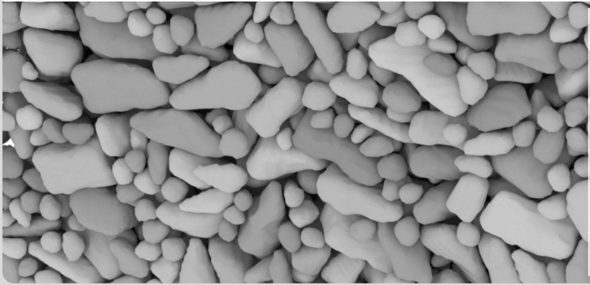
Karlsruhe Institute of Technology

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Development of a massive parallel and optimized phase-field solver for the sinter process

J. Hötzer, H. Hierl, F. Hafner, M. Seiz, L. Promberger, C. Seer, M. Kellner, W. Rheinheimer, M. Berghoff, B. Nestler

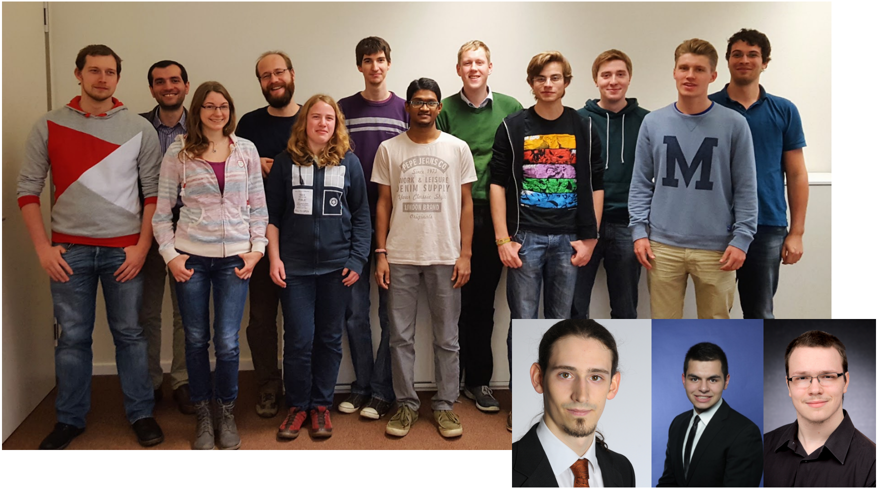
INSTITUTE OF APPLIED MATERIALS - COMPUTATIONAL MATERIALS SCIENCE



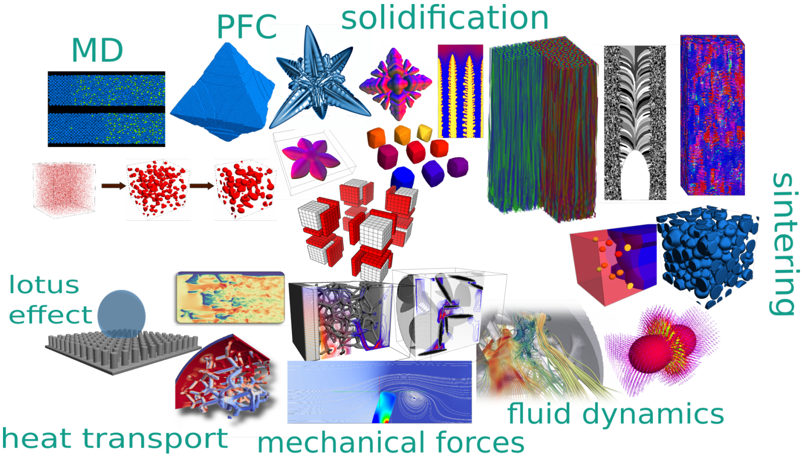
Contents:

Group - High Performance Materials Computing and Data Science

Karlsruhe Institute of Technology



Overview


MD
PFC
solidification
sintering
lotus
effect
heat transport
mechanical forces
fluid dynamics

Overview

Reality/Experiments

Physical Parameter

Mathematical Model

Numerical Scheme

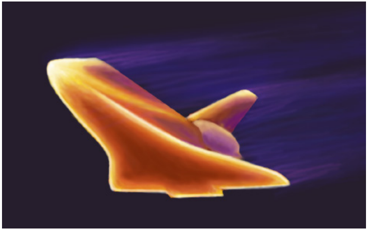
Application Program

Parallel Programming Mod-
els (OpenMP, MPI, OpenCl)

Hardware Architecture

Applications of ceramics

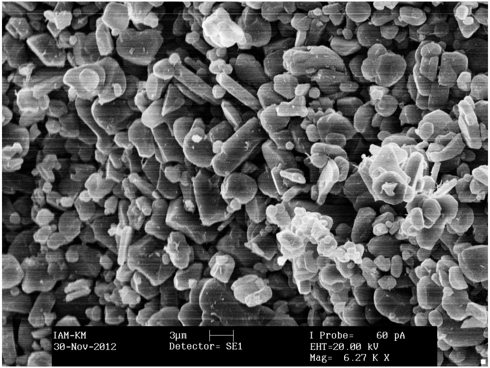
- everyday items (e.g. plates, cups) are "simple" to produce
- however high performance materials
  - sensors (e.g oxygen)
  - spark plugs
  - artificial hip joint
  - batteries
  - electronics
  - ...
- require a tailored microstructure with defined properties
- microstructure is directly influenced by various process and material properties



