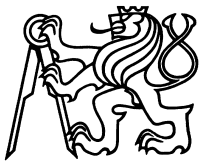
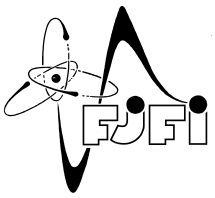


Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering



High performance computing methods for numerical solution of problems with phase transitions

Metody vysoce výkonného počítání pro numerické řešení úloh s fázovými přechody

Bachelor's Degree Project

Author: **Kryštof Jakůbek**

Supervisor: **Ing. Pavel Strachota, Ph.D.**

Language advisor: **Copeland, Darren, MSc.**

Academic year: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jakůbek** Jméno: **Kryštof** Osobní číslo: **509234**

Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**

Zadávající katedra/ústav: **Katedra matematiky**

Studijní program: **Matematické inženýrství**

Specializace: **Matematická informatika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Metody vysoce výkonného počítání pro numerické řešení úloh s fázovými přechody

Název bakalářské práce anglicky:

High performance computing methods for numerical solution of problems with phase transitions

Pokyny pro vypracování:

1. Prostudujte modely dynamiky kontinua popisující fyzikální procesy s fázovými přechody. Zaměřte se na popis tuhnutí a tání, proudění a silové interakce mezi pevnou a kapalnou fází.
2. Seznamte se s modelováním fázových přechodů metodou fázového pole.
3. Seznamte se základy numerických metod pro řešení matematických modelů výše zmíněných procesů.
4. Diskutujte možnosti paralelního zpracování výsledných algoritmů na různých typech architektur pro vysoce výkonné počítání (mnohajádrové systémy, distribuované systémy, GPU akcelerátory).
5. Implementujte numerický řešič typové úlohy tuhnutí a tání materiálu a pokuste se o jeho paralelizaci. Použijte vhodnou numerickou metodu, např. metodu konečných diferencí nebo metodu konečných objemů.
6. Prezentujte a komentujte výsledky numerických simulací.

Seznam doporučené literatury:

[1] N. Provatas, K. Elder, Phase-Field Methods in Materials Science and Engineering. WILEY-VCH Verlag GmbH & Co. KGaA, 2010.
[2] Y. Wang et al., Simulation of Microstructure Evolution in Mg Alloys by Phase-Field Methods: A Review. Crystals 12, 2022, 1305.
[3] P. Strachota, A. Wodecki, M. Beneš, Focusing the latent heat release in 3D phase field simulations of dendritic crystal growth. Modelling Simul. Mater. Sci. Eng. 29, 2021, 065009.
[4] I. Steinbach, Phase-field models in materials science. Modelling Simul. Mater. Sci. Eng. 17, 2009, 073001.
[5] T. Takaki, S. Sakane, R. Suziki, High-performance GPU computing of phase-field lattice
Boltzmann simulations for dendrite growth with natural convection. IOP Conf. Series: Materials Science and Engineering 1281, 2023, 012056.
[6] F. Moukalled, L. Mangani, M. Darwish et al., The finite volume methods in computational fluid dynamics. Springer, 2016.
[7] J. Palán, Modely fázového pole v materiálových vědách a jejich numerické řešení. Diplomová práce, FJFI ČVUT v Praze, 2023.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Strachota, Ph.D. katedra matematiky FJFI

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **24.10.2023** Termín odevzdání bakalářské práce: **05.08.2024**

Platnost zadání bakalářské práce: **30.09.2025**

Ing. Pavel Strachota, Ph.D.	prof. Ing. Zuzana Masáková, Ph.D.	doc. Ing. Václav Čuba, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

28. 11. 2023 — Datum převzetí zadání

Podpis studenta

Acknowledgment:
I would like to thank ..... for (his/her expert guidance) and express my gratitude to ..... for (his/her language assistance).

Author's declaration:
I declare that this Bachelor's Degree Project is entirely my own work and I have listed all the used sources in the bibliography.

Prague, August 5, 2024 Kryštof Jakůbek

Název práce:

Metody vysoce výkonného počítání pro numerické řešení úloh s fázovými přechody

Autor: Kryštof Jakůbek

Specializace: Matematické inženýrství

Specializace: Matematická informatika

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Pavel Strachota, Ph.D., pracoviště školitele (název instituce, fakulty, katedry...)

Abstrakt: Abstrakt max. na 10 řádků.

Klíčová slova: klíčová slova (nebo výrazy) seřazená podle abecedy a oddělená čárkou

Title:

High performance computing methods for numerical solution of problems with phase transitions

Author: Kryštof Jakůbek

Abstract: Max. 10 lines of English abstract text.

Key words: keywords in alphabetical order separated by commas

Contents

Introduction

Paragraphs of the Introduction...

