A Review of Level-Set Methods and Some Recent Applications

Frederic Gibou[a,b], Ronald Fedkiw[c], Stanley Osher[d]

[a] *Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106*
[b] *Department of Computer Science, University of California, Santa Barbara, CA 93106*
[c] *Department of Computer Science, Stanford University, CA 94305*
[d] *Department of Mathematics, University of California, Los Angeles, CA 90095*

Abstract

We review some of the recent advances in level-set methods and their applications. In particular, we discuss how to impose boundary conditions at irregular domains and free boundaries, as well as the extension of level-set methods to adaptive Cartesian grids and parallel architectures. Illustrative applications are taken from the physical and life sciences. Fast sweeping methods are briefly discussed.

Keywords: Level-Set Method, Ghost-Fluid Method, Voronoi Interface Method, Jump Condition, Robin Boundary Condition, Dirichlet Boundary Condition, Octrees, Adaptive Mesh Refinement, Parallel Computing.

1. Introduction

Representing and tracking the evolution of interfaces is a fundamental component of computer simulations. An efficient way to do so is to use the level-set method introduced by Osher and Sethian [1]. It consists in representing the interface as the level-set of a higher dimensional function. The main advantage of this implicit representation of a moving front is its ability to naturally handle changes in topology, as illustrated in figure 1. This is in contrast to explicit methods [2–6] for which changes in topology require extra work for detecting and subsequently treating numerically the merging or pinching of fronts. We note, however, that explicit methods have the advantage of accuracy (e.g. front-tracking preserve volumes better than level-set methods for the same grid resolution) and we refer the interested reader to the work of [7] for a front-tracking method that handle changes in topology. Volume of fluid methods also adopt an implicit formulation using the volume fraction of one phase in each computational cells (see e.g. [8–19] and the references therein). These methods have the advantage of conserving the total volume by construction. They are however more complicated than level-set methods in three spatial dimensions and it is difficult to compute accurately smooth geometric properties such as curvatures from the volume fraction alone, although we refer the reader to the interesting work of Popinet on this issue [20]. Also, we note that phase-field models have been extensively used to tackle free boundary problems, particularly in the case of solidification processes [21–23, 23–29]. However, these models do not represent the interface in a sharp fashion, which in turn leads to a degradation of the accuracy where it matters most and impose sometimes stringent time step restrictions. In what follows, we review the level-set method, including the treatment of boundary conditions in that framework and extensions to adaptive Cartesian grids and parallel architectures. The Fast Sweeping Method, which is often associated with the level-set method for its ability to compute the signed distance function and solutions to other Hamilton-Jacobi equations, is also briefly discussed. We then present some recent applications of the level-set and the fast sweeping methods.

*Corresponding author: fgibou@engineering.ucsb.edu

2. Level-Set Representation and Equations

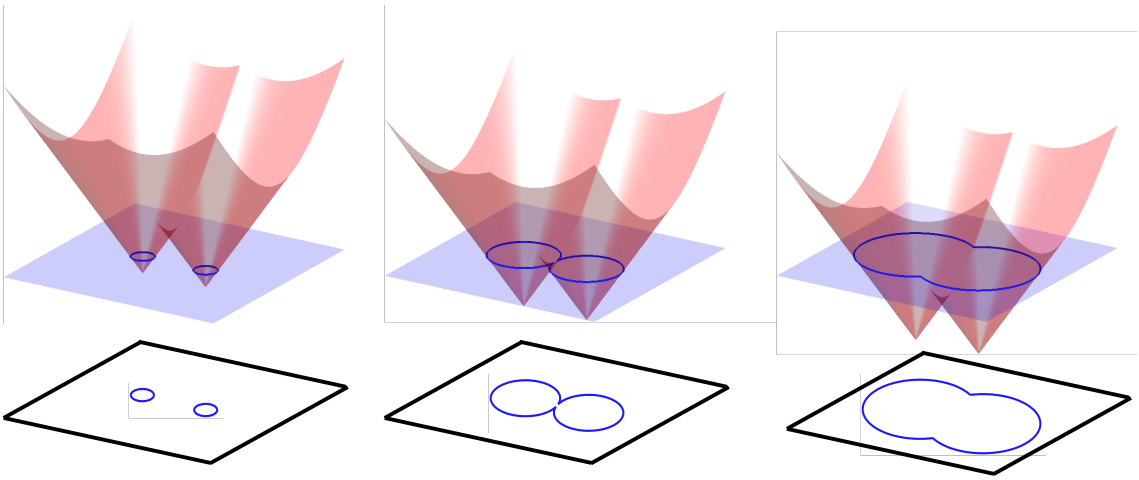


Figure 1: *Level-set representation of a free boundary (blue solid line) in two spatial dimensions, moving in its normal direction, and subsequent changes in topology that are handled automatically. The level-set function is depicted in red. (Color online).*

The level-set method of Osher and Sethian [1] represents an interface, Γ, (i.e. a curve in two spatial dimensions or a surface in three spatial dimensions) as the zero-contour of a higher dimensional function, ϕ, called the level-set function, which is defined as the signed distance function to Γ:

$$\phi(\boldsymbol{x}) = \begin{cases} -\ d & \text{for } \boldsymbol{x} \in \Omega^-, \\ +\ d & \text{for } \boldsymbol{x} \in \Omega^+, \\ \quad 0 & \text{for } \boldsymbol{x} \in \Gamma, \end{cases}$$

where d is the Euclidian distance to Γ. The level-set function can also be used to compute the normal to the interface $\boldsymbol{n}$ and the interface's mean curvature κ:

$$\boldsymbol{n} = \nabla\phi/|\nabla\phi| \qquad \text{and} \qquad \kappa = \nabla \cdot \boldsymbol{n}.$$

Under a velocity field $\boldsymbol{v} = (u, v, w)$, the interface deforms according to the level-set equation:

$$\frac{\partial \phi}{\partial t} + v_n|\nabla\phi| = 0, \qquad \text{where } v_n = \boldsymbol{v} \cdot \boldsymbol{n}. \tag{1}$$

Although the level function can be chosen to be any Lipschitz continuous function, in practice the signed distance function is chosen for its properties of improved mass conservation and accuracy, especially in the computations of geometrical quantities. Since in general the level-set function does not retain its signed-distance-function property as it evolves in time through equation (1), Sussman *et al.* [30] introduced the reinitialization equation:

$$\phi_\tau + \mathtt{sgn}(\phi^0)\,(|\nabla\phi| - 1) = 0, \tag{2}$$

to transform a level set function $\phi^0 : \mathbb{R}^n \to \mathbb{R}$ into the signed distance function ϕ. Here, $\mathtt{sgn}$ is a smoothed-out signum function and τ represents a fictitious time that controls the width of the band around the zero-level set where ϕ will be sign-distanced. Finally, if the level-set function is a signed distance function, the projection onto Γ of any given point $\boldsymbol{x}$ is easily computed as:

$$\boldsymbol{x}_\Gamma = \boldsymbol{x} - \phi(\boldsymbol{x})\nabla\phi(\boldsymbol{x}).$$

3. Approximation of Equations

On uniform grids, the level-set advection equation (1) and the reinitialization equation (2) are discretized with a HJ-WENO scheme in space [31–33] and a TVD-RK3 in time [34]. We gives the details of those schemes next.

3.1. WENO Schemes

The WENO schemes [31, 33] are based on the ENO schemes [35], both of which are used to compute the one-sided backward $D_x^-\phi$ and forward $D_x^+\phi$ first-order derivatives of a scalar function ϕ. The philosophy behind ENO is to choose between three different stencils depending on the upwind direction and on which one will avoid differentiating across discontinuities. In smooth regions, however, a weighted convex combination of those stencils produces higher-order accuracy (fourth-order accurate for conservation laws and fifth-order for Hamilton-Jacobi); this is the idea behind the Weighted ENO (WENO) schemes.

Let's consider the WENO approximation of $D_x^-\phi$; the construction of $D_x^+\phi$ is similar. The three possible ENO approximations of $D_x^-\phi$ are:

$$\phi_x^1 = \frac{d_1}{3} - \frac{7d_2}{6} + \frac{11d_3}{6}, \quad \phi_x^2 = -\frac{d_2}{6} + \frac{5d_3}{6} + \frac{d_4}{3}, \quad \phi_x^3 = \frac{d_3}{3} + \frac{5d_4}{6} - \frac{d_5}{6},$$

where the finite differences d_i are defined as:

$$d_1 = \frac{\phi_{i-2} - \phi_{i-3}}{\Delta x}, \quad d_2 = \frac{\phi_{i-1} - \phi_{i-2}}{\Delta x}, \quad d_3 = \frac{\phi_i - \phi_{i-1}}{\Delta x}, \quad d_4 = \frac{\phi_{i+1} - \phi_i}{\Delta x}, \quad d_5 = \frac{\phi_{i+2} - \phi_{i+1}}{\Delta x}.$$

The WENO approximation of $D_x^-\phi$ is given by:

$$D_x^-\phi = \omega_1\phi_x^1 + \omega_2\phi_x^2 + \omega_3\phi_x^3, \tag{3}$$

where the coefficients ω_k are chosen in such a way that the approximation of (3) is fifth-order accurate in smooth region, while retaining the ENO philosophy near discontinuities. This is achieved by estimating the smoothness of the solution via the smoothness of the stencils as follows:

- Define the smoothness coefficient S_i of each stencil ϕ_x^i:

$$\begin{aligned} S_1 &= \frac{13}{12}(d_1 - 2d_2 + d_3)^2 + \frac{1}{4}(d_1 - 4d_2 + 3d_3)^2, \\ S_2 &= \frac{13}{12}(d_2 - 2d_3 + d_4)^2 + \frac{1}{4}(d_2 - d_4)^2, \\ S_3 &= \frac{13}{12}(d_3 - 2d_4 + d_5)^2 + \frac{1}{4}(3d_3 - 4d_4 + d_5)^2. \end{aligned}$$

- Define coefficients α_i as:

$$\alpha_1 = \frac{.1}{(S_1+\varepsilon)^2}, \ \alpha_2 = \frac{.6}{(S_2+\varepsilon)^2}, \ \alpha_3 = \frac{.3}{(S_3+\varepsilon)^2}, \text{with } \varepsilon = 10^{-6}\max(d_1^2, d_2^2, d_3^2, d_4^2, d_5^2) + 10^{-99}.$$

- Define the ω_i as:

$$\omega_1 = \frac{\alpha_1}{\alpha_1+\alpha_2+\alpha_3}, \quad \omega_2 = \frac{\alpha_2}{\alpha_1+\alpha_2+\alpha_3}, \quad \omega_3 = \frac{\alpha_3}{\alpha_1+\alpha_2+\alpha_3}.$$

The construction of D_x^+u is similar, except for the definition of the d_i's:

$$d_1 = \frac{\phi_{i+3} - \phi_{i+2}}{\Delta x}, \quad d_2 = \frac{\phi_{i+2} - \phi_{i+1}}{\Delta x}, \quad d_3 = \frac{\phi_{i+1} - \phi_i}{\Delta x}, \quad d_4 = \frac{\phi_i - \phi_{i-1}}{\Delta x}, \quad d_5 = \frac{\phi_{i-1} - \phi_{i-2}}{\Delta x}.$$

3.2. Approximation of the Level-Set Equation

In the case of the level-set equation (1), a TVD Runge-Kutta scheme in time is used along with a Godunov spatial discretization, H_G, of the Hamiltonian $H(\nabla\phi) = v_n\,|\nabla\phi|$:

$$H_G(a,b,c,d) = \begin{cases} v_n\sqrt{\max(|a^+|^2,|b^-|^2)+\max(|c^+|^2,|d^-|^2)} & \text{if } v_n \le 0 \\ v_n\sqrt{\max(|a^-|^2,|b^+|^2)+\max(|c^-|^2,|d^+|^2)} & \text{if } v_n > 0 \end{cases},$$

with $\cdot^+ = \max(\cdot,0)$ and $\cdot^- = \min(\cdot,0)$. The one-sided derivatives, $D_x^{\pm}\phi$ and $D_y^{\pm}\phi$ are discretized with the WENO scheme of section 3.1.

Time discretizations use the third-order accurate Total Variation Diminishing Runge-Kutta schemes (TVD RK3), which is defined as a combination of Euler steps [35]. For example, in the case of the level-set equation $\phi_t + v_n|\nabla\phi| = 0$, where the Godunov Hamiltonian H_G is used, one first performs an Euler step from ϕ^n to find a temporary solution, $\tilde{\phi}^{n+1}$, at time t^{n+1}:

$$\frac{\tilde{\phi}^{n+1}-\phi^n}{\Delta t} + H_G(\phi^n) = 0,$$

and another Euler step from $\tilde{\phi}^{n+1}$ to find a temporary solution, $\tilde{\phi}^{n+2}$, at t^{n+2}:

$$\frac{\tilde{\phi}^{n+2}-\tilde{\phi}^{n+1}}{\Delta t} + H_G(\tilde{\phi}^{n+1}) = 0.$$

Averaging gives a temporary solution $\tilde{\phi}^{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\tilde{\phi}^{n+2}$ at time $t^{n+\frac{1}{2}}$. Finally, an Euler step is performed to find a temporary solution, $\tilde{\phi}^{n+\frac{3}{2}}$, at time $t^{n+\frac{3}{2}}$, from the solution $\tilde{\phi}^{n+\frac{1}{2}}$:

$$\frac{\tilde{\phi}^{n+\frac{3}{2}}-\tilde{\phi}^{n+\frac{1}{2}}}{\Delta t} + H_G(\tilde{\phi}^{n+\frac{1}{2}}) = 0,$$

which is used to update ϕ^{n+1} by linear averaging:

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\tilde{\phi}^{n+\frac{3}{2}}.$$

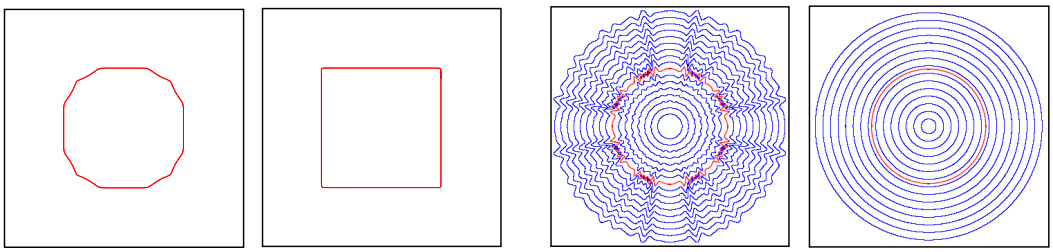
3.3. Approximation of the Reinitialization Equation

In the case of the reinitialization equation (2), a TVD Runge-Kutta scheme (see section 3.2) in time is also used along with a Godunov spatial discretization, H_G of the Hamiltonian $H(\nabla\phi) = \mathrm{sgn}(\phi^0)\,(|\nabla\phi|-1)$:

$$H_G(a,b,c,d) = \begin{cases} \mathrm{sgn}(\phi^0)\sqrt{\max(|a^+|^2,|b^-|^2)+\max(|c^+|^2,|d^-|^2)}-1 & \text{if } \mathrm{sgn}(\phi^0) \le 0 \\ \mathrm{sgn}(\phi^0)\sqrt{\max(|a^-|^2,|b^+|^2)+\max(|c^-|^2,|d^+|^2)}-1 & \text{if } \mathrm{sgn}(\phi^0) > 0 \end{cases},$$

with $\cdot^+ = \max(\cdot,0)$ and $\cdot^- = \min(\cdot,0)$. However, in the reinitialization equation, the one-sided derivatives, $D_x^{\pm}\phi$ and $D_y^{\pm}\phi$ are discretized using a modification of the WENO scheme of section 3.1: In [36], Russo and Smereka pointed out that the WENO scheme of [31] for the reinitialization equation ignores the exact interface location, which produces a mass loss. They solved this problem by using the exact interface location and the fact that $\phi = 0$ at the interface in the construction of the numerical stencils. They developed a second-order accurate reinitialization procedure that significantly improved mass conservation properties (see figure 2(a)). Later, du Chene *et al.* [37] further exploited the idea of Russo and Smereka in order to develop a fourth-order accurate scheme for the reinitialization equation, which in turns produces second-order accurate computations of the mean curvature (see figure 2(b)).

Finally, we mention that other techniques can be used to reinitialize ϕ as a distance function, [38–44], each with their pros and cons. We refer the interested readers to the book by Osher and Fedkiw [45] as well to the book by Sethian [46] for general methods associated with the level-set method.



(a) Interface location (b) Interface curvature

Figure 2: *(a): Interface representing a square after being reinitialized on a* 100×100 *grid using the original HJ-WENO scheme of [31] (left) and the second-order accurate scheme of [36] (right). The scheme of Russo and Smereka significantly improves mass conservation. (b): Comparison of the isocontour of the mean curvature for a circular interface using the HJ-WENO scheme from Jiang and Peng [31] and the modified HJ-WENO scheme of du Chene et al. [37]. The scheme of du Chene et al. produces accurate curvature computations, free of numerical noise. (Color online).*

3.4. Reinitialization using the Hopf-Lax Formula

In this section, we discuss a novel approach to compute a signed distance function, that makes use of the Hopf-Lax formula. This method is particularly appealing for parallel processing for its property that the computation at any grid point is independent of that at any other point, hence producing an "embarrassingly parallel" approach.

Darbon *et al.* developed a numerical approach based on the Hopf formula to solve Hamilton-Jacobi equations in high dimensions when the initial data is convex [47]. Based on this work, Lee *et al.* [48] introduced a fast algorithm using the Hopf-Lax formula for solving the Eikonal equation, $|\nabla\phi| = 1$, in the case where the interface Γ is not necessarily convex. Their work considered the reinitialization of a level-set function in the case where the interface Γ is known analytically. Recently, Royston *et al.* extended the methodology to the practical case where the interface Γ is defined on a grid [49], which we describe next.