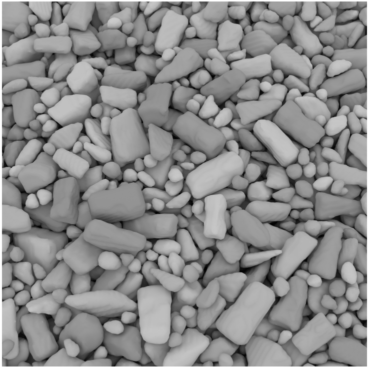
https://upload.wikimedia.org/wikipedia/commons/6/6a/Sparkplug.jpg
https://upload.wikimedia.org/wikipedia/commons/f/fd/Stsheat.jpg

Initial structure - Green body

- Experiment

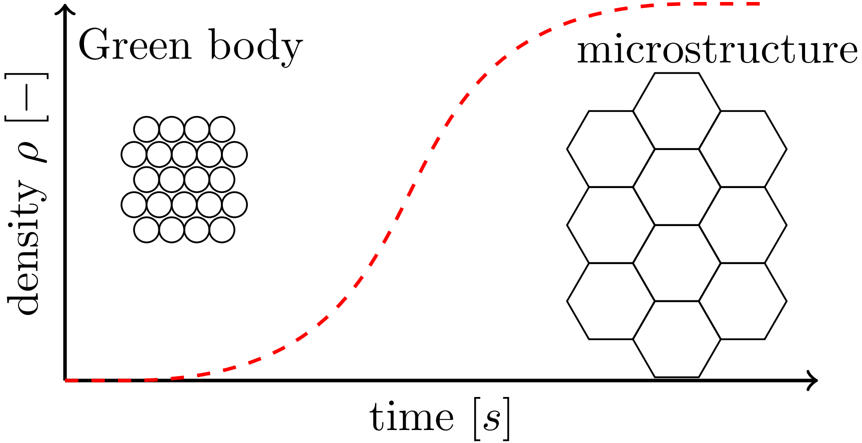


- Generated



experimental image: Fabian Lemke - IAM

Sintering process



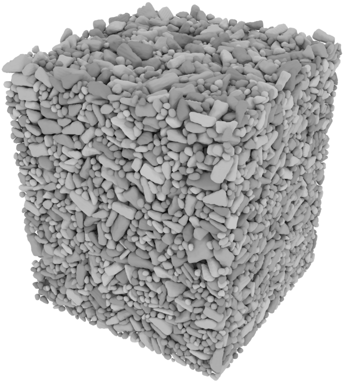
Sintering process in experiments

Video: Experiment of the sinter process

F. Lemke, IAM, KIT

Simulation setting

- solid state sintering
- coupled phase-field and concentration model
- different diffusion paths
- number of grains/particles $N \gg 1000$ with size distribution
- large domain sizes ($> 500^3$ cells)
- large parameter matrices (N^2, N^3)
- parallel PACE3D framework (MPI)

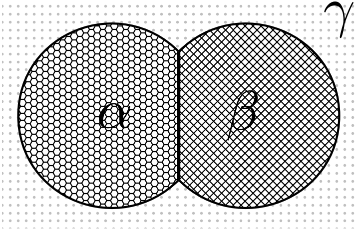


Phase-field model derivation

- total system energy (Laypounov functional)

$$\mathcal{L}(s_1, s_2, \ldots) = \sum_{\substack{\beta=1 \\ \alpha<\beta}}^{N} \int_{\partial V_\alpha} \gamma_{\alpha\beta}(\vec{n})\, dA_{\alpha\beta} + \sum_{\alpha=1}^{N} \int_{V_\alpha} f_{\text{bulk}}(s_1, s_2, \ldots)\, dV_\alpha$$

- surface energy $\gamma_{\alpha\beta}$ in direction $\vec{n}$
- bulk energy of a "phase"



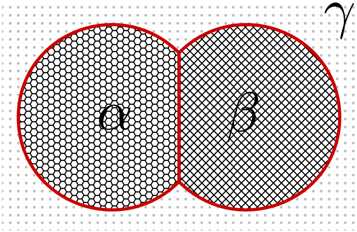
Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field model derivation

- total system energy (Laypounov functional)

$$\mathcal{L}(s_1, s_2, \dots) = \sum_{\substack{\beta=1 \\ \alpha<\beta}}^{N} \int_{\partial V_\alpha} \gamma_{\alpha\beta}(\vec{n}) dA_{\alpha\beta} + \sum_{\alpha=1}^{N} \int_{V_\alpha} f_{\text{bulk}}(s_1, s_2, \dots) dV_\alpha$$

- surface energy $\gamma_{\alpha\beta}$ in direction $\vec{n}$
- bulk energy of a "phase"



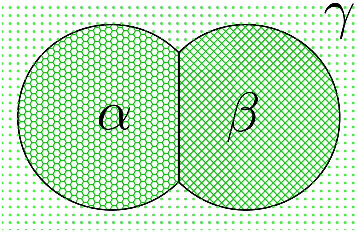
Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field model derivation

- total system energy (Laypounov functional)

$$\mathcal{L}(s_1, s_2, \ldots) = \sum_{\substack{\beta=1 \\ \alpha<\beta}}^{N} \int_{\partial V_\alpha} \gamma_{\alpha\beta}(\vec{n}) dA_{\alpha\beta} + \sum_{\alpha=1}^{N} \int_{V_\alpha} f_{\text{bulk}}(s_1, s_2, \ldots) dV_\alpha$$

