Institute for Machine Tools and Industrial Management. PhD Thesis. Technical University of Munich. 246. Forschungsberichte IWB. Munich: Utz. 199 pp. ISBN: 978-3-8316-4071-3.

A. N. Brooks & T. J. Hughes 1982
Brooks, A. N.; Hughes, T. J. (1982): Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. In: Computer Methods in Applied Mechanics and Engineering 32.1-3, pp. 199–259. ISSN: 0045-7825. DOI: 10.1016/0045-7825(82)90071-8.

Bugatti & Semeraro 2018
Bugatti, M.; Semeraro, Q. (2018): Limitations of the inherent strain method in simulating powder bed fusion processes. In: Additive Manufacturing 23, pp. 329–346. ISSN: 2214-8604. DOI: 10.1016/j.addma.2018.05.041.

Bundesministerium fuer Finanzen 2000
Bundesministerium fuer Finanzen (2000): *AfA-Tabelle für die allgemein verwendbaren Anlagegüter*. ⟨https://www.bundesfinanzministerium.de/Content/DE/Standardartikel/Themen/Steuern/Weitere_Steuerthemen/Betriebspruefung/AfA-Tabellen/2000-12-15-afa-103.pdf;jsessionid=A4B1075A1BDF14954B7953DF3DCBC4F0?__blob=publicationFile&v=3⟩ - 11/22/2018.

Butcher 1986
Butcher, J. C. (1986): The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods. A Wiley-Interscience publication. Chichester: John Wiley & Sons, Ltd. 512 pp. ISBN: 047191046-5.

Carraturo et al. 2019
Carraturo, M.; Giannelli, C.; Reali, A.; Vázquez, R. (2019): Suitably graded THB-spline refinement and coarsening: Towards an adaptive isogeometric analysis of additive manufacturing processes. In: Computer Methods in Applied Mechanics and Engineering 348, pp. 660–679. ISSN: 0045-7825. DOI: 10.1016/j.cma.2019.01.044.

Carslaw & Jaeger 2008
Carslaw, H. S.; Jaeger, J. C. (2008): Conduction of heat in solids. 2nd ed. Oxford science publications. Oxford: Clarendon Press. 510 pp. ISBN: 978-019853368-9.

City Council Munich 2018
City Council Munich (2018): *Festsetzung kalkulatorischer Zinssatz 2019*. ⟨https://www.muenchen-transparent.de/dokumente/4861578⟩ - 11/22/2018.

Cline & Anthony 1977
Cline, H. E.; Anthony, T. R. (1977): Heat treating and melting material with a scanning laser or electron beam. In: Journal of Applied Physics 48.9, pp. 3895–3900. ISSN: 0018-9197. DOI: 10.1063/1.324261.

CONTUZZI et al. 2011
CONTUZZI, N.; CAMPANELLI, S. L.; LUDOVICO, A. D. (2011): 3D Finite Element Analysis in the selective laser melting process. In: International Journal of Simulation Modelling 10.3, pp. 113–121. ISSN: 1726-4529. DOI: 10.2507/IJSIMM10(3)1.169.

CORMEN 2007
CORMEN, T. H. (2007): Introduction to algorithms. 2nd ed. Cambridge, Massachusetts: MIT Press. 1180 pp. ISBN: 0-262-03293-7.

COWLES et al. 2012
COWLES, B.; BACKMAN, D.; DUTTON, R. (2012): Verification and validation of ICME methods and models for aerospace applications. In: Integrating Materials and Manufacturing Innovation 1.1, p. 70. ISSN: 2193-9764. DOI: 10.1186/2193-9772-1-2.

DAMJANIĆ & OWEN 1982
DAMJANIĆ, F.; OWEN, D. (1982): Practical considerations for thermal transient finite element analysis using isoparametric elements. In: Nuclear Engineering and Design 69.1, pp. 109–126. ISSN: 0029-5493. DOI: 10.1016/0029-5493(82)90285-0.

DAVIS & HAGER 2001
DAVIS, T. A.; HAGER, W. W. (2001): Multiple-Rank Modifications of a Sparse Cholesky Factorization. In: SIAM Journal on Matrix Analysis and Applications 22.4, pp. 997–1013. ISSN: 0895-4798. DOI: 10.1137/S0895479899357346.

DENLINGER 2017
DENLINGER, E. R. (2017): Development and Numerical Verification of a Dynamic Adaptive Mesh Coarsening Strategy for Simulating Laser Power Bed Fusion Processes. In: Thermo-mechanical modeling of additive manufacturing. Ed. by M. GOUGE; P. MICHALERIS. Oxford and Cambridge, Massachusetts: Butterworth-Heinemann, pp. 199–213. ISBN: 978-012811820-7. DOI: 10.1016/B978-0-12-811820-7.00015-X.

DENLINGER et al. 2015
DENLINGER, E. R.; HEIGEL, J. C.; MICHALERIS, P. (2015): Residual stress and distortion modeling of electron beam direct manufacturing Ti-6Al-4V. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 229.10, pp. 1803–1813. ISSN: 0954-4054. DOI: 10.1177/0954405414539494.

DEUFLHARD & BORNEMANN 2002
DEUFLHARD, P.; BORNEMANN, F. (2002): Scientific Computing with Ordinary Differential Equations. 42. Texts in Applied Mathematics. New York: Springer. 486 pp. ISBN: 978-0-387-21582-2. DOI: 10.1007/978-0-387-21582-2.

DEUFLHARD & WEISER 2012
DEUFLHARD, P.; WEISER, M. (2012): Adaptive Numerical Solution of PDEs. 1st ed. Berlin: De Gruyter. 432 pp. ISBN: 978-3-11-028311-2. DOI: 10.1515/9783110283112.

DHONDT 2004
DHONDT, G. (2004): The finite element method for three-dimensional thermomechanical applications. Chichester: John Wiley & Sons, Ltd. 340 pp. ISBN: 9780470857526. DOI: 10.1002/0470021217.

DIN EN ISO 80000 2013
DIN EN ISO 80000 (2013): Größen und Einheiten. Berlin: Beuth.

DOUGLAS 1962
DOUGLAS, J. (1962): Alternating direction methods for three space variables. In: Numerische Mathematik 4.1, pp. 41–63. ISSN: 0029-599X. DOI: 10.1007/BF01386295.

ECKERT & DRAKE 1972
ECKERT, E. R. G.; DRAKE, R. M. (1972): Analysis of heat and mass transfer. McGraw-Hill series in mechanical engineering. New York: McGraw-Hill. 806 pp. ISBN: 978-0070189256.

EVANS 2015
EVANS, L. C. (2015): Partial differential equations. 2nd ed. 19. Graduate studies in mathematics. Providence, Rhode Island: American Mathematical Society. 754 pp. ISBN: 978-0821849743.

FACHINOTTI & BELLET 2006
FACHINOTTI, V. D.; BELLET, M. (2006): Linear tetrahedral finite elements for thermal shock problems. In: International Journal of Numerical Methods for Heat & Fluid Flow 16.5, pp. 590–601. ISSN: 0961-5539. DOI: 10.1108/09615530610669120.

FEDERAL INSTITUTE FOR OCCUPATIONAL SAFETY AND HEALTH 2013
FEDERAL INSTITUTE FOR OCCUPATIONAL SAFETY AND HEALTH (2013): *ASR A1.2 Technische Regeln für Arbeitsstätten: Raumabmessungen und Bewegungsflächen.* ⟨https://www.baua.de/DE/Angebote/Rechtstexte-und-Technische-Regeln/Regelwerk/ASR/pdf/ASR-A1-2.pdf?__blob=publicationFile⟩ - 11/22/2018.

FELL & WILLEKE 2010
FELL, A.; WILLEKE, G. P. (2010): Fast simulation code for heating, phase changes and dopant diffusion in silicon laser processing using the alternating direction explicit (ADE) method. In: Applied Physics A 98.2, pp. 435–440. ISSN: 0947-8396. DOI: 10.1007/s00339-009-5416-7.

FREE SOFTWARE FOUNDATION 2018
FREE SOFTWARE FOUNDATION, I. (2018): *Philosophy of the GNU Project*. ⟨https://www.gnu.org/philosophy/philosophy.en.html⟩ - 11/22/2018.

FREY & GEORGE 2000
FREY, P. J.; GEORGE, P.-L. (2000): Mesh generation: Application to finite elements. Oxford: Hermes Science Publ. 814 pp. ISBN: 978-1903398005.

GEAR & WELLS 1984
GEAR, C. W.; WELLS, D. R. (1984): Multirate linear multistep methods. In: BIT 24.4, pp. 484–502. ISSN: 0006-3835. DOI: 10.1007/BF01934907.

GEBHARDT 2013
GEBHARDT, A. (2013): Generative Fertigungsverfahren. Munich: Carl Hanser. ISBN: 978-3-446-43651-0. DOI: 10.3139/9783446436527.

GIBSON et al. 2015
GIBSON, I.; ROSEN, D.; STUCKER, B. (2015): Additive manufacturing technologies: 3D printing, rapid prototyping and direct digital manufacturing. 2nd ed. New York: Springer. 498 pp. ISBN: 978-1-4939-2112-6.

GILL et al. 1972
GILL, P. E.; GOLUB, G. H.; MURRAY, W. A.; SAUNDERS, M. A. (1972): Methods for modifying matrix factorizations. Stanford University.

GOLUB & O'LEARY 1989
GOLUB, G. H.; O'LEARY, D. P. (1989): Some History of the Conjugate Gradient and Lanczos Algorithms: 1948–1976. In: SIAM Review 31.1, pp. 50–102. ISSN: 0036-1445. DOI: 10.1137/1031003.

GOLUB & VAN LOAN 1983
GOLUB, G. H.; VAN LOAN, C. F. (1983): Matrix computations. Oxford: North Oxford Academic. 476 pp. ISBN: 978-094653605-4.

GONZÁLEZ PINTO et al. 1997
GONZÁLEZ PINTO, S.; PÉREZ RODRÍGUEZ, S.; MONTIJANO TORCAL, J. I. (1997): On the numerical solution of stiff IVPs by Lobatto IIIA Runge-Kutta methods. In: Journal of Computational and Applied Mathematics 82.1-2, pp. 129–148. ISSN: 0377-0427. DOI: 10.1016/S0377-0427(97)00086-1.

GRIFFITHS & HIGHAM 2010
GRIFFITHS, D. F.; HIGHAM, D. J. (2010): Numerical Methods for Ordinary Differential Equations: Initial Value Problems. Springer Undergraduate Mathematics Series. London: Springer. DOI: 10.1007/978-0-85729-148-6.

GUENTHER & RENTROP 1994
GUENTHER, M.; RENTROP, P. (1994): Partitioning and Multirate Strategies in Latent Electric Circuits. In: Mathematical Modelling and Simulation of Electrical Circuits and Semiconductor Devices. Proceedings of a Conference held at the Mathematisches Forschungsinstitut. Oberwolfach / Germany, July 5–11, 1992. Ed. by R. E. BANK; H. GAJEWSKI; R. BULIRSCH; K. MERTEN. ISNM International Series of Numerical Mathematics 117. Springer, pp. 33–60. DOI: 10.1007/978-3-0348-8528-7_3.

HACHEM et al. 2010
HACHEM, E.; DIGONNET, H.; KOSSEIFI, N.; MASSONI, E.; COUPEZ, T. (2010): Enriched finite element spaces for transient conduction heat transfer. In: Applied Mathematics and Computation 217.8, pp. 3929–3943. ISSN: 0096-3003. DOI: 10.1016/j.amc.2010.09.057.

HE et al. 2016
HE, K.; TAN, S. X.-D.; WANG, H.; SHI, G. (2016): GPU-Accelerated Parallel Sparse LU Factorization Method for Fast Circuit Analysis. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24.3, pp. 1140–1150. ISSN: 1063-8210. DOI: 10.1109/TVLSI.2015.2421287.

HELD & PFLIGERSDORFFER 2009
HELD, M.; PFLIGERSDORFFER, C. (2009): Correcting warpage of laser-sintered parts by means of a surface-based inverse deformation algorithm. In: Engineering with Computers 25.4, pp. 389–395. ISSN: 0177-0667. DOI: 10.1007/s00366-009-0136-3.

HOGGE & GERREKENS 1983
HOGGE, M.; GERREKENS, P. (1983): "Steep Gradient Modelling in Diffusion Problems". In: *Numerical methods in heat transfer*. Ed. by R. W. LEWIS. 2. John Wiley & Sons, Ltd, pp. 73–97. ISBN: 978-0471900641.

HUANG et al. 2016
HUANG, R.; RIDDLE, M.; GRAZIANO, D.; WARREN, J.; DAS, S.; NIMBALKAR, S.; CRESKO, J.; MASANET, E. (2016): Energy and emissions saving potential of additive manufacturing: the case of lightweight aircraft components. In: Journal of Cleaner Production 135, pp. 1559–1570. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2015.04.109.