Considering a level-set function ϕ^0 defined at grid points $\boldsymbol{i}$, a bilinear interpolation scheme is used to define $\phi^0(\boldsymbol{x}) = \sum_{\boldsymbol{i}} \phi^0_{\boldsymbol{i}} N_{\boldsymbol{i}}(\boldsymbol{x})$ at any space location $\boldsymbol{x}$, where $N_{\boldsymbol{i}}$ refers to the interpolation kernel at $\boldsymbol{i}$. The distance function is then found through the computation of a function $\tilde{\phi}$ that satisfies:

$$\begin{aligned} \frac{\partial}{\partial t}\tilde{\phi}(\boldsymbol{x},t) + |\nabla\tilde{\phi}(\boldsymbol{x},t)| &= 0, \\ \tilde{\phi}(\boldsymbol{x},0) &= \phi^0(\boldsymbol{x}). \end{aligned}$$

Since this equation evolves the zero-level set of $\tilde{\phi}$ normal to itself at speed 1, the distance, $d_{\boldsymbol{i}}$, to the interface at a grid point $\boldsymbol{i}$ is given by the time, $\hat{t}$, it takes for $\tilde{\phi}$ to be zero at that grid point, i.e. $d_{\boldsymbol{i}} = \hat{t} : \tilde{\phi}(\boldsymbol{x}_{\boldsymbol{i}}, \hat{t}) = 0$. This root finding problem is solved using the secant method:

$$t^{k+1} = t^k - \tilde{\phi}(\boldsymbol{x}_{\boldsymbol{i}}, t^k)\frac{t^k - t^{k-1}}{\tilde{\phi}(\boldsymbol{x}_{\boldsymbol{i}}, t^k) - \tilde{\phi}(\boldsymbol{x}_{\boldsymbol{i}}, t^{k-1})},$$

with initial guesses $t^0 = 0$ (or given by the updated values at neighboring grid points) and $t^1 = t^0 + O(\Delta x)$. In the case where the denominator vanishes and the secant method has not converged, the next time iterate is found with:

$$\begin{aligned} t^{k+1} &= t^k + O(\Delta x) \quad \text{if } \tilde{\phi}(\boldsymbol{x}_{\boldsymbol{i}}, t^k) > 0, \\ t^{k+1} &= t^k - O(\Delta x) \quad \text{Otherwise.} \end{aligned}$$

The evaluation of $\tilde{\phi}$ is performed using the Hopf-Lax formula, which amounts to finding the minimum of ϕ^0 over a ball, $\mathcal{B}(\boldsymbol{x}_{\boldsymbol{i}}, t^k)$, of radius t^k and centered at $\boldsymbol{x}_{\boldsymbol{i}}$:

$$\tilde{\phi}(\boldsymbol{x}_{\boldsymbol{i}}, t^k) = \min_{\boldsymbol{y}\in\mathcal{B}(\boldsymbol{x}_{\boldsymbol{i}}, t^k)} \phi^0(\boldsymbol{y}). \tag{4}$$

In [48], Lee *et al.* used the split Bregman iterations to minimize (4), while Royston *et al.* used a projected gradient descent [49]:

$$\begin{cases} \tilde{\boldsymbol{y}}_k^{j+1} = \tilde{\boldsymbol{y}}_k^{j} - \Delta x \, \nabla \phi^0(\boldsymbol{y}_k^j), \\ \boldsymbol{y}_k^{j+1} = \underset{\boldsymbol{y} \in \mathcal{B}(\boldsymbol{x_i}, t^k)}{\texttt{PROJ}}(\tilde{\boldsymbol{y}}_k^{j+1}), \end{cases}$$

where

$$\underset{\boldsymbol{y} \in \mathcal{B}(\boldsymbol{x_i}, t^k)}{\texttt{PROJ}}(\boldsymbol{y}) = \begin{cases} \boldsymbol{y} & \text{if } |\boldsymbol{x_i} - \boldsymbol{y}| \leq t^k, \\ \boldsymbol{x_i} - t^k \dfrac{\boldsymbol{x_i} - \boldsymbol{y}}{|\boldsymbol{x_i} - \boldsymbol{y}|} & \text{Otherwise.} \end{cases}$$

Since this approach to reinitializing a level-set function is based on a minimization process, the typical problem of converging to a local minimum is avoided by randomly seeding multiple initial guesses $\boldsymbol{y}^0$ (one initial guess per grid cell in $\mathcal{B}(\boldsymbol{x_i}, t^k)$) and selecting the minimum in magnitude over all computed distances. A notable strength of this approach is that it is embarrassingly parallel. Royston *et al.* demonstrate the effectiveness of this approach with an parallel implementation on a Graphics Processing Unit (GPU) [49], for which a typical reinitialization takes about 70 ms on a 512×512 grid. We also refer the interested reader to the recent work of Chow *et al.* [50, 51], that further considers minimization approaches for solving Hamilton-Jacobi systems in high dimensions.

4. Level-Set on Adaptive Grids and Parallel Architectures

The democratization of high performance computing facilities and the need for resolving small length scales while reducing the computational cost of simulations, has led researchers to develop the level-set technology on adaptive grids and massively parallel architectures. We focus here on adaptive Cartesian grids and refer the interested reader to [18, 52, 53] and the references therein for description of algorithms on unstructured grids.

Adaptive mesh refinement technique for Cartesian grids date back to the work of Berger and Oliger [54] who considered blocks of uniform grids that are recursively added to regions of a background uniform grid according to some refinement criteria. Originally devised for compressible flows, this strategy has been successfully applied to a wide range of solvers, e.g. [55–57] and the references therein. Later, numerical methods based on the Quadtree and Octree data structures gained popularity for their versatility in mesh generation for classes of problems where shocks do not occur. For example Strain [58] introduced a node-based semi-Lagrangian method for solving the advection equation on Quadtree Cartesian grids, with application to the linear level-set equation. Rather recently, Min and Gibou developed a second-order accurate level-set methodology, including the case of the nonlinear level-set equation, as well as extrapolation and reinitialization schemes. Their node-based approach was also used to develop adaptive solvers for a wide range of PDEs and applications [59–76].

Quad-/Oc-tree grids are particularly well-suited for the level-set method because most of the computational work is concentrated near the interface, where high resolution is needed both for the level-set function itself and for typical physical quantities that usually need high resolution near evolving fronts. The coarsening away from the interface leads to a computational method that is very efficient. In fact, Brun *et al.* [77] pointed out that a tree structure is more efficient than a local level-set method using a hashtable data structure, i.e. a method that only stores a band of uniform grid near the interface. Quad-/Oc-tree Cartesian grids have also been used for solving equation in fluid dynamics using a MAC grid sampling, as e.g. [78–82] and the references therein.

4.1. Quad-/Oc-tree Cartesian Grids

A Cartesian Quad-/Oc-tree grid refers to one that uses the Quad-/Oc-tree data structure for its digital representation. These data structures have been originally introduced in the computer graphics community and used in many contexts before being exploited in scientific computing. We

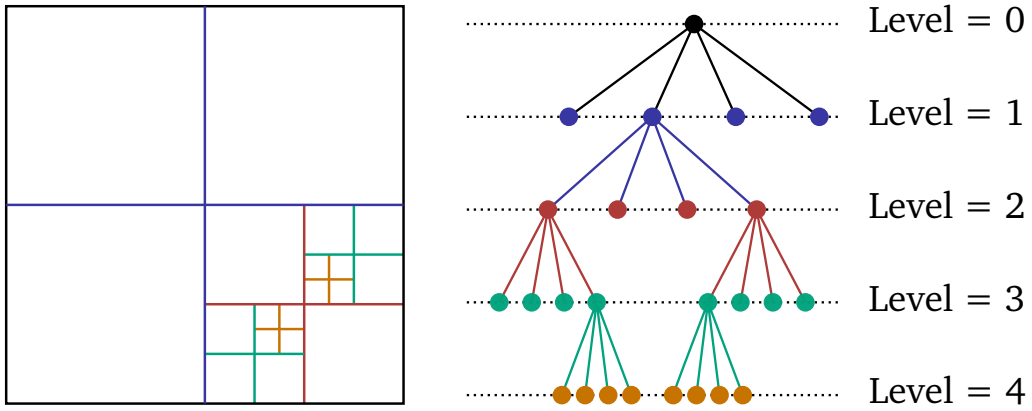


Figure 3: *Quadtree Cartesian grid (left) and its representation (right). The entire domain corresponds to the root of the tree (level 0). Each cell can then be recursively subdivided further into four children. In this example, the tree is non-graded. (Color online).*

refer the interested reader to the excellent books by Samet [83, 84] for a detailed presentation of Quadtree (2D) and Octree (3D) data structures. An example of an adaptive Cartesian grid and its Quadtree representation is given in figure 3. We call a (m, n) Quadtree Cartesian grid one for which the coarsest level is m and finest one is n. A grid is said to be non-graded if the difference of levels between adjacent cells is unconstrained. Graded grids add a constraint to mesh generation procedures and, to some extent, degrade the computational efficiency of solvers by adding potentially unnecessary degrees of freedom (extreme cases are discussed in [85, 86]). An Octree Cartesian grid is the corresponding construct in three spatial dimensions. The Quadtree and Octree data structures provides a $O(\ln(n))$ access to the data stored at their leaves.

In the case of the level-set method, where the finest resolution is needed only near the zero-level set of the level-set function ϕ, a simple refinement criteria is [87–89]:

$$\min_{v \in \text{vertices}(\mathcal{C})} |\phi(v)| \leq \text{Lip}(\phi) \cdot \text{diag-size}(\mathcal{C}), \tag{5}$$

where $\text{Lip}(\phi)$ is the Lipschitz constant associated with ϕ, $\text{diag-size}(\mathcal{C})$ refers to the length of the diagonal of the current cell $\mathcal{C}$ and v refers to a vertex (or node) of the current cell.

4.2. Finite Difference Discretizations

Nonuniform Cartesian grids require one to define level-set values at T-junction nodes, as for example ϕ_G in figure 4. In [89], this value in two spatial dimensions is defined to third-order accuracy by:

$$\phi_g^G = \frac{\phi_3\, s_4 + \phi_4\, s_3}{s_3 + s_4} - \frac{s_3\, s_4}{s_1 + s_2}\left(\frac{\phi_1 - \phi_0}{s_1} + \frac{\phi_2 - \phi_0}{s_2}\right).$$

Similarly, third-order accurate ghost values are defined in three spatial dimensions ([89]). Finite difference formulas can then be formulated in a dimension-by-dimension framework and used in the Godunov approximations of the Hamiltonians, as in section 3.2 and 3.3. In particular, the second-order accurate one-sided approximations of the first-order derivatives are given by:

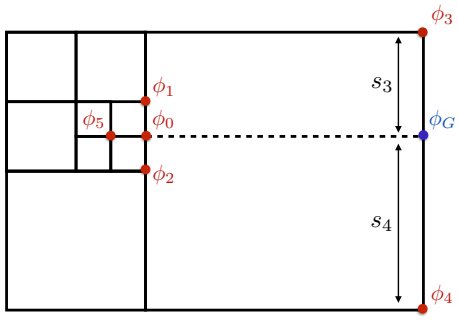


Figure 4: *A T-junction configuration.*

$$D_x^+\phi_0 = \frac{\phi_G - \phi_0}{s_G} - \frac{s_G}{2}\texttt{minmod}\left(D_{xx}^0\phi_0, D_{xx}^0\phi_G\right),$$
$$D_x^-\phi_0 = \frac{\phi_0 - \phi_1}{s_1} + \frac{s_1}{2}\texttt{minmod}\left(D_{xx}^0\phi_0, D_{xx}^0\phi_1\right),$$

where the `minmod` slope limiter [33, 34] is defined as:

$$\texttt{minmod}(x,y) = \begin{cases} x & \text{if } |x| > |y|, \\ y & \text{otherwise,} \end{cases}$$

and the second-order derivative are approximated by central difference, e.g. $D^0_{xx}\phi_0$ is approximated as:

$$D^0_{xx}\phi_0 = \frac{\phi_G - \phi_0}{s_G} \cdot \frac{2}{s_5 + s_G} - \frac{\phi_0 - \phi_5}{s_5} \cdot \frac{2}{s_5 + s_G}.$$

Min and Gibou also used the idea of Russo and Smereka with slight modifications in the context of adaptive mesh refinement [89], and Min pointed out that it is advantageous in terms of speed and memory to replace the traditional Runge-Kutta scheme in time with a Gauss-Seidel iteration of the forward Euler scheme [90].

4.3. Semi-Lagrangian Schemes

The level-set method is often used to capture the evolution of an interface that moves according to an externally generated velocity field. In this case, the level-set equation is linear and the standard unconditionally stable semi-Lagrangian methods can be used. The semi-Lagrangian method is based on tracing back along characteristic curves given a velocity field $\boldsymbol{u}$. Specifically, for any grid node $\boldsymbol{x}$, the departure point $\boldsymbol{x}_d$ along the characteristic curve at time t^n is found by approximating the equation (e.g. with the second order mid-point method):

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}.$$

The update is then simply $\phi^{n+1}(\boldsymbol{x}) = \phi^n(\boldsymbol{x}_d)$, where interpolation schemes can be used to define the value of ϕ at an arbitrary location $\boldsymbol{x}_d$. In, [91] first-order accurate bilinear interpolations where used; in [89] a second-order accurate non-oscillatory interpolation scheme was proposed: for example in two spatial dimensions, scaling any computational cell to a unit cell $\mathcal{C}$, one writes:

$$\begin{aligned}\phi(x,y) = {} & \phi(0,0)(1-x)(1-y) + \phi(0,1)(1-x)(y) + \phi(1,0)(x)(1-y) + \phi(1,1)(x)(y) \\ & - \phi_{xx}\frac{x(1-x)}{2} - \phi_{yy}\frac{y(1-y)}{2},\end{aligned}$$

where the second-order derivatives ϕ_{xx} and ϕ_{yy} are defined as:

$$\phi_{xx} = \underset{v \in \text{vertices}(\mathcal{C})}{\texttt{minmod}} D^0_{xx}\phi(v) \qquad \text{and} \qquad \phi_{yy} = \underset{v \in \text{vertices}(\mathcal{C})}{\texttt{minmod}} D^0_{yy}\phi(v).$$

Semi-Lagrangian methods are not conservative, which when applied to the level-set evolution equation leads to mass loss. In Lentine *et al.* [92], the authors modified the interpolation weights to produce a conservative method as follows. A typical update formula for ϕ reads:

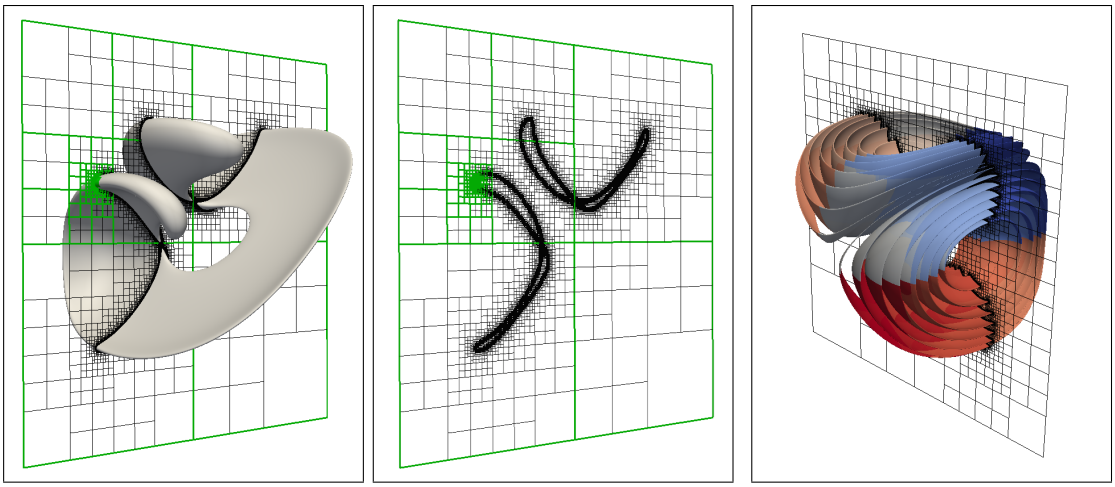
$$\phi^{n+1}(\boldsymbol{x}_j) = \phi^n(\boldsymbol{x}_d) = \sum_i w_{i,j}\phi^n(\boldsymbol{x}_i),$$

where the $w_{i,j}$'s are the interpolation coefficients of cells $\mathcal{C}_i$'s used to update ϕ at $\boldsymbol{x}_j$. After updating ϕ at every grid point $\boldsymbol{x}_j$, one can define the total contribution, σ_i, of each cell $\mathcal{C}_i$ as $\sigma_i = \sum_i w_{i,j}$. Usually, that sum does not equal 1 because of truncation errors; however it needs to be if mass is to be exactly conserved. Lentine *et al.* enforce this it in two steps:

1. Go through all donor cells for which $\sigma_i \geq 1$ and scale the weights as $\hat{w}_{i,j} = w_{i,j}/\sigma_i$.
2. Go through all other cells, i.e. those for which $\sigma_i < 1$, apply a *forward* semi-Lagrangian ray casting giving forward weights $f_{i,j}$ and use them to distribute the remaining $(1-\sigma_i)\,\phi_{i,j}$ to the cells j that are used to perform the interpolation.

The final weights are defined as $\hat{w}_{i,j} = w_{i,j} + (1-\sigma_i)\,f_{i,j}$ and the final update is $\phi^{n+1}(\boldsymbol{x}_j) = \sum_i \hat{w}_{i,j}\phi^n(\boldsymbol{x}_i)$. This scheme is valuable beyond linear level-set equations; in particular [92] showed that it produces the correct speed of propagation for solutions of conservation laws.

4.4. Level-Set on a Forest of Octrees



(a) Local (Green) and global (black) grids. (b) Snapshot at $t = 9$

Figure 5: *(a): global and local Octree grids. The processor with rank "green" stores its local Octree grid, which is only as fine as the global Octree for the region it is responsible for and significantly coarser elsewhere. (b): level-set at* $t = 9$*, color-coded by processors rank. Results using the* `parCASL` *library [93].*

In [94], Mirzadeh *et al.* introduced the level-set method on a forest of Quad-/Oc-trees. This work makes use of the `p4est` library of Burstedde *et al.* [95] for the partitioning of the grid, combined with a suite of algorithms that constructs local trees to each processor, which then enable the discretizations discussed in sections 4.1-4.3.

When considering parallel computation, where the communication of data between processes is orders of magnitude more time consuming than computation, the main focus of algorithm design is to reduce communication or/and hide their cost by intertwining them with computation. Reducing communication is achieved by grouping, or partitioning, the data in the local memory of each process. The strategy of Burstedde *et al.* [95] is illustrated in figure 6: (1) a macromesh of uniform cells is created and replicated on each process; (2) a forest of Quad-/Oc-trees is created recursively using all processes and partitioned among them. The partitioning is using a Z-ordering of the trees' leaves, which are recorded in a one-dimensional array before being split equally among the available processes. Using the Z-ordering clusters the data contiguously, as depicted in figure 6, and thus subsequently minimizes the amount to communication during the discretization phase.

