Moving iso-contour method for solving partial differential equations

Aarne Pohjonen[1]

[1]Affiliation not available

July 10, 2023

Moving iso-contour method for solving partial differential equations

Aarne Pohjonen, Aarne.Pohjonen@Oulu.fi, Aarne.Pohjonen@iki.fi

July 7, 2023

1 Abstract

The numerical solution of partial differential equations is often performed on a numerical grid, where the grid points are used for estimating the partial derivatives. The grid can be fully static as in Eulerian type of solution method, or the grid points can move during the solution, which is the case in Lagrangian type of method. In the current article, a numerical solution method is presented, where the grid points are located on iso-contours of the two-dimensional field. The method calculates the local movement of the iso-contours according to an evolution equation described by the PDE, and the solution proceeds by moving the grid points towards the calculated direction. Additional stability is obtained by setting the grid points to move along the iso-contour line. To exemplify the application of the method, numerical examples are calculated for the two-dimensional diffusion equation.

2 Introduction

In the previous study [1], the movement of a phase interface was simulated with level-set type method using a physical science based model which takes into account the transformation strains and carbon partitioning and diffusion. In that study the connection to the Allen-Cahn equation [1] was made, which connected it's solution to the level-set approach. As a extension of this idea, the connection between a general partial differential equation (PDE) with first order time derivative and the level set formulation is investigated in the current study.

The current approach is closely connected with the level-set method [2], which is often applied for simulations involving phase boundary movement. Level set methods usually perform the solution of a partial differential equation (PDE) in a grid that is not based on the points contained on the isocontours. The level-set method can be used in any types of grids, even in completely Eulerian framework [3, 4], where the computational grid does not need to move.

Sometimes it is important to be able to adapt the mesh to allow for more precision at certain locations that require fine grids for accurate solution, and to allow to the grid become sparse at regions that do not require dense grid for the solution accuracy so that the computational time does not increase too much. For generating adaptive meshes, a class of methods, called deformation methods have been created based on the level-set concept. In the deformation methods, the grid points are moved according to a specific monitor function, which allows to refine the grid locally [5].

In the current study, the mesh points are contained in the iso-contour lines and the focus is on the movement of the grid points, where the function retains it's value. Thus the method calculates the movement of the iso-contour lines, and the points contained in these iso-contours are used as the computational mesh gridpoints. This has certain advantages: first, the the iso-contours and their movement can have actual physical or application specific meaning, and it is often the desired result of the simulation. The current study provides the equations relating the PDE, gradient of the field and the first order time deriative and it provideds the means to calculate the iso-contour movement directly. Secondly, as the iso-contour points are used as the grid-points, this method provides a good basis for generating adaptive meshes.

3 Theory

The method described in this article provides a numerical solution procedure to PDE described by Eq. (1)

$$\frac{\partial}{\partial t}u = L(u) \tag{1}$$

where t is time and $L(u)$ is general PDE operator. For example, in diffusion equation with constant diffusion coefficient D the operator is $L(u) = D(\partial_{xx}u + \partial_{yy}u)$.

The method is based on the fact that advection equation can be used to calculate the movement of isocontours and it contains first order time derivative, which can be equated with the left hand side of Eq. (1). The advection equation is $\partial_t u = -\vec{v} \cdot \nabla u$, which describes the movement of a field u in the direction $\vec{v}$. If the propagation velocity $\vec{v}$ changes as function of position, the shape of the field changes with time. Let us now choose the vector $\vec{v}$ so that it is directed in the negative direction of the gradient, $\vec{v} = -s\hat{n}$, where $\hat{n} = \nabla u/|\nabla u|$ is a unit vector that has the same direction as ∇u ($|\nabla u|$ is the length of the gradient vector). In this case $= \partial_t u = -\vec{v} \cdot \nabla u = s\hat{n} \cdot \nabla u = s\frac{\nabla u \cdot \nabla u}{|\nabla u|} = s|\nabla u|$, so that Eq. (2) is true.

$$\frac{\partial}{\partial t}u = s|\nabla u| \tag{2}$$

Eq. (2) is applied in the Level-Set (LS) method to describe shape evolution of diffuse interface, when the local propagation speed s is first determined based on physical principles. In the level set method, signed distance from the region interface is usually used as the propagating field, and re-calculation of this field can be required.