- surface energy $\gamma_{\alpha\beta}$ in direction $\vec{n}$
- bulk energy of a "phase"



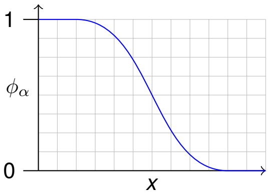
Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field model derivation

- Grand chemical potential functional:

$$\Psi(\phi, \boldsymbol{\mu}, T) = \int_\Omega \underbrace{\left(\epsilon a(\phi, \nabla\phi) + \frac{1}{\epsilon}\omega(\phi) \right)}_{\text{surface energy}} + \underbrace{\psi(\phi, \boldsymbol{\mu}, T)}_{\text{bulk potential}} \, d\Omega$$

- phase-field vector $\phi = (\phi_1, \phi_2, ..., \phi_N)^T$
- order parameter ϕ_α represents the volume fraction of each phase
- volumetric interface at the surface
- smooth transition between the order parameters
- Allen-Cahn type variational differentiation of the functional
- $\rightarrow$ **no interface tracking needed**



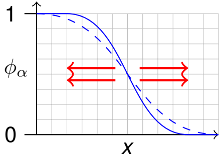
Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field model derivation

- Grand chemical potential functional:

$$\Psi(\phi, \boldsymbol{\mu}, T) = \int_\Omega \underbrace{\left(\epsilon a(\phi, \nabla\phi) + \frac{1}{\epsilon}\omega(\phi) \right)}_{\text{surface energy}} + \underbrace{\psi(\phi, \boldsymbol{\mu}, T)}_{\text{bulk potential}} \, d\Omega$$

- Gradient energy density



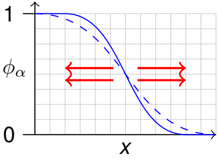
Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field model derivation

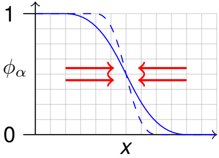
- Grand chemical potential functional:

$$\Psi(\phi, \boldsymbol{\mu}, T) = \int_\Omega \underbrace{\left(\epsilon a(\phi, \nabla\phi) + \frac{1}{\epsilon}\omega(\phi) \right)}_{\text{surface energy}} + \underbrace{\psi(\phi, \boldsymbol{\mu}, T)}_{\text{bulk potential}} \, d\Omega$$

- Gradient energy density



- Interfacial free energy density



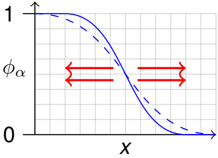
Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field model derivation

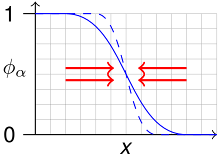
- Grand chemical potential functional:

$$\Psi(\phi, \mu, T) = \int_\Omega \underbrace{\left(\epsilon a(\phi, \nabla\phi) + \frac{1}{\epsilon}\omega(\phi) \right)}_{\text{surface energy}} + \underbrace{\psi(\phi, \mu, T)}_{\text{bulk potential}} \, d\Omega$$

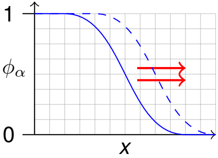
- Gradient energy density



- Interfacial free energy density



- Bulk free energy density



Choudhury, A., Nestler, B (2012). Physical Review E, 85 (2), 71(4), 021602

Phase-field algorithm

- lattice fields
  - two AoS for phase-field (ϕ_{src}, ϕ_{dst})
  - two SoA for chemical potential ($\boldsymbol{\mu}_{src}$, $\boldsymbol{\mu}_{dst}$)
- storing new values calculated from *src* in *dst*

Algorithm 1 calculation of one time step

$\phi_{dst} \leftarrow \phi$-kernel$\left(\phi_{src}, \boldsymbol{\mu}_{src}\right)$

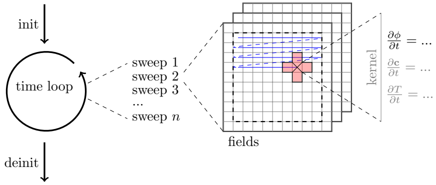
$\boldsymbol{\mu}_{dst} \leftarrow \mu$-kernel$\left(\boldsymbol{\mu}_{src}, \phi_{src}, \phi_{dst}\right)$

ϕ_{dst}-boundary conditions

$\boldsymbol{\mu}_{dst}$-boundary conditions

$\phi_{dst}, \boldsymbol{\mu}_{dst}$-ghost layer exchange

swap $\phi_{src} \leftrightarrow \phi_{dst}$ and $\boldsymbol{\mu}_{src} \leftrightarrow \boldsymbol{\mu}_{dst}$