Since the `p4est` library only stores the one-dimensional array of the forest's leaves, an algorithm for constructing the local trees on each processor is introduced in order to use the discretizations described in sections 3, 4.2 and 4.3. The procedure introduced in Mirzadeh *et al.* [94] is to create a local tree in such a way as the levels of its leaves correspond to that of the one-dimensional array locally. For example figure 5(a) depicts the local and global Octrees on one out of 4096 cores in the case of the standard vortex test [96] on an Octree with a maximum resolution level of 12. In this case, the global grid (i.e. the forest) has 231,905,632 grid points, while the local tree depicted has 1,566,272 grid points. A global grid on a uniform grid would amount to 68,719,476,736 grid points, corresponding to about 300 times many more grid points. Figure 5(b) depicts the solution at $t = 9$, color-coded by the processors' rank.

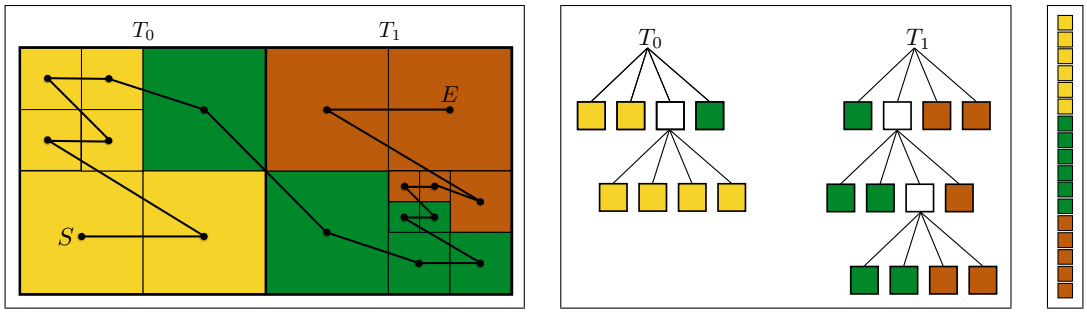


Figure 6: *Two trees, T_0 and T_1, constitute the 'forest'. The Z-ordering (left) of the Quadtree's leaves (center) is used to partition the data among the available processes (right). In this figure, the different colors correspond to different processes. S represents the starting cell and E the last cell visited. In the* `p4est` *library, only the one-dimension array (right) is stored among the available processes.*

5. Applying Boundary Conditions

Since the level-set function gives an implicit representation of an irregular free boundary, special care is needed to apply boundary conditions in a sharp manner. In that regard, the Ghost-Fluid Method has been key to providing a simple and general procedure. Introduced in the context of compressible inviscid flows, its philosophy is based on introducing two copies of the solution, one corresponding to the real fluid and another one to conveniently impose the boundary condition. The particular strength of the GFM is that it enables a sharp treatment of boundary conditions, hence eliminating spurious oscillations of the solution or its unphysical smearing. The GFM was first introduced to treat contact discontinuities in inviscid compressible flows [97, 98] and then extended to the treatment of deflagration and detonation in compressible flows [99], including a conservative treatment for stiff detonation waves [100]. It has also been applied to the coupling of Eulerian/Lagrangian frameworks [101] and compressible/incompressible flows [102]. Finally, it has been applied to flame fronts propagating in incompressible flows [103], to multiphase flows [104], for which a GFM approach to solve the Poisson equation with jump in the solution and its gradient [105] was introduced, and to multiphase flows with phase change [106].

The GFM idea has also been applied to the case where one wants to treat Dirichlet boundary conditions [107, 108] with application to the Stefan problem [109] and has been refined in order to treat jump boundary conditions with higher accuracy [76]. Imposing Robin boundary conditions in a GFM framework is not natural and one prefers a finite volume treatment instead [67, 69, 110].

The treatment of boundary conditions in a level-set framework is often associated with solving a Poisson equation, e.g. as part of a projection method in fluids, for the simulation of materials processing or for simulating biological phenomena. Therefore, we describe in this section the numerical techniques that enable the treatment of jump, Dirichlet, and Robin boundary conditions for the Poisson equation.

5.1. Jump Boundary Conditions

An essential component of solving multiphase flows or other scientific problems where the solution and its gradient jump over an interface can be written as:

$$\begin{cases} \nabla \cdot (\beta \nabla u) &= f & \text{in } \Omega^- \cup \Omega^+ \\ [u] &= g_\Gamma & \text{on } \Gamma \\ [\beta \nabla u \cdot \boldsymbol{n}] &= h_\Gamma & \text{on } \Gamma \end{cases}, \tag{6}$$

where the computational domain Ω is composed of two subdomains, Ω^- and Ω^+, separated by a co-dimension one interface Γ, with $\boldsymbol{n}$ its outward normal. Here, $\beta = \beta(\boldsymbol{x})$, with $\boldsymbol{x} \in \mathbb{R}^n$ $(n \in \mathbb{N})$, is continuous in each subdomains and bounded from below by a positive constant and $[q] = q_\Gamma^+ - q_\Gamma^-$

indicates a discontinuity in the quantity q across Γ, f is in $L^2(\Omega)$, g_Γ, h_Γ and k are given. Note that this general formulation includes possible discontinuities in the coefficient β and in the gradient of the solution ∇u. Dirichlet or Neumann boundary conditions are applied on the boundary of Ω, denoted by $\partial\Omega$.

5.1.1. The Ghost-Fluid Method

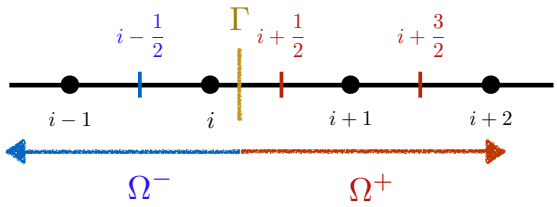


Figure 7: *Local grid near an interface* Γ.

The GFM for the Poisson equation is best described first in one spatial dimension and in the case of a constant β. Consider a domain $\Omega = \Omega^- \cup \Omega^+$, with the region Ω^- defined by the set of points x with $\phi(x) \leq 0$; the interface between the two subdomains is called Γ. The computational domain is discretized into cells of size Δx, with the grid nodes x_i located at the cells' center. The cell edges are referred to as faces, and the two faces bounding the grid node x_i are located at $x_{i\pm\frac{1}{2}}$. The numerical solution of the Poisson equation is computed at the grid nodes and is denoted by $u_i = u(x_i)$. Referring to figure 7, the discretization of $u_{xx} = f$ must be changed for grid points i and $i+1$ that are adjacent to Γ in such a way that the jump conditions $[u] = g_\Gamma$ and $[u_x] = h_\Gamma$ are imposed. For the sake of presentation, we consider the case where $x_{i+\frac{1}{2}}$ is in Ω^+.

- Discretization at grid point i: We use the standard central differencing formula:

$$\frac{(u_x)^{+R}_{i+\frac{1}{2}} - (u_x)^{-R}_{i-\frac{1}{2}}}{\Delta x} = f_i.$$

Since there is a jump in u_x, the expression above is $O(\frac{1}{\Delta x})$. In order to avoid differencing across discontinuities, the GFM replaces $(u_x)^{+R}_{i+\frac{1}{2}}$ by $(u_x)^{-G}_{i+\frac{1}{2}}$:

$$\frac{(u_x)^{-G}_{i+\frac{1}{2}} - (u_x)^{-R}_{i-\frac{1}{2}}}{\Delta x} = f_i.$$

Using the jump condition $[u_x] = (u_x)^+ - (u_x)^- = h_\Gamma$, we have $(u_x)^{-G}_{i+\frac{1}{2}} = (u_x)^{+R}_{i+\frac{1}{2}} - h_\Gamma$, which gives:

$$\frac{(u_x)^{+R}_{i+\frac{1}{2}} - h_\Gamma - (u_x)^{-R}_{i-\frac{1}{2}}}{\Delta x} = f_i \iff \frac{(u_x)^{+R}_{i+\frac{1}{2}} - (u_x)^{-R}_{i-\frac{1}{2}}}{\Delta x} = f_i + \Delta x\, h_\Gamma.$$

The above expression requires the approximation of $(u_x)^{-R}_{i-\frac{1}{2}}$ and $(u_x)^{+R}_{i+\frac{1}{2}}$, given by central differencing:

$$(u_x)^{-R}_{i-\frac{1}{2}} = \frac{u_i^{-R} - u_{i-1}^{-R}}{\Delta x} \qquad \text{and} \qquad (u_x)^{+R}_{i+\frac{1}{2}} = \frac{u_{i+1}^{+R} - u_i^{-R}}{\Delta x}.$$

Again, while the approximation of $(u_x)^{-R}_{i-\frac{1}{2}}$ involves quantities on the same side of the interface and thus will produce accurate results, the approximation of $(u_x)^{+R}_{i-\frac{1}{2}}$ mixes quantities that experience a jump across the interface and thus will produce $O(\frac{1}{\Delta x})$ errors. Again, the GFM replaces u^{+R}_{i+1} by

a ghost value u_{i+1}^{-G} defined from the jump condition $[u] = (u)^+ - (u)^- = g_\Gamma$, i.e. $u_{i+1}^{-G} = u_{i+1}^{+R} - g_\Gamma$. The expression for $(u_x)_{i-\frac{1}{2}}^{+R}$ thus become:

$$(u_x)_{i+\frac{1}{2}}^{+R} = \frac{u_{i+1}^{-G} - u_i^{-R}}{\Delta x} = \frac{u_{i+1}^{+R} - g_\Gamma - u_i^{-R}}{\Delta x}.$$

Putting everything together, the GFM approximation of $u_{xx} = f$ is given by:

$$\frac{1}{\Delta x}\left(\frac{u_{i+1}^{+R} - u_i^{+R}}{\Delta x} - \frac{u_i^{+R} - u_{i-1}^{+R}}{\Delta x}\right) = f_i + \Delta x\, h_\Gamma + \Delta x^2\, g_\Gamma. \tag{7}$$

- The discretization at $i+1$ is similar. We first write:

$$\frac{(u_x)_{i+\frac{3}{2}}^{+R} - (u_x)_{i+\frac{1}{2}}^{-R}}{\Delta x} = f_i.$$

In this case, both $(u_x)_{i+\frac{3}{2}}^{+R}$ and $(u_x)_{i+\frac{1}{2}}^{+R}$ are on the same side of the interface and thus no special treatment is needed for approximating the fluxes. The approximation of $(u_x)_{i+\frac{3}{2}}^{-R}$ and $(u_x)_{i+\frac{1}{2}}^{+R}$ are given by central differencing as:

$$(u_x)_{i+\frac{3}{2}}^{+R} = \frac{u_{i+2}^{+R} - u_{i+1}^{+R}}{\Delta x} \qquad \text{and} \qquad (u_x)_{i+\frac{1}{2}}^{+R} = \frac{u_{i+1}^{+R} - u_i^{-R}}{\Delta x}.$$

In this case, the approximation of $(u_x)_{i+\frac{1}{2}}^{+R}$ depends on the solution u on both side of the interface, hence producing $O(\frac{1}{\Delta x})$ errors. The GFM replaces u_i^{-R} by a ghost value u_i^{+G} defined from the jump condition $[u] = (u)^+ - (u)^- = g_\Gamma$, i.e. $u_i^{+G} = u_i^{-R} + g_\Gamma$. The expression for $(u_x)_{i+\frac{1}{2}}^{+R}$ thus become:

$$(u_x)_{i+\frac{1}{2}}^{+R} = \frac{u_{i+1}^{+R} - u_i^{+G}}{\Delta x} = \frac{u_{i+1}^{+R} - (u_i^{-R} + g_\Gamma)}{\Delta x}.$$

Putting everything together, the GFM approximation of $u_{xx} = f$ is given by:

$$\frac{1}{\Delta x}\left(\frac{u_{i+2}^{+R} - u_{i+1}^{+R}}{\Delta x} - \frac{u_{i+1}^{+R} - u_i^{+R}}{\Delta x}\right) = f_i - \Delta x^2\, g_\Gamma. \tag{8}$$

Remarks: In practice the jump conditions are often given at grid nodes. For example, in a two-phase flow simulation the jump in the solution is proportional to the interface's curvature, which is computed at grid point. In this case, one defines g_Γ and h_Γ by interpolation:

$$g_\Gamma = \frac{g_i|\phi_{i+1}| + g_{i+1}|\phi_i|}{|\phi_i| + |\phi_{i+1}|} \qquad \text{and} \qquad h_\Gamma = \frac{h_i|\phi_{i+1}| + h_{i+1}|\phi_i|}{|\phi_i| + |\phi_{i+1}|}.$$

We also refer the interested reader to [105] for details on subcell resolution and for the discretization of the variable coefficient Poisson equation $(\beta u_x)_x = f$.

Equations (7) and (8) show that the GFM produces a linear system with the standard symmetric positive definite matrix for the Poisson equation on regular domains in the absence of jumps. Only the right-hand side of the linear system is modified to include the jump conditions. In two spatial dimensions, [105] proposed a simple dimension-by-dimension approach, where the system (6) is modified as follows:

$$\begin{cases} (\beta u_x)_x + (\beta u_y)_y &= f & \text{in } \Omega^- \cup \Omega^+ \\ [u] &= g_\Gamma & \text{on } \Gamma \\ [\beta u_x] &= h_\Gamma\, n_1 & \text{on } \Gamma \\ [\beta u_y] &= h_\Gamma\, n_2 & \text{on } \Gamma \end{cases},$$

where n_1 and n_2 are the x- and y- components of the normal vector $\boldsymbol{n}$. This gives a straightforward procedure to solve the Poisson equation in two and three spatial dimensions since the GFM can be applied independently on the term $(\beta u_x)_x$ with the jump condition $[\beta u_x] = h_\Gamma\, n_1$ and on the term $(\beta u_y)_y$ with the jump condition $[\beta u_x] = h_\Gamma\, n_2$. As pointed out in [105], this treatment preserves the jump in the normal derivative but smears out the jump in the tangential derivative. Nonetheless, the method has been shown to be convergent with first-order accuracy [111]. Unfortunately, the gradients do not converge in general and thus this approach is not appropriate in problems for which the gradient of the solution drives the accuracy of the problem.

5.1.2. The Voronoi Interface Method

In [76], the authors introduced an approach based on building a Voronoi tessellation local to the interface, which then enables the direct discretization of the jump conditions in the normal direction. This Voronoi Interface Method (VIM)[1] produces second-order accurate solutions in the L^∞-norm, first-order accurate gradients and, similar to the GFM, produces a symmetric positive definite linear system with the jump conditions only influencing its right-hand side. This enables the linear system to be solved with fast iterative solvers, e.g. the Conjugate Gradient of the Petsc libraries [113, 114] preconditioned with the Hypre multigrid [115, 116].

The procedure for creating the grid relies on first finding the projection onto Γ of the degrees of freedoms adjacent to the interface. Then, those degrees of freedom are replaced by two new degrees of freedom located in the normal direction on either side of the interface at a (arbitrary) distance $d_\Gamma = \texttt{diag}/5$ from Γ, where `diag` refers to the length of the local cell's diagonal. A Voronoi tessellation is then build from this new set of degrees of freedom as illustrated in figure 8. Since the original grid is only modified near the interface, building the Voronoi tessellation is a computationally efficient procedure that can be easily applied to the adaptive Cartesian grids of section 4. In addition this procedure is embarrassingly parallel and thus can be applied to the parallel framework of section 4.4. In [76], the authors used the `Voro++` library [117] to construct the local Voronoi tessellation.

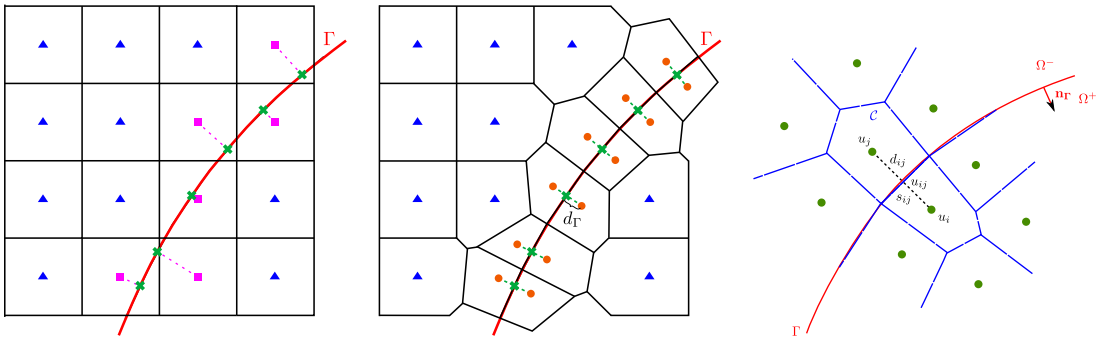


Figure 8: *Left: projections onto* Γ *of all the nodes adjacent to the interface. Center: two additional degrees of freedom are created on each side of* Γ *and the original node adjacent to the interface have been deleted. The final grid is obtained from constructing a Voronoi tessellation of the new set of degrees of freedom. Right: typical local control volume* $\mathcal{C}$ *used in the discretization of* (6).

Referring to figure 8 (right), the system (6) is discretized at each degree of freedom i (with control volume $\mathcal{C}$) with a finite volume approach:

$$\int_{\mathcal{C}} \nabla \cdot (\beta \nabla u)\, dV = \int_{\partial \mathcal{C}} (\beta \nabla u) \cdot \boldsymbol{n}_{\mathcal{C}}\, dl \approx \sum_j s_{ij} \beta_i \frac{u_{ij} - u_i}{d_{ij}/2},$$

[1] Not to be confused with the Voronoi Implicit Interface Method, which is an interface tracking methodology [112].

where u_{ij} is the value of u at the middle of the segment $[i,j]$, and thus can be considered as u_Γ. By definition, $\boldsymbol{n}_\mathcal{C}$ is the outer normal to the faces of $\mathcal{C}$ connecting the local grid node i and its neighbors j and s_{ij} is the measure of that face; β_i is the value of the variable coefficient $\beta(\boldsymbol{x})$ at i; d_{ij} is the distance between the degrees of freedom i and j.

