Solving partial differential equations in deformed grids by estimating local average gradients with planes

Aarne Pohjonen
Materials and Mechanical Engineering Faculty of Technology University of Oulu PL4200, 90014 Oulun Yliopisto Oulu, Finland
E-mail: Aarne.Pohjonen@Oulu.fi

Abstract. For constructing physical science based models in irregular numerical grids, an easy-to-implement method for solving partial differential equations has been developed and its accuracy has been evaluated by comparison to analytical solutions that are available for simple initial and boundary conditions. The method is based on approximating the local average gradients of a field by fitting equation of plane to the field quantities at neighbouring grid positions and then calculating an estimate for the local average gradient from the plane equations. The results, comparison to analytical solutions, and accuracy are presented for 2-dimensional cases.

1. Introduction

Aim of the present study is to describe a numerical method for solving partial differential equations (PDE’s) in deformed grids, which is relatively easy to implement. The method is designed for simulating the physical phenomena which occur at a microstructural level during thermomechanical processing of materials, such as diffusional [1] and displacive phase transformations [2, 3, 4], elasticity and plasticity [5], recovery, recrystallization and grain growth [6, 7, 8], but it could be used also in other contexts where PDEs need to be solved in irregular numerical grids. The physical science based full field modelling of material microstructure is important for a quantitative understanding of the associated phenomena, and for linking the microstructure evolution to the resulting material properties. As an example, we recently found that there is clear need for such detailed simulations in order to clarify the details associated with mechanical and chemical stabilization of austenite during bainite formation [9]. In order to mathematically model the spatio-temporal evolution of microstructure during thermo-mechanical processing, the model needs to have capability to calculate the change in the quantities at material positions which are displaced from a regular grid due to elastic and plastic deformations. For incorporating the fundamental physical laws, described by partial differential equations, the model needs to solve them despite that the numerical grid can be disordered due to the movements of the material. As an important step towards physics based full field modelling the microstructure evolution during thermo-mechanical processing, a numerical method which is relatively easy to implement and which can be used for solving partial differential equations in the deformed grids has been developed and its accuracy has been assessed by comparison to known analytical solutions.
2. Theory and calculations

2.1. Theory

The process of calculating the average gradient of a function $u$ for deformed numerical grids is presented. The successive application of the method to calculate the gradients of the obtained functions, $\partial_x u$ and $\partial_y u$, yields $(\partial_{xx} u, \partial_{xy} u)$ and $(\partial_{xy} u, \partial_{yy} u)$, hence also the second order partial derivatives of the function are obtained for deformed grids. A correction for the second order derivatives near the maxima and minima of the function is introduced to further enhance the accuracy and stability of the solution near these points, which is beneficial especially for sparse grids.

2.1.1. Calculation of local average gradient

The computational method is based on approximating the local average gradient of a function by plane equation determined by the positions and function values at the neighbouring grid points. Since application of the plane equation does not require regular shaped grid, the method works even if the grid is deformed. To describe the method in 2-dimensional case, consider the grid geometries depicted in Figs. 1 and 2.

![Figure 1](image1.png)

**Figure 1.** In the undeformed numerical grid, the points neighbouring a given grid point $\vec{p}_o$ are distributed at equal distance from point $\vec{p}_o$ parallel to the coordinate unit vectors.

![Figure 2](image2.png)

**Figure 2.** In the deformed numerical grid, the points neighbouring a given grid point $\vec{p}_o$ are displaced from their undeformed position.

The first objective is to estimate the gradient of a function $u(x, y)$ at point $\vec{p}_o$. For this purpose, we first estimate the gradients in the regions $R_i$, $i = 1, 2, 3, 4$ (see Figs. 1 and 2), by applying the equation of plane, and then calculate the average gradient in the union of the regions as the weighted average of these gradients. The areas of the individual regions are used as the weights. Although the method is described for 2-dimensional case in space, it can be extended to 3-dimensional case by defining two additional points in the third dimension and fitting equation of a hyperplane.