T. J. R. HUGHES 1987
HUGHES, T. J. R. (1987): The finite element method: Linear static and dynamic finite element analysis. Englewood Cliffs: Prentice-Hall. 803 pp. ISBN: 013317025-X.

T. J. R. HUGHES & A. BROOKS 1979
HUGHES, T. J. R.; BROOKS, A. (1979): A multi-dimensional upwind scheme with no crosswind diffusion. In: American Society of Mechanical Engineers, Applied Mechanics Division, AMD 34, pp. 19–35. ISSN: 0160-8835.

T. J. R. HUGHES et al. 1979
HUGHES, T. J. R.; PISTER, K. S.; TAYLOR, R. L. (1979): Implicit-explicit finite elements in nonlinear transient analysis. In: Computer Methods in Applied Mechanics and Engineering 17-18, pp. 159–182. ISSN: 0045-7825. DOI: `10.1016/0045-7825(79)90086-0`.

T. J. R. HUGHES et al. 1989
HUGHES, T. J. R.; FRANCA, L. P.; HULBERT, G. M. (1989): A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations. In: Computer Methods in Applied Mechanics and Engineering 73.2, pp. 173–189. ISSN: 0045-7825. DOI: `10.1016/0045-7825(89)90111-4`.

INCROPERA et al. 2007
INCROPERA, F. P.; DEWITT, D. P.; BERGMAN, T. L.; LAVINE, A. S. (2007): Fundamentals of Heat and Mass Transfer. 6th ed. Hoboken: John Wiley & Sons, Ltd. 997 pp. ISBN: 978-0-471-45728-2.

ISO/ASTM52900 2015
ISO/ASTM52900 (2015): Terminology for Additive Manufacturing - General Principles - Terminology. West Conshohocken, PA: ASTM International. DOI: `10.1520/ISOASTM52900-15`.

JAOUEN 1998
JAOUEN, O. (1998): Modélisation tridimensionnelle par éléments finis pour l'analyse thermo-mécanique du refroidissement des pièces coulées. École Nationale Supérieure des Mines de Paris.

JOULAIAN et al. 2014
JOULAIAN, M.; DUCZEK, S.; GABBERT, U.; DUESTER, A. (2014): Finite and spectral cell method for wave propagation in heterogeneous materials. In: Computational Mechanics 54.3, pp. 661–675. ISSN: 0178-7675. DOI: `10.1007/s00466-014-1019-z`.

KANAPADY et al. 2005
KANAPADY, R.; JAIN, A.; TAMMA, K.; SRINIVASAN, S. (2005): Local Discontinuous Galerkin Method for Heat Conduction Problems Involving High Gradients and Imperfect Contact Surfaces. In: 43rd AIAA Aerospace Sciences Meeting and Exhibit. Reston (Virginia) / USA, January 10-13, 2005. American Institute of Aeronautics and Astronautics, pp. 1–47. ISBN: 978-1-62410-064-2. DOI: 10.2514/6.2005-762.

KATO et al. 2009
KATO, T.; INOUE, K.; FUKUTANI, T.; KANDA, Y. (2009): Multirate Analysis Method for a Power Electronic System by Circuit Partitioning. In: IEEE Transactions on Power Electronics 24.12, pp. 2791–2802. ISSN: 0885-8993. DOI: 10.1109/TPEL.2009.2030959.

KELLER & PLOSHIKHIN 2014
KELLER, N.; PLOSHIKHIN, V. (2014): New Method for Fast Prediction of Residual Stress and Distortion of AM Parts. In: Proceedings of the 25th Solid Freeform Fabrication Symposium. Austin (Texas) / USA, August 4-6, 2014. Ed. by D. L. BOURELL; J. J. BEAMAN; R. H. CRAWFORD; C. C. SEEPERSAD; H. MARCUS, pp. 1229–1237.

KELLNER 2014
KELLNER, T. (2014): *Fit to Print: New Plant Will Assemble World's First Passenger Jet Engine With 3D Printed Fuel Nozzles, Next-Gen Materials*. Ed. by GE. ⟨https://www.ge.com/reports/post/80701924024/fit-to-print/⟩ - 01/17/2019.

KHOEI 2014
KHOEI, A. R. (2014): Extended Finite Element Method: Theory and applications. Wiley Series in Computational Mechanics. Chichester, UK: John Wiley & Sons, Ltd. 1584 pp. DOI: 10.1002/9781118869673.

KING et al. 2014
KING, W.; ANDERSON, A. T.; FERENCZ, R. M.; HODGE, N. E.; KAMATH, C.; KHAIRALLAH, S. A. (2014): Overview of modelling and simulation of metal powder bed fusion process at Lawrence Livermore National Laboratory. In: Materials Science and Technology 31.8, pp. 957–968. ISSN: 0267-0836. DOI: 10.1179/1743284714Y.0000000728.

KNOWLES et al. 2012
KNOWLES, C. R.; BECKER, T. H.; TAIT, R. B. (2012): Residual stress measurements and structural integrity implications for selective laser melted Ti-6Al-4V. In: South African Journal of Industrial Engineering 23.3, p. 119. ISSN: 2224-7890. DOI: 10.7166/23-3-515.

KNUPP 2001
KNUPP, P. M. (2001): Hexahedral and Tetrahedral Mesh Untangling. In: Engineering with Computers 17.3, pp. 261–268. ISSN: 0177-0667. DOI: 10.1007/s003660170006.

KOLLMANNSBERGER et al. 2018
KOLLMANNSBERGER, S.; ÖZCAN, A.; CARRATURO, M.; ZANDER, N.; RANK, E. (2018): A hierarchical computational model for moving thermal loads and phase changes with applications to selective laser melting. In: Computers & Mathematics with Applications 75.5, pp. 1483–1497. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2017.11.014.

KOLOSSOV et al. 2004
KOLOSSOV, S.; BOILLAT, E.; GLARDON, R.; FISCHER, P.; LOCHER, M. (2004): 3D FE simulation for temperature evolution in the selective laser sintering process. In: International Journal of Machine Tools and Manufacture 44.2-3, pp. 117–123. ISSN: 0890-6955. DOI: 10.1016/j.ijmachtools.2003.10.019.

KRAUSS 2016
KRAUSS, H. (2016): Qualitätssicherung beim Laserstrahlschmelzen durch schichtweise thermografische In-Process-Überwachung. Institute for Machine Tools and Industrial Management. PhD Thesis. Technical University of Munich. 325. Forschungsberichte IWB. Munich: Utz. 272 pp. ISBN: 978-3-8316-4628-9.

KROL et al. 2012
KROL, T. A.; ZAEH, M. F.; SEIDEL, C. (2012): Optimization of support in metal-based additive manufacturing by means of finite element models. In: Proceedings of the 23rd Solid Freeform Fabrication Symposium. Austin (Texas) / USA, August 6-8, 2012. Ed. by D. L. BOURELL; J. J. BEAMAN; R. H. CRAWFORD; C. C. SEEPERSAD; H. MARCUS, pp. 707–718.

KROL et al. 2013
KROL, T. A.; SEIDEL, C.; ZAEH, M. F. (2013): Prioritization of Process Parameters for an Efficient Optimisation of Additive Manufacturing by Means of a Finite Element Method. In: Procedia CIRP 12, pp. 169–174. ISSN: 2212-8271. DOI: 10.1016/j.procir.2013.09.030.

KROL et al. 2011
KROL, T. A.; WESTHAEUSER, S.; ZAEH, M. F.; SCHILP, J.; GROTH, G. (2011): Development of a Simulation-Based Process Chain – Strategy for Different Levels of Detail for the Preprocessing Definitions. In: SNE Simulation Notes Europe 21.3-4, pp. 135–140. ISSN: 2305-9974. DOI: 10.11128/sne.21.tn.10081.

KRUTH et al. 2004
KRUTH, J. P.; FROYEN, L.; VAN VAERENBERGH, J.; MERCELIS, P.; ROMBOUTS, M.; LAUWERS, B. (2004): Selective laser melting of iron-based powder. In: Journal of Materials Processing Technology 149.1-3, pp. 616–622. ISSN: 0924-0136. DOI: 10.1016/j.jmatprotec.2003.11.051.

KUNDAKCIOGLU et al. 2016
KUNDAKCIOGLU, E.; LAZOGLU, I.; RAWAL, S. (2016): Transient thermal modeling of laser-based additive manufacturing for 3D freeform structures. In: The International Journal of Advanced Manufacturing Technology 85.1-4, pp. 493–501. ISSN: 0268-3768. DOI: 10.1007/s00170-015-7932-2.

LETNIOWSKI 1992
LETNIOWSKI, F. W. (1992): Three-Dimensional Delaunay Triangulations for Finite Element Approximations to a Second-Order Diffusion Operator. In: SIAM Journal on Scientific and Statistical Computing 13.3, pp. 765–770. ISSN: 0196-5204. DOI: 10.1137/0913045.

LITTLE et al. 2015
LITTLE, C. H.; TEO, K. L.; VAN BRUNT, B. (2015): Real analysis via sequences and series. Undergraduate Texts in Mathematics. New York: Springer. 476 pp. ISBN: 978-1-4939-2650-3. DOI: 10.1007/978-1-4939-2651-0.

LUO & ZHAO 2018
LUO, Z.; ZHAO, Y. (2018): A survey of finite element analysis of temperature and thermal stress fields in powder bed fusion Additive Manufacturing. In: Additive Manufacturing 21, pp. 318–332. ISSN: 2214-8604. DOI: 10.1016/j.addma.2018.03.022.

MA & BIN 2007
MA, L.; BIN, H. (2007): Temperature and stress analysis and simulation in fractal scanning-based laser sintering. In: The International Journal of Advanced Manufacturing Technology 34.9-10, pp. 898–903. ISSN: 0268-3768. DOI: 10.1007/s00170-006-0665-5.

MANYIKA et al. 2013
MANYIKA, J.; CHUI, M.; BUGHIN, J.; DOBBS, R.; BISSON, P.; MARRS, A. (2013): *Disruptive technologies: Advances that will transform life, business, and the global economy*. Ed. by MCKINSEY GLOBAL INSTITUTE. ⟨https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/disruptive%20technologies/mgi_disruptive_technologies_full_report_may2013.ashx⟩ - 01/17/2019.

MARSDEN et al. 2008
MARSDEN, J. E.; SIROVICH, L.; ANTMAN, S. S.; ALLAIRE, G.; KABER, S. M. (2008): Numerical Linear Algebra. 55. New York: Springer. ISBN: 978-0-387-34159-0. DOI: 10.1007/978-0-387-68918-0.

MARTUKANITZ et al. 2014
MARTUKANITZ, R.; MICHALERIS, P.; PALMER, T.; DEBROY, T.; LIU, Z.-K.; OTIS, R.; HEO, T. W.; CHEN, L.-Q. (2014): Toward an integrated computational system for describing the additive manufacturing process for metallic materials. In: Additive Manufacturing 1-4, pp. 52–63. ISSN: 2214-8604. DOI: 10.1016/j.addma.2014.09.002.

MATLAB AND SYMBOLIC MATH TOOLBOX 2016
MATLAB AND SYMBOLIC MATH TOOLBOX (2016): Version 9.1.0.441655 (R2016b). Natick, Massachusetts: The MathWorks Inc.

MATSUMOTO et al. 2002
MATSUMOTO, M.; SHIOMI, M.; OSAKADA, K.; ABE, F. (2002): Finite element analysis of single layer forming on metallic powder bed in rapid prototyping by selective laser processing. In: International Journal of Machine Tools and Manufacture 42.1, pp. 61–67. ISSN: 0890-6955. DOI: 10.1016/S0890-6955(01)00093-1.

MEGAHED et al. 2016
MEGAHED, M.; MINDT, H.-W.; N'DRI, N.; DUAN, H.; DESMAISON, O. (2016): Metal additive-manufacturing process and residual stress modeling. In: Integrating Materials and Manufacturing Innovation 5.1, pp. 1–33. ISSN: 2193-9764. DOI: 10.1186/s40192-016-0047-2.

MERCELIS & KRUTH 2006
MERCELIS, P.; KRUTH, J. P. (2006): Residual stresses in selective laser sintering and selective laser melting. In: Rapid Prototyping Journal 12.5, pp. 254–265. ISSN: 1355-2546. DOI: 10.1108/13552540610707013.

MEYER et al. 2018
MEYER, T.; LI, P.; LU, D.; SCHWEIZER, B. (2018): Implicit co-simulation method for constraint coupling with improved stability behavior. In: Multibody System Dynamics 44.2, pp. 135–161. ISSN: 1384-5640. DOI: 10.1007/s11044-018-9632-9.