The jump conditions affect the discretization in the case where the grid nodes i and j belong to different subdomains. The jump in the normal flux can be used to define:

$$s_{ij}\beta_i\frac{u_{ij}^+ - u_i}{d_{ij}/2} = s_{ij}\beta_j\frac{u_j - u_{ij}^-}{d_{ij}/2} - s_{ij}\left[\beta\nabla u\cdot\boldsymbol{n}\right],$$

since $\boldsymbol{n}$ points towards i, and using the jump in u (i.e., $u_{ij}^+ = u_{ij}^- + [u]$), one gets:

$$\begin{aligned} s_{ij}\beta_i\frac{u_{ij}^+ - u_i}{d_{ij}/2} &= s_{ij}\beta_j\frac{u_j - u_{ij}^+ + [u]}{d_{ij}/2} - s_{ij}\left[\beta\nabla u\cdot\boldsymbol{n}\right] \\ \iff u_{ij}^+ &= \frac{1}{\beta_i+\beta_j}\left(\beta_j u_j + \beta_i u_i + \beta_j[u] - \frac{d_{ij}}{2}\left[\beta\nabla u\cdot\boldsymbol{n}\right]\right). \end{aligned}$$

For a degree of freedom i, the contribution of its neighboring degree of freedom j is given by:

$$\tilde{\beta}_{ij}s_{ij}\frac{u_j - u_i}{d_{ij}} = \tilde{\beta}_{ij}\frac{s_{ij}}{d_{ij}}\left(-\text{sign}(\phi_i)[u] + \frac{d_{ij}}{2\beta_j}\left[\beta\nabla u\cdot\boldsymbol{n}\right]\right) + \texttt{Vol}(\mathcal{C})\cdot f_i,$$

where $\texttt{Vol}(\mathcal{C})$ is the volume of the Voronoi cell associated to the degree of freedom i and $\tilde{\beta}_{ij} = \dfrac{2\beta_i\beta_j}{\beta_i+\beta_j}$ is the harmonic mean between β_i and β_j. Specifically, for the degree of freedom i, each neighboring degree of freedom j contributes the following expression to the matrix coefficient of the linear system:

$$\tilde{\beta}_{ij}s_{ij}\frac{u_j - u_i}{d_{ij}},$$

and the following expression to its right-hand side:

$$\tilde{\beta}_{ij}\frac{s_{ij}}{d_{ij}}\left(-[u] + \frac{d_{ij}}{2\beta_j}\left[\beta\nabla u\cdot\boldsymbol{n}\right]\right).$$

Since the two domains are decoupled, one can readily consider cases where the jumps in β, u and $\nabla u\cdot\boldsymbol{n}$ are several orders of magnitude. For example [76] present an example where the jump in the variable coefficient β is 10^5.

We include an illustrative example from [76], who considered a computational domain $\Omega = [-1,1]^2$ decomposed into four subdomains by the following three contours:

$$\begin{aligned} \Gamma^0 &= \left\{(x,y),\ \phi^0(x,y) = \sqrt{x^2+y^2} - 0.2\right\}, \\ \Gamma^1 &= \left\{(x,y),\ \phi^1(x,y) = \sqrt{x^2+y^2} - 0.5 + 0.1\cos(5\theta)\right\}, \\ \Gamma^2 &= \left\{(x,y),\ \phi^2(x,y) = \sqrt{x^2+y^2} - 0.8\right\}, \end{aligned}$$

where θ is the angle between (x,y) and the x-axis. The exact solution is defined as:

$$u(x,y) = \begin{cases} e^x + 1.3 & \text{if } (x,y)\in\Omega^0 = \left\{(x,y),\ \phi^0(x,y)\le 0\right\}, \\ \cos(y) + 1.8 & \text{if } (x,y)\in\Omega^1 = \left\{(x,y),\ \phi^1(x,y)\le 0\right\}, \\ \sin(x) + 0.5 & \text{if } (x,y)\in\Omega^2 = \left\{(x,y),\ \phi^2(x,y)\le 0\right\}, \\ -x + \ln(y+2) & \text{if } (x,y)\in\Omega^3 = \left\{(x,y),\ \phi^3(x,y)\le 0\right\}, \end{cases}$$

while the diffusion coefficient is defined as:

$$\beta(x,y) = \begin{cases} y^2+1 & \text{if } (x,y)\in\Omega^0, \\ e^x & \text{if } (x,y)\in\Omega^1, \\ y+1 & \text{if } (x,y)\in\Omega^2, \\ x^2+1 & \text{if } (x,y)\in\Omega^3. \end{cases}$$

The solution is depicted in figure 9, while table 1 gives the order of accuracy of the method in the L^∞-norm.

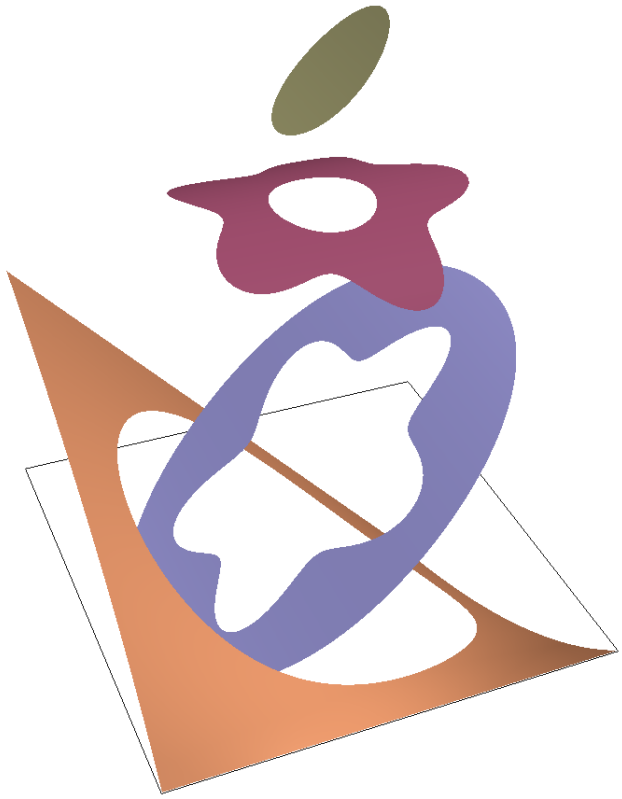


Figure 9: *Solution of section 5.1.2 using the VIM (from [76]).*

resolution	solution	order	gradient	order
2^4	$1.00 \cdot 10^{-3}$	-	$3.33 \cdot 10^{-3}$	-
2^5	$2.33 \cdot 10^{-4}$	2.11	$1.01 \cdot 10^{-3}$	1.72
2^6	$6.23 \cdot 10^{-5}$	1.90	$3.46 \cdot 10^{-4}$	1.54
2^7	$1.56 \cdot 10^{-5}$	2.00	$1.59 \cdot 10^{-4}$	1.12
2^8	$4.00 \cdot 10^{-6}$	1.96	$6.82 \cdot 10^{-5}$	1.22
2^9	$1.01 \cdot 10^{-6}$	1.99	$4.00 \cdot 10^{-5}$	0.77
2^{10}	$2.55 \cdot 10^{-7}$	1.99	$2.24 \cdot 10^{-5}$	0.84

Table 1: Accuracy of the solution of section 5.1.2 and its gradient in the L^∞-norm using the VIM (from [76]).

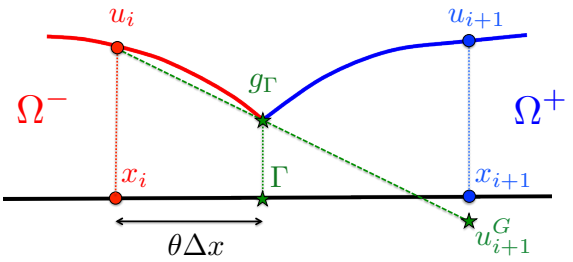


Figure 10: *Definition of the ghost value u^G_{i+1} using a linear extrapolation. First, construct a linear interpolant $\tilde{u}(x) = ax + b$ of u such that $\tilde{u}(0) = u_i$ and $\tilde{u}(\theta\Delta x) = g_\Gamma$. Then define $u^G_{i+1} = \tilde{u}(\Delta x)$. (Color online).*

Consider again the Poisson equation, but this time with a Dirichlet boundary condition of g_Γ at Γ. In this case, it is natural to impose the boundary condition in a dimension by dimension approach since the boundary condition on u can be fixed in each Cartesian directions. Therefore, without loss of generality, we only describe the discretization for the one-dimensional Poisson equation:

$$\left\{ \begin{array}{rcll} \dfrac{\partial}{\partial x}\left(\beta \dfrac{\partial u}{\partial x}\right) & = & f & \text{in } \Omega^- \cup \Omega^+ \\ u & = & g_\Gamma & \text{on } \Gamma \end{array} \right. , \tag{9}$$

for which the following central differencing scheme is used:

$$\frac{1}{\Delta x}\left(\beta_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x}\right) = f_i. \tag{10}$$

As it is the case of section 5.1, one avoids differentiating the fluxes across the interface where the solution presents a kink, by using a ghost value u^G_{i+1} at x_{i+1} across the interface and rewrite equation (10) as:

$$\frac{1}{\Delta x}\left(\beta_{i+\frac{1}{2}} \frac{u^G_{i+1} - u_i}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x}\right) = f_i.$$

The ghost value is defined by first constructing an interpolant $\tilde{u}(x)$ of u on the left of the interface with origin at x_i, and then defining $u^G_{i+1} = \tilde{u}(\Delta x)$. Figure 10 illustrates the definition of the ghost cells in the case of a linear extrapolation. Linear, quadratic and cubic extrapolations are defined by[2]:

Linear Extrapolation: Take $\tilde{u}(x) = ax + b$ with:

$$\tilde{u}(0) = u_i \text{ and } \tilde{u}(\theta\Delta x) = g_\Gamma.$$

Quadratic Extrapolation: Take $\tilde{u}(x) = ax^2 + bx + c$ with:

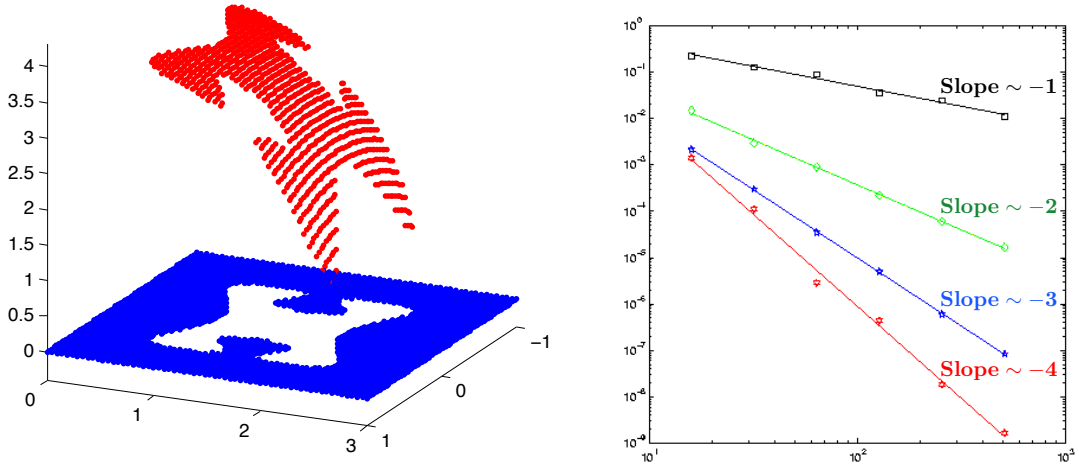
$$\tilde{u}(-\Delta x) = u_{i-1}, \quad \tilde{u}(0) = u_i \text{ and } \tilde{u}(\theta\Delta x) = g_\Gamma.$$

Cubic Extrapolation: Take $\tilde{u}(x) = ax^3 + bx^2 + cx + d$ with:

$$\tilde{u}(-2\Delta x) = u_{i-2}, \quad \tilde{u}(-\Delta x) = u_{i-1}, \quad \tilde{u}(0) = u_i \text{ and } \tilde{u}(\theta\Delta x) = g_\Gamma.$$

[2]One may prefer a Newton's form for constructing the interpolant $\tilde{u}(x)$.

In these equations, $\theta \in [0,1]$ refers to the cell fraction occupied by the subdomain Ω^-. The interface location (and therefore θ) is found by first constructing a linear or higher-order interpolant of the level-set function ϕ and then finding the zero of the interpolant. Similar constructions define u_i^G using values to the right of x_{i+1}. This procedures produces a linear system of equations that is symmetric in the case of a linear extrapolation only. The quadratic extrapolation is equivalent to the Shortley-Weller method [118]. Recent theoretical work have proved the accuracy of the solution in the L^∞-norm and studied the condition number of the linear system [119]. We also refer the interested reader to the work of Min *et al.* who have proved the accuracy of the gradients in the L^2-norm [120] and L^∞-norm [121].



(a) Solution profile. (b) Accuracy.

Figure 11: *Solution of section 5.2 (left) and its accuracy in the L^∞-norm for constant (black), linear (green), quadratic (blue) and cubic (red) extrapolations. (Color online). From [108].*

We include an illustrative example from [107], who considered the system (9) on a computational domain $\Omega = [-1,1] \times [0,3]$. The exact solution is defined as $u = \exp(x^2 + y^2 - \frac{\pi^2}{25})$ and the irregular domain is parametrized by:

$$\begin{cases} x(\alpha) &= .6\cos(\alpha) - .3\cos(3\alpha) \\ y(\alpha) &= 1.5 + .7\sin(\alpha) - .07\sin(3\alpha) + .2\sin(7\alpha) \end{cases},$$

where $\alpha \in [0, 2\pi]$. The numerical solution is depicted in figure 11(a) and the slopes in figure 11(b) give the order of accuracy obtained using different definition of the ghost values. As it was the case for the solution of (6), the solutions of (9) inside and outside the interface satisfy the correct boundary conditions and are decoupled, making the methods applicable in the most general case.

5.3. Robin Boundary Conditions

Consider the following system:

$$\begin{cases} \nabla \cdot (\beta \nabla u) &= f & \text{in } \Omega^- \cup \Omega^+ \\ \nabla u \cdot \boldsymbol{n} + \alpha u &= r_\Gamma & \text{on } \Gamma \end{cases}, \tag{11}$$

where α is a positive constant. Imposing Robin boundary condition in the Ghost-Fluid framework is less amenable than following a finite volume approach, which is well-suited when the boundary condition is expressed as a function of the solution's flux. In that case, the two subdomains are treated independently; we describe here the approach introduced in Papac *et al.* [110] who considered the integration of (11) in the dual cell $\mathcal{C}$ centered at the local grid node c (see figure 12):

$$\int_{\mathcal{C}\cap\Omega^-} \nabla\cdot(\beta\,\nabla u)\,d\Omega = \int_{\mathcal{C}\cap\Omega^-} f\,d\Omega,$$
$$\iff \int_{\partial(\mathcal{C}\cap\Omega^-)} \beta\,\nabla u\cdot\boldsymbol{n}\,d\Gamma = \int_{\mathcal{C}\cap\Omega^-} f\,d\Omega.$$

The boundary integral is split into two parts: the boundary of the computational cell $\mathcal{C}$ that belongs to Ω^- and the part of the boundary Γ that is located in $\mathcal{C}$:

$$\int_{\partial(\mathcal{C}\cap\Omega^-)} \beta\,\nabla u\cdot\boldsymbol{n}\,d\Gamma = \int_{\partial\mathcal{C}\cap\Omega^-} \beta\,\nabla u\cdot\boldsymbol{n}\,d\Gamma + \int_{\mathcal{C}\cap\Gamma} \beta\,(r_\Gamma-\alpha u)\,d\Gamma,$$

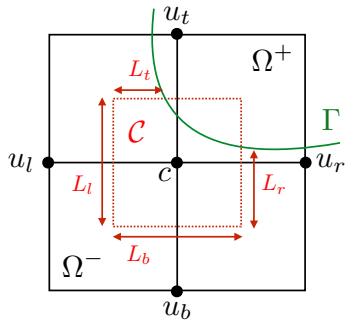


Figure 12: *Nomenclature for imposing a Robin boundary condition.*

where we have invoked the Robin boundary condition in the last term. Referring to figure 12, we approximate the first integral as:

$$\int_{\partial\mathcal{C}\cap\Omega^-} \nabla u^{n+1}\cdot\boldsymbol{n}\,d\Gamma = \frac{u_r-u_c}{s_r}L_r - \frac{u_c-u_l}{s_r}L_l + \frac{u_t-u_c}{s_r}L_t - \frac{u_c-u_b}{s_r}L_b,$$

where L_r (resp. L_l, L_t and L_b) is the length fraction of the right (resp. left, top and bottom) face that is in Ω^-. Finally, one writes:

$$\int_{\mathcal{C}\cap\Gamma} \alpha\,\beta\,u\,d\Gamma \approx \alpha\,\beta_c\,u_c \int_{\mathcal{C}\cap\Gamma} d\Gamma,$$

where u_c refers to the value of u at the center of the cell $\mathcal{C}$. The integrations

$$\int_{\mathcal{C}\cap\Gamma} r_\Gamma\,d\Gamma \quad \text{and} \quad \int_{\mathcal{C}\cap\Gamma} d\Gamma,$$

are performed with the geometric integration of Min and Gibou [62]. These approximations define the coefficients in the linear system associated with grid node c. This linear system is symmetric positive definite. We include an illustrative example from [110], who considered the system (11) on $\Omega = [-1,1]^2$. The exact solution is defined as $u = e^{xy}$ and the irregular domain defined by $\phi = r - 0.5 - \dfrac{y^5+5x^4y-10x^2y^3}{3r^5}$. The Robin boundary condition in (11) has a coefficient $\alpha = 1$. Figure 13 depicts the solution (left) and its accuracy in the L^∞-norm (right). As in the previous sections, the solution of (11) in Ω^- is decoupled from the solution in Ω^+.