Chapter 1

Models with phase transition's

1.1 Notation

$L_{0x}, L_{0y} \in \mathbb{R}$ ... Real size of simulation region

$L_0 := (0, L_{0x}) \times (0, L_{0y})$ ... Simulation region

$a, b, \alpha, \beta, \xi \in \mathbb{R}$ ... Simulation parameters

K, L ... Some grid cells

$x, y \in L_0$ ... Some positions within the region

$\sigma = K|L$ ... The boundary between cells K and L

$m(K), m(\sigma)$ ... Lebesgue measure of K and σ

$\rho(x, y)$ ... Distance between x and y

$\rho(x, \sigma) := \inf_{y \in \sigma} \rho(x, y)$ ... Distance between x and the closest point on boundary σ

$x_K := \frac{1}{m(K)} \int_K \boldsymbol{x}\, dx$ ... Position in the centroid of cell K

$x_\sigma := \frac{1}{m(\sigma)} \int_\sigma \boldsymbol{x}\, dx$ ... Position in the centroid of boundary σ

$\Phi_K := \Phi(x_K), T_K := T(x_K)$

$\Phi_\sigma := \Phi(x_\sigma)$

E, W, N, S ... Cells to the East, West, North and South relative to cell K respectively

1.2 Stefan Problem

1.3 Allen-Cahn equation

[At least a bit of the theory behind the model]

The problem statement in dimensionless formulation is given by equations for heat $T(t, \boldsymbol{x}) \in \mathbb{R}$ and phase $\Phi(t, \boldsymbol{x}) \in \mathbb{R}$ on some open region $\Omega \in \mathbb{R}^n$ and time interval $\mathcal{T} = [0, 1]$ as

$$\frac{\partial T}{\partial t} = \nabla^2 T + L\frac{\partial \Phi}{\partial t} \qquad \text{on } \Omega \times \mathcal{T} \tag{1.1}$$

$$\xi^2\alpha\frac{\partial \Phi}{\partial t} = \xi^2\nabla^2\Phi + f(\Phi, T, \xi) \qquad \text{on } \Omega \times \mathcal{T} \tag{1.2}$$

$$T|_{t=0} = T_{ini} \qquad \text{on } \Omega \tag{1.3}$$

$$\Phi|_{t=0} = \Phi_{ini} \qquad \text{on } \Omega \tag{1.4}$$

$$T|_{\partial\Omega} = T_{\partial\Omega} \text{ or } \nabla T\boldsymbol{n} = 0 \qquad \text{on } \partial\Omega \times \mathcal{T} \tag{1.5}$$

$$\Phi|_{\partial\Omega} = \Phi_{\partial\Omega} \text{ or } \nabla\Phi\boldsymbol{n} = 0 \qquad \text{on } \partial\Omega \times \mathcal{T} \tag{1.6}$$

where $f(\Phi, T, \xi)$ is the reaction term. There are multiple choices for the form of reaction term with different physical and numerical properties. We will use

$$f(\Phi, T, \xi) = af_0(\Phi) - b\xi^2\beta|\nabla\Phi|(T - T^m)$$
$$f_0(\Phi) = \Phi(1 - \Phi)(\Phi - \frac{1}{2})$$

in the rest of the text.

1.3.1 Anisotropy

The equations above are not sufficient to generate the complex crystal shapes found in real materials. One common characteristic of such materials is anisotropy. There have been a few [SUCH AS?] proposed models incorporating anisotropy into the Allen-Cahn equation. Because this work only considers a 2D case we choose the simple model

$$\xi^2\alpha\frac{\partial \Phi}{\partial t} = g(\theta)[\xi^2\nabla^2\Phi + af_0(\Phi)] - b\xi^2\beta|\nabla\Phi|(T - T^m) \tag{1.7}$$

$$g(\theta) := 1 - S\,cos(m(\theta_0 - \theta)) \tag{1.8}$$

where $\theta \in \mathbb{R}$ denotes angle between the direction of crystal interface movement and axis x, $S \in [0, 1)$ is the anisotropy strength, $m \in \mathbb{N}$ the frequency of anisotropy [???] and $\theta_0 \in \mathbb{R}$ the orientation of anisotropy [Cite Benes]. Note that θ can be computed as

$$\theta = \arccos\left(\frac{(\nabla\Phi)_x}{||(\nabla\Phi)_x||}\right) \tag{1.9}$$

We can rewrite the equation for Φ to simplify the subsequent algebraic transformations as:

$$\frac{\partial \Phi}{\partial t} = \underbrace{\frac{g(\theta)}{\alpha}}_{k_1(\nabla\Phi)}\nabla^2\Phi + \underbrace{\frac{g(\theta)af_0(\Phi)}{\xi^2\alpha}}_{k_0(\nabla\Phi)} + \underbrace{\frac{b\beta|\nabla\Phi|}{\alpha}}_{k_2(\nabla\Phi)}(T^m - T) \tag{1.10}$$

$$\frac{\partial \Phi}{\partial t} = k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T) \tag{1.11}$$

Note that $k1, k2, k3$ are all non linear.

Chapter 2

Solid Fluid interaction models

Chapter 3

Finite volume method

3.1 General mesh

[Mathematical definition of the discretized region, mesh, cells, admissibility and such]

3.2 Regular grid

From now on we will consider only the region $\Omega = (0, L_{0x}) \times (0, L_{0y}) \in \mathbb{R}^2$. We split this region into $n_x, n_y \in \mathbb{N}$ same sized rectangular cells C in the x and y directions respectively and define $\Delta x := L_{0x}/n_x$, $\Delta y := L_{0y}/n_y$.

In order not to repeat ourselves in the following sections, we will precalculate approximations of some integral forms over a single cell $K \in C$ for a general function $\Psi : \Omega \to \mathbb{R}$.