Focusing on the 2-dimensional case, the equation of the plane is described by Eq. 1.

$$u(x, y) = u_0 + a(x - x_0) + b(y - y_0)$$
When the coordinates of the three corner points of a triangular region are \( \vec{p}_0 = (x_0, y_0) \), \( \vec{p}_1 = (x_1, y_1) \) and \( \vec{p}_2 = (x_2, y_2) \) and the corresponding function values at these points are \( u_0 \), \( u_1 \), \( u_2 \), the coefficients \( a \) and \( b \) in Eq. (1) can be evaluated using Eq. (2), which is obtained by solving the equation group \( u(\vec{p}_0) = u_0, u(\vec{p}_1) = u_1 \) and \( u(\vec{p}_2) = u_2 \).

\[
a = \frac{(u_1 - u_0)y_2 + (u_0 - u_2)y_1 + (u_2 - u_1)y_0}{(x_1 - x_0)y_2 + (x_0 - x_2)y_1 + (x_2 - x_1)y_0},
b = \frac{(u_1 - u_0)x_2 + (u_0 - u_2)x_1 + (u_2 - u_1)x_0}{(x_1 - x_0)y_2 + (x_0 - x_2)y_1 + (x_2 - x_1)y_0}.
\]  

(2)

The coefficients of the plane equation are used for estimating the average gradient \( \nabla u = (\partial_x u, \partial_y u) \) inside the triangle whose corners are the points \( \vec{p}_0, \vec{p}_1 \) and \( \vec{p}_2 \), as described by Eq. (3)

\[
\partial_x u = a, \partial_y u = b
\]

(3)

After the average gradients in the regions \( R_i \) are calculated using Eqs. (1), (2) and (3), the average of the gradient in the union of the regions is obtained as the weighted average. The weights \( A_i \) are the areas of the regions, and they are obtained by applying the cross product of the vectors originating from \( p_o \) and ending to the corners of the region. For example for the region \( R_1 \) the area is obtained as

\[
A_1 = \frac{(\vec{p}_w - \vec{p}_0) \times (\vec{p}_n - \vec{p}_0)}{2}
\]

(4)

Similarly \( A_2, A_3 \) and \( A_4 \) are obtained using the corner points of the respective regions \( R_2, R_3 \) and \( R_4 \). The gradient at \( p_0 \), \( (\partial_x u|_{p_0}, \partial_y u|_{p_0}) \) is estimated as the average gradient over the union of the regions, which is obtained by the weighted average as

\[
\partial_x u|_{p_0} = \frac{A_1 a_1 + A_2 a_2 + A_3 a_3 + A_4 a_4}{A_1 + A_2 + A_3 + A_4}, \quad \partial_y u|_{p_0} = \frac{A_1 b_1 + A_2 b_2 + A_3 b_3 + A_4 b_4}{A_1 + A_2 + A_3 + A_4}
\]

(5)

where \( A_i \) are the areas of the regions \( R_i \), \( a_i = \partial_x u|_{R_i} \) are the partial derivatives of \( u \) with respect to \( x \) in the regions \( R_i \), and \( b_i = \partial_y u|_{R_i} \) are the partial derivatives with respect to \( y \) in the regions. The partial derivatives are obtained from equations (2) and (3).

2.1.2. Calculation of second order partial derivatives Once the partial derivative functions, \( \partial_x u \) and \( \partial_y u \), are obtained at each grid point using the procedure described in the previous section 2.1.1, the second order partial derivatives can be obtained by repeating the same procedure to calculate the gradient of \( \partial_x u \) and \( \partial_y u \),

\[
\nabla \partial_x u = (\partial_{xx} u, \partial_{yx} u), \quad \nabla \partial_y u = (\partial_{xy} u, \partial_{yy} u)
\]

(6)

As described in the calculations and results section, this approach already converges to the analytical solution, when the numerical grid is refined. However, in the vicinity of the local maxima, oscillations around the analytical solution are present when the numerical grid is too sparse. Since such oscillations negatively affect the accuracy of the solution for sparse grids, can cause stability issues and lead to numerical artefacts when simulating real physical systems where the field values are changing dynamically, the second order derivatives were recalculated in the vicinity of the local maxima or minima to enhance the accuracy and stability of the result.
The recalculation of the second order differentials at the local maxima, and at the neighbouring grid points of the maxima and minima was performed as follows: on the neighbouring points located on the left and on the right to the maxima (i.e. when \( \vec{p}_o \) is neighbouring point on left or right from the maxima/minima), instead of calculating the gradient at point \( \vec{p}_o \) and \( \vec{p}_e \), the vector \( \vec{v} \) in Eq. (7) was used as the gradient at points \( \vec{p}_o + \frac{\vec{v} - \vec{p}_e}{2} \) and \( \vec{p}_o + \frac{\vec{v} - \vec{p}_e}{2} \). The x-component of this vector was used for calculating the second order differentials \( \partial_{xx} u \) and \( \partial_{xy} u \) at these points (instead of using the gradients calculated at points \( \vec{p}_o \) and \( \vec{p}_e \)). In the Eq. (7) the index \( i \in \{w, e\} \) for points located on the right and left side of the maxima/minima.