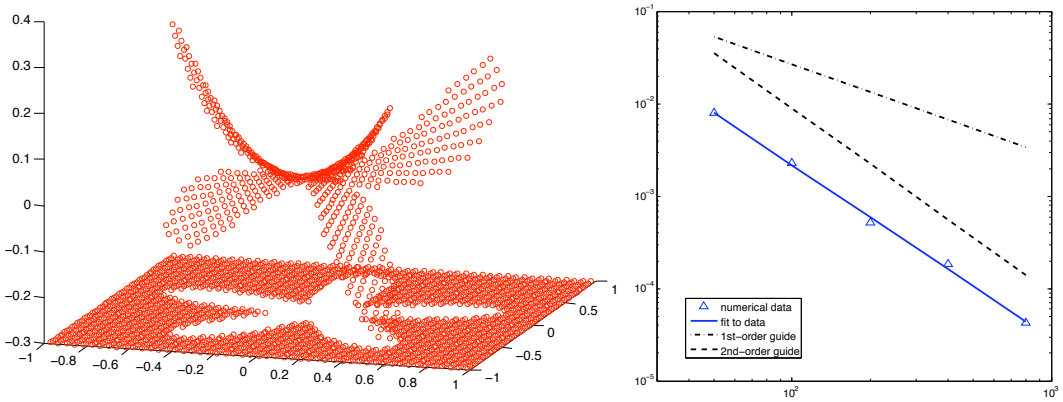


Figure 13: *Solution of section 5.3 (right) and its accuracy (left) in the L^∞-norm. From [110].*

Notes:

- The case where $\alpha < 0$ or where different boundary conditions are imposed on different parts of the interface is more involved and we refer the interested reader to methodology introduced in [122].
- The approximation $\int_{\mathcal{C}\cap\Gamma} \alpha\,\beta\,u\,d\Gamma \approx \alpha\,\beta_c\,u_c \int_{\mathcal{C}\cap\Gamma} d\Gamma$ can be replaced to further use the Robin boundary condition [123] as:

$$\beta_c\,\alpha\,u_\gamma \int_{\mathcal{C}\cap\Gamma} d\Gamma,$$

where u_γ is the orthogonal projection of u_c onto Γ. Taylor's approximation gives:

$$u_c = u_\gamma + |\phi_c| \left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_\gamma + O(|\phi_c|^2),$$

so that one can write the following approximation:

$$u_\gamma \approx \frac{u_c - |\phi_c|\,r_\gamma}{1 - \alpha|\phi_c|},$$

In the case of (near) singularity in this expression, u_c is used instead of u_γ.

- We refer the interested reader to the review by Gibou, Min and Fedkiw [124] for a discussion of common misconceptions when applying Dirichlet boundary conditions.

6. Fast Sweeping Methods

Within the level-set community, there exists two classes of fast numerical methods to solve the Eikonal equation and more generally Hamilton-Jacobi equations: the Fast Marching Method (FMM) [38, 39, 125] and the Fast Sweeping Method (FSM) [40]. We also note that a new approach based on the Hopf-Lax formula is being developed (see section 3.4). We focus here on the FSM, including its parallel version and refer the interested reader to [126] for a parallel implementation of the FMM.

6.1. Sequential FSM

The Eikonal equation defined on a computational domain Ω reads:

$$\begin{aligned} |\nabla u(\mathbf{x})| &= f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^n, \\ u(\mathbf{x}) &= g_\Gamma(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma \subset \Omega, \end{aligned} \tag{12}$$

where $f(\mathbf{x})$ is a given function, and $g_\Gamma(\mathbf{x})$ is the boundary value prescribed on the co-dimension one boundary Γ.

The idea behind the FSM, introduced by Zhao [40], is to initialize the procedure by assigning exact (or interpolated [127]) values of g_Γ at the grid points nearest the interface Γ, and large positive values at all other grid nodes, and then to use a succession of sweeps to propagate the information from Γ to the rest of the domain. At each grid point during these sweeps, a Godunov upwind differencing approximation [128] of (12) is used:

$$[(u_{i,j}^{new} - u_{xmin})^+]^2 + [(u_{i,j}^{new} - u_{ymin})^+]^2 = f_{i,j}^2 \Delta x^2, \tag{13}$$

where Δx is the grid spacing and one uses:

$$u_{xmin} = \min(u_{i-1,j}, u_{i+1,j}) \qquad \text{and} \qquad (x)^+ = \begin{cases} x & x > 0, \\ 0 & x \le 0. \end{cases}$$

Figure 14 (left) illustrates a snapshot of a standard sweeping, at the end of which the data on Γ (here a single point) will be propagated to the top right quadrant. Zhao showed that this procedure will converge in a finite number of sweep (2^n sweeps in $\mathbb{R}^n$), producing a method that is $O(N)$, where N is the total number of grid points [40]. In the case where obstacles are present, for example describing regions of space where propagation is prohibited, the sweeping process must be further iterated to reach convergence.

6.2. Parallel FSM

In [129], Zhao introduced a parallel implementation of the FSM method on shared memory machines by assigning each ordering to a separate thread and synchronizing the data at each grid point by taking the minimum value of all threads at the end of each sweep. In addition, a domain decomposition strategy is used to take advantage of an arbitrary number of threads.

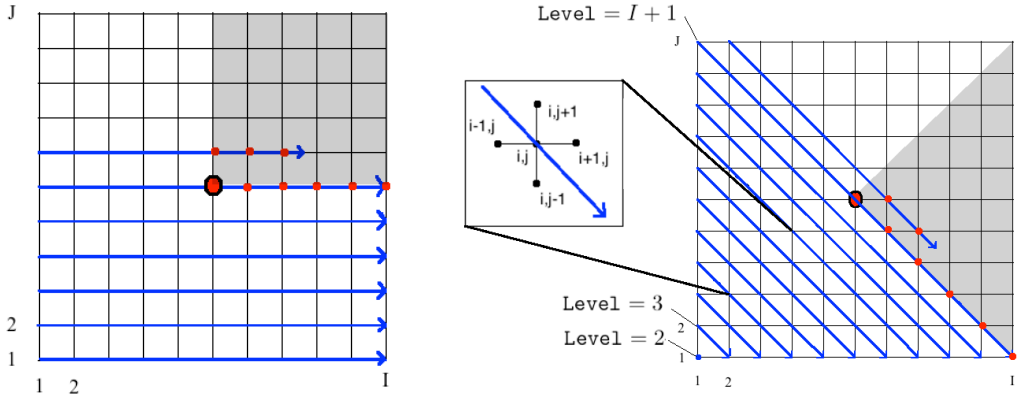


Figure 14: *Fast Sweeping in serial (left) and in parallel (right), where the data from* Γ *(the point in the center) will propagate to the entire shaded region. The inset demonstrates that all the nodes along the same level can be processed independently.*

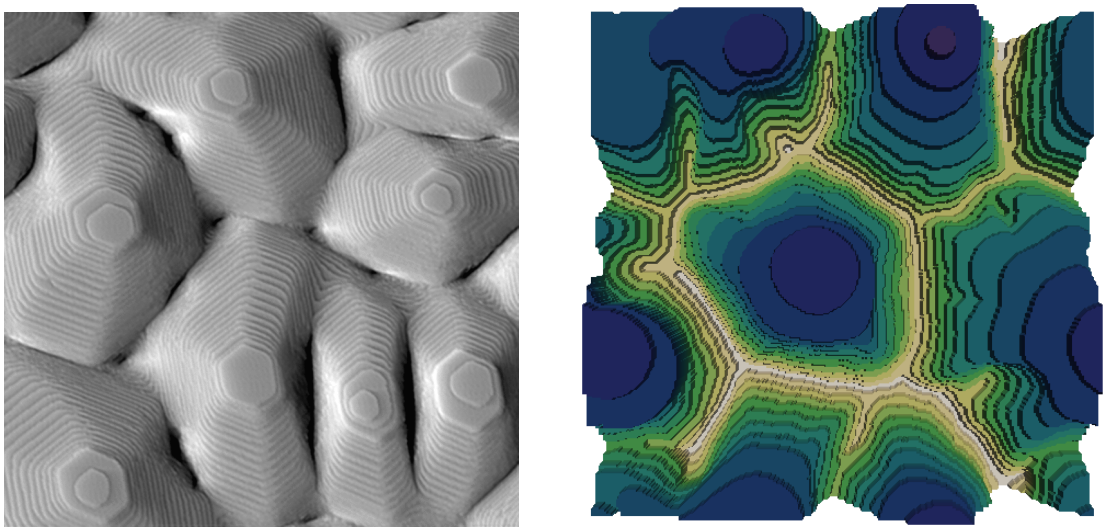
In [130], Detrixhe *et al.* proposed to consider sweeping directions that are not those used in the serial version of FSM. Defining the `level` as the set of grid points `(i,j)` such that `i+j=level`, the sweeps are performed along the 'diagonals' defined by `level=constant` (see figure 14). Since the stencil of (13) at each grid point `(i,j)` does not involve any other grid point on that `level`, all the grid points that belong to the same `level` can be processed independently; the source of parallelism.

In [131], Detrixhe *et al.* combined the domain decomposition technique of [129] with the parallel approach of [130] to propose a hybrid parallelization of the FSM. In addition, this approach was extended to the more general class of static Hamilton-Jacobi equations:

$$\begin{aligned} H(\nabla u(\mathbf{x}), \mathbf{x}, u(\mathbf{x})) &= 1 \quad \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^n, \\ u(\mathbf{x}) &= g(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma \subset \Omega. \end{aligned}$$

7. Some Recent Applications

7.1. Epitaxial growth and the Ehrlich-Schwoebel Barrier



(a) Scanning Tunneling Microscope image from [132]

(b) Simulation of [123]

Figure 15: *Mound formation of epitaxially grown thin film.*

Epitaxial growth is a process where a material is deposited on top of another. It is crucial in the fabrication of most modern opto-electronic devices such as lasers and in the fabrication of devices used for memory storage and optical coherence tomography. Other important applications include catalysts used in the energy sector, food processing, biology and environmental science. Deposited atom diffuse on the substrate they are deposited on and eventually nucleate stochastically, forming dimmers that are mostly stable. Other diffusing atoms further aggregate to form islands, producing a layer by layer growth.

A level-set approach to the simulation of epitaxial growth, called the Island Dynamics Model (IDM), was introduced in [133] and then further refined in [134–139]. This model treats the evolution of islands as a free boundary problem where each boundary moves with a normal velocity proportional to the mass flux towards that boundary, reminiscent to the Burton Cabrera Frank model [140]:

$$\boldsymbol{v} = \left(D^- \frac{\partial \rho^-}{\partial \boldsymbol{n}} - D^+ \frac{\partial \rho^+}{\partial \boldsymbol{n}} \right) \boldsymbol{n} \ , \tag{14}$$

where the $^+$ and $^-$ signs refer to the the upper and lower terraces, respectively.

The IDM considers the density of adatoms (diffusing atoms), $\rho(\mathbf{x}, t)$, as a continuous variable and describes its dynamics by conservation law:

$$\frac{\partial \rho}{\partial t} = F + \nabla \cdot (D \nabla \rho) - 2 \frac{dN}{dt} \ , \tag{15}$$

which accounts for the deposition flux F, the adatom diffusion with diffusion coefficient D, and the nucleation of islands with rate dN/dt where $N(t)$ is the island density. This rate is given by:

$$\frac{dN}{dt} = \sigma_1 D \langle \rho^2(\mathbf{x}) \rangle \ , \tag{16}$$

where σ_1 is a capture number [141, 142] and $\langle \cdot \rangle$ denotes the average taken over the entire domain. The growth of island is thus discrete in the vertical direction but continuous in the lateral direction.

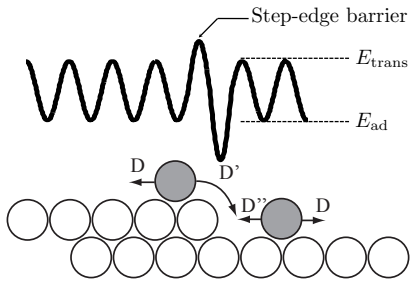
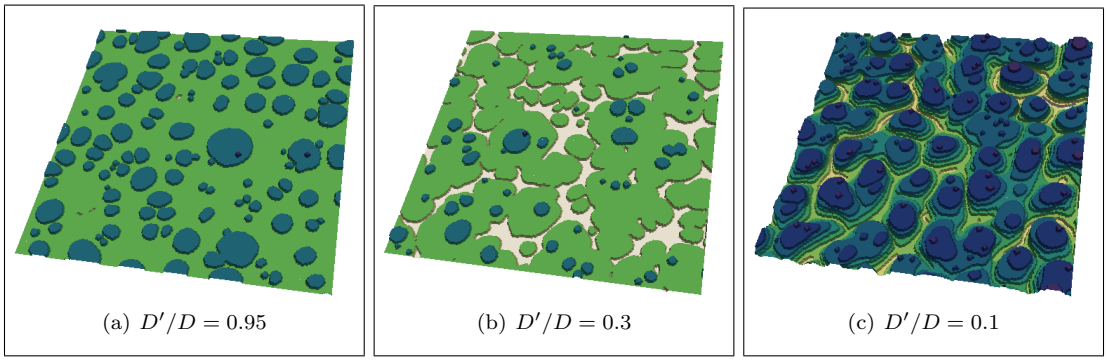


Figure 16: *Schematic of an asymmetric (ES) step-edge barrier. The lower panel shows the atomistic configuration of a step edge, to which an adatom attaches with rate D' (D'') from the upper (lower) terrace. The upper panel depicts the corresponding energy landscape in one spatial dimension.*

The boundary conditions for the adatom density expresses the effects of the Ehrlich-Schwoebel barrier through a Robin boundary condition [143–145] (see Figure 16):

$$\nabla\rho \cdot \boldsymbol{n} + \frac{D'}{D - D'}\rho = \frac{D'}{D - D'}\rho_{\text{eq}} \ , \tag{17}$$

where D' describes the energy barrier for adatom diffusion over the island boundary, ρ_{eq} is an equilibrium adatom density, and $\boldsymbol{n}$ is the outward normal to the islands' boundary. When $D' = D$, the boundary condition is replaced by the Dirichlet boundary condition $\rho = \rho_{\text{eq}}$ [140, 146, 147]. Otherwise, the smaller D', the stronger the Ehrlich-Schwoebel barrier, which translate into the formation of mounds (see figure 15(a)). The need to impose Robin boundary conditions is clear in this case. In addition, adaptive grids and parallel computing are desired to address the need for resolving the adatom density on small terraces while simulating meaningful physical domain. Figure 15 depicts the results from [123] of a multilayer growth using the adaptive parallel framework of section 4.4, with the treatment of the Ehrlich-Schwoebel barrier with the method outlined in section 5.3. The simulation uses a small ratio $D'/D = 0.1$, which promotes the formation of mounds. The results shown in figure 15(b) are similar to what is experimentally observed (see figure15(a)). Figure 17 further depicts the effect of increasing the ratio D'/D that results in less and less pronounced mounding.



(a) $D'/D = 0.95$ (b) $D'/D = 0.3$ (c) $D'/D = 0.1$

Figure 17: *From left to right: effects of increasing the Ehrlich-Schwoebel barrier (i.e. decreasing D'/D). This simulation (from [123]) uses a (5, 8)-Quadtree and 256 processors.*

7.2. Solidification of Binary Alloys

Solidification is an indispensable manufacturing process that enables the production of materials with arbitrary shapes and unique properties. The solidification of multicomponent alloys used to fabricate turbine blades in the aerospace and energy sectors is particularly important. This process

can be modeled by the diffusion of different material components in their liquid phase (region Ω^l) that solidify at the interface of a solid-liquid interface Γ; the solid region is denoted by Ω^s.

In the case of a binary system, conservation laws express the dynamics of the solute's concentration C^l and C^s in the liquid and solid phases, respectively, as well as the temperature, T:

$$\left\{\begin{array}{lcll} \partial T/\partial t & = & k/\rho c \Delta T & , \quad \boldsymbol{x} \in \Omega^s \cup \Omega^l, \\ \partial C^l/\partial t & = & D^l \Delta C^l & , \quad \boldsymbol{x} \in \Omega^l, \\ \partial C^s/\partial t & = & D^s \Delta C^s & , \quad \boldsymbol{x} \in \Omega^s, \end{array}\right. \tag{18}$$

where ρ is the density, c is the heat capacities, D^l and D^s the solutal diffusion coefficients in the liquid and solid phase, respectively, and where k is the thermal conductivity. The temperature is continuous at Γ and satisfies the Gibbs-Thompson condition:

$$T\big|_\Gamma \quad = \quad T_m + m_L C^l\big|_\Gamma + \varepsilon_c \kappa, \tag{19}$$

where T_m is the melting temperature, m_L is the liquidus slope, κ the interface curvature and ε_c a coefficient that captures the crystallography of the system [148]. The solidification front evolves with a normal velocity given by the heat flux balance and the solute-rejection equation:

$$V_\Gamma \quad = \quad \frac{k}{L}\left[\frac{\partial T}{\partial \boldsymbol{n}}\right], \tag{20}$$

$$V_\Gamma \quad = \quad \left[D\frac{\partial C}{\partial \boldsymbol{n}}\right] \Big/ (C^s - C^l), \tag{21}$$

where L is the latent heat. One of the main questions relevant to solidification is the prediction of the growth regimes as a function of the applied temperature gradient G_S and the interface velocity V_Γ [149]. In [71], the adaptive Quadtree framework of section 4 and the boundary treatments of section 5 were used to simulate the solidification of binary alloys and to generate the map of growth regimes as a function of the parameters G_S and V_Γ, as illustrated in figure 18.