3.2.1 Laplacian

$$\int_K \nabla^2\Psi \, dx = \int_{\partial K} \nabla\Psi \cdot \boldsymbol{n} \, ds = \tag{3.1}$$

$$= \sum_{\sigma\in\partial K} \int_\sigma \nabla\Psi \cdot \boldsymbol{n_{k,\sigma}} \, ds \approx \tag{3.2}$$

$$\approx \sum_{\sigma\in\partial K} m(\sigma)\nabla\Psi_\sigma \cdot \boldsymbol{n_{k,\sigma}} \approx \tag{3.3}$$

$$\approx \sum_{\sigma\in\partial K,\sigma=K|L} \frac{m(\sigma)}{\rho(x_K, x_L)}(\Psi_L - \Psi_K) \tag{3.4}$$

Because of the regular uniform grid C $\forall\sigma = K|L$ holds $m(\sigma) = \Delta x$, $\rho(x_K, x_L) = \Delta y$ if cells K and L are adjacent in x or $m(\sigma) = \Delta y$, $\rho(x_K, x_L) = \Delta x$ if are adjacency in y. Further $\forall K \in C : m(K) = \Delta x \Delta y$. Thus by expanding the sum above we obtain

$$\int_K \nabla^2\Psi \, dx = \frac{\Delta y}{\Delta x}(\Psi_E - \Psi_K) - \frac{\Delta y}{\Delta x}(\Psi_W - \Psi_K) + \frac{\Delta x}{\Delta y}(\Psi_N - \Psi_K) - \frac{\Delta x}{\Delta y}(\Psi_S - \Psi_K) \tag{3.5}$$

$$= \frac{\Delta y}{\Delta x}(\Psi_E + \Psi_W - 2\Psi_K) + \frac{\Delta x}{\Delta y}(\Psi_N + \Psi_S - 2\Psi_K) \tag{3.6}$$

Note that by approximating this integral using the value at the cell centroid

$$(\nabla^2\Psi)_K \approx \frac{1}{m(K)}\int_K \nabla^2\Psi\, dx \approx \frac{\Psi_E + \Psi_W - 2\Psi_K}{\Delta x^2} + \frac{\Psi_N + \Psi_S - 2\Psi_K}{\Delta y^2} \tag{3.7}$$

we obtain the second order central difference approximation.

3.2.2 Gradient

$$\int_K \nabla\Psi\, dx = \int_{\partial K} \Psi \cdot \boldsymbol{n}\, dS = \sum_{\sigma\in\epsilon_K}\int_\sigma \Psi\cdot\boldsymbol{n}\, dS \tag{3.8}$$

$$\approx \sum_{\sigma\in\epsilon_K} m(\sigma)\Psi_\sigma\cdot\boldsymbol{n_{k,\sigma}} \tag{3.9}$$

$$= \Delta y\Psi_{\sigma E}\begin{pmatrix}1\\0\end{pmatrix} + \Delta y\Psi_{\sigma W}\begin{pmatrix}-1\\0\end{pmatrix} + \Delta x\Psi_{\sigma N}\begin{pmatrix}0\\1\end{pmatrix} + \Delta x\Psi_{\sigma S}\begin{pmatrix}0\\-1\end{pmatrix} \tag{3.10}$$

$$= \begin{pmatrix}\Delta y(\Psi_{\sigma E} - \Psi_{\sigma W})\\ \Delta x(\Psi_{\sigma N} - \Psi_{\sigma S})\end{pmatrix} \tag{3.11}$$

We use second order approximation of Ψ the cell boundary

$$\forall\sigma = K|L : \Psi_\sigma \approx \frac{\rho(x_K,\sigma)\Psi_L + \rho(x_L,\sigma)\Psi_K}{\rho(x_K,x_L)} = \frac{1}{2}(\Psi_L + \Psi_K)$$

where the mesh C properties were used to write $\rho(x_K,\sigma) = \frac{1}{2}\rho(x_K,x_L)$.
Thus we obtain the final approximation

$$\int_K \nabla\Psi\, dx \approx \frac{1}{2}\begin{pmatrix}\Delta y(\Psi_E - \Psi_W)\\ \Delta x(\Psi_N - \Psi_S)\end{pmatrix} \tag{3.12}$$

and the cell centroid approximation yields

$$(\nabla\Psi)_K \approx \frac{1}{m(K)}\int_K \nabla\Psi\, dx \approx \begin{pmatrix}\frac{\Psi_E - \Psi_W}{2\Delta x}\\ \frac{\Psi_N - \Psi_S}{2\Delta y}\end{pmatrix} \tag{3.13}$$

which is again the second [Is it?] order central difference approximation.

Chapter 4

Numerical solvers

4.1 Explicit solver

We attempt to numerically integrate the Allen-Cahn equation in time given by

$$\frac{\partial T}{\partial t} = \nabla^2 T + L\frac{\partial \Phi}{\partial t} \tag{4.1}$$

$$\frac{\partial \Phi}{\partial t} = k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T) \tag{4.2}$$