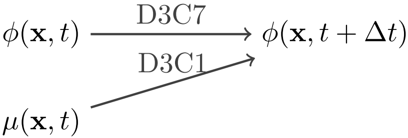
ϕ-kernel

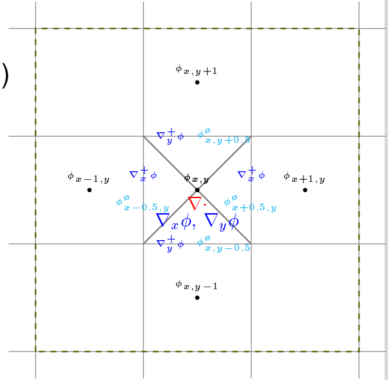
$$\tau\varepsilon\frac{\partial\phi_\alpha}{\partial t} = \underbrace{-\varepsilon\left(\frac{\partial a(\phi,\nabla\phi)}{\partial\phi_\alpha} + \nabla\cdot\frac{\partial a(\phi,\nabla\phi)}{\partial\nabla\phi_\alpha}\right)}_{\text{D3C7}}\underbrace{-\frac{1}{\varepsilon}\frac{\partial\omega(\phi)}{\partial\phi_\alpha} - \frac{\partial\psi(\phi,\mu,T)}{\partial\phi_\alpha} + \lambda}_{\text{D3C1}}$$

- finite differences scheme for space
- explicit Euler scheme for the time discretization
- roofline performance model:

$^{\text{FLOP}}/_{cell}$ (likwid)	893 / 7812 ($N = 4$ / $N = 8$)
loads & stores	40 $^{\text{byte}}/_{\text{phase}}$

⟶ compute bound





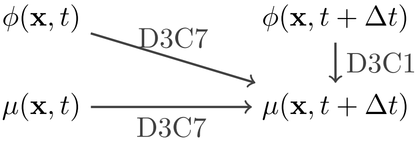
μ-kernel

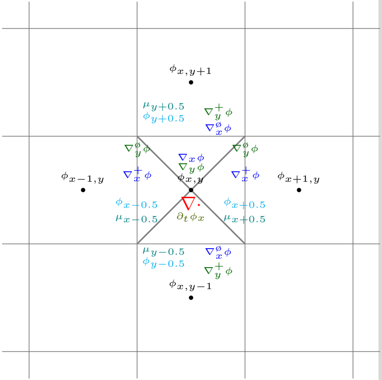
$$\frac{\partial \boldsymbol{\mu}}{\partial t} = \underbrace{\left[\sum_{\alpha=1}^{N} h_\alpha(\vec{\phi})\left(\frac{\partial \vec{c}^{\alpha}(\boldsymbol{\mu}, T)}{\partial \boldsymbol{\mu}}\right)\right]^{-1}}_{\text{D3C1}} \left(\underbrace{\nabla \cdot \left(\boldsymbol{M}(\vec{\phi}, \boldsymbol{\mu}, T)\nabla \boldsymbol{\mu}\right)}_{\text{D3C7}} \underbrace{- \sum_{\alpha=1}^{N} \vec{c}^{\alpha}(\boldsymbol{\mu}, T)\frac{\partial h_\alpha(\phi)}{\partial t} - \sum_{\alpha=1}^{N} h_\alpha(\vec{\phi})\left(\frac{\partial \vec{c}^{\alpha}(\boldsymbol{\mu}, T)}{\partial T}\right)\frac{\partial T}{\partial t}}_{\text{D3C1}}\right)$$

- finite differences scheme for space
- explicit Euler scheme for the time discretization
- roofline performance model:

FLOP/cell	467
loads & stores	144

⟶ compute bound





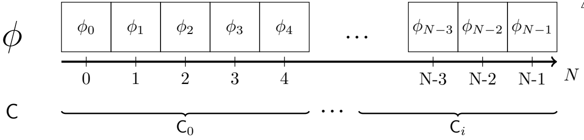
Optimizations layer

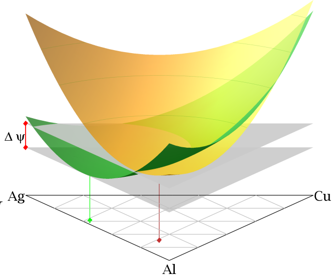
- Parameter layer
- Model layer
- Algorithm layer
- Hardware layer

Optimizations I

Parameter layer

- fitting of Gibbs energies with parabolic approach from CALPHAD databases to calculate the driving forces
- reduction of the parameter matrices with the size $N \times N$ and $N \times N \times N$ to a class based concept of 2×2 and $2 \times 2 \times 2$





Optimizations II

Karlsruhe Institute of Technology

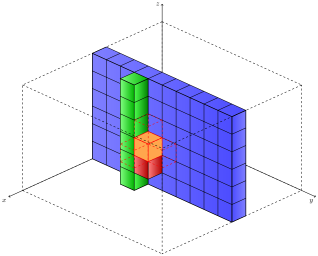
Model layer

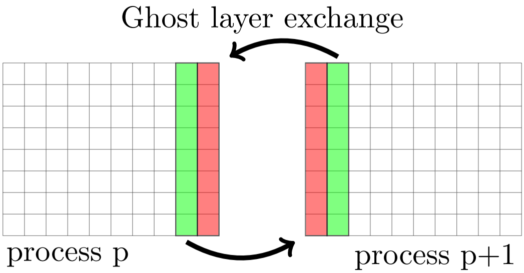
- simplifications due to defined setup (e.g. fix number of concentrations)
- classification of cells $\longrightarrow$ skip terms
  $\partial_t \phi = \ldots$ needs only calculated in the diffuse interface
- elimination and pre-calculation of common subexpressions (e.g. 1/2 $\longrightarrow$ 0.5)