In the current article, we apply the Eq. (2) to describe the evolution of the iso-contours of a general field whose evolution equations can be calculated with an equation of the form given by Eq. (1). Iso-contour normal vector is parallel to the gradient. Since the local velocity $\vec{v} = s\hat{n}$ of the field u was chosen in the negative direction of the gradient of u, the Eq. (2) describes the local movement of the field iso-contour with speed s.

Now, comparing Eq. (1) and (2), it can be seen that since both equations contain the first order time derivative, they can be equated, which yields the propagation speed of the iso-contour to the negative direction of the gradient, Eq. (3), which is the direction of the iso-contour normal.

$$s = \frac{L(u)}{|\nabla u|} \quad \text{when } |\nabla u| \neq 0 \tag{3}$$

Once the local speed of an iso-contour point $\vec{p}(t) = (p_x, p_y)$ is calculated using Eq. (3), the time derivative of the isocontour point position is obtained using Eq. (4).

$$\frac{\partial \vec{p}}{\partial t} = \vec{v} = -s\hat{n} \tag{4}$$

Further, by noting that $\hat{n} = \nabla u / |\nabla u|$, Eq. (4) can be also written out as Eq. (5)

$$\frac{\partial \vec{p}}{\partial t} = -\frac{L(u)}{|\nabla u|}\hat{n} = -L(u)\frac{\nabla u}{|\nabla u|^2} \tag{5}$$

Now, the movement of the point $\vec{p}$ contained at the iso-contour can be calculated, using the time derivative $\partial_t \vec{p} = (\partial_t p_x, \partial_t p_y)$ which is obtained either from Eqs. (3) and (4), or directly from Eq. (5).

However, as described in the section 4, it appeared that the numerical solution was rather unstable if simple Euler forward time step was used. This is most likely because of the same issues as associated with the solution of the advection equation, which is notorious for its instability in the basic Euler forward time-stepping procedure. To overcome this issue, the time-stepping described by Eqs. (6) and (7) was applied, which is similar to the one used for obtaining stable solution for advection equation in [6], but here the average time derivative $\left\langle \frac{\partial \vec{p}}{\partial t} \right\rangle$ was used instead of $\left.\frac{\partial \vec{p}}{\partial t}\right|_{t=t_i}$. In general, the average time derivative could provide better solution also in the case if variable timesteps were used.

$$\vec{p}(t_i + \Delta t) = \vec{p}(t_i - \Delta t) + \left\langle \frac{\partial \vec{p}}{\partial t} \right\rangle 2\Delta t \tag{6}$$

where

$$\left\langle \frac{\partial \vec{p}}{\partial t} \right\rangle = \frac{1}{2}\left(\left.\frac{\partial \vec{p}}{\partial t}\right|_{t=t_i-\Delta t} + \left.\frac{\partial \vec{p}}{\partial t}\right|_{t=t_i} \right) \tag{7}$$

After moving the isocontour gridpoint in the direction of the isocontour normal (i.e. the direction of the gradient) using either Eqs. (3, 4, 6) or Eq.

(5, 6), the gridpoint can be additionally moven along the isocontour line. This is because the value of the function u does not change when the gridpoint is moved along the isocontour - when the gridpoint is moved to a new position along the isocontour, the new position will also belong to the isocontour. This fact was used to maintain the grid shape more regular, as shown in the section 4.

For maintaining better grid shape, the gridpoints were moved along the isocontours after the time evolution step, so that the grid would remain suitably regular. This was achieved by calculating the difference of the current x-coordinate of the gridpoint from it's initial x-coordinate $p_{x(i,j)}(t) - p_{x(i,j)}(0)$. If this value was negative, the gridpoint was moved towards the right hand side gridpoint $(i+1,j)$ which is on the same isocontour. On the other hand, if $p_{x(i,j)}(t) - p_{x(i,j)}(0) > 0$, the gridpoint was moved towards the left gridpoint $(i-1,j)$, located on the same isocontour. More precisely, the vector $\vec{b}$ descibing the movement of the gridpoint along the isocontour is stated Eq. (8).

$$
\begin{aligned}
&\vec{b} = \gamma \hat{h} \\
&\text{where} \\
&\hat{h} = \frac{\vec{p}_{(i+1,j)}(t) - \vec{p}_{(i,j)}(0)}{|\vec{p}_{(i+1,j)}(t) - \vec{p}_{(i,j)}(0)|} \text{ if } p_{x(i,j)}(t) - p_{x(i,j)}(0) < 0 \\
&\text{and} \\
&\hat{h} = \frac{\vec{p}_{(i-1,j)}(t) - \vec{p}_{(i,j)}(0)}{|\vec{p}_{(i-1,j)}(t) - \vec{p}_{(i,j)}(0)|} \text{ if } p_{x(i,j)}(t) - p_{x(i,j)}(0) > 0
\end{aligned}
\tag{8}
$$

where the adjustment parameter γ controls how much the gridpoint moves along the isocontour line during a timestep. The value $\gamma = 0.05$ was used in the current example.