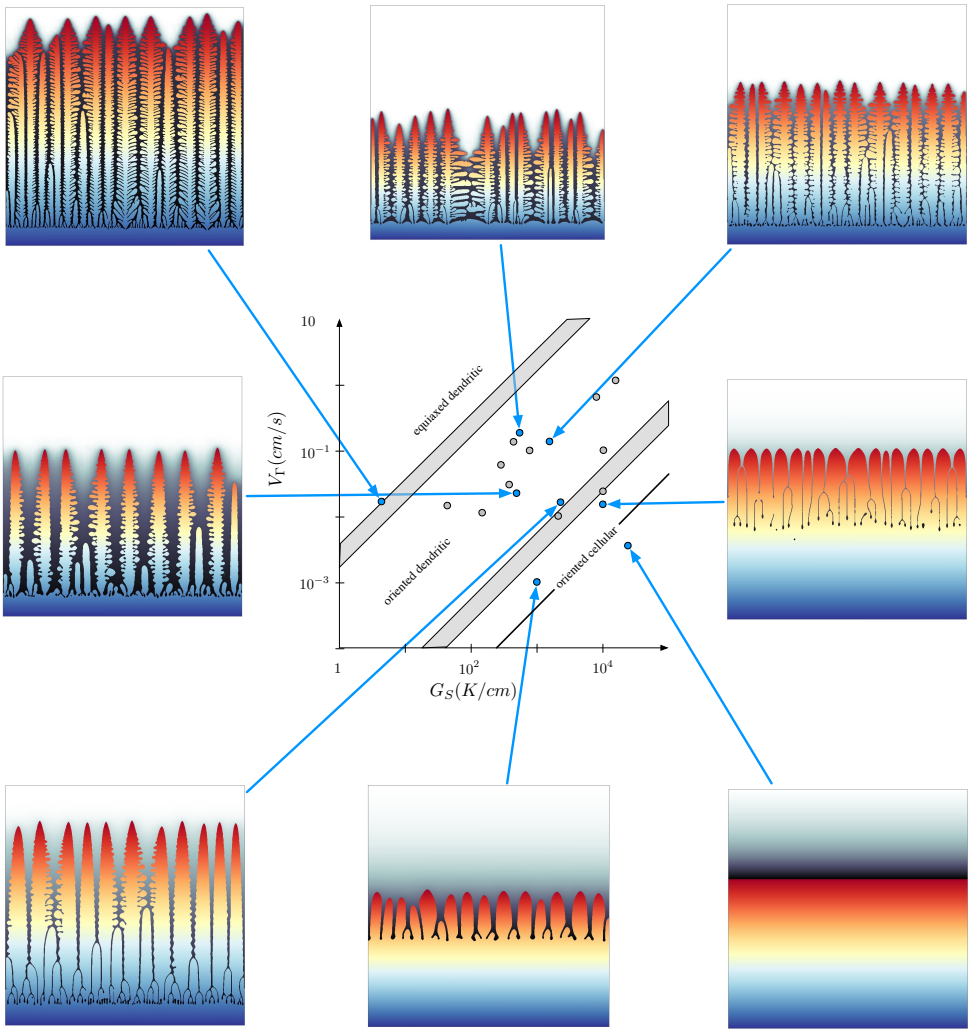


Figure 18: *Resulting crystalline configuration as a function of G_S and V_Γ, illustrating the planar, cellular and dendritic regimes. From [71].*

7.3. Electroporation of Cell Membranes

The efficacy of the treatments of some cancers depends on the proper delivery of therapeutic molecules directly into malignant cells. Electroporation, which is achieved by applying a large electric pulse, is an increase in the permeability of a cell's membrane. In turn, it promotes the diffusion of drugs directly into cancer cells through their permeabilized membrane [150–152]. Electroporation, is a technique that is currently used in the treatment of some skin or pancreatic cancers [153]. Proper treatment relies on the accurate knowledge of the distribution of the electroporated region. Numerical simulations can play a crucial role in that quest and can inform scientist about the effects of cell screening, the macroscopic behavior of the electrical potential in cell aggregates or the validity of transmission models [154–162]. However, one of the main goals of developing a computational approach is to be able to consider con-

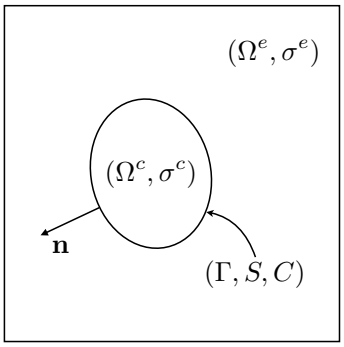


Figure 19: *Electroporation computational domain.*

figurations that are close to experimental ones, i.e. one needs to consider clusters of at least a few thousands to a few tens of thousands cells [163, 164].

As a first step in that direction, Guittet *et al.* [75] used the VIM of [76] described in section 5.1.2 to solve the model of Schwan, Stuchly *et al.* [165, 166]: Consider a domain Ω composed of two subdomains, Ω^c describing the cell cytoplasm and Ω^e describing the extracellular medium (see figure 19). The electrical potential obeys the following boundary value problem:

$$\Delta u = 0, \quad \text{in } (0,T) \times (\Omega^e \cup \Omega^c)\,, \tag{22a}$$

$$u(t,\cdot) = g(t,\cdot) \quad \text{on } (0,+\infty) \times \partial\Omega, \tag{22b}$$

with the jump conditions:

$$[\sigma \partial_{\mathbf{n}} u] = 0, \quad \text{on } (0,T) \times \Gamma, \tag{22c}$$

$$C\partial_t [u](t,\cdot) + S(t,[u])[u] = \sigma \partial_{\mathbf{n}} u(t,\cdot)|_\Gamma\,, \quad \text{on } (0,T) \times \Gamma. \tag{22d}$$

where the conductivity σ has a jump across the cells' membrane Γ. The jump condition (22c) enforces the continuity of the flux while (22d) describes the combined effect of the capacitance, C, and of the surface conductance, S, of the membrane. Dirichlet boundary conditions are given on $\partial\Omega$. In addition to those equations, a model must be provided for the membrane conductance (see e.g. [156, 157, 161]). In [75], the authors focused in particular on the model of Poignard *et al.* [161]:

$$S(t,\lambda) = S_{\mathrm{L}} + S_0 X_0(t,\lambda) + S_1 X_1(t, X_0(t,\lambda)), \tag{23}$$

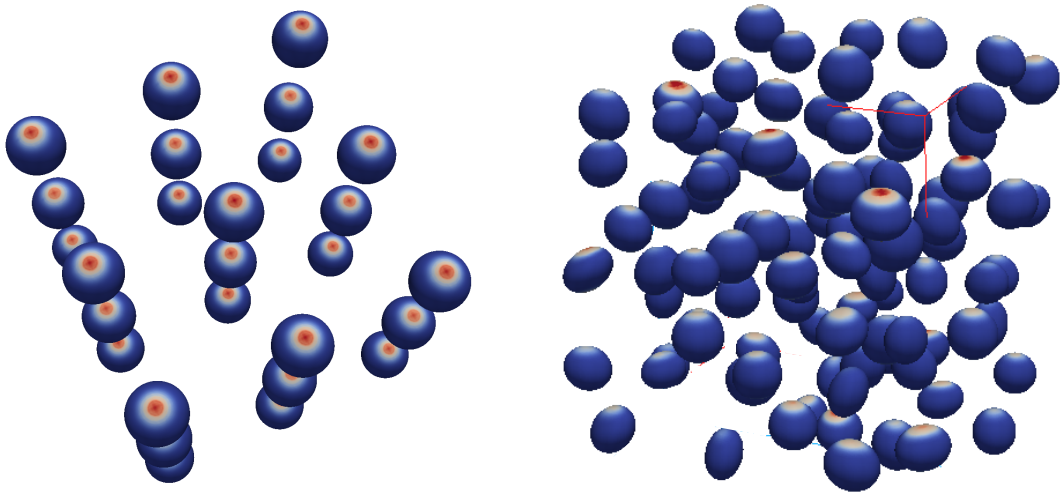
which considers three different states for the surface conductance: resting (S_{L}), porated (S_0) and permeabilized (S_1) states. This model also gives an evolution equation for the degree of poration X_0:

$$\begin{cases} \dfrac{\partial X_0(t,\lambda)}{\partial t} & = \dfrac{\beta_0(\lambda(t)) - X_0}{\tau_{\mathrm{ep}}}, \\ X_0(0,\lambda) & = 0, \end{cases}$$

and another evolution equation for the degree of permeabilization X_1:

$$\begin{cases} \dfrac{\partial X_1(t,X_0)}{\partial t} & = \max\left(\dfrac{\beta_1(X_0) - X_1}{\tau_{\mathrm{perm}}}, \dfrac{\beta_1(X_0) - X_1}{\tau_{\mathrm{res}}}\right), \\ X_1(0,\lambda) & = 0, \end{cases}$$

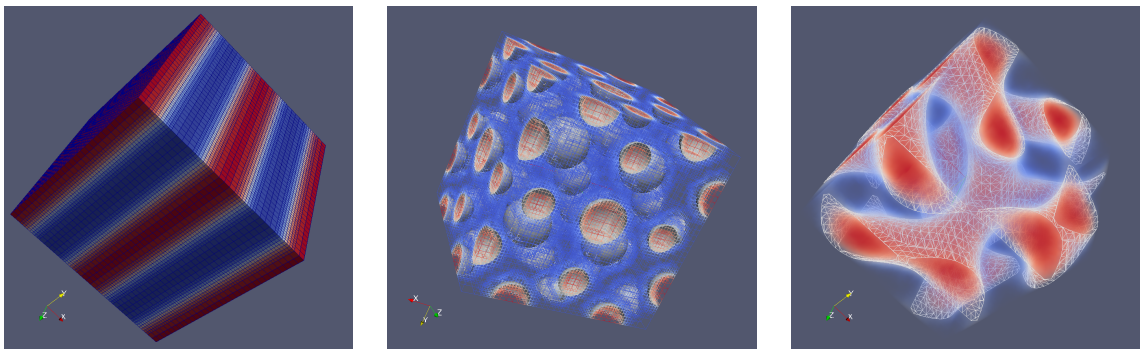
where β_0 and β_1 are given step functions. Figures 20 depicts the results of two simulations from [75]. In particular, this framework is capable of considering many cells and study the effects of shadowing on the permeabilization of cells. This work could potentially be extended to the case of parallel computations, as described in section 4.4, and thus could pave the way to considering experimental condition with clusters of a few tens of thousands cells.



(a) $3 \times 3 \times 3$ array of cells (b) Cluster of 100 cells

Figure 20: *Visualization of the permeabilization X_1 for two different cells' distributions. From [75].*

7.4. Inverse Problem for Directed Self-Assembly



(a) Lamellar phase (b) BCC phase (c) Gyroid phase

Figure 21: *Simulation from [72] using a self-consistent field theoretic solver on adaptive grids.*

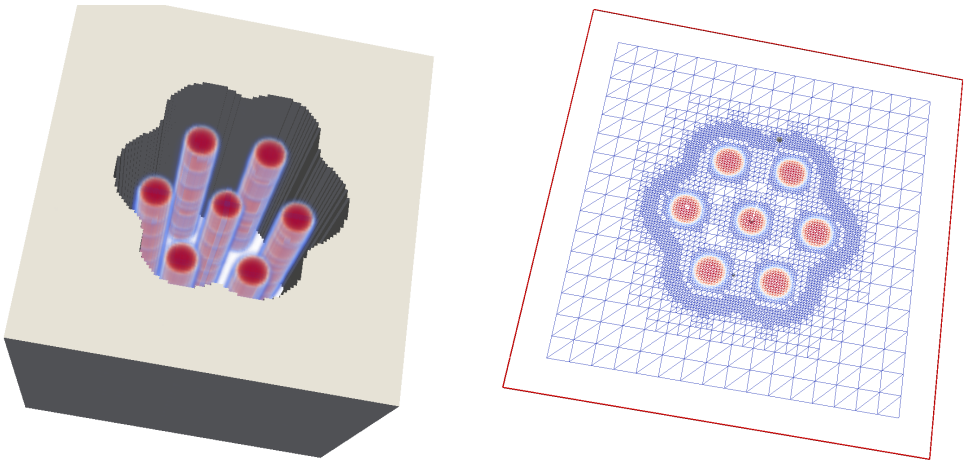
Diblock copolymers are melts of molecular chains A and B that self-assemble into ordered structures. They are therefore extremely useful in generating geometries with a length scale (~ 5 to 100 nm) that is not accessible with other processes. Applications can be found in the electronics sector as well as in the energy and in the health industries [167]. One application of particular interest is in the context of Directed Self-Assembly (DSA), which is a patterning technique for next generation lithography [168, 169]. In this context, the challenge is to find a template shape that will direct the self-assembly into cylindrical structures with a targeted topology.

The level-set method is a powerful approach to solve inverse problems where the optimum geometry is the targeted outcome. It has been used for example for shape optimization in the context of acoustic, fluids and structural systems among others [46, 170–175]. Recently, Ouaknin *et al.* introduced a level-set approach to solve the inverse problem associated with DSA [176]: given a target for the position and radius of cylindrical structures, what is the optimum mask that will drive the self-assembly to that target. In this work, the adaptive framework of section 4 is used to solve the equation of the self-consistent field theory, an accurate model that describes how inhomogeneous phases self-assemble [177–179].

Within the SCFT framework, the densities of the A and B components are found through the solution of chain propagators q and $q^\dagger$, which represent the statistical weight of a polymer chain at a given location, $\boldsymbol{r}$, and contour length, s, and which satisfy the Fokker-Planck equations:

$$\begin{cases} \partial_s q(s,\boldsymbol{r}) = \bigtriangledown^2 q(s,\boldsymbol{r}) - q(s,\boldsymbol{r}) \times w(\boldsymbol{r}) & \text{forward,} \\ \partial_s q^\dagger(s,\boldsymbol{r}) = \bigtriangledown^2 q^\dagger(s,\boldsymbol{r}) - q^\dagger(s,\boldsymbol{r}) \times w^\dagger(\boldsymbol{r}) & \text{backward.} \end{cases}$$

In these equations, w and $w^\dagger$ are respectively related to the pressure potential that enforces the incompressibility constraint and to the exchange potential describing the interaction between A and B. In the case of an infinite domain, the boundary conditions for q and $q^\dagger$ are periodic whereas for a confined domain, homogeneous Neumann conditions are used [180]. Examples of structures obtained for a periodic domain within this framework are illustrated in figure 21 while figure 22 depicts the results of self-assembly in an irregular confined domain.



(a) 3D confined domain. (b) Cross-section.

Figure 22: *Simulation from [72] of self-assembly of block-copolymer in a confined domain using a self-consistent field theoretic solver on adaptive grids.*

The inverse problem associated with DSA can be solved by considering a level-set function ϕ that evolves in such a way as to minimize the following functional:

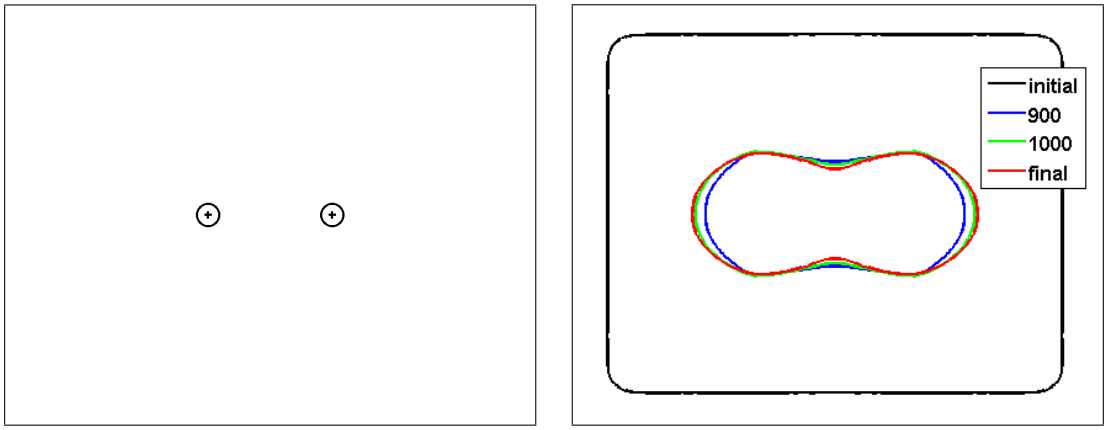
$$J[\phi(\boldsymbol{r}), w_+(\boldsymbol{r}), w_-(\boldsymbol{r})] = \int_{\Omega_-} \frac{1}{2} \left(\rho_A(\boldsymbol{r}) - \rho_A^{\text{target}}(\boldsymbol{r})\right)^2 d\Omega_-,$$

where ρ_A^{target} represents the targeted density of species A and ρ_A represents the density of species A responding to the confined domain enclosed in $\Omega^- = \{\boldsymbol{x} : \phi(\boldsymbol{x}) \leq 0\}$. In addition to depending on the level-set function, the functional J depends on the exchange potential w^- and the pressure potential w^+. In [181], Ouaknin *et al.* derived the functional derivative $\dfrac{\delta J}{\delta \phi(\boldsymbol{r})}$, which can then be used to define the velocity at which ϕ must evolve in order to minimize the functional J. In [176], the authors used instead the normal velocity defined as $\boldsymbol{v}(\boldsymbol{r}) = w_+(\boldsymbol{r})\,\boldsymbol{n}$, emulating a shape that responds to the pressure field w^+.

In this context, it is thus particularly important to solve the SCFT equations in confined domains in a way that w^+ is properly defined near the evolving boundary. Confined domains can be emulated on a periodic domain (and thus by using a periodic boundary condition) by using a compressible assumption. In this case, the walls of the confined domain are given a width δ_w, across which the solution varies rapidly. As a consequence, the value for w^+ near the wall follows a steep profile

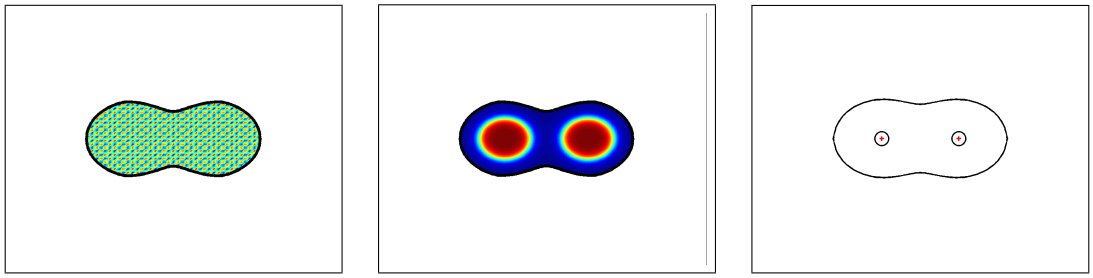
that sharpens as the value of δ_w decreases. This solution process thus imposes an arbitrary density profile in a region of length δ_w, which in turn make the computation of the shape optimization velocity $\boldsymbol{v} = w^+ \boldsymbol{n}$ difficult. On the other hand, using a sharp representation of the confined domain and imposing sharp boundary condition as described in section 5.3, provides the correct profile (as demonstrated in [72]) that one can exploits to define accurately the shape optimization velocity.

Figure 23(b) gives an example of the evolution of ϕ until it converges to the optimum mask given the target densities of figure 23(a) (only the two cylinders' center are depicted - the radius of the cylinders are also given as input). This optimum mask used as a confined domain drives the self-assembly of a random melt to the target design, as illustrated in figure 24. In particular, figure 24(c) demonstrates that the centers of the self-assembly process are consistent with the centers of the targeted cylinders.



(a) Target design. (b) ϕ at different iteration of shape optimization.

Figure 23: *A result of the shape optimization algorithm (From [176]).*



(a) Initial random melt. (b) Resulting self-assembly. (c) Centers of target (circles) .vs. self-assembly (pluses).