MICHALERIS 2014
MICHALERIS, P. (2014): Modeling metal deposition in heat transfer analyses of additive manufacturing processes. In: Finite Elements in Analysis and Design 86, pp. 51–60. ISSN: 0168-874X. DOI: 10.1016/j.finel.2014.04.003.

MIFUNE et al. 2002
MIFUNE, T.; IWASHITA, T.; SHIMASAKI, M. (2002): A fast solver for FEM analyses using the parallelized algebraic multigrid method. In: IEEE Transactions on Magnetics 38.2, pp. 369–372. ISSN: 0018-9464. DOI: 10.1109/20.996099.

MOS et al. 1999
MOS, N.; DOLBOW, J.; BELYTSCHKO, T. (1999): A finite element method for crack growth without remeshing. In: International Journal for Numerical Methods in Engineering 46.1, pp. 131–150. ISSN: 1097-0207. DOI: 10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J.

MTU AERO ENGINES AG 2018
MTU AERO ENGINES AG (2018): *Additive manufacturing: MTU Aero Engines steps up its efforts*. ⟨https://www.mtu.de/fileadmin/EN/7_News_Media/1_Press/1_Latest_Press_Realeases/2018/2018_05_22_MTU_Additive_en.doc⟩ - 09/04/2018.

MUNSCH et al. 2013
MUNSCH, M.; EMMELMANN, C.; HINTZE, W. (2013): Reduzierung von Eigenspannungen und Verzug in der laseradditiven Fertigung. 1st ed. 6. Schriftenreihe Lasertechnik. Göttingen: Cuvillier. 133 pp. ISBN: 978-3954045013.

MUNSON 2013
MUNSON, B. R. (2013): Fundamentals of fluid mechanics. 7th ed. Hoboken: John Wiley & Sons, Ltd. ISBN: 978-111839971-2.

NAKSHATRALA et al. 2009
NAKSHATRALA, P. B.; NAKSHATRALA, K. B.; TORTORELLI, D. A. (2009): A time-staggered partitioned coupling algorithm for transient heat conduction. In: International Journal for Numerical Methods in Engineering 78.12, pp. 1387–1406. ISSN: 0029-5981. DOI: 10.1002/nme.2524.

N'DRI et al. 2015
N'DRI, N.; MINDT, H.-W.; SHULA, B.; MEGAHED, M.; PERALTA, A.; KANTZOS, P.; NEUMANN, J. (2015): DMLS Process Modelling and Validation. In: TMS 2015 144th Annual Meeting & Exhibition. Supplemental Proceedings. Orlando (Florida) / USA, March 15-19, 2015. Springer International Publishing, pp. 389–396. ISBN: 978-3-319-48608-6. DOI: 10.1007/978-3-319-48127-2_49.

NEUGEBAUER et al. 2014
NEUGEBAUER, F.; KELLER, N.; XU, H.; KOBER, C.; PLOSHIKHIN, V. (2014): Simulation of Selective Laser Melting Using Process Specific Layer Based Meshing. In: DDMC 2014. Fraunhofer Direct Digital Manufacturing Conference. Berlin / Germany, March 12-13, 2014. Ed. by A. DEMMER. Fraunhofer Verlag, pp. 297–302. ISBN: 978-3-8396-9128-1.

NIKOUKAR et al. 2013
NIKOUKAR, M.; PATIL, N.; PAL, D.; STUCKER, B. (2013): Methods for Enhancing the Speed of Numerical Calculations for the Prediction of the Mechanical Behavior of Parts Made Using Additive Manufacturing. In: Proceedings of the 24th Solid Freeform Fabrication Symposium. Austin (Texas) / USA, August 12-14, 2013. Ed. by D. L. BOURELL; J. J. BEAMAN; R. H. CRAWFORD; C. C. SEEPERSAD; H. MARCUS, pp. 525–534.

OBERKAMPF et al. 2002
OBERKAMPF, W. L.; DELAND, S. M.; RUTHERFORD, B. M.; DIEGERT, K. V.; ALVIN, K. F. (2002): Error and uncertainty in modeling and simulation. In: Reliability Engineering & System Safety 75.3, pp. 333–357. ISSN: 0951-8320. DOI: 10.1016/S0951-8320(01)00120-X.

OJO et al. 2015
OJO, S. O.; GRIVET-TALOCIA, S.; PAGGI, M. (2015): Model order reduction applied to heat conduction in photovoltaic modules. In: Composite Structures 119, pp. 477–486. ISSN: 0263-8223. DOI: 10.1016/j.compstruct.2014.09.008.

ORTEGA & RHEINBOLDT 1970
ORTEGA, J. M.; RHEINBOLDT, W. C. (1970): Iterative Solution of Nonlinear Equations in Several Variables. Burlington: Elsevier. 593 pp. DOI: 10.1016/C2013-0-11263-9.

PAL et al. 2014
PAL, D.; PATIL, N.; TENG, C.; ZENG, K.; XU, S., SUBLETTE, T.; STUCKER, B. (2014): Enhancing Simulations of Additive Manufacturing Processes Using Spatiotemporal Multiscaling. In: Proceedings of the 25th Solid Freeform Fabrication Symposium. Austin (Texas) / USA, August 4-6, 2014. Ed. by D. L. BOURELL; J. J. BEAMAN; R. H. CRAWFORD; C. C. SEEPERSAD; H. MARCUS, pp. 1213–1228.

PAL et al. 2015
PAL, D.; PATIL, N.; ZENG, K.; TENG, C.; STUCKER, B. (2015): An Efficient Multi-Scale Simulation Architecture for the Prediction of Performance Metrics of Parts Fabricated Using Additive Manufacturing. In: Metallurgical and Materials Transactions A 46.9, pp. 3852–3863. ISSN: 1073-5623. DOI: 10.1007/s11661-015-2903-7.

PAL et al. 2016
PAL, D.; PATIL, N.; KUTTY, K. H.; ZENG, K.; MORELAND, A.; HICKS, A.; BEELER, D.; STUCKER, B. (2016): A Generalized Feed-Forward Dynamic Adaptive Mesh Refinement and Derefinement Finite-Element Framework for Metal Laser Sintering - Part II: Nonlinear Thermal Simulations and Validations 2. In: Journal of Manufacturing Science and Engineering 138.6. ISSN: 1087-1357. DOI: 10.1115/1.4032078.

PAPADAKIS, LOIZOU, RISSE & SCHRAGE 2014
PAPADAKIS, L.; LOIZOU, A.; RISSE, J.; SCHRAGE, J. (2014): Numerical Computation of Component Shape Distortion Manufactured by Selective Laser Melting. In: Procedia CIRP 18, pp. 90–95. ISSN: 2212-8271. DOI: 10.1016/j.procir.2014.06.113.

PAPADAKIS, LOIZOU, RISSE, BREMEN, et al. 2014
PAPADAKIS, L.; LOIZOU, A.; RISSE, J.; BREMEN, S.; SCHRAGE, J. (2014): A computational reduction model for appraising structural effects in selective laser melting manufacturing. In: Virtual and Physical Prototyping 9.1, pp. 17–25. ISSN: 1745-2759. DOI: 10.1080/17452759.2013.868005.

PARVIZIAN et al. 2007
PARVIZIAN, J.; DÜSTER, A.; RANK, E. (2007): Finite cell method. In: Computational Mechanics 41.1, pp. 121–133. ISSN: 0178-7675. DOI: 10.1007/s00466-007-0173-y.

PATIL et al. 2015
PATIL, N.; PAL, D.; KHALID RAFI, H.; ZENG, K.; MORELAND, A.; HICKS, A.; BEELER, D.; STUCKER, B. (2015): A Generalized Feed Forward Dynamic Adaptive Mesh Refinement and Derefinement Finite Element Framework for Metal Laser Sintering - Part I: Formulation and Algorithm Development. In: Journal of Manufacturing Science and Engineering 137.4. ISSN: 1087-1357. DOI: 10.1115/1.4030059.

PEACEMAN & RACHFORD 1955
PEACEMAN, D. W.; RACHFORD, J. H. H. (1955): The Numerical Solution of Parabolic and Elliptic Differential Equations. In: Journal of the Society for Industrial and Applied Mathematics 3.1, pp. 28–41. ISSN: 0368-4245. DOI: 10.1137/0103003.

PENG et al. 2018
PENG, H.; GHASRI-KHOUZANI, M.; GONG, S.; ATTARDO, R.; OSTIGUY, P.; GATRELL, B. A.; BUDZINSKI, J.; TOMONTO, C.; NEIDIG, J.; SHANKAR, M. R.; BILLO, R.; GO, D. B.; HOELZLE, D. (2018): Fast prediction of thermal distortion in metal powder bed fusion additive manufacturing: Part 1, a thermal circuit network model. In: Additive Manufacturing 22, pp. 852–868. ISSN: 2214-8604. DOI: 10.1016/j.addma.2018.05.023.

PHILLIPS 2003
PHILLIPS, J. R. (2003): Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 22.2, pp. 171–187. ISSN: 0278-0070. DOI: 10.1109/TCAD.2002.806605.

PICHELIN & COUPEZ 1999
PICHELIN, E.; COUPEZ, T. (1999): A Taylor discontinuous Galerkin method for the thermal solution in 3D mold filling. In: Computer Methods in Applied Mechanics and Engineering 178.1-2, pp. 153–169. ISSN: 0045-7825. DOI: 10.1016/S0045-7825(99)00011-0.

POLIFKE & KOPITZ 2009
POLIFKE, W.; KOPITZ, J. (2009): Waermeuebertragung: Grundlagen, analytische und numerische Methoden. 2nd ed. Always learning. Munich: Pearson Studium. ISBN: 978-3-8273-7349-6.

POTTLACHER et al. 2002
POTTLACHER, G.; HOSAEUS, H.; KASCHNITZ, E.; SEIFTER, A. (2002): Thermophysical properties of solid and liquid Inconel 718 Alloy. In: Scandinavian Journal of Metallurgy 31.3, pp. 161–168. ISSN: 0371-0459. DOI: 10.1034/j.1600-0692.2002.310301.x.

PROKAJ & CHOUDHURY 2006
PROKAJ, J.; CHOUDHURY, S. R. (2006): Multi-rate numerical techniques for diffusion problems in one spatial dimension. In: International Journal of Applied and Computational Mathematics 13.3, pp. 126–137. ISSN: 2349-5103.

PUTTI & CORDES 1998
PUTTI, M.; CORDES, C. (1998): Finite Element Approximation of the Diffusion Operator on Tetrahedra. In: SIAM Journal on Scientific Computing 19.4, pp. 1154–1168. ISSN: 1064-8275. DOI: 10.1137/S1064827595290711.

READY 1965
READY, J. F. (1965): Effects Due to Absorption of Laser Radiation. In: Journal of Applied Physics 36.2, pp. 462–468. ISSN: 0018-9197. DOI: 10.1063/1.1714012.

REWIENSKI & WHITE 2003
REWIENSKI, M.; WHITE, J. (2003): A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 22.2, pp. 155–170. ISSN: 0278-0070. DOI: 10.1109/TCAD.2002.806601.

RIEDLBAUER et al. 2014
RIEDLBAUER, D.; STEINMANN, P.; MERGHEIM, J. (2014): Thermomechanical finite element simulations of selective electron beam melting processes: performance considerations. In: Computational Mechanics 54.1, pp. 109–122. ISSN: 0178-7675. DOI: 10.1007/s00466-014-1026-0.

ROMBOUTS et al. 2005
ROMBOUTS, M.; FROYEN, L.; GUSAROV, A. V.; BENTEFOUR, E. H.; GLORIEUX, C. (2005): Photopyroelectric measurement of thermal conductivity of metallic powders. In: Journal of Applied Physics 97.2. ISSN: 0018-9197. DOI: 10.1063/1.1832740.

ROSENTHAL 1946
ROSENTHAL, D. (1946): The theory of moving sources of heat and its application to metal treatments. In: Transactions of the ASME 11.68, pp. 849–866.

ROSSI & TOIVANEN 1999
ROSSI, T.; TOIVANEN, J. (1999): A Parallel Fast Direct Solver for Block Tridiagonal Systems with Separable Matrices of Arbitrary Dimension. In: SIAM Journal on Scientific Computing 20.5, pp. 1778–1793. ISSN: 1064-8275. DOI: 10.1137/S1064827597317016.

RUESS et al. 2012
RUESS, M.; TAL, D.; TRABELSI, N.; YOSIBASH, Z.; RANK, E. (2012): The finite cell method for bone simulations: verification and validation. In: Biomechanics and modeling in mechanobiology 11.3-4, pp. 425–437. ISSN: 1617-7959. DOI: 10.1007/s10237-011-0322-2.