4 Numerical experiments

To test the application of the method in practice, numerical experiments were conducted. To make sure that the method produces correct result, it is useful to compare the numerical solution obtained with the method to an analytical solution. For this purpose, the solution for the one dimensional diffusion equation was calculated for a case that has a simple analytical solution. By using the separation of variables, the usual Fourier solution for the following case can be found: The analytical solution for a diffusion equation $\partial_t u = \partial_{xx} u$ with Dirichlet boundary conditions $u(0) = 1$ and $u(2\pi) = 0$ and initial condition $u(x, 0) = 1 - x/(2\pi) + 0.15\sin(x)$ has the analytical solution $u(x, t) = 1 - x/(2\pi) + 0.15\exp(-t)\sin(x)$.

To apply the method described in the theory section for this one dimensional case, the range of values of the initial function were divided to 100 equally spaced iso-contours (0, 0.01, 0.02, 0,03, ...). The second order spatial derivative $\partial_{xx} u$

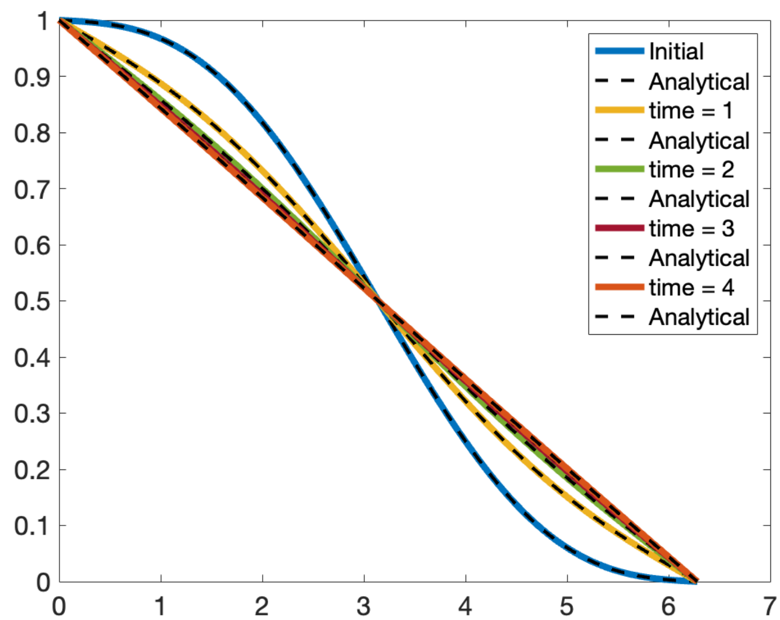

Figure 1: Comparison of the numerical solution to the analytical solution for $\partial_t u = \partial_{xx} u$ with Dirichlet boundary conditions $u(0) = 1$, $u(2\pi) = 0$ and initial condition $u(x, 0) = 1 - x/(2\pi) + 0.15\sin(x)$. The coloured solid lines represent the numerical solution at different times, and the dashed black line shows the corresponding analytical solution, which fully overlaps with the numerical solution for each time instant.

was calculated at each point by applying the region weighted average method [7]. In one dimensional case the regions are the lengths of the segments separating the isocontour points. The speed of the movement of the isocontour is calculated applying Eq. (3) (isocontour is a straigh line represented by a point in the 1 dimensional case). The direction of the movement $\hat{n} = \nabla u/|\nabla u|$ is the sign of the gradient, and the velocity of the isocontour is obtained from Eq. (5). For each timestep Δt, the new position of the isocontour is calculated as

$$\vec{p}(t_i + \Delta t) = \vec{p}(t_i) + \frac{\partial \vec{p}}{\partial t}\Delta t \tag{9}$$

The calculated solution compared to the analytical solution for the one dimensional diffusion equation is shown in Fig. 1, where the solid lines represent the numerical solution at different times, and the dashed black line shows the corresponding analytical solution. It can be seen that the method reproduces the analytical solution.