Optimizations III

Algorithm layer

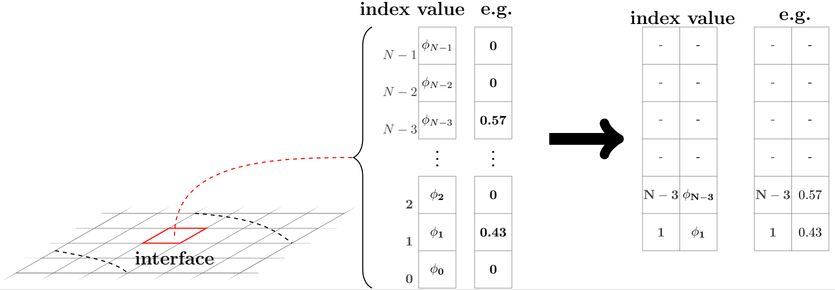
- access patterns / stencils (streaming)
- domain decomposition (MPI)
- buffering of staggered values - point line plane buffer
- local reduction of order parameter (LROP) for ϕ





Local reduced order parameter (LROP)

- in models **maximal six phases** in one cell enough
  (Kim, Kim, Kim and Park, (2006), Physical Review E, 74, 061605)
- only **storage** phase **values** $\phi_\alpha \neq 0$ **and their index** in the phase-field vector ϕ instead of all *N* elements
- **other phases** are assumed to be **zero**
  $\longrightarrow$ memory requirements independent from number of phases
  $\longrightarrow$ reduction of calculation time $\sum_\alpha^N \ldots \rightarrow \sum_\alpha^{\max(6)} \ldots$



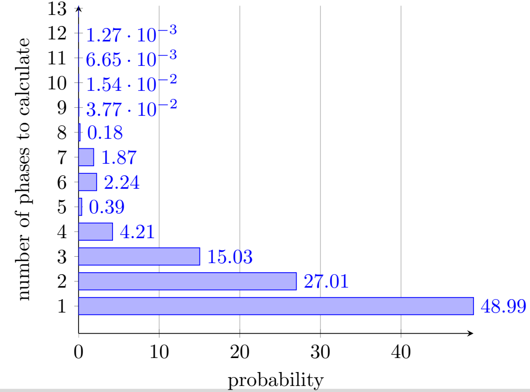
Optimizations IV

Hardware layer

- **explicit** vectorization with SIMD **intrinsics**
- light weight macro layer to support **SSE** and **AVX**
- for $\partial_t \mu$ **classical approach**, calculate multiple cells at once
- for $\partial_t \phi$ the **calculation per cell** is **vectorized**
  $\longrightarrow$ calculate multiple phases at once
  $\longrightarrow$ still possible to use all optimizations (e.g. classification)
  $\longrightarrow$ **LROP cells differ** between neighboring cells, but for **vectorization** they **need the same structure** which results in **complex sorting**
  $\longrightarrow$ good experience with vectorization of four phases (**up to** 25% **peak performance**)

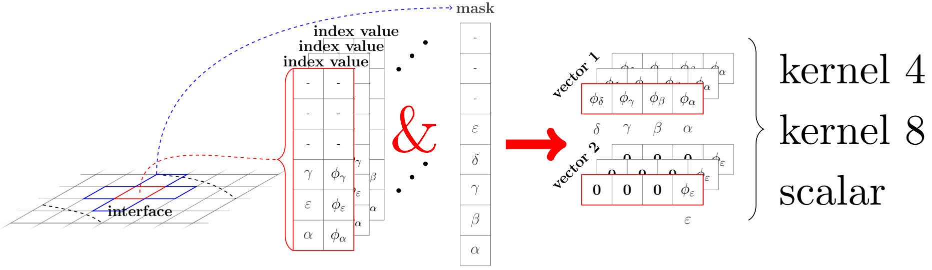
Vectorization

- many **vector matrix multiplications** of the form $\mathbf{y} = \sum \mathbf{Ax}$
- optimized **pattern approach** to pre-rotate all combinations of x for four and eight phases
- **three kernels** depending on the number of phases N to calculate in current cell
  - ⟶ vectorized kernel for **4** phases
  - ⟶ vectorized kernel for **8** phases
  - ⟶ scalar kernel for **more** than eight phases

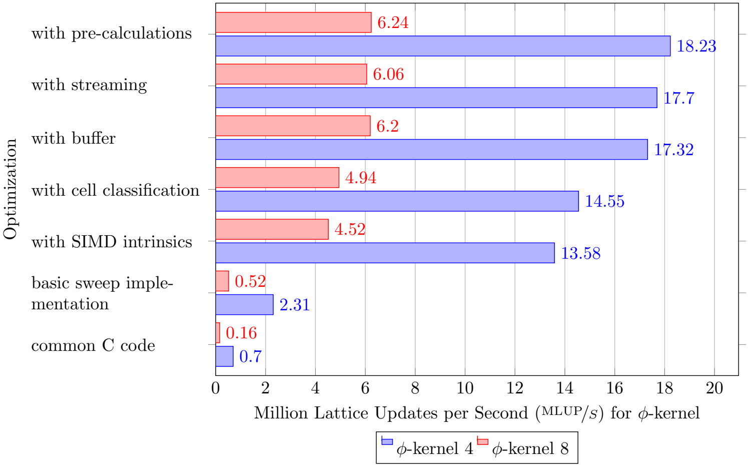


Vectorization

- **mapping** of **LROP cell** to **SIMD vector**
  ⟶ all local ϕ vectors of the stencil and matrices need the same order to calculate e.g. $\nabla\phi$
- create **mask** depending on stencil
- create **SSE/AVX vectors** from LROP cell **based on mask**
- depending on size of mask select the optimal kernel