SAAD 2003
SAAD, Y. (2003): Iterative methods for sparse linear systems. 2$^{\text{nd}}$ ed. Philadelphia: Society for Industrial and Applied Mathematics. 528 pp. ISBN: 978-0898715347.

ŞAHIN et al. 1995
ŞAHIN, A. Z.; KAVRANOĞLU, D.; BETTAYEB, M. (1995): Model reduction in numerical heat transfer problems. In: Applied Mathematics and Computation 69.2-3, pp. 209–225. ISSN: 0096-3003. DOI: 10.1016/0096-3003(94)00128-Q.

SCHAFSTALL et al. 2016
SCHAFSTALL, H.; KHAZAN, P.; TÖPPEL, T.; KORDASS, R. (2016): Comparison of Approaches for Structural Simulation of Additively Manufactured Metal Parts. In: DDMC 2016. Fraunhofer Direct Digital Manufacturing Conference. Berlin / Germany, March 16 - 17, 2016. Fraunhofer Verlag, pp. 1–7. ISBN: 978-3-8396-1001-5.

SCHENK 2000
SCHENK, O. (2000): Scalable parallel sparse LU factorization methods on shared memory multiprocessors. DOI: 10.3929/ETHZ-A-003876213.

SCHILP et al. 2014
SCHILP, J.; SEIDEL, C.; KRAUSS, H.; WEIRATHER, J. (2014): Investigations on Temperature Fields during Laser Beam Melting by Means of Process Monitoring and Multiscale Process Modelling. In: Advances in Mechanical Engineering 6, pp. 1–7. ISSN: 1687-8140. DOI: 10.1155/2014/217584.

SCHLEGEL et al. 2009
SCHLEGEL, M.; KNOTH, O.; ARNOLD, M.; WOLKE, R. (2009): Multirate Runge–Kutta schemes for advection equations. In: Journal of Computational and Applied Mathematics 226.2, pp. 345–357. ISSN: 0377-0427. DOI: 10.1016/j.cam.2008.08.009.

SCHMIDT et al. 2017
SCHMIDT, M.; MERKLEIN, M.; BOURELL, D.; DIMITROV, D.; HAUSOTTE, T.; WEGENER, K.; OVERMEYER, L.; VOLLERTSEN, F.; LEVY, G. N. (2017): Laser based additive manufacturing in industry and academia. In: CIRP Annals - Manufacturing Technology 66.2, pp. 561–583. ISSN: 0007-8506. DOI: 10.1016/j.cirp.2017.05.011.

SCHOINOCHORITIS et al. 2017
SCHOINOCHORITIS, B.; CHANTZIS, D.; SALONITIS, K. (2017): Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 1, pp. 96–117. ISSN: 0954-4054. DOI: 10.1177/0954405414567522.

SCHOTT et al. 2015
SCHOTT, B.; RASTHOFER, U.; GRAVEMEIER, V.; WALL, W. A. (2015): A face-oriented stabilized Nitsche-type extended variational multiscale method for incompressible two-phase flow. In: International Journal for Numerical Methods in Engineering 104.7, pp. 721–748. ISSN: 1097-0207. DOI: 10.1002/nme.4789.

SEIDEL 2016
SEIDEL, C. (2016): Finite-Elemente-Simulation des Aufbauprozesses beim Laserstrahlschmelzen. Institute for Machine Tools and Industrial Management. PhD Thesis. Technical University of Munich.

SEIDEL & ZAEH 2018
SEIDEL, C.; ZAEH, M. F. (2018): Multi-scale Modelling Approach for Contributing to Reduced Distortion in Parts Made by Laser-based Powder Bed Fusion. In: Procedia CIRP 67, pp. 197–202. ISSN: 2212-8271. DOI: 10.1016/j.procir.2017.12.199.

SEIDEL et al. 2014
SEIDEL, C.; ZAEH, M. F.; WUNDERER, M.; WEIRATHER, J.; KROL, T. A.; OTT, M. (2014): Simulation of the Laser Beam Melting Process – Approaches for an Efficient Modelling of the Beam-material Interaction. In: Procedia CIRP 25, pp. 146–153. ISSN: 2212-8271. DOI: 10.1016/j.procir.2014.10.023.

SOCHALSKI-KOLBUS et al. 2015
SOCHALSKI-KOLBUS, L. M.; PAYZANT, E. A.; CORNWELL, P. A.; WATKINS, T. R.; BABU, S. S.; DEHOFF, R. R.; LORENZ, M.; OVCHINNIKOVA, O.; DUTY, C. (2015): Comparison of Residual Stresses in Inconel 718 Simple Parts Made by Electron Beam Melting and Direct Laser Metal Sintering. In: Metallurgical and Materials Transactions A 46.3, pp. 1419–1432. ISSN: 1073-5623. DOI: 10.1007/s11661-014-2722-2.

SPECIAL METALS CORPORATION 2007
SPECIAL METALS CORPORATION (2007): *Inconel alloy 718 - Publication Number SMC-045*. ⟨http://www.specialmetals.com/assets/smc/documents/inconel_alloy_718.pdf⟩ - 03/10/2019.

STALLMAN & GAY 2002
STALLMAN, R. M.; GAY, J., eds. (2002): *Free software, free society: selected essays*. 1st ed. Boston, Massachusetts: Free Software Foundation. 220 pp. ISBN: 978-1882114986.

STATHATOS & VOSNIAKOS 2018
STATHATOS, E.; VOSNIAKOS, G.-C. (2018): A computationally efficient universal platform for thermal numerical modeling of laser-based additive manufacturing. In: Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 232.13, pp. 2317–2333. ISSN: 0954-4062. DOI: 10.1177/0954406217720230.

STATISTA GMBH 2018
STATISTA GMBH (2018): *Entwicklung der Durchschnittsmieten für Büroflächen in München von 2004 bis 2017*. ⟨`https://de.statista.com/statistik/daten/studie/209635/umfrage/entwicklung-der-mietpreise-fuer-bueroflaechen-in-muenchen/`⟩ - 11/22/2018.

STEWART 1998
STEWART, G. W. (1998): Matrix algorithms: Vol. I: Basic decompositions. Philadelphia: Society for Industrial and Applied Mathematics. 460 pp. ISBN: 978-0898714142.

STRANTZA et al. 2018
STRANTZA, M.; GANERIWALA, R. K.; CLAUSEN, B.; PHAN, T. Q.; LEVINE, L. E.; PAGAN, D.; KING, W. E.; HODGE, N. E.; BROWN, D. W. (2018): Coupled experimental and computational study of residual stresses in additively manufactured Ti-6Al-4V components. In: Materials Letters 231, pp. 221–224. ISSN: 0167-577X. DOI: `10.1016/j.matlet.2018.07.141`.

STRATASYS DIRECT 2015
STRATASYS DIRECT, I. (2015): *Trend forecast: 3D Printing's Imminent Impact on Manufacturing*. ⟨`https://advancedmanufacturing.org/wp-content/uploads/2016/04/Top-Challenges-to-Widespread-3D-Printing-Adoption.pdfkell`⟩ - 01/17/2019.

Systems and software engineering– System life cycle processes 2015
Systems and software engineering– System life cycle processes (2015). ISO/IEC/IEEE15288:2015(E). International standard. New York: Institute of Electrical and Electronics Engineers. 108 pp. ISBN: 978-073819532-2.

TABATA 1978
TABATA, M. (1978): Uniform convergence of the upwind finite element approximation for semilinear parabolic problems. In: Journal of Mathematics of Kyoto University 18.2, pp. 327–351. ISSN: 0023-608X. DOI: `10.1215/kjm/1250522577`.

TANNEHILL et al. 1997
TANNEHILL, J. C.; ANDERSON, D. A.; PLETCHER, R. H. (1997): Computational fluid mechanics and heat transfer. 2nd ed. Series in computational and physical processes in mechanics and thermal sciences. Philadelphia: Taylor & Francis. 792 pp. ISBN: 978-1560320463.

UEDA et al. 1975
UEDA, Y.; FUKUDA, K.; NAKACHO, K.; ENDO, S. (1975): A New Measuring Method of Residual Stresses with the Aid of Finite Element Method and Reliability of Estimated Values. In: Journal of the Society of Naval Architects of Japan 138, pp. 499–507. ISSN: 1884-2070. DOI: 10.2534/jjasnaoe1968.1975.138_499.

URIONDO et al. 2015
URIONDO, A.; ESPERON-MIGUEZ, M.; PERINPANAYAGAM, S. (2015): The present and future of additive manufacturing in the aerospace sector: A review of important aspects: A review of important aspects. In: Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 229.11, pp. 2132–2147. ISSN: 0954-4100. DOI: 10.1177/0954410014568797.

VAN BELLE et al. 2012
VAN BELLE, L.; VANSTEENKISTE, G.; BOYER, J. C. (2012): Comparisons of Numerical Modelling of the Selective Laser Melting. In: Key Engineering Materials 504-506, pp. 1067–1072. ISSN: 1662-9795. DOI: 10.4028/www.scientific.net/KEM.504-506.1067.

VAN HENSON & U. M. YANG 2002
VAN HENSON, E.; YANG, U. M. (2002): BoomerAMG: A parallel algebraic multigrid solver and preconditioner. In: Applied Numerical Mathematics 41.1, pp. 155–177. ISSN: 0168-9274. DOI: 10.1016/S0168-9274(01)00115-5.

WALL et al. 2010
WALL, W. A.; GERSTENBERGER, A.; KÜTTLER, U.; MAYER, U. M. (2010): An XFEM Based Fixed-Grid Approach for 3D Fluid-Structure Interaction. In: Fluid Structure Interaction II. Ed. by H.-J. BUNGARTZ. Lecture Notes in Computational Science and Engineering 73. Berlin, Heidelberg: Springer, pp. 327–349. ISBN: 978-3-642-14205-5. DOI: 10.1007/978-3-642-14206-2_12.

WOHLERS et al. 2016
WOHLERS, T. T.; CAFFREY, T.; CAMPBELL, R. I. (2016): Wohlers report 2016: 3D printing and additive manufacturing state of the industry : annual worldwide progress report. Fort Collins, Colorado: Wohlers Associates. 335 pp. ISBN: 978-0-9913332-2-6.

WOOD & LEWIS 1975
WOOD, W. L.; LEWIS, R. W. (1975): A comparison of time marching schemes for the transient heat conduction equation. In: International Journal for Numerical Methods in Engineering 9.3, pp. 679–689. ISSN: 0029-5981. DOI: 10.1002/nme.1620090314.

X. YAN 2002
YAN, X. (2002): On the Penetration Depth in Fourier Heat Conduction. In: 8th AIAA/ASME Joint Thermophysics and Heat Transfer Conference. St. Louis (Missouri), USA, June 24-26, 2002. Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2002-2881.

Z. YAN et al. 2018
YAN, Z.; LIU, W.; TANG, Z.; LIU, X.; ZHANG, N.; LI, M.; ZHANG, H. (2018): Review on thermal analysis in laser-based additive manufacturing. In: Optics & Laser Technology 106, pp. 427–441. ISSN: 0030-3992. DOI: 10.1016/j.optlastec.2018.04.034.

H.-J. YANG et al. 2002
YANG, H.-J.; HWANG, P.-J.; LEE, S.-H. (2002): A study on shrinkage compensation of the SLS process by using the Taguchi method. In: International Journal of Machine Tools and Manufacture 42.11, pp. 1203–1212. ISSN: 0890-6955. DOI: 10.1016/S0890-6955(02)00070-6.

ZAEH 2013
ZAEH, M. F. (2013): Wirtschaftliche Fertigung mit Rapid-Technologien: Anwender-Leitfaden zur Auswahl geeigneter Verfahren. 1st ed. Munich: Hanser. 274 pp. ISBN: 978-3446439160. DOI: 10.3139/9783446439573.

ZAEH & LUTZMANN 2010
ZAEH, M. F.; LUTZMANN, S. (2010): Modelling and simulation of electron beam melting. In: Production Engineering 4.1, pp. 15–23. ISSN: 0944-6524. DOI: 10.1007/s11740-009-0197-6.

ZANDER et al. 2014
ZANDER, N.; BOG, T.; ELHADDAD, M.; ESPINOZA, R.; HU, H.; JOLY, A.; WU, C.; ZERBE, P.; DUESTER, A.; KOLLMANNSBERGER, S.; PARVIZIAN, J.; RUESS, M.; SCHILLINGER, D.; RANK, E. (2014): FCMLab: A finite cell research toolbox for MATLAB. In: Advances in Engineering Software 74, pp. 49–63. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2014.04.004.