Figure 24: *Example showing a SCFT simulation from a random melt using the mask of figure 23. From [176].*

7.5. Parkinson's Disease and Isochrons

A recent application of the parallel fast sweeping method of section 6.2 is in the context of Deep Brain Stimulation (DBS), a remarkably effective treatment to relieve the tremor, rigidity, and bradykinesia of Parkinson's disease (PD) in many individuals. The exact mechanism of action of DBS is unknown, though it is likely that it disrupts ongoing pathological neuronal activity, and results as well in neurotransmitter release. However, there is no consensus on a systematic

or theoretical basis by which to optimize the many adjustable DBS parameters such as energy levels, stimulus rate, waveform shape and duration [182, 183] and it is almost certainly the case that improving the stimulus protocol will result in better symptom control and in reduced side effects. Skeletal motor activity is controlled by a series of pathways in the basal ganglia and cortex that include the globus pallidus, subthalamic nucleus, and striatum, and it is well-accepted that abnormal firing in this circuit is at the core of many of the symptoms of PD. Tremors associated with Parkinson's disease stem from abnormal electrical activity in this circuit, and it is believe that desynchronizing the neuronal firing results in treatment of this symptom.

The concept of isochrons, which are sets of points with trajectories that converge with the same phase on a periodic orbit, is powerful in this context [184]. Models based on isochrons can be used to design an energy-optimal stimulus that drives the states of the neurons within the brain to a region of phase space where the isochrons are clustered tightly (the phaseless set), thereby desynchronizing neurons activity.

In [185], Detrixhe *et al.* considered the four dimensional Hodgkin-Huxley neuron's model of the form:

$$\frac{d\mathbf{x}}{dt} = \boldsymbol{F}(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^4, \tag{24}$$

where the states variable $\boldsymbol{x}$ stores the transmembrane voltage potential and the three gating variables describing the physiology of the neuron, and $\boldsymbol{F}$ describes their dynamics [186]. Considering the asymptotic phase function $\theta(\mathbf{x})$ that takes values in $[0, 2\pi)$, the associated isochrons were computed by solving the corresponding Eikonal equations:

$$\nabla\theta \cdot \mathbf{F}(\mathbf{x}) = \frac{2\pi}{T},$$

with the parallel FSM of [130] (see section 6.2) and by considering its isocontours.

Figure 25 depicts a 3D cross-section of the four-dimensional isochrons computed on a grid with 160 points in each spatial dimensions. In particular, one can easily compute the almost phaseless set by finding the region in space where $|\nabla\theta| > \tau$, where τ is a chosen threshold. This knowledge can be used to design control strategies in the context of DBS.

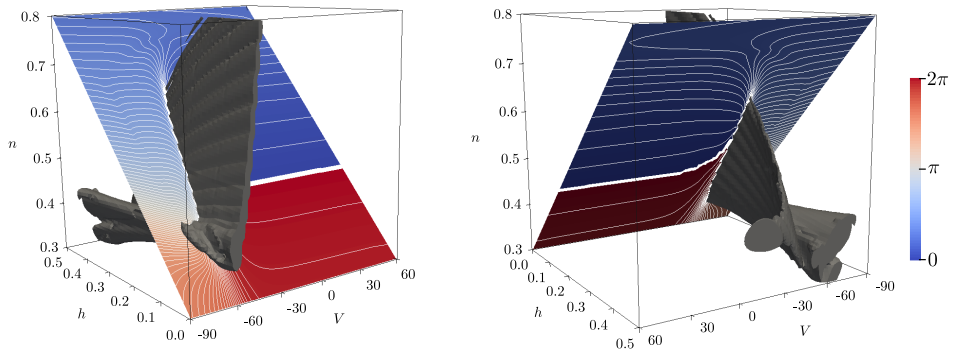


Figure 25: *3D cross-sections of the almost phaseless set (gray) of the four dimensional Hodgkin-Huxley model. The phaseless set is defined by* $|\nabla\theta| > 30$. *The diagonal 2D slices depicts 100 equally spaced isochrons. From [185].*

8. Conclusion

Level-set and fast sweeping methods are powerful numerical methods that have been applied to a wide range of applications. In that review, we have discussed some recent progress related to the numerical treatments of boundary conditions, adaptive grids and parallel strategies in the level-set

framework. We have illustrated some of the numerical treatments with some recent applications in materials science and neuroscience. It goes without saying that recent progress and applications are too numerous to be included in this review, and we have chosen to discuss mainly those closest to us.

Acknowledgment

The work of Frederic Gibou was supported by ONR N00014-11-1-0027, ARO W911NF-16-1-0136, NSF DMS 1620471, NSF DMS-1412695 and by the DMREF program under DMR 1534264. The work of Ron Fedkiw was supported by ONR N00014-13-1-0346, ONR N00014-17-1-2174, ARL AHPCRC W911NF-07-0027 and NSF CNS1409847. The work of Stanley Osher was supported by ONR N000141410683, N000141210838, N000141712162 and DOE DE-SC00183838.

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

[2] D Juric and G Tryggvason. A Front Tracking Method for Dendritic Solidification. *J. Comput. Phys*, 123:127–148, 1996.

[3] D Juric and G Tryggvason. Computations of Boiling Flows. *Int. J. Multiphase. Flow.*, 24:387–410, 1998.

[4] J. Qian, G. Tryggvason, and C.K. Law. A Front Tracking Method for the Motion of Premixed Flames. *Journal of Computational Physics*, 144(1):52–69, July 1998.

[5] G Tryggvason, B Bunner, A Esmaeeli, D Juric, N Al-Rawahi, W Tauber, J Han, S Nas, and Y.-J. Jan. A Front-Tracking Method for the Computations of Multiphase Flow. *J. Comput. Phys.*, 169:708–759, 2001.

[6] S O Unverdi and G Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100:25–37, 1992.

[7] Wurigen Bo, Xingtao Liu, James Glimm, and Xiaolin Li. A robust front tracking method: Verification and application to simulation of the primary breakup of a liquid jet. *SIAM J. Sci. Comput.*, 33(4):1505–1524, 2011.

[8] Eugenio Aulisa, Sandro Manservisi, Ruben Scardovelli, and Stephane Zaleski. A geometrical area-preserving volume-of-fluid advection method. *Journal of Computational Physics*, 192(1):355 – 364, 2003.

[9] D Benson. Volume of Fluid Interface Reconstruction Methods for Multimaterial Problems. *Applied Mechanics Reviews*, 52:151–165, 2002.

[10] D Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput. Meth. in Appl. Mech. and Eng.*, 99:235–394, 1992.

[11] R. DeBar. Fundamentals of the KRAKEN code. Technical report, Lawrence Livermore National Laboratory (UCID- 17366), 1974.

[12] V Dyadechko and M Shashkov. Moment-of-Fluid Interface Reconstruction. Technical report, Los Alamos National Laboratory (LA-UR-05-7571), 2006.

[13] C Hirt and B Nichols. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *J. Comput. Phys.*, 39:201–225, 1981.

[14] W Noh and P Woodward. SLIC (simple line interface calculation). In *5th International Conference on Numerical Methods in Fluid Dynamics*, pages 330–340, 1976.

[15] Yuriko Renardy and Michael Renardy. Prost: A parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics*, 183(2):400 – 421, 2002.

[16] Mark Sussman and Elbridge Gerry Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301 – 337, 2000.

[17] Zhaoyuan Wang, Jianming Yang, and Frederick Stern. A new volume-of-fluid method with a constructed distance function on general structured grids. *Journal of Computational Physics*, 231(9):3703 – 3722, 2012.

[18] Xiaofeng Yang, Ashley J. James, John Lowengrub, Xiaoming Zheng, and Vittorio Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217(2):364 – 394, 2006.

[19] D Youngs. An interface tracking method for a 3D Eulerian hydrodynamics code. Technical report, AWRE (44/92/35), 1984.

[20] Stéphane Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228(16):5838–5866, September 2009.

[21] R Braun and M Murray. Adaptive Phase-Field Computations of Dendritic Crystal Growth. *J. Cryst Growth.*, 174:41, 1997.

[22] K Elder, M Grant, N Provatas, and J Kosterlitz. Sharp interface limits of phase-field models. *SIAM J. Appl. Math*, 64:21604, 2001.

[23] J.-H. Jeong, N Goldenfeld, and J Dantzig. Phase Field Model for Three-Dimensional Dendritic Growth with Fluid Flow. *Phys. Rev. E*, 64:41602, 2001.

[24] A Karma and W.-J Rappel. Phase-Field Modeling Method for Computationally Efficient Modeling of Solidification with Arbitrary Interface Kinetics. *Phys. Rev. E*, 53, 1996.

[25] A Karma and W.-J Rappel. Quantitative Phase-Field Modeling of Dendritic Growth in Two and Three Dimensions. *Phys. Rev. E*, 57:4323–4349, 1997.

[26] A Karma. Phase-Field Formulation for Quantitative Modeling of Alloy Solidification. *Phys. Rev. Lett.*, 87:115701, 2001.

[27] B Nestler, D Danilov, and P Galenko. Crystal growth of pure substances: Phase-field simulations in comparison with analytical and experimental results. *J. Comput. Phys.*, 207:221–239, 2005.

[28] N Provatas, N Goldenfeld, and J Dantzig. Efficient Computation of Dendritic Microstructures using Adaptive Mesh Refinement. *Phys. Rev. Lett.*, 80:3308, 1998.

[29] N Provatas, N Goldenfeld, and J Dantzig. Adaptive Mesh Refinement Computation of Solidification Microstructure using Dynamic Data Structures. *J. Comput. Phys*, 148:265, 1999.

[30] M Sussman, P Smereka, and S Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114:146–159, 1994.

[31] G.-S. Jiang and D Peng. Weighted ENO Schemes for Hamilton-Jacobi Equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.

[32] G.-S. Jiang and C.-W. Shu. Efficient Implementation of Weighted ENO Schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[33] X.-D. Liu, S Osher, and T Chan. Weighted Essentially Non-Oscillatory Schemes. *J. Comput. Phys.*, 126:202–212, 1996.

[34] C.-W. Shu and S Osher. Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II. *J. Comput. Phys.*, 83:32–78, 1989.

[35] Chi-Wang Shu. Efficient Implementation of Essentially Non-oscillatory Schemes , II. *Journal of Computational Physics*, 78:32–78, 1989.

[36] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.

[37] A. du Chene, C. Min, and F. Gibou. Second-order accurate computation of interface curvature in a level set framework. *J. Sci. Computing.*, 35:114–131, 2008.

[38] J Tsitsiklis. Efficient Algorithms for Globally Optimal Trajectories. *IEEE Trans. on Automatic Control*, 40:1528–1538, 1995.

[39] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.*, 93:1591–1595, 1996.

[40] Hongkai Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74:603–627, 2004.

[41] Yen-Hsi Richard Tsai. Rapid and accurate computation of the distance function using grids. *J. Comp. Phys.*, 178(1):175–195, 2002.

[42] L-T. Cheng and Y.-H. Tsai. Redistancing by flow of time dependent eikonal equation redistancing by flow of time dependent eikonal equation redistancing by flow of time dependent eikonal equation redistancing by flow of time dependent Eikonal equation. *J. Comp. Phys.*, pages 4002–4017, 2008.

[43] Y.-H. Tsai, L.-T. Cheng, and S Osher. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 41(2):673–694, 2003.

[44] J Helmsen, E G Puckett, P Colella, and M Dorr. Two new methods for simulating photolithography development in 3D. In *Proceedings of the SPIE - The International Society for Optical Engineering Optical Microlithography IX, Santa Clara, CA, USA*, volume 13-15, pages 253–261.

[45] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.

[46] J. A. Sethian. *Level set methods and fast marching methods*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, second edition, 1999. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.

[47] Jerome Darbon and Stanley Osher. Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere. *Res Math Sci*, 3:19, 2016.

[48] Byungjoon Lee, Jerome Darbon, Stanley Osher, and Myungjoo Kang. Revisiting the redistancing problem using the Hopf–Lax formula. *J. Comput. Phys.*, 330:268–281, 2017.

[49] Michael Royston, Andre Pradhana, Byungjoon Lee, Yat Tin Chow, Wotao Yin, Joseph Teran, and Stanley Osher. Parallel redistancing using the Hopf-Lax formula. *In Preparation*, 2017.

[50] Y-T Chow, S Osher, J Darbon, and W Yin. Algorithms for overcoming the curse of dimensionality for time dependent nonconvex Hamilton-Jacobi equations arising from control and differential games problems. *UCLA CAM report 16-62*, 2016.

[51] Y-T Chow, S Osher, J Darbon, and W Yin. Algorithms for overcoming the curse of dimensionality for certain nonconvex Hamilton-Jacobi equations, projections and differential games. *UCLA CAM report 16-27*, 2016.

[52] Nathaniel R. Morgan and Jacob I. Waltz. 3d level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement. *Journal of Computational Physics*, 336:492 – 512, 2017.

[53] R. Abgrall, H. Beaugendre, and C. Dobrzynski. An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, 257, Part A:83 – 101, 2014.

[54] M Berger and J Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[55] M Berger and P Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[56] M Sussman, A Almgren, J Bell, P Colella, L Howell, and M Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148:81–124, 1999.

[57] P McCorquodale, P Colella, D Grote, and J.-L. Vay. A Node-Centered Local Refinement Algorithm for Poisson's Equation in Complex Geometries. *J. Comput. Phys.*, 201:34–60, 2004.

[58] J Strain. A fast modular semi-Lagrangian method for moving interfaces. *J. Comput. Phys.*, 161:512–536, 2000.

[59] Chohong Min, Frederic Gibou, and Hector Ceniceros. A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids. *Journal of Computational Physics*, 218(1):123–140, October 2006.

[60] Chohong Min and Frederic Gibou. A second order accurate projection method for the incompressible Navier-Stokes equations on non-graded adaptive grids. *Journal of Computational Physics*, 219(2):912–929, December 2006.

[61] Han Chen, Chohong Min, and Frederic Gibou. A Supra-Convergent Finite Difference Scheme for the Poisson and Heat Equations on Irregular Domains and Non-Graded Adaptive Cartesian Grids. *Journal of Scientific Computing*, 31(1-2):19–60, March 2007.

[62] Chohong Min and Frederic Gibou. Geometric integration over irregular domains with application to level-set methods. *Journal of Computational Physics*, 226(2):1432–1443, October 2007.

[63] Chohong Min and Frederic Gibou. Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions. *Journal of Computational Physics*, 227(22):9686–9695, November 2008.

[64] Han Chen, Chohong Min, and Frederic Gibou. A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate. *J. Comput. Phys.*, 228(16):5803–5818, 2009.

[65] M. Theillard, C. H. Rycroft, and F. Gibou. A multigrid method on non-graded adaptive octree and quadtree cartesian grids. *J. Scientific Comput.*, 55:1–15, 2012.

[66] Maxime Theillard, Landry Fokoua Djodom, Jean-Léopold Vié, and Frédéric Gibou. A second-order sharp numerical method for solving the linear elasticity equations on irregular domains and adaptive grids – application to shape optimization. *Journal of Computational Physics*, 233:430 – 448, 2013.

[67] Joseph Papac, Asdis Helgadottir, Christian Ratsch, and Frederic Gibou. A level set approach for diffusion and stefan-type problems with Robin boundary conditions on quadtree/octree adaptive cartesian grids. *J. Comput. Phys.*, 233:241, 2013.

[68] M. Mirzadeh, M. Theillard, A. Helgadottir, D. Boy, and F. Gibou. An adaptive, finite difference solver for the nonlinear Poisson-Boltzmann equation with applications to biomolecular computations. *Communications in Computational Physics*, 13(1):150–173, 2012.

[69] Ásdís Helgadóttir and Frederic Gibou. A Poisson-Boltzmann solver on irregular domains with Neumann or Robin boundary conditions on non-graded adaptive grid. *Journal of Computational Physics*, 230(10):3830–3848, May 2011.

[70] M. Mirzadeh and F. Gibou. A conservative discretization of the Poisson–Nernst–Planck equations on adaptive cartesian grids. *J. Comput. Phys.*, 274:633–653, 2014.

[71] M. Theillard, F. Gibou, and T. Pollock. A sharp computational method for the simulation of the solidification of binary alloys. *J. Sci. Comput.*, 63:330–354, 2014.

[72] Gaddiel Ouaknin, Nabil Laachi, Kris Delaney, Glenn H. Fredrickson, and Frederic Gibou. Self-consistent field theory simulations of polymers on arbitrary domains. *Journal of Computational Physics*, 327:168 – 185, 2016.

[73] Mohammad Mirzadeh, Maxime Theillard, and Frédéric Gibou. A second-order discretization of the nonlinear Poisson–Boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids. *Journal of Computational Physics*, 230(5):2125 – 2140, 2011.

[74] Arthur Guittet, Maxime Theillard, and Frédéric Gibou. A stable projection method for the incompressible navier–stokes equations on arbitrary geometries and adaptive quad/octrees. *Journal of Computational Physics*, 292:215 – 238, 2015.

[75] Arthur Guittet, Clair Poignard, and Frederic Gibou. A voronoi interface approach to cell aggregate electropermeabilization. *Journal of Computational Physics*, 332:143 – 159, 2017.

[76] Arthur Guittet, Mathieu Lepilliez, Sebastien Tanguy, and Frédéric Gibou. Solving elliptic problems with discontinuities on irregular domains – the voronoi interface method. *Journal of Computational Physics*, 298:747 – 765, 2015.

[77] Emmanuel Brun, Arthur Guittet, and Frederic Gibou. A local level-set method using a hash table data structure. *J. Comp. Phys.*, 231:2528–2536, 2012.

[78] M J Aftosmis, M J Berger, and J E Melton. Adaptive Cartesian mesh generation. In *CRC Handbook of Mesh Generation (Contributed Chapter)*, 1998.

[79] S Popinet. Gerris: A Tree-Based Adaptive Solver for the Incompressible Euler Equations in Complex Geometries. *J. Comput. Phys.*, 190:572–600, 2003.

[80] Frank Losasso, Frederic Gibou, and Ron Fedkiw. Simulating Water and Smoke with an Octree Data Structure. *ACM Trans. Graph. (SIGGRAPH Proc.)*, pages 457–462, 2004.

[81] Frank Losasso, Ron Fedkiw, and Stanley Osher. Spatially Adaptive Techniques for Level Set Methods and Incompressible Flow. *Computers and Fluids*, 35:995–1010, 2006.

[82] A. Guittet, M. Theillard, and F. Gibou. An unconditionally stable incompressible Navier-Stokes solver on arbitrary geometries and adaptive Quad-/Oc-trees. *J. Comp. Phys. (in press)*.

[83] H Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, New York, 1989.