\[
\vec{v}_i = \frac{\vec{p}_i - \vec{p}_o}{|\vec{p}_i - \vec{p}_o|} u(\vec{p}_i) - u(\vec{p}_o)
\]

Similarly, on the neighbouring points located above and below of the maxima (i.e. \( \vec{p}_o \) is the neighbouring point above of below of the maxima), the vector in Eq. (7), where \( i \in \{n, s\} \), was used as the gradient at points \( \vec{p}_o + \frac{\vec{v}_x - \vec{p}_e}{2} \) and \( \vec{p}_o + \frac{\vec{v}_x - \vec{p}_e}{2} \). The y-component of this vector was used for calculating the second order differentials \( \partial_{yy} u \) and \( \partial_{xy} u \).

The application of the vector \( \vec{v}_i \) in the equation (7) means that, at the points next to the local maxima/minima, the gradient is assumed to be directed from or towards the neighbouring gridpoint and its magnitude is given by the difference of the function value at the gridpoints. The locations of the points \( \vec{p}_o + \frac{\vec{v}_x - \vec{p}_e}{2} \) and \( \vec{p}_o + \frac{\vec{v}_x - \vec{p}_e}{2} \) relative to a grid point \( \vec{p}_o \), which is on the left or right to the maxima/minima are shown in Fig. 3 as red filled circles. The locations of the points \( \vec{p}_o + \frac{\vec{v}_y - \vec{p}_e}{2} \) and \( \vec{p}_o + \frac{\vec{v}_y - \vec{p}_e}{2} \) are shown as green unfilled circles. When \( \vec{p}_o \) is the grid point located on the left or right side of the maxima/minima, the red points were used and when \( \vec{p}_o \) is located above or below of the maxima/minima, the green unfilled points were used.

At the grid point where the local maxima/minima appears in the numerical simulation, the gradient vector quantities were calculated using Eq. (7) at both the points denoted with red filled circles, as well as at the points denoted with green unfilled circles in Fig. 3 instead of using the previously calculated gradient at the points \( \vec{p}_n, \vec{p}_e, \vec{p}_s, \vec{p}_w \). These values were used when calculating the second order derivative to obtain the correction which helped to avoid the oscillations observed in the simulations with sparse grids.

2.2. Calculations

The numerical grid was constructed initially with square geometry before introducing the deformation of the grid to the simulations. The positions of the gridpoints were set at \((x, y) = -\frac{1}{2}(L_x, L_y) + i(\Delta X, 0) + j(0, \Delta Y)\), where the segments \( \Delta X = L_x/npoints \) and \( \Delta Y = L_y/npoints \) were obtained by dividing the lengths \( L_x \) and \( L_y \) of the simulation domain in \( x \) and \( y \) directions by the number of grid points in one direction, \( npoints \). The integers \( i \) and \( j \) range from 1 to the value \( npoints \). For square geometry \( L_x = L_y \) and \( \Delta X = \Delta Y \). The deformations of the grid were introduced by displacing the grid points both systematically and randomly. The systematic deformation was introduced by adding shear strain \( \epsilon_{xy} = \epsilon_{yx} = 0.3 \) to the coordinates, and the random displacement was introduced by adding a random value \( \pm 0.16 \Delta Y \) and \( \pm 0.16 \Delta X \) to the coordinates, i.e. by displacing the coordinates with 16% of the segment length. These displacements were realized by calculating the coordinates using Eq. (8), where \( r_x(i, j) \) and \( r_y(i, j) \) are random numbers between 0 and 1.

\[
(x(i, j), y(i, j)) = -\frac{1}{2}(L_x, L_y) + i(\Delta X, 0) + j(0, \Delta Y) + (j \epsilon_{xy} \Delta Y, i \epsilon_{yx} \Delta X) + 0.16([r_x - 0.5] \Delta X, [r_x - 0.5] \Delta Y)
\]
Figure 3. When calculating the second order partial derivatives for points $\vec{p}_n$ located on the left or right side of a local maxima or minima, the gradients were calculated using Eq. 7 at the red filled circles instead $\vec{p}_w$ and $\vec{p}_e$. Similarly for the grid points located above or below of the maxima, the gradients were calculated at the green open circles instead of $\vec{p}_n$ and $\vec{p}_s$. At the local maxima, the gradient was calculated in both the red filled circles and the green unfilled circles.

