methods if small CFL numbers are acceptable. Indeed several algorithms of this type have been proposed with applications to modeling dendritic crystal growth [47], multiphase flows [41, 19, 33], and atomization process [21].

In many applications, it is desirable that the level-set function has signed-distance property, i.e. $|\nabla\phi| = 1$. Generally, there are two approaches to enforce this property, either by solving the pseudo-time transient reinitialization equation [40, 30]

$$\phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0,$$

or by solving the Eikonal equation

$$F(x)|\nabla\phi| = 1$$

with constant speed function $F(x) \equiv 1$. The transient reinitialization equation can be solved using explicit finite differences and thus can easily be parallelized in a domain decomposition approach. Moreover, only a few iterations may be needed if the signed-distance property is only required close to the interface [28]. This is the approach we have chosen in this paper. However, if the signed-distance property is required in the entire domain, solving the Eikonal equation is more computationally efficient. Unfortunately, the most popular algorithm for solving the Eikonal equation, i.e. the Fast Marching Method [37, 38], is inherently sequential due to causal relationship between grid points and cannot be easily parallelized. The Fast Sweeping Method (FSM) [49] is an alternative method for solving the Eikonal equation iteratively. The FSM can be more computationally efficient for simple choices of speed function, e.g. as in this context, and for simple interfaces. Moreover, FSM has more potential for parallelization compared to the FMM.

One of the earliest attempt in parallelizing the FMM is reported in [20] where a domain decomposition algorithm was introduced. Unlike the serial FMM, however, parallel FMM potentially requires multiple iterations or "rollback operations" to enforce causality across processes. Similar ideas are described in details in [46]. It should be noted that the number of iterations needed for the parallel FMM to converge greatly depends on the complexity of the interface and on the parallel partitioning and, in general, fewer iterations are required if the domains are aligned with the normals to the interface. Due to the nature of the Eikonal equation, shared memory machines might be a better environment for parallelization. For instance, in [10] the authors use an "adaptive" technique where individual threads implicitly define a domain decomposition at runtime. Unfortunately, this approach does not seem to be more effective than a simple static decomposition. In [50] a parallel FSM method was presented for the first time, which suffered from a plateau in the speed up. A scalable FSM was more recently proposed in [16], where the Cuthill-McKee numbering was utilized to exploit more parallelism. More recently a two-scale, hybrid FMM-FSM was presented in [15] which, albeit being more complicated to implement, promises even better scalability. Finally, a parallel Fast Iterative Method (FIM) was proposed in [23]. The FIM is similar to FMM in that it also maintains a list of "active nodes". However, unlike FMM, FIM avoids sorting the list and allows for concurrent updating of all nodes in an iterative fashion. In this article we simply choose the pseudo-time transient formulation for two reasons: 1) it is considerably easier to parallelize on Quadtrees and Octrees and 2) we are merely interested in the sign-distance property close to the interface, which only requires a few iterations.

The rest of this article is organized as follows. In section 2, we briefly review the sequential algorithms and discretization methods for the level-set equation on adaptive tree-based grids. These ideas are then extended in section 3 to parallel environments using a domain decomposition method. In section 4, we provide several examples that illustrate the scalability of our algorithms. Finally, we close by providing an important application of our strategy by considering the simulation of the solidification process by solving a Stefan problem in section 5.

2. The level-set method

The level-set method, introduced in [31], is an implicit framework for tracking interfaces that undergo complicated topological changes. In this framework, an interface is represented by the zero contour of a higher dimensional function, e.g. a curve in two spatial dimensions can be described as $\Gamma = \{(x,y)|\phi(x,y) = 0\}$, where $\phi(x,y)$ is the level-set function. The evolution of the curve under a velocity field $\underline{\mathbf{u}}$ is then obtained by solving the level-set equation:

$$\phi_t + \underline{\mathbf{u}} \cdot \underline{\nabla}\phi = 0. \tag{1}$$