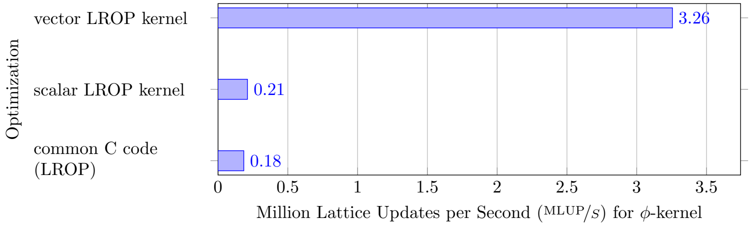


Optimization results – ϕ-kernel 4 / 8 – Hazel Hen



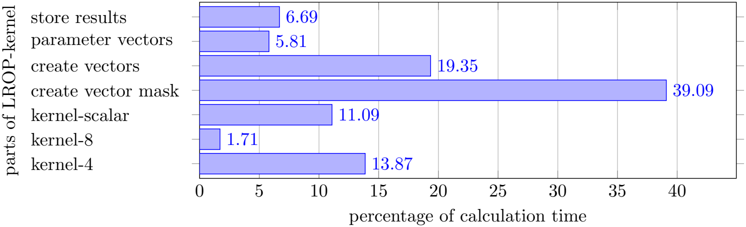
- $60 \times 60 \times 60$ cells per block
- only kernel without mapping from LROP to SSE/AVX vectors

Optimization results – LROP-kernel – Hazel Hen



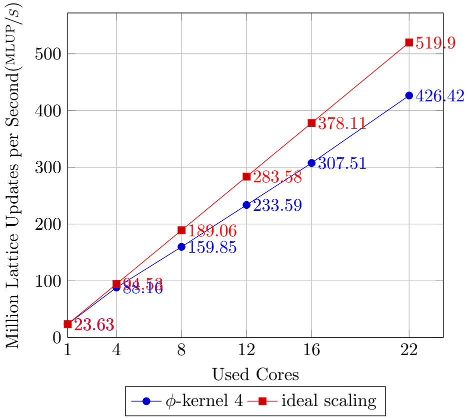
- preliminary results of LROP-kernel with mapping
- 17.9 % to 52.2 % of single ϕ-kernels

LROP-kernel analysis of typical simulation



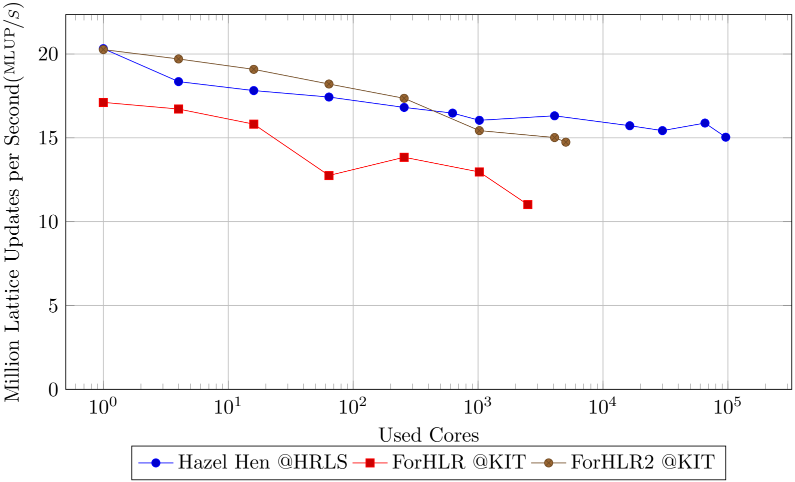
- preliminary results of LROP-kernel with mapping
- mapping from LROP cell to vectors requires 71.75 %
- calculation requires 27.48 %

Single node scaling – ϕ-kernel 4 – Hazel Hen



- preliminary results of ϕ-kernel 4
- $60 \times 60 \times 60$ cells per block

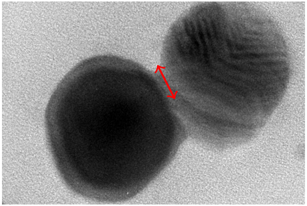
Scaling results – ϕ-kernel 4 – Hazel Hen



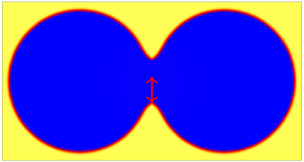
- preliminary results for ϕ-kernel 4 only

Validation

- classical two particle system
- measure parameter: neck radius X
- analytics:
  $X = At^n$, $n \in [0.14, 0.33]$