To check that the numerical solution converges to the correct solution as the numerical grid is refined, we compared the numerical solution of the time-dependent diffusion equation, Eq. (9), to the analytical solution, Eq. (10).[10]

$$\frac{\partial u}{\partial t} = D \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$  \hspace{1cm} (9)

Let us consider the initial condition where the concentration described by function $u$ is located at a singular point $(x_c, y_c)$ and elsewhere the concentration is zero. In detail, when time $t = 0$ the concentration $u(x_c, y_c) = M$ and $u(x, y) = 0$ when $(x, y) \neq (x_c, y_c)$. The analytical solution to Eq. (9) for this condition in the infinite plane $\mathbb{R}^2$ is described by Eq. (10). Values $D = 1$ and $M = 0.1$ were chosen for the calculations.

$$u(x, y, t) = \frac{M}{4\pi t D} \exp \left( -\frac{(x - x_c)^2 + (y - y_c)^2}{4Dt} \right)$$  \hspace{1cm} (10)

In the numerical simulations, the partial differentials $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$ in Eq. (9) were evaluated as described in the section 2.1. Their sum yields the value for the time derivative $\frac{\partial u}{\partial t}$ at each grid point. The value of $u$ after a time step $\Delta t$ was then calculated using the simple forward Euler approximation, Eq. (11), which turned out to be sufficient in the studied case.

$$u(x, y, t + \Delta t) = u(x, y, t) + \frac{\partial u}{\partial t} \Delta t$$  \hspace{1cm} (11)

For the purpose of checking the capability of the presented method to produce the correct solution in deformed grids when an initial condition was chosen, we applied the solution given
by Eq. (10), using the time $t = 4.0$ to define the initial simulation condition. This means that at the beginning of the simulation the singular concentration has diffused for the time period of 4.0. The corresponding initial value for the function $u$ was calculated for every grid point using Eq. (10). The length dimensions in simulation domain in $x$- and $y$- directions were $L_x = L_y = 162$. The values of the field for the initial condition as well as for the duration of the simulation were such that the value of $u$ near the boundaries was negligible for all presented simulation cases, so that the numerical solution corresponds to the analytical solution for the infinite plane. The simulation domain was divided into $npoints - 1$ segments in both $x$ and $y$ dimensions, where the number of grid points in one dimension, $npoints$, was varied. For all simulation cases the time step size $\Delta t = 0.25$ was chosen. In principle any initial time could be chosen instead of the value 4.0. The accuracy of the solution however depends on the number of grid points used for dividing the simulation domain, and the numerical stability requires smaller time step size for larger number of grid points (smaller length segment requires smaller step size for stability).

3. Results and Discussion
It was checked that the numerical solution converges to the result given by analytical solution when the numerical grid is refined. The comparison for the case where simulation time was 25.0 after the initial condition, using the 121 gridpoints in both spatial dimensions is shown in Fig. 4. The colors in Fig. 4 a) represent the values of the function $u$ between the gridpoints. The figure was created with Matlab surf routine. The black lines connect the simulation gridpoints. The contour plot of the same condition is shown in Fig. 4 b), where the contours obtained from the simulation results are plotted together with the contours of the analytical solution. It can be seen that the lines in the contour plots overlap almost everywhere, indicating that the numerical solution provides a good approximation for the simulated case for the deformed grid.