ZELLER & ZAEH 2019
ZELLER, C.; ZAEH, M. F. (2019): Algorithm for the Penetration Depth in Laser Beam Melting and Algorithm for Optimal Time Steps in the Simulation of Laser Beam Melting [Code book]. In: mediaTUM - Media and Publications Repository of the Technical University of Munich. DOI: 10.14459/2019mp1515159.

ZELLER et al. 2018
ZELLER, C.; SURENDRAN, B.; ZAEH, M. F. (2018): Parameterized Extended Finite Element Method for high thermal gradients. In: Journal of Computational Design and Engineering 5.3, pp. 329–336. ISSN: 2288-4300. DOI: 10.1016/j.jcde.2017.12.001.

ZELLER et al. 2019
ZELLER, C.; ZHAO, X. F.; ZAEH, M. F. (2019): Source Code of the Parameterized eXtended Finite Element Method [Code book]. In: mediaTUM - Media and Publications Repository of the Technical University of Munich. DOI: 10.14459/2019mp1515157.

ZENG et al. 2012
ZENG, K.; PAL, D.; STUCKER, B. (2012): A review of thermal analysis methods in laser sintering and selective laser melting. In: Proceedings of the 23rd Solid Freeform Fabrication Symposium, Austin (Texas) / USA, August 6-8, 2012. Ed. by D. L. BOURELL; J. J. BEAMAN; R. H. CRAWFORD; C. C. SEEPERSAD; H. MARCUS, pp. 796–814.

L. ZHANG et al. 2004
ZHANG, L.; REUTZEL, E.; MICHALERIS, P. (2004): Finite element modeling discretization requirements for the laser forming process. In: International Journal of Mechanical Sciences 46.4, pp. 623–637. ISSN: 0020-7403. DOI: 10.1016/j.ijmecsci.2004.04.001.

L. ZHANG et al. 2007
ZHANG, L.; MICHALERIS, P.; MARUGABANDHU, P. (2007): Evaluation of Applied Plastic Strain Methods for Welding Distortion Prediction. In: Journal of Manufacturing Science and Engineering 129.6, pp. 1000–1010. ISSN: 1087-1357. DOI: 10.1115/1.2716740.

Y. ZHANG et al. 2018
ZHANG, Y.; GUILLEMOT, G.; BERNACKI, M.; BELLET, M. (2018): Macroscopic thermal finite element modeling of additive metal manufacturing by selective laser melting process. In: Computer Methods in Applied Mechanics and Engineering 331, pp. 514–535. ISSN: 0045-7825. DOI: 10.1016/j.cma.2017.12.003.

ZIENKIEWICZ & TAYLOR 2002
ZIENKIEWICZ, O. C.; TAYLOR, R. L. (2002): The finite element method. 5th ed. Oxford: Butterworth-Heinemann. 689 pp. ISBN: 978-0750650496.

A Appendix

A.1 Determination of the Penetration Depth for the Solution of the Semi-Infinite Heat Equation with an Initial Energy Deposition

A domain with constant temperature distribution is assumed, which only undergoes a temperature change in the region $0 \leq x \leq d_{\text{ed}}$ at time $t = 0$, i. e. $T^0 = T^{0,2}$. The temperature at point $x = 0$ is used for the wall temperature $T_w(t)$, as it corresponds to the highest temperature difference compared to the initial temperature throughout entire period under consideration.

$$
\begin{aligned}
T_w(t) &= \frac{\left(T^{0,1} - T^{0,2}\right)}{2} \left[\operatorname{erf}\left(\frac{d_{\text{ed}}}{\sqrt{4at}}\right) + \operatorname{erf}\left(\frac{d_{\text{ed}}}{\sqrt{4at}}\right)\right] + T^{0,2} && \text{(A.1)}\\
&= \frac{\left(T^{0,1} - T^{0,2}\right)}{2} 2\operatorname{erf}\left(\frac{d_{\text{ed}}}{\sqrt{4at}}\right) + T^{0,2} && \text{(A.2)}
\end{aligned}
$$

Since the region from 0 to d_{ed} has already undergone a temperature change, the penetration depth is shifted by d_{ed}. Thus, for the determination of the penetration depth for the analytical solution of an initial energy deposition (Equation 2.25), the equation

$$
\begin{aligned}
& \frac{T_w(t) - T(d_{\text{ed}} + \delta(t), t)}{T_w(t) - T^{0,2}} = 0.99 && \text{(A.3)}\\
\Rightarrow \quad & \frac{2\operatorname{erf}\left(\frac{1}{\sqrt{4at}}\right) - \left[\operatorname{erf}\left(\frac{2+\delta}{\sqrt{4at}}\right) + \operatorname{erf}\left(\frac{-\delta}{\sqrt{4at}}\right)\right]}{2\operatorname{erf}\left(\frac{1}{\sqrt{4at}}\right)} = 0.99 && \text{(A.4)}
\end{aligned}
$$

has to be solved. Without loss of generality, $d_{\text{ed}} = 1$ was used due to the proven convertibility of different solutions by a characteristic factor $\bar{x}$ (cf. Subsection 2.3.2). Summarizing the terms results in

$$
0.02\operatorname{erf}\left(\frac{1}{\sqrt{4at}}\right) - \left[\operatorname{erf}\left(\frac{2+\delta}{\sqrt{4at}}\right) + \operatorname{erf}\left(\frac{-\delta}{\sqrt{4at}}\right)\right] = 0 \qquad \text{(A.5)}
$$

Due to the numerous appearing error functions, a direct solution depending on the variables a and t, as in Equation 2.33, is not possible. The equation is therefore solved for different explicit values at and a regression analysis is performed.

The analytical solutions for a heating rod (cf. Equation 2.21) as well as for the considered deposition of energy at the boundary (cf. Equation 2.25) are thermal processes in which the energy propagates from one side of the domain. For this reason it is expected that the temporal behavior of the penetration depth is similar. Therefore, a root function as in the penetration depth for the heating rod in Equation 2.33 is used for the model function $f_{\text{model}} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ of the regression analysis, but it is supplemented by a correction polynomial of first order to take into account differences in the propagation of the temperature front. Two regression analyses are performed with the model function

$$f_{\text{model}}(x) = p_1\sqrt{x} + p_2 x + p_3 \tag{A.6}$$

and the parameters $p_1, p_2, p_3 \in \mathbb{R}$ are to be determined.

First, an analysis is carried out to determine the penetration depth for small values $at \ll 3\,\text{mm}^2$ and then for large values $1\,\text{mm}^2 \ll at$. In the former case, the parameter $p_3 = 0$ is selected in advance to ensure that the function searched for passes through the origin. In both analyses, 30 data points are generated from Equation A.3. In the case of small values at, the data points are logarithmically distributed in the interval from $1 \times 10^{-7}\,\text{mm}^2$ to $3\,\text{mm}^2$ to assure especially accurate approximations close to the origin. The data points of the regression analysis for the large values at are distributed equidistantly in the interval from $1\,\text{mm}^2$ to $50\,\text{mm}^2$. The results[36] of the regression analyses using MATLAB are displayed in Table A.1. In both cases, a *standard fit error* of less than 1.4×10^{-2} was achieved, i. e. the data points are, on average, $1.4 \times 10^{-2}\,\%$ of their δ value away from the fitted function. The best possible approximation is not claimed by the model function in Equation A.6, but the accuracies achieved are sufficient for the considerations in the context of this thesis.

Table A.1: Determined parameter by the regression analyses.

	$at \ll 3\,\text{mm}^2$	$1\,\text{mm}^2 \ll at$
p_1	2.86	3.86
p_2 in mm^{-1}	3.74×10^{-1}	8.12×10^{-3}
p_3 in mm	0	6.60×10^{-1}

[36] ZELLER & ZAEH (2019), penetration_depth.m

A.2 Cardinality of a Full Factorial Design of Experiments for an Arbitrary Number of Components of the Cuboid Structure

A parameter combination of the LBM-MM with h_{mr} components corresponds to the selection of $h_{\mathrm{mr}} - 1$ values from the set Π of possible component-specific calculation intervals as well as from the set Ψ of possible numbers of layers before a component starts. The sequence is not relevant, since it results from the selected values. In addition, the values for the first component are not selected, as they are fixed at $[(0, 1)]$. This is a combined *urn problem* and the number of possible parameter combinations for each kind of boundary condition is therefore calculated via

$$\sum_{s=1}^{\min(|\Pi|,|\Psi|)} \binom{|\Pi|}{s} \binom{|\Psi|}{s} . \tag{A.7}$$

In the case of the cuboid structure (cf. Figure 5.8) the following applies:

$$\Pi = \{2, 4, 8, 16, 32, 64\}, \qquad |\Pi| = 6, \tag{A.8}$$

$$\Psi = \{1, 2, \ldots, 20\}, \qquad |\Psi| = 20, \tag{A.9}$$

and this results in

$$\sum_{s=1}^{6} \binom{6}{s} \binom{20}{s} = 230\,229. \tag{A.10}$$

This applies both to the LBM-MM with the Dirichlet boundary condition and with the adiabatic boundary condition. Overall, this leads to 460 458 possible parameter combinations.

A.3 Data of the Chapter 4: Parameterized XFEM

A.3.1 Subsection 4.3.2: Spatial gradient within one layer

Nodal temperature values T_n^m

Table A.2: Data of Figure 4.4, FEM.

z in mm	T^1 in K, $t = 0.09\,\mathrm{s}$	T^2 in K, $t = 0.18\,\mathrm{s}$	T^3 in K, $t = 0.27\,\mathrm{s}$	T^4 in K, $t = 0.36\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15
5	385.67	415.88	444	470.22
10	345.98	340.74	337.15	335
15	354.73	355.57	355.85	355.69
20	352.8	352.68	352.72	352.86
25	353.23	353.24	353.21	353.16

Table A.3: Data in Figure 4.4, 'Exp'.

z in mm	T^1 in K, $t = 0.09\,\mathrm{s}$	T^2 in K, $t = 0.18\,\mathrm{s}$	T^3 in K, $t = 0.27\,\mathrm{s}$	T^4 in K, $t = 0.36\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15
5	342.63	329.03	315.64	307.95
10	355.47	358.01	359.95	360.13
15	352.64	352.18	351.96	352.19
20	353.26	353.34	353.35	353.25
25	353.13	353.11	353.12	353.15

Table A.4: Data in Figure 4.4, 'Erf'.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.09\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 0.18\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 0.27\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 0.36\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15
5	353.15	353.76	357.08	364.23
10	353.15	353.02	352.31	350.9
15	353.15	353.18	353.33	353.6
20	353.15	353.14	353.11	353.06
25	353.15	353.15	353.16	353.17

Table A.5: Data in Figure 4.5, 'Exp diff'.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.09\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 0.18\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 0.27\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 0.36\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15
5	353.14	353.02	352.78	353.52
10	353.15	353.18	353.23	353.05
15	353.15	353.14	353.13	353.18
20	353.15	353.15	353.15	353.14
25	353.15	353.15	353.15	353.15

Table A.6: Data in Figure 4.7, 'Exp diff trans'.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.09\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 0.18\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 0.27\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 0.36\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15
5	353.08	355.81	362.61	373.23
10	353.17	352.56	351.18	349.24
15	353.15	353.28	353.56	353.9
20	353.15	353.12	353.07	353.01
25	353.15	353.16	353.17	353.18

Table A.7: Data in Figure 4.7, ’Erf trans’.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.09\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 0.18\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 0.27\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 0.36\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15
5	353.15	353.15	353.15	353.29
10	353.15	353.15	353.15	353.12
15	353.15	353.15	353.15	353.16
20	353.15	353.15	353.15	353.15
25	353.15	353.15	353.15	353.15

Parameterization variables $\zeta_{n,o}$

Table A.8: Data in Figure 4.6, 'Exp' and 'Exp diff'.

t in s	ζ_{opt}	$\zeta_{1,2}$, Exp	$\zeta_{1,2}$, Exp diff
$9 \cdot 10^{-2}$	6.14	28.11	31.25
0.18	4.17	10.7	12.58
0.27	3.2	6.39	7.8
0.36	2.6	4.52	5.67

Table A.9: Data in Figure 4.6, 'Erf'.

t in s	ζ_{opt}	$\zeta_{1,2}$, Erf
$9 \cdot 10^{-2}$	$9 \cdot 10^{-2}$	$9.01 \cdot 10^{-2}$
0.18	0.18	0.18
0.27	0.27	0.27
0.36	0.36	0.37

Table A.10: Data in Figure 4.8, 'Exp trans diff'.

t in s	ζ_{opt}	$\zeta_{1,2}$, Exp trans diff
$9 \cdot 10^{-2}$	$2.65 \cdot 10^{-2}$	$2.11 \cdot 10^{-2}$
0.18	$5.76 \cdot 10^{-2}$	$4.89 \cdot 10^{-2}$
0.27	$9.75 \cdot 10^{-2}$	$8.53 \cdot 10^{-2}$
0.36	0.15	0.13

Table A.11: Data in Figure 4.8, 'Erf trans'.

t in s	ζ_{opt}	$\zeta_{1,2}$, Erf trans
$9 \cdot 10^{-2}$	3.33	12.48
0.18	2.36	5.4
0.27	1.92	3.49
0.36	1.67	2.65

Error e

Table A.12: Data in Figure 4.9 and Figure 4.10, FEM.