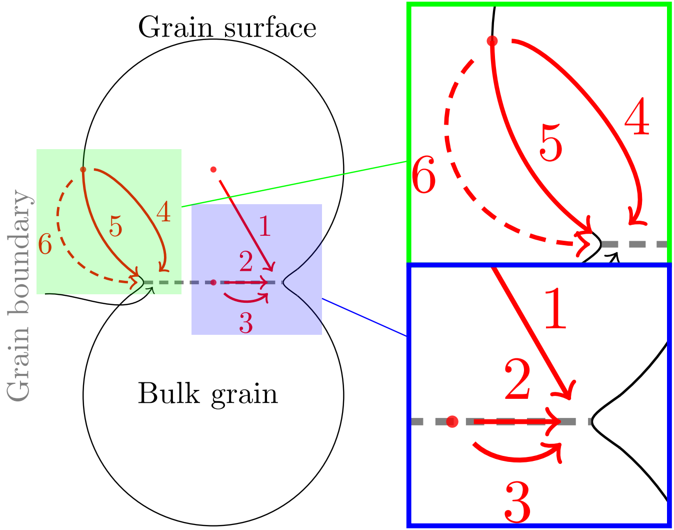


Asoro et al., Acta Materialia 81 (2014): 173-183.

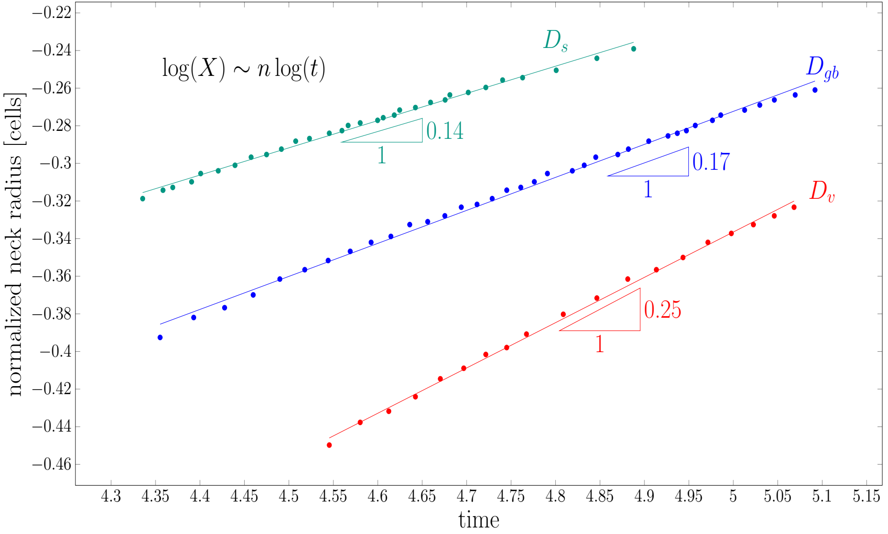


Benjamin Ehreiser, IAM, KIT/Ceramic Processing and Sintering; Wang et al. 2006 Acta Materialia