The isocontour method is particularly useful in a context, where the function is defined on one boundary region, and the points contained in this region move. Such condition was defined and the time evolution of the system was calculated. For the test case, the two dimensional diffusion equation with constant diffusivity $D = 1$ was picked, i.e. the PDE is described by Eq. (10). The method [7], which is capable of handling irregular grids, was used to calculate the first and second order partial derivatives for the two dimensional case.

$$\frac{\partial}{\partial t} u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{10}$$

The calculation domain was defined as a rectangular grid with grid point spacing in the x and y directions $\Delta x = 0.02 = \Delta y$ and the grid $x_{i,j}$ points were numbered with indices i and j in the horizontal x and vertical y direction. The initial condition of the system was defined as $u(x, y) = 1 - y$, as shown in Fig. 2 a). This is the steady state condition for Eq. (10) for the boundary values described by the function at their initial position. The values of the function on the upper, left and right boundaries were not allowed to change during the simulation, and the positions of the points at these boundaries were held fixed. On the bottom boundary, where initially $y = \Delta y$, the function values were defined as $u = \Delta y$ for all t, but the positions of the gridpoints of this iso-contour were changed during the calculation according to Eq. (11).

$$\begin{aligned} y(x) &= a(t)\mathrm{exp}\left(\frac{-(x-x_0)^2}{b}\right) + \Delta y \\ a(t) &= \frac{t}{2 \cdot 10^{-2}} 0.2 \text{ when } 0 \leq t < 2 \cdot 10^{-2} \\ a(t) &= 0.2 \text{ when } t \geq 2 \cdot 10^{-2} \end{aligned} \tag{11}$$

where $x_0 = 0.48$ and $b = 0.1$. This means that the points contained on the isocontour $u = 0$ were moving in the y-direction, so that the y-coordinates as function of time were described by the gaussian function, Eq. (11). The x-coordinates of the boundary gridpoints retained their original value for all t.

As a first instance, the movement of the the internal gridpoints were calculated using Eqs. (3, 4, 6) and they were not moved along the isocontours. It was found that when the boundary changed significantly, the grid became deformed, as shown in Fig. 2.

The approach of moving the gridpoints along the isocontour lines, descried in section 3 helped in maintaining a suitable grid shape, as shown in Fig. 3.