By integrating the Allen-Cahn equations over a cell $K \in C$ and using the centroid approximation for the unknown $\frac{\partial \Phi}{\partial t}$ and all nonlinear terms

$$(\frac{\partial T}{\partial t})_K \approx \frac{1}{m(K)}\int_K \nabla^2 T + L\frac{\partial \Phi}{\partial t}\, dx \tag{4.3}$$

$$\approx (\nabla^2 T)_K + L(\frac{\partial \Phi}{\partial t})_K \tag{4.4}$$

$$(\frac{\partial \Phi}{\partial t})_K \approx \frac{1}{m(K)}\int_K k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T)\, dx \tag{4.5}$$

$$\approx k_1((\nabla\Phi)_K)(\nabla^2\Phi)_K + k_0(\Phi_K, (\nabla\Phi)_K) + k_2((\nabla\Phi)_K)(T^m - T_K) \tag{4.6}$$

where we use the approximations derived earlier to calculate $(\nabla^2 T)_K, (\nabla^2\Phi)_K, (\nabla\Phi)_K$.

$$(\nabla^2 T)_K \approx \frac{T_E + T_W - 2T_K}{\Delta x^2} + \frac{T_N + T_S - 2T_K}{\Delta y^2} \tag{4.7}$$

$$(\nabla^2\Phi)_K \approx \frac{\Phi_E + \Phi_W - 2\Phi_K}{\Delta x^2} + \frac{\Phi_N + \Phi_S - 2\Phi_K}{\Delta y^2} \tag{4.8}$$

$$(\nabla\Phi)_K \approx \begin{pmatrix} \frac{\Phi_E - \Phi_W}{2\Delta x} \\ \frac{\Phi_N - \Phi_S}{2\Delta y} \end{pmatrix} \tag{4.9}$$

For time discretization we evaluated Explicit Euler method, fourth order Runge-Kutta method and fourth order Runge-Kutta with adaptive timestep.

For illustration we include example of the solver procedure going from time step n to $n+1$ using the explicit Euler method defined as

$$\frac{\partial \Phi^n}{\partial t} = \frac{\Phi^{n+1} - \Phi^n}{\Delta t} \tag{4.10}$$

where $\Phi^n := \Phi(n\Delta t)$.
Thus the algorhitm is as follows:

1. calculate $\frac{\partial \Phi^n}{\partial t}$ using T^n, Φ^n
2. calculate $\frac{\partial T^n}{\partial t}$ using $T^n, \frac{\partial \Phi^n}{\partial t}$
3. $\Phi^{n+1} = \Phi^n + \Delta t \frac{\partial \Phi^n}{\partial t}$
4. $T^{n+1} = T^n + \Delta t \frac{\partial T^n}{\partial t}$

4.2 Semi-implicit solver

4.2.1 Coupled solver

For simplicity of notation in this subsection without stated otherwise we will only consider values at the cell centroids so we associate $\Phi := \Phi_K$, $\nabla\Phi := (\nabla\Phi)_K$, $\nabla^2\Phi := (\nabla^2\Phi)_K$ for some cell K and similarly for T.

We begin with the approximated form of the Allen-Cahn equations, as derived above

$$\frac{\partial T}{\partial t} \approx \nabla^2 T + L\frac{\partial \Phi}{\partial t} \tag{4.11}$$

$$\frac{\partial \Phi}{\partial t} \approx k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T) \tag{4.12}$$

and attempt to formulate them into a matrix equation $A\boldsymbol{X} = \boldsymbol{b}$ where $A \in \mathbb{R}^{2N\times 2N}$, $\boldsymbol{X}, \boldsymbol{b} \in \mathbb{R}^{2N}$ by applying the generalized Crank-Nicolson time discretization given by

$$\frac{\Psi^{n+1} - \Psi^n}{\Delta t} = \gamma\frac{\partial \Psi}{\partial t}^{n+1} + (1-\gamma)\frac{\partial \Psi}{\partial t}^{n} \tag{4.13}$$

for some general function $\Psi : \mathbb{R} \to \mathbb{R}$ and $\gamma \in [0, 1]$. By setting $\gamma = 1$ we obtain Backwards Euler method, by setting $\gamma = 1/2$ we obtain the Crank-Nicolson method and by setting $\gamma = 0$ the Forward Euler method. Thus we obtain:

$$\frac{T^{n+1} - T^n}{\Delta t} \approx \gamma(\nabla^2 T^{n+1} + L\frac{\partial \Phi^{n+1}}{\partial t}) + (1-\gamma)(\nabla^2 T^n + L\frac{\partial \Phi^n}{\partial t}) \tag{4.14}$$

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} \approx \gamma(k_1(\nabla\Phi^{n+1})\nabla^2\Phi^{n+1} + k_0(\Phi^{n+1}, \nabla\Phi^{n+1}) + k_2(\nabla\Phi^{n+1})(T^m - T^{n+1})) \tag{4.15}$$

$$+ (1-\gamma)(k_1(\nabla\Phi^n)\nabla^2\Phi^n + k_0(\Phi^n, \nabla\Phi^n) + k_2(\nabla\Phi^n)(T^m - T^n)) \tag{4.16}$$

Since k_0, k_1, k_2 are non-linear we cannot use them to calculate the future value Φ^{n+1}, thus we approximate them by using the current value Φ^n. Because of this we drop argument notation and associate $k_0 := k_0(\Phi^n, \nabla\Phi^n)$, $k_1 := k_1(\nabla\Phi^n)$, $k_2 := k_2(\nabla\Phi^n)$. Thus