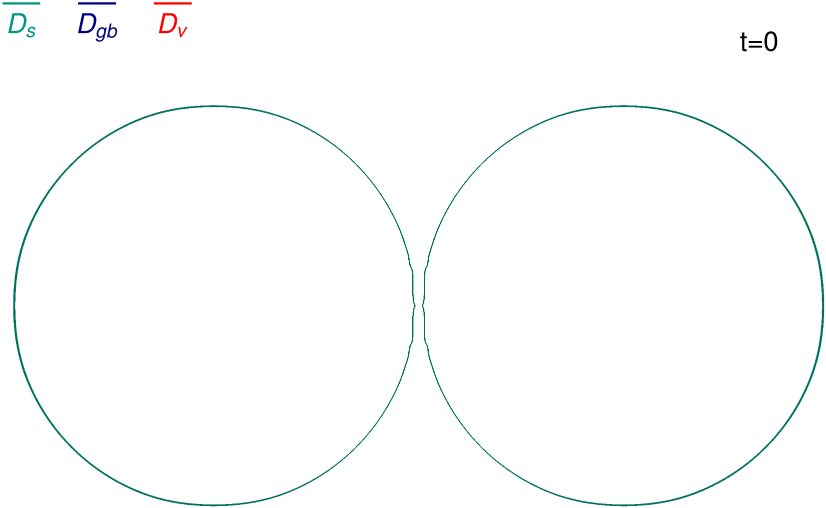
Diffusion paths



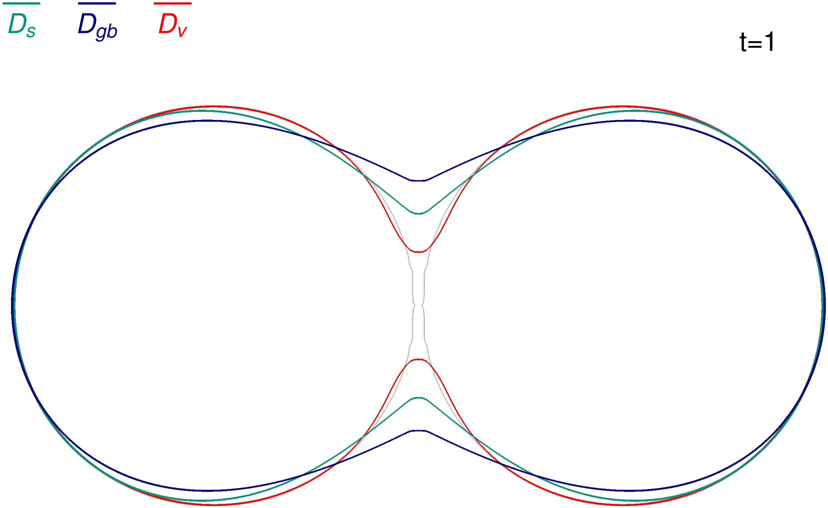
Validation: neck radius



Validation: contour lines

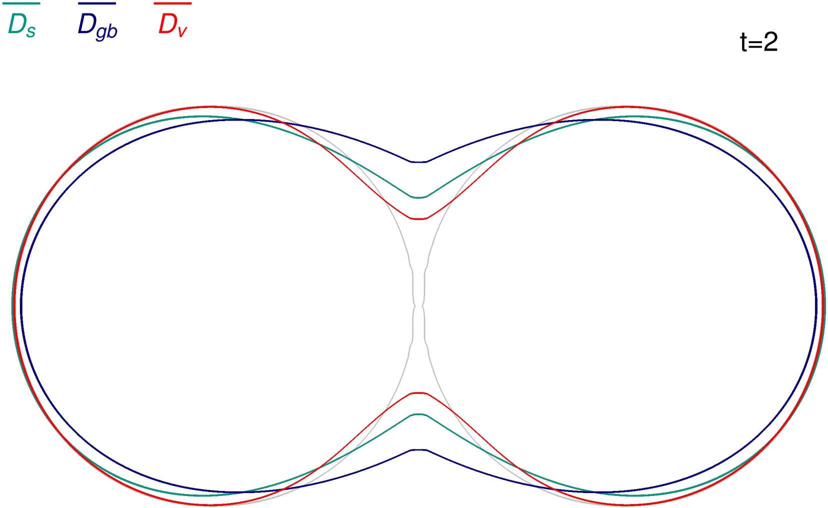


Validation: contour lines

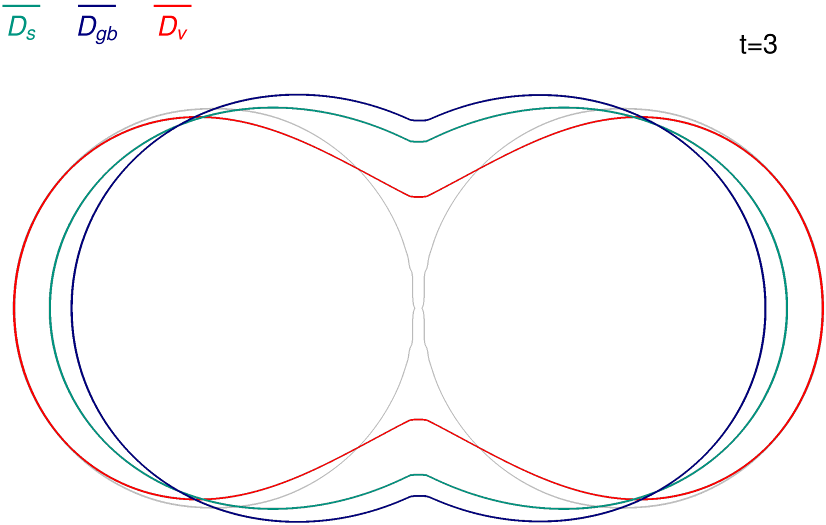


Validation: contour lines

Karlsruhe Institute of Technology

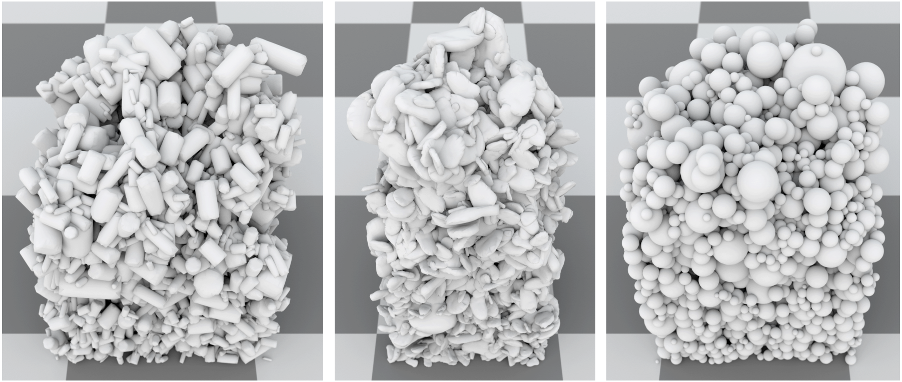


Validation: contour lines



Green body generator

- generation of packings with defined
  - ⟶ density
  - ⟶ particle size distribution
  - ⟶ particle shapes



Hötzer et al., Forschung Aktuell, Hochschule Karlsruhe, 2016

Simumation of the sinter process

Video: Simulation of the sinter process

- 400^3 cells, 1333 cores, 24h

Conclusions

Karlsruhe Institute of Technology

Preliminary summary

- efficient calculation of multi phase-field models
- connecting of highly optimized and vectorized kernels
- still optimization potential

Future work

- optimize mapping of LROP cells to vector cells
- optimize mask creation
  $\longrightarrow$ additional field indicating change of mask
  $\longrightarrow$ test only cells ahead of iteration direction
- buffering of parameter vectors depending on mask
- communication hiding for MPI

END

Thank you for your attention!

Open questions? Ideas? Improvements?

Contact:

- Johannes Hötzer
  `johannes.hoetzer@kit.edu`
- Britta Nestler
  `britta.nestler@kit.edu`