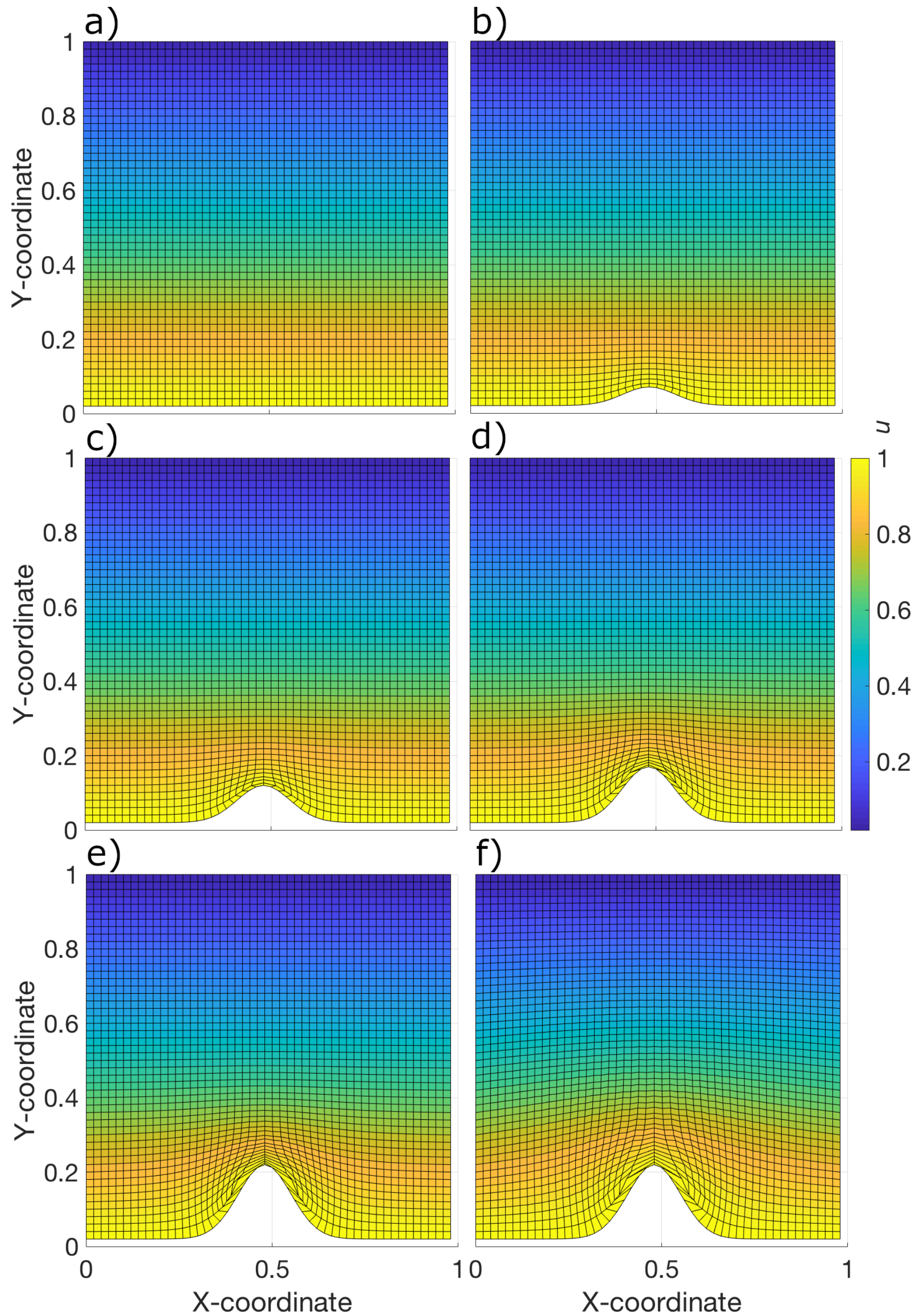

Figure 2: Simulation results where the gridpoints were not moved along the isocontour lines. a) $t = 5 \cdot 10^{-3}$, b) $t = 10 \cdot 10^{-3}$, c) $t = 15 \cdot 10^{-3}$, d) $t = 20 \cdot 10^{-3}$, e) $t = 25 \cdot 10^{-3}$, f) $t = 0.2$.