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} \approx \gamma(k_1\nabla^2\Phi^{n+1} + k_0 + k_2(T^m - T^{n+1})) \tag{4.17}$$

$$+ (1-\gamma)(k_1\nabla^2\Phi^n + k_0 + k_2(T^m - T^n)) \tag{4.18}$$

We multiply the equation by Δx and isolate all terms containing T^{n+1}, Φ^{n+1}

$$T^{n+1} - \Delta t\gamma(\nabla^2 T^{n+1} + L\frac{\partial \Phi^{n+1}}{\partial t}) \approx T^n + \Delta t(1-\gamma)(\nabla^2 T^n + L\frac{\partial \Phi^n}{\partial t}) \tag{4.19}$$

$$\Phi^{n+1} - \Delta t\gamma(k_1\nabla^2\Phi^{n+1} - k_2 T^{n+1}) \approx \Phi^n + \Delta t(1-\gamma)(k_1\nabla^2\Phi^n - k_2 T^n) + \Delta t(k_0 + k_2 T^m) \tag{4.20}$$

Substituting the T equation for $\frac{\partial \Phi}{\partial t}^{n+1} = \frac{\Phi^{n+1}-\Phi^n}{\gamma \Delta t} - \frac{1-\gamma}{\gamma}\frac{\partial \Phi}{\partial t}^n$ taken from the Crank-Nicolson discretization and shuffling terms, yields

$$T^{n+1} - \Delta t\gamma \nabla^2 T^{n+1} - L\Phi^{n+1} \approx T^n - L\Phi^n + \Delta t(1-\gamma)\nabla^2 T^n \tag{4.21}$$

Now we transition from a notation in terms of general cell K using its indeces $(i, j) \in \hat{n_y} \times \hat{n_x}$. We define general vector $\boldsymbol{P} \in \mathbb{R}^{n_x \times n_y}$ such that $\forall j \in \hat{n_x}, i \in \hat{n_y}$: $\boldsymbol{P}_{j+in_x} = \Psi(j\Delta x, i\Delta x) = \Psi_K$. Using the approximation for $\nabla^2\Psi$ and $\nabla\Psi$ derived earlier and realizing what cells E, W, N, S means in respect to cell at (i, j) allows us to defined discrete versions of the differential oeprators

$$\nabla_d^2 \boldsymbol{P}_{j+n_x i} := \frac{\boldsymbol{P}_{j+1+n_x i} + \boldsymbol{P}_{j-1+n_x i} - 2\boldsymbol{P}_{j+n_x i}}{\Delta x^2} + \frac{\boldsymbol{P}_{j+n_x(i-1)} + \boldsymbol{P}_{j+n_x(i+1)} - 2\boldsymbol{P}_{j+n_x i}}{\Delta y^2} \approx (\nabla^2\Psi)_K \tag{4.22}$$

$$\nabla_d \boldsymbol{P}_{j+n_x i} := \begin{pmatrix} \frac{P_{j+1+in_x} - P_{j-1+in_x}}{2\Delta x} \\ \frac{P_{j+(i+1)n_x} - P_{j+(i-1)n_x}}{2\Delta y} \end{pmatrix} \approx (\nabla\Psi)_K \tag{4.23}$$

We define $\boldsymbol{F}$ with respect to Φ and $\boldsymbol{U}$ in respect to T in the same way as $\boldsymbol{P}$ with respect to Ψ. Rewriting equations 4.24 and 4.20 int terms of $\boldsymbol{U}$ and $\boldsymbol{F}$ yields

$$\boldsymbol{U}^{n+1} - \Delta t\gamma \nabla_d^2 \boldsymbol{U}^{n+1} - L\boldsymbol{F}^{n+1} \approx \boldsymbol{U}^n - L\boldsymbol{F}^n + \Delta t(1-\gamma)\nabla_d^2 \boldsymbol{U}^n \tag{4.24}$$

$$\boldsymbol{F}^{n+1} - \Delta t\gamma(k_1 \nabla_d^2 \boldsymbol{F}^{n+1} - k_2 T^{n+1}) \approx \boldsymbol{F}^n + \Delta t(1-\gamma)(k_1 \nabla_d^2 \boldsymbol{F}^n - k_2 T^n) + \Delta t(k_0 + k_2 T^m) \tag{4.25}$$

which can be expressed as a system of linear equations in the form

$$\begin{pmatrix} A_F & B_F \\ B_U & A_U \end{pmatrix} \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{U} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_U \end{pmatrix} \tag{4.26}$$

with the values for $N := n_x n_y \in \mathbb{N}$, A_F, B_F, B_U, $A_U \in \mathbb{R}^{N \times N}$ and vectors of right hand side $\boldsymbol{b}_F$, $\boldsymbol{b}_U \in \mathbb{R}^N$. $\forall (k, l) \in \hat{N} \times \hat{N}$

$$(A_F)_{k,l} = \begin{cases} 1 + k_1(\frac{2\gamma\Delta t}{\Delta x^2} + \frac{2\Delta t}{\Delta y^2}) \text{ when } l = k, \\ -k_1 \frac{\gamma \Delta t}{\Delta x^2} \text{ when } l + 1 = k, \\ -k_1 \frac{\gamma \Delta t}{\Delta x^2} \text{ when } l - 1 = k, \\ -k_1 \frac{\gamma \Delta t}{\Delta y^2} \text{ when } l + n_x = k, \\ -k_1 \frac{\gamma \Delta t}{\Delta y^2} \text{ when } l - n_x = k, \\ 0 \text{ otherwise} \end{cases} \tag{4.27}$$