Element ϵ_u	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.09$ s	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.18$ s	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.27$ s	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.36$ s
1	1,050.21	885.3	771.28	684.1
2	38.21	74.2	107.28	135.99
3	8.42	14.71	19.15	21.97
4	1.86	2.88	3.25	3.12
5	0.41	0.55	0.52	0.36

Table A.13: Data in Figure 4.9 and Figure 4.10, 'Exp'.

Element ϵ_u	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.09$ s	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.18$ s	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.27$ s	Error e in $\text{mm}^{\frac{1}{2}}$ K, $t = 0.36$ s
1	579.37	513.32	439.11	375.52
2	12.36	28.6	45.87	59.51
3	2.73	5.75	8.12	8.46
4	0.6	1.15	1.43	1.18
5	0.13	0.23	0.24	0.13

Table A.14: Data in Figure 4.9 and Figure 4.10, 'Erf'.

Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.09\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.18\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.27\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.36\,\mathrm{s}$
1	0.34	1.48	5.21	8.91
2	$2.42 \cdot 10^{-3}$	0.65	3.63	8.97
3	$5.43 \cdot 10^{-4}$	0.16	0.99	2.66
4	$1.2 \cdot 10^{-4}$	$3.49 \cdot 10^{-2}$	0.21	0.54
5	$2.64 \cdot 10^{-5}$	$7.7 \cdot 10^{-3}$	$4.5 \cdot 10^{-2}$	0.11

Table A.15: Data in Figure 4.9 and Figure 4.10, 'Exp diff'.

Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.09\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.18\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.27\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.36\,\mathrm{s}$
1	588.2	541.43	484.28	432.48
2	$9.66 \cdot 10^{-3}$	0.25	1.97	6.02
3	$2.12 \cdot 10^{-3}$	$3.27 \cdot 10^{-2}$	$8.91 \cdot 10^{-2}$	0.12
4	$4.68 \cdot 10^{-4}$	$7.11 \cdot 10^{-3}$	$1.81 \cdot 10^{-2}$	$3.16 \cdot 10^{-2}$
5	$1.03 \cdot 10^{-4}$	$1.55 \cdot 10^{-3}$	$3.67 \cdot 10^{-3}$	$8.06 \cdot 10^{-3}$

Table A.16: Data in Figure 4.9 and Figure 4.10, 'Exp diff trans'.

Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.09\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.18\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.27\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.36\,\mathrm{s}$
1	102.39	98.77	89.64	79.83
2	$8.48 \cdot 10^{-2}$	3.05	10.04	19.19
3	$1.87 \cdot 10^{-2}$	0.69	2.33	4.64
4	$4.12 \cdot 10^{-3}$	0.15	0.48	0.89
5	$9.09 \cdot 10^{-4}$	$3.4 \cdot 10^{-2}$	0.1	0.17

Oscillations

Table A.17: Data in Figure 4.11a.

time step m	FEM T in K	Exp T in K	Exp diff T in K	Exp trans diff T in K	Erf T in K
1	-7.17	-10.52	$-8.19 \cdot 10^{-3}$	$-7.22 \cdot 10^{-2}$	$-4.62 \cdot 10^{-4}$
2	-12.41	-24.12	-0.13	-0.59	-0.13
3	-16	-37.51	-0.37	-1.97	-0.84
4	-18.15	-45.2	-0.1	-3.91	-2.25

Table A.18: Data in Figure 4.11b.

time step m	Exp in %	Exp diff in %	Exp trans diff in %	Erf in %
1	146.75	0.11	1.01	$6.44 \cdot 10^{-3}$
2	194.31	1.03	4.75	1.09
3	234.49	2.31	12.33	5.25
4	249.02	0.57	21.55	12.38

A.3.2 Subsection 4.3.3: Spatial gradient over multiple layers

Nodal temperature values T_n^m

Table A.19: Data in Figure 4.12, FEM.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.5\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 1\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 1.5\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 2\,\mathrm{s}$	$\boldsymbol{T}^5$ in K, $t = 2.5\,\mathrm{s}$	$\boldsymbol{T}^6$ in K, $t = 3\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15	1,523.15	1,523.15
5	492.67	596.32	675.98	739.07	790.34	832.92
10	343.04	354.77	377.39	404.98	434.34	463.77
15	353.88	351.64	350.6	352.13	356.38	363.04
20	353.1	353.36	353.2	352.78	352.44	352.52
25	353.15	353.13	353.17	353.2	353.17	353.07

Table A.20: Data in Figure 4.12, 'Exp diff'.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.5\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 1\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 1.5\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 2\,\mathrm{s}$	$\boldsymbol{T}^5$ in K, $t = 2.5\,\mathrm{s}$	$\boldsymbol{T}^6$ in K, $t = 3\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15	1,523.15	1,523.15
5	405.52	504.41	593.43	666.75	727.08	777.32
10	353.65	359.13	373.66	393.88	417.72	443.35
15	353.11	352.79	353.82	355.94	359.54	364.73
20	353.15	353.17	353.05	353	353.11	353.5
25	353.15	353.15	353.16	353.15	353.12	353.1

Table A.21: Data in Figure 4.12, 'Erf'.

z in mm	$\boldsymbol{T}^1$ in K, $t = 0.5\,\mathrm{s}$	$\boldsymbol{T}^2$ in K, $t = 1\,\mathrm{s}$	$\boldsymbol{T}^3$ in K, $t = 1.5\,\mathrm{s}$	$\boldsymbol{T}^4$ in K, $t = 2\,\mathrm{s}$	$\boldsymbol{T}^5$ in K, $t = 2.5\,\mathrm{s}$	$\boldsymbol{T}^6$ in K, $t = 3\,\mathrm{s}$
0	1,523.15	1,523.15	1,523.15	1,523.15	1,523.15	1,523.15
5	425.73	527.11	613.76	684.32	742.14	790.26
10	353.91	361.09	376.31	397.63	422.32	448.49
15	353.09	352.68	353.51	355.79	359.7	365.26
20	353.15	353.18	353.05	352.96	353.04	353.44
25	353.15	353.15	353.16	353.15	353.12	353.09

Parameterization variables $\zeta_{n,o}$

Table A.22: Data in Figure 4.13, 'Exp diff'.

t in s	$\zeta_{\text{opt},1,2}$	$\zeta_{1,2}$, Exp diff	$\zeta_{\text{opt},2,3}$	$\zeta_{2,3}$, Exp diff	$\zeta_{\text{opt},3,4}$	$\zeta_{3,4}$, Exp diff
0	0	$3.16 \cdot 10^{-3}$	0	$3.16 \cdot 10^{-3}$	0	0
0.5	0.5	0.36	0.16	0.21	0	0
1	0.91	0.68	0.3	0.33	0	0.33
1.5	1.32	0.98	0.44	0.44	0.26	0.37
2	1.74	1.27	0.58	0.55	0.35	0.43
2.5	2.15	1.57	0.72	0.67	0.43	0.5
3	2.57	1.87	0.86	0.78	0.51	0.57

Table A.23: Data in Figure 4.13, 'Erf'.

t in s	ζ_{opt}	$\zeta_{1,2}$, Erf	$\zeta_{2,3}$, Erf	$\zeta_{3,4}$, Erf
0	0	$1 \cdot 10^{-4}$	0.1	0
0.5	0.5	0.57	0.8	0
1	1	1.09	1.23	1
1.5	1.5	1.56	1.68	1.91
2	2	2.02	2.12	2.48
2.5	2.5	2.47	2.58	2.96
3	3	2.92	3.04	3.42

Error e

Table A.24: Data in Figure 4.14 and Figure 4.15, FEM.

Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.5\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 1\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 1.5\,\mathrm{s}$
1	563.95	346.27	240.27
2	156.61	205.06	205.81
3	12.61	1.42	20.08
4	0.91	1.83	3.36
5	$6.61 \cdot 10^{-2}$	0.26	$8.12 \cdot 10^{-2}$
Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 2\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 2.5\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 3\,\mathrm{s}$
1	179.44	140.79	114.42
2	192.42	175.73	159.35
3	37.93	51.21	60.21
4	2.42	1.05	4.97
5	0.46	0.97	1.17

Table A.25: Data in Figure 4.14 and Figure 4.15, 'Exp diff'.

Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.5\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 1\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 1.5\,\mathrm{s}$
1	139.41	70.61	51.96
2	17.27	18.65	14.26
3	0.62	6.25	8.7
4	$4.55 \cdot 10^{-2}$	0.46	0.7
5	$3.3 \cdot 10^{-3}$	$2.68 \cdot 10^{-2}$	0.13
Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 2\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 2.5\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 3\,\mathrm{s}$
1	42.78	36.58	31.76
2	12.68	13.26	14.51
3	11.45	11.92	10.9
4	2.69	5.2	7.8
5	0.2	0.14	0.14

Table A.26: Data in Figure 4.14 and Figure 4.15, 'Erf'.

Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 0.5\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 1\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 1.5\,\mathrm{s}$
1	99.56	62.68	34.11
2	36.66	45.49	39.37
3	0.94	8.7	9.99
4	$6.87 \cdot 10^{-2}$	0.59	0.32
5	$4.98 \cdot 10^{-3}$	$3.33 \cdot 10^{-2}$	0.12
Element ϵ_u	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 2\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 2.5\,\mathrm{s}$	Error e in $\mathrm{mm}^{\frac{1}{2}}\,\mathrm{K}$, $t = 3\,\mathrm{s}$
1	18.24	9.47	4.51
2	31.44	24.29	18.37
3	14.58	16.66	16.88
4	2.47	5.35	8.42
5	0.26	0.24	$9.45 \cdot 10^{-2}$

Oscillations

Table A.27: Data in Figure 4.16a.

time step m	FEM T in K	Exp diff T in K	Erf T in K
1	-10.11	$-3.65 \cdot 10^{-2}$	$-5.51 \cdot 10^{-2}$
2	-1.51	-0.36	-0.47
3	-2.55	-0.1	$-9.64 \cdot 10^{-2}$
4	-1.02	-0.15	-0.19
5	-0.71	$-4.04 \cdot 10^{-2}$	-0.11
6	-0.63	$-5.25 \cdot 10^{-2}$	$-6.17 \cdot 10^{-2}$

Table A.28: Data in Figure 4.16b.

time step m	Exp in %	Erf in %
1	0.36	0.55
2	24.08	31.12
3	4	3.78
4	14.63	18.88
5	5.74	16.14
6	8.4	9.87

A.3.3 Section 4.4: Validation

Table A.29: Data in Figure 4.17.

d_{refine}	$t^{m_Y} = 0.36\,\text{s}$, $m_Y = 4$, Error e in $\text{mm}^{\frac{1}{2}}\,\text{K}$	$t^{m_Y} = 3\,\text{s}$, $m_Y = 6$, Error e in $\text{mm}^{\frac{1}{2}}\,\text{K}$
1	1,050.21	563.95
2	567.56	180.72
3	335.62	92.54
4	208.76	173.08
5	130.33	237.26
6	86.46	288.63
7	91.71	332.26
8	114.94	369.58
9	138.39	402

A.4 Data of the Chapter 5: LBM Multirate Method

Table A.30: Data in Figure 5.5.

time step m	κ_w^m in K	$\frac{\partial^2 T}{\partial t^2}(t, 10d_1)$ in $\text{K}\,\text{s}^{-2}$	$T(t, 10d_1)$ in K
1	$8.24 \cdot 10^{-21}$	$1.31 \cdot 10^{-26}$	80
2	$1.51 \cdot 10^{-20}$	$6.86 \cdot 10^{-24}$	80
3	$1.51 \cdot 10^{-18}$	$1.63 \cdot 10^{-21}$	80
4	$8.9 \cdot 10^{-17}$	$2.05 \cdot 10^{-19}$	80
5	$3.4 \cdot 10^{-15}$	$1.5 \cdot 10^{-17}$	80
6	$8.96 \cdot 10^{-14}$	$6.91 \cdot 10^{-16}$	80
7	$1.71 \cdot 10^{-12}$	$2.15 \cdot 10^{-14}$	80
8	$2.47 \cdot 10^{-11}$	$4.73 \cdot 10^{-13}$	80
9	$2.78 \cdot 10^{-10}$	$7.68 \cdot 10^{-12}$	80
10	$2.51 \cdot 10^{-9}$	$9.5 \cdot 10^{-11}$	80
11	$1.87 \cdot 10^{-8}$	$9.22 \cdot 10^{-10}$	80
12	$1.16 \cdot 10^{-7}$	$7.19 \cdot 10^{-9}$	80
13	$6.15 \cdot 10^{-7}$	$4.6 \cdot 10^{-8}$	80
14	$2.82 \cdot 10^{-6}$	$2.45 \cdot 10^{-7}$	80
15	$1.13 \cdot 10^{-5}$	$1.11 \cdot 10^{-6}$	80
16	$4 \cdot 10^{-5}$	$4.31 \cdot 10^{-6}$	80
17	$1.27 \cdot 10^{-4}$	$1.45 \cdot 10^{-5}$	80
18	$3.6 \cdot 10^{-4}$	$4.31 \cdot 10^{-5}$	80