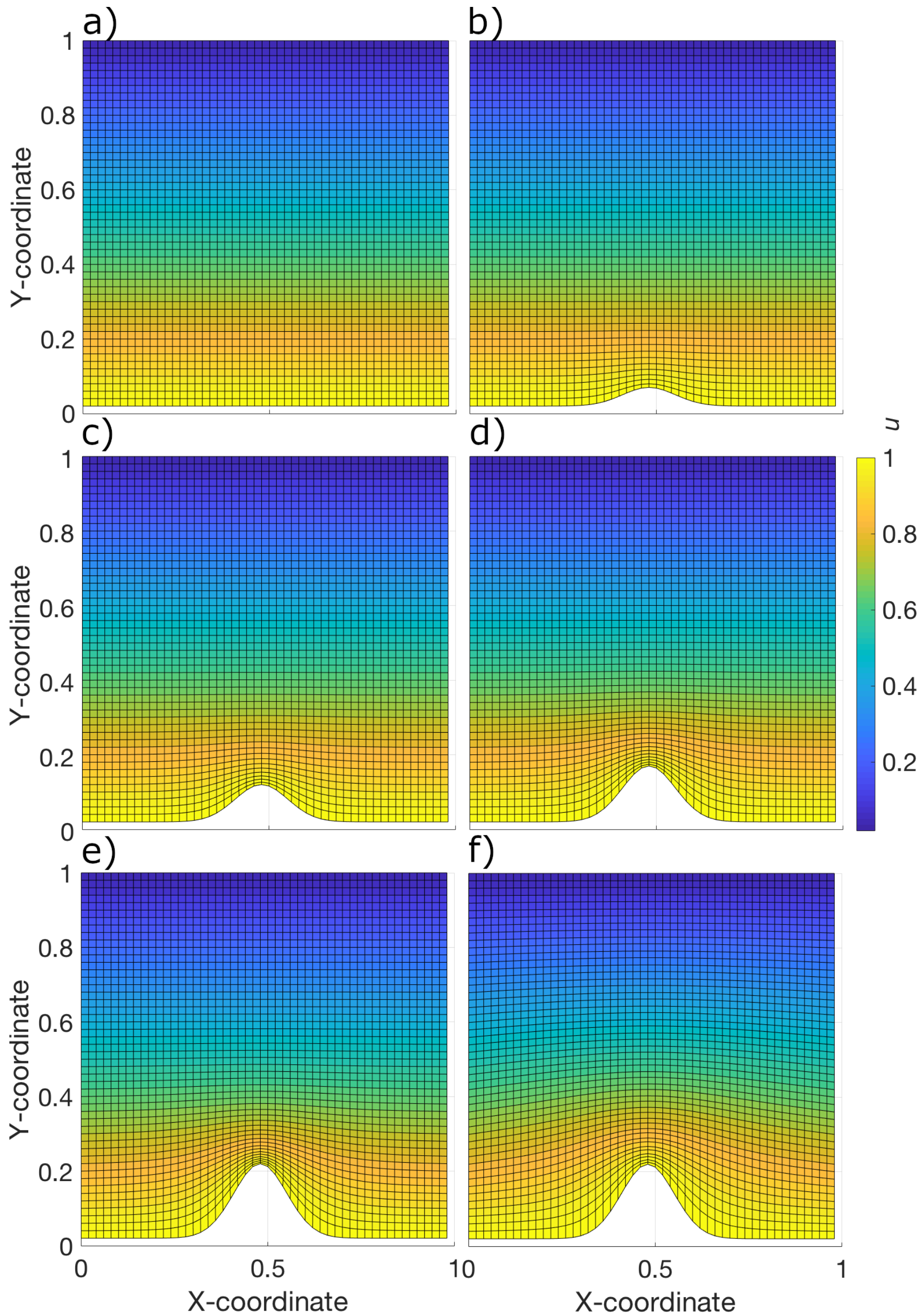

Figure 3: Simulation results where the gridpoints were moved along the isocontour lines to maintain a more regular grid shape. a) $t = 5 \cdot 10^{-3}$, b) $t = 10 \cdot 10^{-3}$, c) $t = 15 \cdot 10^{-3}$, d) $t = 20 \cdot 10^{-3}$, e) $t = 25 \cdot 10^{-3}$, f) $t = 0.2$.

When t increased, $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ approached zero, as the system approached steady state, since the boundary movement was stopped at time $t = 2 \cdot 10^{-2}$ as defined in Eq. (11).

5 Conclusions and outlook

A moving isocontour method was described and numerical experiments were performed. The method was compared to an analytic solution for one dimensional diffusion equation and it's applicability was demonstrated also for two dimensional diffusion equation. The method provides a way for constructing a moving mesh, based on the movement on the field iso-contours to calculate the time evolution of a PDE with first order time derivative. The method can also be used for the PDEs containing higher order time derivatives, if the first order time derivative can be calculated numerically. The method is useful for calculating time evolution of the iso-contours when the domain boundary is moving. In future the method could be applied in calculating the solution to a first order time PDEs for a sharp interface problems with moving boundary, such as diffusional growth of precipitates or phases in steels [8, 9].

References

References

[1] Aarne Pohjonen. Full field model describing phase front propagation, transformation strains, chemical partitioning, and diffusion in solid–solid phase transformations. *Advanced Theory and Simulations*, 6(3):2200771, 2023.

[2] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[3] Mark Sussman, Emad Fatemi, S Osher, and P Smereka. A level set approach for computing solutions to incompressible two-phase flow ii. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 1995.

[4] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[5] Guojun Liao, Feng Liu, C Gary, Danping Peng, and Stanley Osher. Level-set-based deformation methods for adaptive grids. *Journal of Computational Physics*, 159(1):103–122, 2000.

[6] Neil Gershenfeld. *The Nature of Mathematical Modeling.* Cambridge University Press, 2011.

[7] Aarne Pohjonen. Solving partial differential equations in deformed grids by estimating local average gradients with planes. *Journal of Physics: Conference Series*, 2090(1):012069, nov 2021.

[8] Aarne Pohjonen, Shashank Ramesh Babu, and Ville-Valtteri Visuri. Coupled model for carbon partitioning, diffusion, cottrell atmosphere formation and cementite precipitation in martensite during quenching. *Computational Materials Science*, 209:111413, 2022.

[9] O. Seppälä, A. Pohjonen, J. Mendonça, V. Javaheri, R. Podor, H. Singh, and J. Larkiola. In-situ sem characterization and numerical modelling of bainite formation and impingement of a medium- carbon, low-alloy steel. *Materials & Design*, 230:111956, 2023.