$$(B_F)_{k,l} = \begin{cases} -\gamma \Delta t k_2 \text{ when } l = k, \\ 0 \text{ otherwise} \end{cases} \tag{4.28}$$

$$(A_U)_{k,l} = \begin{cases} 1 + \frac{2\gamma\Delta t}{\Delta x^2} + \frac{2\Delta t}{\Delta y^2} \text{ when } l = k, \\ -\frac{\gamma \Delta t}{\Delta x^2} \text{ when } l + 1 = k, \\ -\frac{\gamma \Delta t}{\Delta x^2} \text{ when } l - 1 = k, \\ -\frac{\gamma \Delta t}{\Delta y^2} \text{ when } l + n_x = k, \\ -\frac{\gamma \Delta t}{\Delta y^2} \text{ when } l - n_x = k, \\ 0 \text{ otherwise} \end{cases} \tag{4.29}$$

$$(B_U)_{k,l} = \begin{cases} -L \text{ when } l = k, \\ 0 \text{ otherwise} \end{cases} \tag{4.30}$$

Note that we used k_1 dependent on the index k to mean $k_1(\nabla_d F_k)$ where ∇F_k is given by 4.23. Similarly $k_2 = k_2(\nabla_d F_k)$.

$$(\boldsymbol{b}_F)_k = \boldsymbol{F}_k^n + \Delta t(1-\gamma)(k_1 \nabla_d^2 \boldsymbol{F}_k^n - k_2 \boldsymbol{U}_k^n) + \Delta t(k_0 + k_2 T^m) \tag{4.31}$$

$$(\boldsymbol{b}_U)_k = \boldsymbol{U}_k^n - L\boldsymbol{F}_k^n + \Delta t(1-\gamma)\nabla_d^2 \boldsymbol{U}_k^n \tag{4.32}$$

[Schema of the block matrix]

Such system of linear equations can be solved by an appropriate matrix solver such as the Conjugate Gradient method. However in practice, with physically accurate choice of simulation parameters this matrix is very numerically unstable. To observe this notice that in the off diagonal matrix B_U the only non-zero values are $-L$ without dependence on the mesh resolution Δx or time step Δt. In our experiments we choose $L = 2$ making the matrix not diagonally dominant and causing the solver to diverge.

4.2.2 Split solver

The poor practical results of the Coupled solver motivates us to use Operator splitting [CITE!] to obtain an approximate solution. [Reformulate!]

We start with the equations 4.24 and 4.25 but approximate the $\boldsymbol{U}^{n+1}$ term with $\boldsymbol{U}^n$ in the $\boldsymbol{F}$ equation and move it to the right side. Thus a new equation is obtained for the right hand side

$$(\boldsymbol{b}_F)_k = \boldsymbol{F}_k^n + \Delta t(1-\gamma)k_1 \nabla_d^2 \boldsymbol{F}_k^n + \Delta t(k_0 + k_2(T^m - \boldsymbol{U}_k^n)) \qquad \forall k \in \hat{N} \tag{4.33}$$

Now $A_F^n \boldsymbol{F}^{n+1} = \boldsymbol{b}_F^n$ can be indpendently solved using only the current state at time $n\Delta t$ and yielding the new $\boldsymbol{F}^{n+1}$. Since $\boldsymbol{F}^{n+1}$ is know we can move it to the right side of the equation 4.24 giving us modified equation for its right hand side

$$(b_U)_k^{n,n+1} = \boldsymbol{U}_k^n - L(\boldsymbol{F}_k^{n+1} - \boldsymbol{F}_k^n) + \Delta t(1-\gamma)\boldsymbol{U}_k^n \qquad \forall k \in \hat{N} \tag{4.34}$$

Solving $A_U^n \boldsymbol{U}^{n+1} = \boldsymbol{U_F}^{n,n+1}$ yields the new T state $\boldsymbol{U}^{n+1}$.

4.3 Step residual

We can rewrite both the Explicit and semi-implicit numerical schemes in the form

$$\boldsymbol{F}^{(n+1)} = S_F(\boldsymbol{F}^{(n)}, \boldsymbol{U}^{(n)})$$
$$\boldsymbol{U}^{(n+1)} = S_U(\boldsymbol{F}^{(n+1)}, \boldsymbol{U}^{(n)})$$

where S_F, S_U are the solvers used to calculate the next iteration of $\boldsymbol{F}, \boldsymbol{U}$ respectively. For the explicit solver the definition of S_F, S_U can be written explicitly in terms of the individual cells. For the semi-implicit solver S_F, S_U are defined in terms of solution to $A_F \boldsymbol{F} = \boldsymbol{b}_F$ and $A_U \boldsymbol{U} = \boldsymbol{b}_U$.

Notice that the calculation of $\boldsymbol{F}^{(n+1)}$ uses $\boldsymbol{U}^{(n)}$ while the calculation $\boldsymbol{U}^{(n+1)}$ uses $\boldsymbol{F}^{(n+1)}$. This asymmetry causes the approximated original equations

$$\partial_t \Phi^{(n)} = \frac{1}{\alpha}\nabla^2 \Phi^{(n)} + \frac{1}{\xi^2 \alpha} f(\Phi^{(n)}, T^{(n)}, \nabla\Phi^{(n)})$$
$$\partial_t T^{(n)} = \nabla^2 T^{(n)} + L \cdot \partial_t \Phi^{(n)}$$

to not be be satisfied for any n. In order for the approximated equations to be satisfied $\Phi^{(n)}$ needs to be calculated using values from the already updated $T^{(n)}$ and vice versa. In other words there is bi-directional dependency between the two equations, but the current solver only models one direction.