The line plot along the red horizontal line in Fig. 4 is shown in Figs. 5 and 6 for different simulation times, and compared to the analytical solution (markers). The simulation results where no correction procedure was applied are shown in Fig. 5, and the results where the correction procedure was applied are shown in Fig. 6. It can be seen that even when no correction was applied the simulation result converges to the analytical solution when the number of grid points is increased. However, there are severe oscillations affecting the accuracy of the results for sparse grids for the simulations where correction was not applied. As shown in Fig. 6, when the correction procedure was applied, the oscillations were removed and the simulation accuracy was improved. This can be seen to be beneficial especially for the sparse grids.
**Figure 4.** Simulation time 25.0, \( npoints = 121 \). a) The function values and the deformed 2-dimensional numerical grid. The color represents the function value between the gridpoints. b) Contour plot from the same result. The contours of the analytical solution (Eq. (10)) are plotted with same color in the figure together with the simulation result, overlapping almost everywhere. The line plot along the red line is shown in Figs. 5 d) and 6 d) (with blue line).
Figure 5. Comparison of numerical solutions (lines) to the analytical solution (points) for the uncorrected method (i.e. no correction at or near local maxima and minima), a) npoints = 41 b) npoints = 61 c) npoints = 81 d) npoints = 121.
Figure 6. Comparison of numerical solutions (lines) to the analytical solution (points) for the corrected method (i.e. correction procedure was applied at and near local maxima and minima), a) $n_{points} = 41$ b) $n_{points} = 61$ c) $n_{points} = 81$ d) $n_{points} = 121$. 
4. Conclusions

As shown in the Results and Discussion section 3, using the procedure described in the Theory section 2.1, it is possible to calculate the first and second order partial derivatives of the function for deformed numerical grids. Although the basic method without a correction already converges to the analytical solution when the numerical grid is refined, it was found that for sparse grids, there were oscillations in the solution near local maxima for a second order PDE (time dependent diffusion equation was used as an example case). These oscillations could be removed by introducing a correction procedure. The correction, which was introduced for second order derivatives for the grid points located next to the maxima/minima, removes the oscillations and improves the accuracy of the solution for sparse grids. The presented method provides a way to incorporate the mathematical description of the physical phenomena to numerical microstructure models via the solution of partial differential equations. The significance of the presented method for future work is that the method enables relatively easy implementation and the development of physics based material microstructure models where the numerical simulation grid points can be displaced from their initial regular positions.

References

[1] Irina Loginova. *Phase-field modeling of diffusion controlled phase transformations*. PhD thesis, Mekanik, 2003.

[2] Amer Malik, Gustav Amberg, Annika Borgenstam, and John Ågren. Effect of external loading on the martensitic transformation – a phase field study. *Acta Materialia*, 61(20):7868–7880, 2013.

[3] Oskari Seppälä, Aarne Pohjonen, and Jari Larkiola. Effect of anisotropic growth and grain boundary impingement on bainite transformation models. 2021.

[4] Oskari Seppälä, Aarne Pohjonen, Antti Kajalainen, Jari Larkiola, and David Porter. Simulation of bainite and martensite formation using a novel cellular automata method. *Procedia Manufacturing*, 15:1856–1863, 2018. Proceedings of the 17th International Conference on Metal Forming METAL FORMING 2018 September 16 – 19, 2018, Loisir Hotel Toyohashi, Toyohashi, Japan.

[5] Dierk Raabe and Richard C Becker. Coupling of a crystal plasticity finite-element model with a probabilistic cellular automaton for simulating primary static recrystallization in aluminium. *Modelling and Simulation in Materials Science and Engineering*, 8(4):445, 2000.

[6] Tomohiro Takaki, Akinori Yamanaka, and Yoshihiro Tomita. *Phase-Field Modeling for Dynamic Recrystallization*, pages 441–459. Springer International Publishing, Cham, 2015.

[7] Håkan Hallberg. Approaches to modeling of recrystallization. *Metals*, 1(1):16–48, 2011.

[8] Ludovic Maire, Benjamin Scholtes, Charbel Moussa, Nathalie Bozzolo, Daniel Pino Muñoz, Amico Settefrati, and Marc Bernacki. Modeling of dynamic and post-dynamic recrystallization by coupling a full field approach to phenomenological laws. *Materials & Design*, 133:498–519, 2017.

[9] Aarne Pohjonen, Pentti Kaikkonen, Oskari Seppälä, Joonas Imlola, Vahid Javaheri, Timo Manninen, and Mahesh Somani. Numerical and experimental study on thermo-mechanical processing of medium-carbon steels at low temperatures for achieving ultrafine-structured bainite. *Materialia*, 18:101150, 2021.

[10] Lecture Notes on Diffusion. [http://web.mit.edu/1.061/www/dream/THREE/THREETHEORY.PDF](http://web.mit.edu/1.061/www/dream/THREE/THREETHEORY.PDF). Accessed: 2021-08-13.