Table A.30: Data in Figure 5.5.

time step m	κ_w^m in K	$\frac{\partial^2 T}{\partial t^2}(t, 10d_1)$ in K s^{-2}	$T(t, 10d_1)$ in K
19	$9.31 \cdot 10^{-4}$	$1.13 \cdot 10^{-4}$	80
20	$2.19 \cdot 10^{-3}$	$2.64 \cdot 10^{-4}$	80
21	$4.73 \cdot 10^{-3}$	$5.55 \cdot 10^{-4}$	80
22	$9.38 \cdot 10^{-3}$	$1.05 \cdot 10^{-3}$	80.01
23	$1.72 \cdot 10^{-2}$	$1.81 \cdot 10^{-3}$	80.02
24	$2.91 \cdot 10^{-2}$	$2.83 \cdot 10^{-3}$	80.04
25	$4.57 \cdot 10^{-2}$	$4.04 \cdot 10^{-3}$	80.1
26	$6.65 \cdot 10^{-2}$	$5.28 \cdot 10^{-3}$	80.21
27	$8.94 \cdot 10^{-2}$	$6.3 \cdot 10^{-3}$	80.43
28	0.12	$6.88 \cdot 10^{-3}$	80.8
29	0.15	$6.83 \cdot 10^{-3}$	81.41
30	0.16	$6.12 \cdot 10^{-3}$	82.35
31	0.17	$4.85 \cdot 10^{-3}$	83.74
32	0.17	$3.26 \cdot 10^{-3}$	85.65
33	0.15	$1.63 \cdot 10^{-3}$	88.17
34	0.11	$2.09 \cdot 10^{-4}$	91.3
35	0.21	$-8.36 \cdot 10^{-4}$	95.03
36	0.33	$-1.45 \cdot 10^{-3}$	99.25
37	0.46	$-1.67 \cdot 10^{-3}$	103.81
38	0.62	$-1.59 \cdot 10^{-3}$	108.49
39	0.75	$-1.34 \cdot 10^{-3}$	113.06
40	0.84	$-1.03 \cdot 10^{-3}$	117.28
41	0.89	$-7.19 \cdot 10^{-4}$	120.93
42	0.89	$-4.63 \cdot 10^{-4}$	123.83
43	0.88	$-2.73 \cdot 10^{-4}$	125.87
44	0.81	$-1.46 \cdot 10^{-4}$	126.97
45	0.69	$-6.81 \cdot 10^{-5}$	127.14
46	0.53	$-2.57 \cdot 10^{-5}$	126.42
47	0.33	$-5.27 \cdot 10^{-6}$	124.9
48	0.15	$2.72 \cdot 10^{-6}$	122.72
49	0.28	$4.62 \cdot 10^{-6}$	119.99
50	0.47	$4.06 \cdot 10^{-6}$	116.88
51	0.65	$2.85 \cdot 10^{-6}$	113.51
52	0.81	$1.74 \cdot 10^{-6}$	110.01
53	0.95	$9.66 \cdot 10^{-7}$	106.49
54	1.07	$4.9 \cdot 10^{-7}$	103.05
55	1.17	$2.29 \cdot 10^{-7}$	99.75
56	1.24	$9.84 \cdot 10^{-8}$	96.67
57	1.3	$3.88 \cdot 10^{-8}$	93.83
58	1.35	$1.39 \cdot 10^{-8}$	91.27
59	1.38	$4.52 \cdot 10^{-9}$	89
60	1.4	$1.31 \cdot 10^{-9}$	87.02

Table A.30: Data in Figure 5.5.

time step m	κ_w^m in K	$\frac{\partial^2 T}{\partial t^2}(t, 10d_1)$ in $\mathrm{K\,s^{-2}}$	$T(t, 10d_1)$ in K
61	1.42	$3.31 \cdot 10^{-10}$	85.33
62	1.43	$7.16 \cdot 10^{-11}$	83.93
63	1.43	$1.29 \cdot 10^{-11}$	82.79
64	1.44	$1.85 \cdot 10^{-12}$	81.89

Table A.31: Data in Figure 5.6 and second derivative in Figure 5.7.

time step m	$\Delta t^m / \Delta t^{m+1}$	$\frac{\partial^2 T}{\partial t^2}$ in $\mathrm{K\,s^{-2}}$
3	1.07	26.71
4	1.08	23.27
5	1.08	20.09
6	1.08	17.22
7	1.09	14.67
8	1.09	12.44
9	1.09	10.49
10	1.09	8.8
11	1.1	7.36
12	1.1	6.12
13	1.1	5.07
14	1.1	4.19
15	1.11	3.44
16	1.11	2.81
17	1.11	2.29
18	1.11	1.86
19	1.12	1.5
20	1.12	1.21
21	1.12	0.96
22	1.12	0.77
23	1.13	0.61
24	1.13	0.48
25	1.13	0.37
26	1.14	0.29
27	1.14	0.23
28	1.14	0.17
29	1.15	0.13
30	1.15	0.1
31	1.16	$7.54 \cdot 10^{-2}$
32	1.16	$5.63 \cdot 10^{-2}$
33	1.17	$4.17 \cdot 10^{-2}$
34	1.17	$3.06 \cdot 10^{-2}$
35	1.18	$2.23 \cdot 10^{-2}$
36	1.18	$1.6 \cdot 10^{-2}$

Table A.31: Data in Figure 5.6 and second derivative in Figure 5.7.

time step m	$\Delta t^m / \Delta t^{m+1}$	$\frac{\partial^2 T}{\partial t^2}$ in $\mathrm{K\,s^{-2}}$
37	1.19	$1.14 \cdot 10^{-2}$
38	1.2	$8.07 \cdot 10^{-3}$
39	1.2	$5.63 \cdot 10^{-3}$
40	1.21	$3.88 \cdot 10^{-3}$
41	1.22	$2.64 \cdot 10^{-3}$
42	1.23	$1.77 \cdot 10^{-3}$
43	1.24	$1.17 \cdot 10^{-3}$
44	1.25	$7.59 \cdot 10^{-4}$
45	1.26	$4.85 \cdot 10^{-4}$
46	1.28	$3.04 \cdot 10^{-4}$
47	1.29	$1.87 \cdot 10^{-4}$
48	1.31	$1.12 \cdot 10^{-4}$
49	1.32	$6.57 \cdot 10^{-5}$
50	1.34	$3.76 \cdot 10^{-5}$
51	1.36	$2.08 \cdot 10^{-5}$
52	1.39	$1.12 \cdot 10^{-5}$
53	1.41	$5.82 \cdot 10^{-6}$
54	1.45	$2.91 \cdot 10^{-6}$
55	1.48	$1.39 \cdot 10^{-6}$
56	1.52	$6.35 \cdot 10^{-7}$
57	1.57	$2.74 \cdot 10^{-7}$
58	1.63	$1.11 \cdot 10^{-7}$
59	1.69	$4.22 \cdot 10^{-8}$
60	1.77	$1.47 \cdot 10^{-8}$
61	1.87	$4.68 \cdot 10^{-9}$
62	2	$1.34 \cdot 10^{-9}$
63	2.15	$3.35 \cdot 10^{-10}$
64	2.36	$7.21 \cdot 10^{-11}$

Table A.32: Maximum appearing nodal temperature deviation of the reduced systems in Figure 5.10b and in Figure 5.11b.

time step m	Dirichlet T in K	Dirichlet Standard deviation	adiabatic T in K	adiabatic Standard deviation
1	$1.39 \cdot 10^{-2}$	$5.5 \cdot 10^{-3}$	$1.41 \cdot 10^{-2}$	$3.96 \cdot 10^{-3}$
3	0.12	$2.69 \cdot 10^{-2}$	$8.18 \cdot 10^{-2}$	$1.02 \cdot 10^{-2}$
5	$2.17 \cdot 10^{-2}$	$3.32 \cdot 10^{-3}$	$1.94 \cdot 10^{-2}$	$4.17 \cdot 10^{-3}$
7	$3.91 \cdot 10^{-2}$	$9.5 \cdot 10^{-3}$	$2.51 \cdot 10^{-2}$	$3.54 \cdot 10^{-3}$
9	$3.46 \cdot 10^{-2}$	$6.97 \cdot 10^{-3}$	$2.84 \cdot 10^{-2}$	$4.57 \cdot 10^{-3}$
11	$1.45 \cdot 10^{-2}$	$3.05 \cdot 10^{-3}$	$1.15 \cdot 10^{-2}$	$2.11 \cdot 10^{-3}$
13	$9.75 \cdot 10^{-3}$	$1.31 \cdot 10^{-3}$	$1.4 \cdot 10^{-2}$	$2.47 \cdot 10^{-3}$
15	$2.96 \cdot 10^{-2}$	$3.31 \cdot 10^{-3}$	$2.74 \cdot 10^{-2}$	$2.36 \cdot 10^{-3}$

Table A.32: Maximum appearing nodal temperature deviation of the reduced systems in Figure 5.10b and in Figure 5.11b.

time step m	Dirichlet T in K	Dirichlet Standard deviation	adiabatic T in K	adiabatic Standard deviation
17	$2.5 \cdot 10^{-2}$	$8.11 \cdot 10^{-3}$	$3.17 \cdot 10^{-2}$	$9.61 \cdot 10^{-3}$
19	$5.52 \cdot 10^{-2}$	$1.37 \cdot 10^{-2}$	$8.68 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$
21	0.31	$1.88 \cdot 10^{-2}$	0.45	$4.59 \cdot 10^{-2}$
23	0.98	$2.82 \cdot 10^{-2}$	1.39	$6.66 \cdot 10^{-2}$
25	2.2	$4.12 \cdot 10^{-2}$	3.1	$8.97 \cdot 10^{-2}$
27	4	$7.05 \cdot 10^{-2}$	5.59	0.19
29	6.24	0.13	8.68	0.39
31	8.57	0.25	11.96	0.74
33	10.88	0.6	14.91	1.21
35	12.87	1.17	17.09	1.79
37	13.7	1.81	18.32	2.45
39	13.59	2.71	18.73	3.1
41	12.35	3.5	18.69	3.65
43	10.26	3.78	18.57	3.95
45	7.84	3.29	18.49	3.85
47	5.2	2.65	18.25	3.15
49	3.25	1.19	17.29	1.94
51	2.74	1	14.84	1.81
53	3.2	0.92	10.46	2.74
55	2.08	1.06	5.22	2.31
57	0.71	0.51	1.54	1
59	0.12	0.1	0.23	0.19
61	$7.26 \cdot 10^{-3}$	$7.51 \cdot 10^{-3}$	$1.27 \cdot 10^{-2}$	$1.23 \cdot 10^{-2}$
63	$1.32 \cdot 10^{-4}$	$1.54 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$1.84 \cdot 10^{-4}$

Table A.33: Maximum appearing nodal temperature deviation of the full systems in Figure 5.10b and in Figure 5.11b.

time step m	Dirichlet T in K	Dirichlet Standard deviation	adiabatic T in K	adiabatic Standard deviation
2	$3.87 \cdot 10^{-3}$	$1.53 \cdot 10^{-3}$	$5.08 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$
4	$5.94 \cdot 10^{-2}$	$1.38 \cdot 10^{-2}$	$6.33 \cdot 10^{-2}$	$9.32 \cdot 10^{-3}$
6	$9.49 \cdot 10^{-3}$	$9.85 \cdot 10^{-4}$	$1.02 \cdot 10^{-2}$	$1.96 \cdot 10^{-3}$
8	$2.71 \cdot 10^{-2}$	$4.98 \cdot 10^{-3}$	$1.67 \cdot 10^{-2}$	$2.79 \cdot 10^{-3}$
10	$2.24 \cdot 10^{-2}$	$4.47 \cdot 10^{-3}$	$2.32 \cdot 10^{-2}$	$3.01 \cdot 10^{-3}$
12	$7.48 \cdot 10^{-3}$	$7.17 \cdot 10^{-4}$	$8.27 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$
14	$7.36 \cdot 10^{-3}$	$6.6 \cdot 10^{-4}$	$8.15 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$
16	$1.83 \cdot 10^{-2}$	$1.55 \cdot 10^{-3}$	$1.46 \cdot 10^{-2}$	$8.39 \cdot 10^{-4}$
18	$1.27 \cdot 10^{-2}$	$3.4 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	$1.51 \cdot 10^{-3}$
20	$4.01 \cdot 10^{-2}$	$6.23 \cdot 10^{-3}$	$2.67 \cdot 10^{-2}$	$3.92 \cdot 10^{-3}$
22	0.27	$1.4 \cdot 10^{-2}$	$9.44 \cdot 10^{-2}$	$9.59 \cdot 10^{-3}$