Thus we define n-th step residual $r_\Phi^{(n)}$ for both solvers as

$$r_\Phi^{(n)} := S_F(\boldsymbol{F}^{(n)}, \boldsymbol{U}^{(n)}) - S_F(\boldsymbol{F}^{(n)}, \boldsymbol{U}^{(n+1)})$$

r_Φ can be read as "*the value we got using the information from the last step minus the value we would have obtained if we already knew the the values from the current step*". Note that r_Φ quantifies only the internal consistency of the approximations. It makes no statement about the quality of the chosen approximations in relation to the exact solution of the original equations.

Using the solver functions we can rewrite r_Φ as

$$\begin{aligned} r_\Phi^{(n)} &= \boldsymbol{F}^{(n+1)} - S_F(\boldsymbol{F}^{(n)}, \boldsymbol{U}^{(n+1)}) \\ &= \boldsymbol{F}^{(n+1)} - S_F(\boldsymbol{F}^{(n)}, S_U(\boldsymbol{F}^{(n+1)}, \boldsymbol{U}^{(n)})) \end{aligned}$$

and generalize its definition for some current candidate $\mathcal{F}$ for the next step $\boldsymbol{F}^{(n+1)}$ as:

$$r_\Phi = \mathcal{F} - S_F(\boldsymbol{F}^{(n)}, S_U(\mathcal{F}, \boldsymbol{U}^{(n)})) \tag{4.35}$$

4.3.1 Correction loop

We can lower r_Φ, thus potentially increasing the accuracy of our model, by iterating the candidate $\mathcal{F}$ for the next step $\boldsymbol{F}^{(n+1)}$ repeatedly until the desired accuracy is reached. The following algorithm uses this to go from $\boldsymbol{F}^{(n)}, \boldsymbol{U}^{(n)}$ to $\boldsymbol{F}^{(n+1)}, \boldsymbol{U}^{(n+1)}$:

$\mathcal{F}^{(0)} \leftarrow \boldsymbol{F}^{(n)}$
$\mathcal{U}^{(0)} \leftarrow \boldsymbol{U}^{(n)}$
for $k = 1, \ldots, M$ **do**
  $\mathcal{F} \leftarrow S_F(\boldsymbol{F}^{(n)}, \mathcal{U}^{(k-1)})$
  $\mathcal{U} \leftarrow S_U(\mathcal{F}, \boldsymbol{U}^{(n)})$
  $r_\Phi \leftarrow \mathcal{F}^{(k-1)} - \mathcal{F}$
  if $\|r_\Phi\| < \varepsilon$ **then**
    $\mathcal{F}^{(k)} \leftarrow \mathcal{F}$
    $\mathcal{U}^{(k)} \leftarrow \mathcal{U}$
    break
  end if
  $\mathcal{F}^{(k)} \leftarrow (1-\alpha)\mathcal{F}^{(k-1)} + \alpha\mathcal{F}$
  $\mathcal{U}^{(k)} \leftarrow (1-\alpha)\mathcal{U}^{(k-1)} + \alpha\mathcal{U}$
end for
$\boldsymbol{F}^{(n+1)} \leftarrow \mathcal{F}^{(k)}$
$\boldsymbol{U}^{(n+1)} \leftarrow \mathcal{U}^{(k)}$

where $\varepsilon \in R$ is the target accuracy, $\alpha \in (0, 1]$ hyperparamter to force convergence and $M \in N$ the maximum number of correction iterations. Realize that the formula used to calculate r_Φ matches the definition 4.35 wtih $\mathcal{F}^{(k-1)}$ as the current candidate. This is because $\mathcal{F} = S_F(\boldsymbol{F}^{(n)}, S_U(\mathcal{F}^{(k-1)}, \boldsymbol{U}^{(n)}))$ (only true with $\alpha = 1$).

From our testing with $\alpha = 0.9$ this algorithm works and drives down the step residual arbitrarily low, however for each iteration we have to solve the equations anew. This means that for M iterations of the above algorithm the total runtime will be $M + 1$ times longer [whats the correct way to say this?].

[DOES IT?! This is an outdated result of a broken test! Test this] This algorithm does not work for the explicit solver, since it only optimizes the solution locally. As such when two cells with opposite signed r_Φ are located next to each other a situation can, and in practice does, where the r_Φ only further grows in the direction of the sign [REWORK THIS]. This quickly results in reaching extremely high cell values for Φ, T and breaking the simulation.

4.3.2 Correction term

We can drastically reduce the step residual r_Φ almost for no runtime overhead by making the the Φ equation "more" implicit. We can take advantage of the linearity of the the T equation and upon chosen time discretization easily factor out the the $T^{(n+1)}$. Substituting it back to the original Φ equation yields a new equation Φ that approximately takes into account the resulting T.