[84] H Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS*. Addison-Wesley, New York, 1990.

[85] D Moore. The cost of balancing generalized quadtrees. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 305–312, 1995.

[86] A Weiser. *Local-Mesh, Local-Order, Adaptive Finite Element Methods with a Posteriori Error Estimators for Elliptic Parital Differential Equations*. PhD thesis, Yale University, June 1981.

[87] J Strain. Fast Tree-Based Redistancing for Level Set Computations. *J. Comput. Phys.*, 152:664–686, 1999.

[88] C Min. Local level set method in high dimension and codimension. *J. Comput. Phys.*, 200:368–382, 2004.

[89] Chohong Min and Frederic Gibou. A second order accurate level set method on non-graded adaptive Cartesian grids. *Journal of Computational Physics*, 225(1):300–321, 2007.

[90] Chohong Min. On reinitializing level set functions. *Journal of Computational Physics*, 229(8):2764–2772, April 2010.

[91] J Strain. Tree Methods for Moving Interfaces. *J. Comput. Phys.*, 151:616–648, 1999.

[92] Michael Lentine, Jón Tómas Grétarsson, and Ronald Fedkiw. An unconditionally stable fully conservative semi-lagrangian method. *Journal of Computational Physics*, 230(8):2857 – 2879, 2011.

[93] http://www.engr.ucsb.edu/~fgibou/research.html, 2017.

[94] Mohammad Mirzadeh, Arthur Guittet, Carsten Burstedde, and Frederic Gibou. Parallel level-set methods on adaptive tree-based grids. *Journal of Computational Physics*, 322:345 – 364, 2016.

[95] Carsten Burstedde, Lucas C. Wilcox, and Omar Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.

[96] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83 – 116, 2002.

[97] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457 – 492, 1999.

[98] Ronald P Fedkiw, Antonio Marquina, and Barry Merriman. An isobaric fix for the overheating problem in multimaterial compressible flows. *Journal of Computational Physics*, 148(2):545 – 578, 1999.

[99] Ronald P Fedkiw, Tariq Aslam, and Shaojie Xu. The ghost fluid method for deflagration and detonation discontinuities. *Journal of Computational Physics*, 154(2):393 – 427, 1999.

[100] Duc Nguyen, Frederic Gibou, and Ronald Fedkiw. A Fully Conservative Ghost Fluid Method and Stiff Detonation Waves. In *12th Int. Detonation Symposium, San Diego, CA*, 2002.

[101] Ronald P. Fedkiw. Coupling an eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method. *Journal of Computational Physics*, 175(1):200 – 224, 2002.

[102] Rachel Caiden, Ronald P. Fedkiw, and Chris Anderson. A numerical method for two-phase flow consisting of separate compressible and incompressible regions. *Journal of Computational Physics*, 166(1):1 – 27, 2001.

[103] Duc Q. Nguyen, Ronald P. Fedkiw, and Myungjoo Kang. A boundary condition capturing method for incompressible flame discontinuities. *Journal of Computational Physics*, 172(1):71 – 98, 2001.

[104] M. Kang, R. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput.*, 15(323-360), 2000.

[105] Xu-Dong Liu, Ronald P. Fedkiw, and Myungjoo Kang. A boundary condition capturing method for poisson's equation on irregular domains. *Journal of Computational Physics*, 160(1):151 – 178, 2000.

[106] Frederic Gibou, Liguo Chen, Duc Nguyen, and Sanjoy Banerjee. A level set based sharp interface method for the multiphase incompressible navier–stokes equations with phase change. *Journal of Computational Physics*, 222(2):536–555, March 2007.

[107] Frederic Gibou, Ronald P. Fedkiw, Li-Tien Cheng, and Myungjoo Kang. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *Journal of Computational Physics*, 176(1):205 – 227, 2002.

[108] Frédéric Gibou and Ronald Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *Journal of Computational Physics*, 202(2):577 – 601, 2005.

[109] F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher. A level set approach for the numerical simulation of dendritic growth. *J. Sci. Comput.*, 19:183–199, 2003.

[110] Joseph Papac, Frederic Gibou, and Christian Ratsch. Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions. *Journal of Computational Physics*, 229(3):875–889, 2010.

[111] Xu-Dong Liu and Thomas Sideris. Convergence of the ghost-fluid method for elliptic equations with interfaces. *Math. Comp.*, 72:1731–1746, 2003.

[112] Robert I. Saye and James A. Sethian. The Voronoi implicit interface method for computing multiphase physics. *PNAS*, 108:19498–19503, 2011.

[113] Satish Balay, Jed Brown, , Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. *PETSc Users Manual*. Argonne National Laboratory, 2012.

[114] Satish Balay, Jed Brown, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. Petsc web page, 2012.

[115] Hypre web page, 2012.

[116] R. D. Falgout and U. M. Yang. *Hypre: A library of high performance preconditioners*, volume 2331. Springer Berlin Heidelberg, 2002.

[117] C. H. Rycroft. Voro++: A three-dimensional voronoi cell library in C++. *Chaos*, 19, 2009.

[118] G. H. Shortley and R. Weller. Numerical solution of laplace's equation. *J. Appl. Phys.*, 9:334–348, 1938.

[119] Gangjoon Yoon and Chohong Min. Analyses on the finite difference method by Gibou *et al.* for Poisson equation. *Journal of Computational Physics*, 280:184 – 194, 2015.

[120] Gangjoon Yoon and Chohong Min. Convergence analysis of the standard central finite difference method for Poisson equation. *J. Sci. Comput.*, 67:602–617, 2016.

[121] Jiwon Seo, Seung yeal Ha, and Chohong Min. Convergence analysis in the l^∞ norm of the numerical gradient of the Shortley-Weller method. *J. Sci. Comput. (in press)*, 2017.

[122] Daniil Bochkov and Frederic Gibou. Solving the poisson equation with Robin boundary conditions on piecewise smooth irregular boundaries. *In Preparation*, 2017.

[123] Pouria Mistani, Arthur Guittet, Daniil Bochkov, Joshua Schneider, Dionisios Margetis, Christian Ratsch, and Frederic Gibou. The island dynamics model on parallel quadtree grids. *In preparation*, 2017.

[124] Frederic Gibou, Chohong Min, and Ronald Fedkiw. High resolution sharp computational methods for elliptic and parabolic problems in complex geometries. *J. Sci. Comput.*, 54:369–413, 2013.

[125] J Sethian. fast marching methods. *SIAM Review*, 41:199–235, 1999.

[126] A. Chacon and A. Vladimirsky. A parallel two-scale method for Eikonal equations. *SIAM J. on Scientific Computing*, 37(A156-A180), 2015.

[127] D. L. Chopp. Some improvements of the fast marching method. *J. Sci. Comput.*, 23(1):230–244, 2001.

[128] E. Rouy and A. Tourin. A viscosity solution approach to shape-from-shading. *SIAM J. Num. Anal.*, 29(3):867–884, June 1992.

[129] Hongkai Zhao. Parallel implementations of the fast sweeping method. *Journal of Computational Mathematics*, 25:421–429, 2007.

[130] Miles Detrixhe, Frédéric Gibou, and Chohong Min. A parallel fast sweeping method for the eikonal equation. *Journal of Computational Physics*, 237:46 – 55, 2013.

[131] Miles Detrixhe and Frédéric Gibou. Hybrid Massively Parallel Fast Sweeping Method for static Hamilton-Jacobi Equations. *In preparation*, 2014.

[132] J. Krug, P. Politi, and T. Michely. Island nucleation in the presence of step-edge barriers: Theory and applications. *Phys. Rev. B*, 61:14037, 2000.

[133] R.E. Caflisch, M.F. Gyure, B. Merriman, S. Osher, C. Ratsch, D.D. Vvedensky, and J.J. Zinck. Island dynamics and the level set method for epitaxial growth. *Applied Mathematics Letters*, 12:13, 1999.

[134] S. Chen, M. Kang, B. Merriman, R.E. Caflisch, C. Ratsch, R. Fedkiw, M.F. Gyure, and S. Osher. Level Set Method for Thin Film Epitaxial Growth . *Journ. Comp. Phys.*, 167:475, 2001.

[135] C. Ratsch, M.F. Gyure, S. Chen, M. Kang, and D.D. Vvedensky. Fluctuation and Scaling in Aggregation Phenomena. *Phys. Rev. B*, 61:R10598, 2000.

[136] X. Niu, R. Vardavas, R.E. Caflisch, and C. Ratsch. A Level Set Simulation of Directed Self-Assembly during Epitaxial Growth . *Phys. Rev. B*, 74:193403, 2006.

[137] Frederic Gibou, Christian Ratsch, Suzan Chen, Mark Gyure, and Russel Caflisch. Rate Equations and Capture Numbers with Implicit Island Correlations. *Phys. Rev. B.*, 63:115401, 2001.

[138] D Vvedensky, C Ratsch, F Gibou, and R Vardavas. Singularities and Spatial Fluctuations in Submonolayer Epitaxy. *Phys. Rev. Lett.*, 90:189601, 2003.

[139] C. Ratsch, M. F. Gyure, R. E. Caflisch, F. Gibou, M. Petersen, M. Kang, J. Garcia, and D. D. Vvedensk. Level-set method for island dynamics in epitaxial growth. *Phys. Rev. B*, 65:195403, 2002.

[140] W. K. Burton, N. Cabrera, and F. C. Frank. The growth of crystals and the equilibrium structure of their surfaces. *Philos. Trans. R. Soc. London Ser. A*, 243:299–358, 1951.

[141] G. S. Bales and D. C. Chrzan. Dynamics of irreversible island growth during submonolayer epitaxy. *Phys. Rev. B*, 50:6057–6067, 1994.

[142] Frederic Gibou, Christian Ratsch, and Russel Caflisch. Capture Numbers in Rate Equations and Scaling Laws for Epitaxial Growth. *Phys. Rev. B.*, 67:155403, 2003.

[143] A. A. Chernov. The spiral growth of crystals. *Soviet Phys. Uspekhi*, 4:116–148, 1961.

[144] J. Villain. Continuum models of crystal growth from atomic beams with and without desorption. *J. Phys. (France) I*, 1, 1991.

[145] H.-C. Jeong and E. D. Williams. Steps on surfaces: experiment and theory. *Surf. Sci. Rep*, 34:171–294, 1999.

[146] A Pimpinelli and J Villain. *Physics of Crystal Growth*. Cambridge University Press, Cambridge, UK, 1999.

[147] J. Lu, J.-G. Liu, and D. Margetis. Emergence of step flow from an atomistic scheme of epitaxial growth in 1 + 1 dimensions. *Phys. Rev. E*, 91:032403, 2015.

[148] S Davis. *Theory of Solidification*. Cambridge University Press, Cambridge, England, 2001.

[149] W. Kurz and D. J. Fisher. *Fundamentals of Solidification*. Trans Tech Publication, 1998.

[150] L.M. Mir. Electroporation of cells in tissues. Methods for detecting cell electropermeabilisation *in vivo*. In *Electroporation based Technologies and Treatment: proceedings of the international scientific workshop and postgraduate course*, pages 32–35, 14-20 November 2005. Ljubljana, SLOVENIA.

[151] B. Gabriel and J. Teissié. Time courses of mammalian cell electropermeabilization observed by millisecond imaging of membrane property changes during the pulse. *Biophys. J.*, 76(4):2158–2165 (electronic), 1999.

[152] M.C. Vernhes, P.A. Cabanes, and J. Teissié. Chinese hamster ovary cells sensitivity to localized electrical stresses. *Bioelectrochem. Bioenerg.*, 48:17–25, 1999.

[153] Anita Gothelf, Lluis M Mir, and Julie Gehl. Electrochemotherapy: results of cancer treatment using enhanced delivery of bleomycin by electroporation. *Cancer treatment reviews*, 29(5):371–387, 2003.

[154] J. Teissié, M. Golzio, and M.P. Rols. Mechanisms of cell membrane electropermeabilization: A minireview of our present (lack of ?) knownledge. *Biochimica et Biophysica Acta*, 1724:270–280, 2005.

[155] J. Teissié. Electropermeabilization of the cell membrane. In S. Li, J. Cutrera, R. Heller, and J. Teissié, editors, *Electroporation Protocols. Preclinical and Clinical Gene Medicine*, chapter 2. Springer, 2014.

[156] K. DeBruin and W. Krassowska. Modelling electroporation in a single cell. I. Effects of field strength and rest potential. *Biophysical Journal*, 77:1213–1224, Sept 1999.

[157] K. DeBruin and W. Krassowska. Modelling electroporation in a single cell. II. Effects of ionic concentrations. *Biophysical Journal*, 77:1225–1233, Sept 1999.

[158] J.C Weaver. Electroporation of cells and tissues. *IEEE Trans. on Plasma Sci.*, 28, 2000.

[159] Z. Vasilkoski, A. T. Esser, T. R. Gowrishankar, and J. C. Weaver. Membrane electroporation: The absolute rate equation and nanosecond time scale pore creation. *Phys Rev E Stat Nonlin Soft Matter Phys*, 74(2 Pt 1):021904, Aug 2006.

[160] Otared Kavian, Michael Leguèbe, Clair Poignard, and Lisl Weynans. "Classical" electropermeabilization modeling at the cell scale. *Journal of Mathematical Biology*, pages 1–31, 2012.

[161] M. Leguèbe, A. Silve, L.M. Mir, and C. Poignard. Conducting and permeable states of cell membrane submitted to high voltage pulses: Mathematical and numerical studies validated by the experiments. *Journal of Theoretical Biology*, 360:83–94, 2014. cited By 0.

[162] C. Poignard, A. Silve, and L. Wegner. Different approaches used in modeling of cell membrane electroporation. In D. Miklavčič, editor, *Handbook on Electroporation*. Springer, To appear.

[163] Franziska Hirschhaeuser, Heike Menne, Claudia Dittfeld, Jonathan West, Wolfgang Mueller-Klieser, and Leoni A. Kunz-Schughart. Multicellular tumor spheroids: an underestimated tool is catching up again. *Journal of Biotechnology*, 148(1):3–15, July 2010.

[164] Laure Gibot, Luc Wasungu, Justin Teissié, and Marie-Pierre Rols. Antitumor drug delivery in multicellular spheroids by electropermeabilization. *Journal of Controlled Release: Official Journal of the Controlled Release Society*, 167(2):138–147, April 2013.

[165] K.R. Foster and H.P. Schwan. Dielectric properties of tissues and biological materials: a critical review. *CRC in Biomedical Engineering*, 17(1):25–104, 1989.

[166] E.C. Fear and M.A. Stuchly. Modelling assemblies of biological cells exposed to electric fields. *IEEE Trans Biomed Eng*, 45(10):1259–1271, Oct 1998.

[167] R. A. Segalman. Patterning with block copolymer thin films. *Materials Science and Engineering: R: Reports*, 48(6):191–226, 2005.

[168] K. Galatsis, K. L. Wang, M. Ozkan, C. S. Ozkan, Y. Huang, J. P. Chang, H. G. Monbouquette, Y. Chen, P. F. Nealey, and Y. Botros. Patterning and templating for nanoelectronics. *Advanced Materials*, 22(6):769–778, 2010.

[169] D. J. C. Herr. Directed block copolymer self-assembly for nanoelectronics fabrication. *Journal of Materials Research*, 26(02):122–139, 2011.

[170] Stanley Osher and Fadil Santosa. Level set methods for optimization problems involving geometry and constraints: frequen- cies of a two-density inhomogeneous drum. *J. Comp. Phys.*, 171:272–288, 2001.

[171] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set mehtod. *Journal of Computational Physics*, 2003.

[172] Maxime Theillard, Landry Fokoua Djodom, Jean-Léopold Vié, and Frédéric Gibou. A second-order sharp numerical method for solving the linear elasticity equations on irregular domains and adaptive grids – application to shape optimization. *Journal of Computational Physics*, In press, 2012.

[173] Stanley Osher and Ronald P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169(2):463 – 502, 2001.

[174] Frédéric Chantalat, Charles-Henri Bruneau, Cédric Galusinski, and Angelo Iollo. Level-set, penalization and cartesian meshes: A paradigm for inverse problems and optimal design. *Journal of Computational Physics*, 228(17):6291 – 6315, 2009.

[175] Angelo Iollo and Manuel D. Salas. Contribution to the optimal shape design of two-dimensional internal flows with embedded shocks. *Journal of Computational Physics*, 125(1):124 – 134, 1996.

[176] Gaddiel Y. Ouaknin, Nabil Laachi, Kris Delaney, Glenn H. Fredrickson, and Frederic Gibou. Level-set strategy for inverse DSA-lithography. *J. Comp. Phys. (submitted)*, 2017.

[177] G. H. Fredrickson. *The equilibrium theory of inhomogeneous polymers*, volume 134. Oxford University Press, USA, 2006.

[178] Eugene Helfand. Block copolymer theory. iii. statistical mechanics of the microdomain structure. *Macromolecules*, 8(4):552–556, 1975.

[179] Eugene Helfand. Theory of inhomogeneous polymers: Fundamentals of the gaussian random walk model. *The Journal of Chemical Physics*, 62(3):999–1005, 1975.

[180] P.-G De Gennes. A rule of sums for semidilute polymer chains near a wall. *C. R. Seances Acad. Sci, Ser. B*, 290(23):509–510, 1980.

[181] Gaddiel Ouaknin, Nabil Laachi, Daniil Bochkov, Kris Delaney, Glenn Fredrickson, and Frederic Gibou. Functional level-set derivative for self consistent field theory. *J. Comp. Phys.*, 345:168–185, 2017.

[182] GK Cooray, B Sengupta, P. Douglas, M. Englund, R. Wickstrom, and K Friston. Characterising seizures in anti-NMDA-receptor encephalitis with dynamic causal modelling. *Neuroimage*, 118:508–519, 2015.

[183] AO Hebb, JJ Zhang, MH Mahoor, C Tsiokos, C Matlack, HJ Chizeck, and N Pouratian. Creating the feedback loop: closed-loop neurostimulation. *Neurosurg Clin N Am*, 25:187–204, 2014.

[184] Arthur Winfree. Patterns of Phase Compromise in Biological Cycles. *Journal of Mathematical Biology*, 95, 1974.

[185] Miles Detrixhe, Marion Doubeck, Jeff Moehlis, and Frederic Gibou. Fast Eulerian approach for computation of global isochrons of high-dimensional biological models. *SIAM J. APPLIED DYNAMICAL SYSTEMS*, 15:1501–1527, 2016.

[186] L. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117:500–544, 1952.