Table A.33: Maximum appearing nodal temperature deviation of the full systems in Figure 5.10b and in Figure 5.11b.

time step m	Dirichlet T in K	Dirichlet Standard deviation	adiabatic T in K	adiabatic Standard deviation
24	0.86	$2.39 \cdot 10^{-2}$	0.39	$2.25 \cdot 10^{-2}$
26	1.93	$4.16 \cdot 10^{-2}$	0.97	$5.47 \cdot 10^{-2}$
28	3.52	$8.89 \cdot 10^{-2}$	1.85	0.1
30	5.62	0.22	3.07	0.13
32	8.03	0.48	4.42	0.26
34	10.21	0.86	5.65	0.47
36	11.7	1.33	6.91	0.63
38	12.32	1.93	7.89	0.84
40	12.02	2.64	8.48	1.11
42	10.69	3.22	8.95	1.51
44	8.77	3.16	9.74	1.96
46	6.6	2.86	11.01	2.18
48	3.91	2.74	11.81	1.9
50	1.79	1.1	11.31	1.18
52	1.52	0.56	8.91	1.76
54	1.61	0.62	5.06	1.97
56	0.71	0.45	1.71	1.02
58	0.13	0.11	0.27	0.21
60	$8.11 \cdot 10^{-3}$	$8.23 \cdot 10^{-3}$	$1.44 \cdot 10^{-2}$	$1.34 \cdot 10^{-2}$
62	$1.46 \cdot 10^{-4}$	$1.69 \cdot 10^{-4}$	$1.85 \cdot 10^{-4}$	$1.94 \cdot 10^{-4}$
64	0	0	0	0

Table A.34: Maximum appearing nodal temperature deviation in each layer in Figure 5.12.

Local layer Λ_w	Dirichlet T in K	Dirichlet Standard deviation	adiabatic T in K	adiabatic Standard deviation
1	−7.76	1.84	15	1.39
2	−7.8	1.85	15.07	1.4
3	−7.88	1.86	15.23	1.45
4	−8.03	1.89	15.48	1.53
5	−8.27	1.87	15.83	1.65
6	−8.68	1.8	16.33	1.88
7	−9.42	1.63	16.99	2.2
8	−10.77	1.43	18.17	2.9
9	−12.79	1.34	19.9	3.36
10	−14.33	1.15	12.64	0.59
11	−14.73	1.15	11.42	0.51
12	−14.38	1.24	10.29	0.6
13	−13.7	1.35	−9.57	0.82
14	−12.77	1.23	−9.54	0.93
15	−11.78	1.29	−9.24	0.87

Table A.34: Maximum appearing nodal temperature deviation in each layer in Figure 5.12.

Local layer Λ_w	Dirichlet T in K	Dirichlet Standard deviation	adiabatic T in K	adiabatic Standard deviation
16	−10.79	1.09	−8.62	0.84
17	−9.65	0.83	−8.07	0.8
18	−8.54	0.72	−7.35	0.56
19	−7.67	0.42	−6.47	0.29
20	−6.72	0	−5.72	0

Table A.35: Size of system of equations in Figure 5.18.

Cooling cycle	size in %	Cooling cycle	size in %	Cooling cycle	size in %
1	100	31	17.01	61	11.92
2	100	32	16.76	62	11.8
3	100	33	16.52	63	11.69
4	100	34	16.28	64	11.59
5	100	35	16.06	65	11.48
6	100	36	15.84	66	11.38
7	100	37	15.63	67	11.28
8	59.21	38	15.42	68	11.18
9	58.97	39	15.22	69	11.08
10	58.75	40	15.02	70	10.98
11	58.54	41	14.83	71	10.89
12	58.33	42	14.65	72	10.8
13	31.1	43	14.47	73	10.71
14	22.98	44	14.29	74	10.62
15	22.5	45	14.13	75	10.54
16	22.04	46	13.96	76	10.45
17	21.61	47	13.8	77	10.37
18	21.19	48	13.64	78	10.29
19	20.79	49	13.49	79	10.21
20	20.41	50	13.34	80	10.13
21	20.04	51	13.19	81	10.05
22	19.68	52	13.05	82	9.97
23	19.34	53	12.91	83	9.9
24	19.01	54	12.78	84	9.83
25	18.69	55	12.65	85	9.76
26	18.39	56	12.52	86	9.68
27	18.09	57	12.39	87	9.62
28	17.81	58	12.27	88	9.55
29	17.53	59	12.15	89	9.48
30	17.27	60	12.03	90	9.41

A.5 Data of the Chapter 6: LBM-specific Direct Solving Strategy

A.5.1 Section 6.3: Verification

Table A.36: Data in Figure 6.7, Cuboid.

T_{fix}	Mean spectral radius ϱ	Standard deviation
80	1.25	0.16
141.84	1.08	0.15
203.68	0.94	0.13
265.53	0.81	0.12
327.37	0.7	0.11
389.21	0.61	$8.88 \cdot 10^{-2}$
451.05	0.53	$7.35 \cdot 10^{-2}$
512.89	0.46	$5.41 \cdot 10^{-2}$
574.74	0.41	$2.42 \cdot 10^{-2}$
636.58	0.42	$3.18 \cdot 10^{-4}$
698.42	0.45	$3.01 \cdot 10^{-4}$
760.26	0.47	$1.73 \cdot 10^{-4}$
822.11	0.5	$1.55 \cdot 10^{-4}$
883.95	0.52	$1.69 \cdot 10^{-4}$
945.79	0.53	$1.79 \cdot 10^{-4}$
1,007.63	0.55	$1.88 \cdot 10^{-4}$
1,069.47	0.56	$1.94 \cdot 10^{-4}$
1,131.32	0.58	$2.01 \cdot 10^{-4}$
1,193.16	0.59	$2.06 \cdot 10^{-4}$
1,255	0.61	$2.11 \cdot 10^{-4}$

Table A.37: Data in Figure 6.7, Panel.

T_{fix}	Mean spectral radius ϱ	Standard deviation
100.61	1.37	0.13
162.46	1.2	0.12
224.3	1.05	0.11
286.14	0.91	0.11
347.98	0.8	$9.81 \cdot 10^{-2}$
409.82	0.7	$8.24 \cdot 10^{-2}$
471.67	0.62	$6.37 \cdot 10^{-2}$
533.51	0.54	$4.82 \cdot 10^{-2}$
595.35	0.47	$3.35 \cdot 10^{-2}$
657.19	0.43	$6.05 \cdot 10^{-3}$
719.04	0.46	$8 \cdot 10^{-5}$
780.88	0.48	$4.64 \cdot 10^{-5}$

Table A.37: Data in Figure 6.7, Panel.

T_{fix}	Mean spectral radius ϱ	Standard deviation
842.72	0.5	$5.12 \cdot 10^{-5}$
904.56	0.52	$5.47 \cdot 10^{-5}$
966.4	0.54	$5.74 \cdot 10^{-5}$
1,028.25	0.56	$5.96 \cdot 10^{-5}$
1,090.09	0.57	$6.14 \cdot 10^{-5}$
1,151.93	0.59	$6.3 \cdot 10^{-5}$
1,213.77	0.6	$6.42 \cdot 10^{-5}$

Table A.38: Data in Figure 6.7, Strut.

T_{fix}	Mean spectral radius ϱ	Standard deviation
121.23	1.32	0.14
183.07	1.16	0.13
244.91	1.02	0.12
306.75	0.88	0.11
368.6	0.78	0.1
430.44	0.68	$8.08 \cdot 10^{-2}$
492.28	0.6	$6.1 \cdot 10^{-2}$
554.12	0.53	$4.38 \cdot 10^{-2}$
615.96	0.46	$2.49 \cdot 10^{-2}$
677.81	0.44	$8.5 \cdot 10^{-5}$
739.65	0.46	$5.71 \cdot 10^{-5}$
801.49	0.49	$4.1 \cdot 10^{-5}$
863.33	0.51	$4.46 \cdot 10^{-5}$
925.18	0.53	$4.75 \cdot 10^{-5}$
987.02	0.55	$4.97 \cdot 10^{-5}$
1,048.86	0.56	$5.14 \cdot 10^{-5}$
1,110.7	0.57	$5.29 \cdot 10^{-5}$
1,172.54	0.59	$5.43 \cdot 10^{-5}$
1,234.39	0.6	$5.54 \cdot 10^{-5}$

Table A.39: Spectral radius for different iterations in Figure 6.7.

Iteration i	$T_{\text{fix}} =$ 353.15 K	$T_{\text{fix}} =$ 483.70 K	$T_{\text{fix}} =$ 614.26 K	$T_{\text{fix}} =$ 744.82 K	$T_{\text{fix}} =$ 1528.15 K
1	0.47	0.27	0.26	0.34	0.61
2	1.04	0.75	0.52	0.36	0.61
3	1.07	0.78	0.55	0.38	0.61
4	1.06	0.77	0.54	0.38	0.61
5	1.07	0.77	0.55	0.38	0.61

Table A.39: Spectral radius for different iterations in Figure 6.7.

Iteration i	$T_{\text{fix}} =$ 353.15 K	$T_{\text{fix}} =$ 483.70 K	$T_{\text{fix}} =$ 614.26 K	$T_{\text{fix}} =$ 744.82 K	$T_{\text{fix}} =$ 1528.15 K
6	1.07	0.77	0.55	0.38	0.61
7	1.07	0.77	0.55	0.38	0.61
8	1.07	0.77	0.55	0.38	0.61
9	1.07	0.77	0.55	0.38	0.61

A.5.2 Section 6.4: Validation

Table A.40: Data in Figure 6.9, Cuboid.

	Standard Cholesky		$T_{\text{fix}} = 1473.15\,\text{K}$		$T_{\text{fix}} = 1073.15\,\text{K}$	
time step m	i	Standard deviation	i	Standard deviation	i	Standard deviation
1	6.45	0.5	13.75	0.7	9.95	0.38
2	5	0	12.85	0.96	11.55	0.97

Table A.41: Data in Figure 6.9, Panel.

	Standard Cholesky		$T_{\text{fix}} = 1473.15\,\text{K}$		$T_{\text{fix}} = 1073.15\,\text{K}$	
time step m	i	Standard deviation	i	Standard deviation	i	Standard deviation
1	6.03	0.18	13.83	1.1	11.27	0.57
2	5	0	13	0.93	10.93	0.36

Table A.42: Data in Figure 6.9, Strut.

	Standard Cholesky		$T_{\text{fix}} = 1473.15\,\text{K}$		$T_{\text{fix}} = 1073.15\,\text{K}$	
time step m	i	Standard deviation	i	Standard deviation	i	Standard deviation
1	6.03	0.31	11.63	1.45	10.2	0.7
2	5	0	11.37	1.28	9.4	0.76

Table A.43: Maximal residual flux per iteration in Figure 6.10, Strut.

Iteration i	$T_{\text{fix}} = 1473.15\,\text{K}$	Standard Cholesky	$T_{\text{fix}} = 1073.15\,\text{K}$
1	5,141.39	5,141.33	5,141.08
2	1,834.4	298.51	1,821.91
3	734.74	40.05	724.38
4	302.82	4.04	292.62
5	152.3	0.46	119.64
6	96.67		49.39
7	55.57		20.56

Table A.43: Maximal residual flux per iteration in Figure 6.10, Strut.

Iteration i	$T_{\text{fix}} = 1473.15\,\text{K}$	Standard Cholesky	$T_{\text{fix}} = 1073.15\,\text{K}$
8	30.28		8.62
9	16.3		3.64
10	8.74		1.55
11	4.62		0.66
12	2.42		
13	1.27		
14	0.66		

A.6 Data of the Chapter 7: Conclusion

Table A.44: Reduction of total computing time in Figure 7.2.

Time share	LBM-DSS	LBM-MM
35	82.68	89.4
33	83.67	90.01
31	84.66	90.61
29	85.65	91.22
27	86.64	91.83
25	87.63	92.43
23	88.62	93.04
21	89.61	93.64
19	90.6	94.25
17	91.59	94.85
15	92.58	95.46