We start by considering the forward Euler time discretization of the T equation

$$\frac{T^{n+1} - T^n}{\Delta t} = \nabla^2 T^n + L\frac{\partial \Phi^n}{\partial t} \tag{4.36}$$

$$\tag{4.37}$$

And factor out $T^{(n+1)}$

$$T^{n+1} = T^n + \Delta t \nabla^2 T^n + \Delta t L \frac{\partial \Phi^n}{\partial t} \tag{4.38}$$

$$\tag{4.39}$$

Subsituting T^{n+1} for T to the space-discretized Φ equation 4.12 we obtain

$$\frac{\partial \Phi}{\partial t} \approx k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T - \Delta t \nabla^2 T^n - \Delta t L \frac{\partial \Phi^n}{\partial t}) \tag{4.40}$$

We dropped the T^n notation in favor of T since we consider the resulting equation a brand new object instead of a discretized version of the previous equation.

Rearranging so that we isolate $\frac{\partial \Phi}{\partial t}$ yields

$$\frac{\partial \Phi}{\partial t} + k_2(\nabla\Phi)\Delta t L \frac{\partial \Phi^n}{\partial t} \approx k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T - \Delta t \nabla^2 T^n) \tag{4.41}$$

$$\frac{\partial \Phi}{\partial t} \approx \frac{k_1(\nabla\Phi)\nabla^2\Phi + k_0(\Phi, \nabla\Phi) + k_2(\nabla\Phi)(T^m - T - \Delta t \nabla^2 T)}{1 + \Delta t L k_2(\nabla\Phi)} \tag{4.42}$$

By defining rescaled nonlinear functions

$$\tilde{k}_0(\Phi, \nabla\Phi) := \frac{k_0(\Phi, \nabla\Phi)}{1 + \Delta t L k_2(\nabla\Phi)} \qquad \tilde{k}_1(\nabla\Phi) := \frac{k_1(\nabla\Phi)}{1 + \Delta t L k_2(\nabla\Phi)} \qquad \tilde{k}_2(\nabla\Phi) := \frac{k_2(\nabla\Phi)}{1 + \Delta t L k_2(\nabla\Phi)} \tag{4.43}$$

We can rewrite the modified equation into a form very closely matching the original

$$\frac{\partial \Phi}{\partial t} \approx \tilde{k}_1(\nabla\Phi)\nabla^2\Phi + \tilde{k}_0(\Phi, \nabla\Phi) + \tilde{k}_2(\nabla\Phi)(T^m - T - \Delta t \nabla^2 T) \tag{4.44}$$

Combined with the space-discretized T equation 4.11 this can be directly solved with an explicit integrator such as the fourth order Runge-Kutta method. Its worth noting that all of the additional terms

in 4.42 were already being calculated in each time step, thus this "corrected" version has negligable added runtime cost.

To obtain a similar result for the semi implicit solver we perform the exact same steps as while deriving the split solver above. The resulting matrix A_F is exactly the same except k_1, k_2 and k_3 are replaced with their rescaled counterparts. Right hand side $\boldsymbol{b}_F$ is also rescaled and contains an additional term from the added $-\tilde{k}_2 \Delta t \nabla^2 T$ yielding

$$(\boldsymbol{b}_F)_k = \boldsymbol{F}_k^n + \Delta t(1-\gamma)k_1 \nabla_d^2 \boldsymbol{F}_k^n + \Delta t(k_0 + k_2(T^m - \boldsymbol{U}_k^n - \Delta t \nabla_d^2 \boldsymbol{U}_k^n)) \qquad \forall k \in \hat{N} \tag{4.45}$$

This additional term has to be calculated explicitly adding non insignificant runtime cost. However in practice its addition makes the employed matrix solver converge in fewer iterations thus making this additional term a net reduction in terms of runtime.

Chapter 5

Numerical methods

5.1 Adaptive RK4

5.2 Jacobi

5.3 Gauss-Seidel

5.4 Conjugate Gradient

Chapter 6

Implementation

6.1 Cuda architecture

6.2 Performance considerations

Chapter 7

Results

7.1 Parameter setting

7.2 Correction impact

7.3 Comparison with analytical solution

Compare with an exact solution give specific boundary and initial conditions Efficient numerical scheme for solving the Allen-Cahn equation

Conclusion

Text of the conclusion...

Bibliography

[1] S. Allen, J. W. Cahn: *A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening*. Acta Metall., 27:1084-1095, 1979.

[2] N. Provatas, K. Elder, *Phase-Field Methods in Materials Science and Engineering*, WILEY-VCH Verlag GmbH & Co. KGaA, 2010.

[3] Y. Wang et al., *Simulation of Microstructure Evolution in Mg Alloys by Phase-Field Methods: A Review*, Crystals (12), 2022, 1305.

[4] P. Strachota, A. Wodecki, M. Beneš, *Focusing the latent heat release in 3D phase field simulations of dendritic crystal growth*. Modelling Simul. Mater. Sci. Eng. 29 065009 (2021). ISSN: 0965-0393.

[5] I. Steinbach, *Phase-field models in materials science*, Modelling Simul. Mater. Sci. Eng. (17), 2009, 073001.

[6] T. Takaki, S. Sakane, R. Suziki, *High-performance GPU computing of phase-field lattice Boltzmann simulations for dendrite growth with natural convection*. IOP Conf. Series: Materials Science and Engineering 1281 (2023), 012056.

[7] F. Moukalled, L. Mangani, M. Darwish et al., *The finite volume methods in computational fluid dynamics*. Springer, 2016.

[8] J. Palán, *Modely fázového pole v materiálových vědách a jejich numerické řešení*. Diplomová práce, FJFI ČVUT v Praze, 2023.