Journal of Computational Mathematics
Vol.33, No.6, 2015, 576–586.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1503-m4532

AN ADAPTIVE FAST INTERFACE TRACKING METHOD*

Yana Di
LSEC, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
Email: yndi@lsec.cc.ac.cn
Jelena Popovic
Department of Numerical Analysis, CSC, KTH, 100 44 Stockholm, Sweden
Email: jelenap@csc.kth.se
Olof Runborg
Department of Mathematics and Swedish e-Science Research Center (SeRC), KTH, 100 44 Stockholm, Sweden
Email: olofr@nada.kth.se

Abstract

An adaptive numerical scheme is developed for the propagation of an interface in a velocity field based on the fast interface tracking method proposed in [2]. A multiresolution stategy to represent the interface instead of point values, allows local grid refinement while controlling the approximation error on the interface. For time integration, we use an explicit Runge-Kutta scheme of second-order with a multiscale time step, which takes longer time steps for finer spatial scales. The implementation of the algorithm uses a dynamic tree data structure to represent data in the computer memory. We briefly review first the main algorithm, describe the essential data structures, highlight the adaptive scheme, and illustrate the computational efficiency by some numerical examples.

Mathematics subject classification: 42C40, 65D10.
Key words: Interface tracking, Multiresolution, adaptivity, Fast algorithms.

1. Introduction

Tracking the evolution of interfaces or fronts is important in many application, for instance wave propagation, multiphase flow, crystal growth, melting, epitaxial growth and flame propagation. The interface in these cases is a manifold of co-dimension one which moves according to some physical law that depends on the shape and location of the interface. We suppose for convenience that it can be parameterized, so that for a fixed time t, the interface is described by the function $\boldsymbol{x}(t,s):\mathbb{R}^+\times\mathbb{R}^q\to\mathbb{R}^d$, with the parameterization $s\in\Omega\subset\mathbb{R}^q$ and $q=d-1$. In this paper we consider the simplified case when the interface is moving in a time-varying velocity field that does not depend on the front itself. Then $\boldsymbol{x}(t,s)$ satisfies the parameterized ordinary differential equation (ODE)

$$\frac{\partial \boldsymbol{x}(t,s)}{\partial t}=F(t,\boldsymbol{x}(t,s)),\qquad \boldsymbol{x}(0,s)=\gamma(s),\qquad s\in\Omega, \tag{1.1}$$

where $F(t,\boldsymbol{x}):\mathbb{R}^+\times\mathbb{R}^d\to\mathbb{R}^d$ is a given function representing the velocity field and $\gamma(s):\mathbb{R}^q\to\mathbb{R}^d$ is the initial interface. We will mostly treat curves in two dimensions, $d=2$,

* Received June 3, 2014 / Revised version received March 20, 2015 / Accepted March 26, 2015 / Published online November 17, 2015 /

$q = 1$, but also discuss extensions to higher dimensions $d = 3$, $q = 2$ and co-dimensions $d = 3$, $q = 1$. Applications could include the tracking of physically motivated interfaces, like wavefronts in high frequency wave propagation problems, or "artificial" fronts of propagation paths parameterized by initial data, where a problem has the structure (1.1) even though the front has no direct physical interpretation. This could be, for instance, iso-distance curves on a surface (front of geodesics), fiber tract bundles in brain imaging or the method of characteristics for the solution graph of hyperbolic PDEs. In many of these problems it is better to numerically consider a front rather than a set of individual paths, since the connectivity between paths is then maintained, which for example simplifies interpolation between them. Numerical methods for this problem include the Lagrangian front tracking method [4]. There are also Eulerian approaches like the level set method [5] and segment projection [6]. For flow problems we should also mention the marker-and-cell (MAC) [7] and volume of fluid (VOF) [8] methods.

We focus here on front tracking, in which the interface is described by a set of marker points that are connected in a known topology. In one dimension one would approximate $\boldsymbol{x}_j(t) \approx \boldsymbol{x}(t, s_j)$ and use a numerical method for ODEs to solve

$$\frac{\mathrm{d}\boldsymbol{x}_j(t)}{\mathrm{d}t} = F(t, \boldsymbol{x}_j(t)), \qquad \boldsymbol{x}_j(0) = \gamma(s_j), \tag{1.2}$$

where $s_0 < s_1 < \ldots < s_N$ is a discretization of Ω. For surfaces in three dimensions, the markers on the interface are typically held together in a triangulation. Propagating one marker numerically with a time step length Δt to a fixed time costs $O(1/\Delta t)$ operations. Hence, if the interface is represented by N points the cost of standard front tracking is $O(N/\Delta t)$. In [1–3], wavelet vectors were used to describe the interface, which correspond to the details of the interface on different scale levels. It was shown that the time derivatives of the wavelet vectors, just as the wavelet vectors themselves, decay exponentially with level of detail. By taking multiscale time steps, i.e. longer time steps for the fine scales than for the coarse scales, the computational cost is reduced to only $O(\log N/\Delta t)$ or even $O(1/\Delta t)$ without affecting the overall accuracy. We should emphasize that this is different from standard wavelet based adaptive schemes where shorter time steps are often used for the fine details, which is the opposite of the method in [2]. With such strategy the cost will be reduced, but it will only be the constant in the complexity estimate that is improved; the complexity itself remains the same order. The reason is that there are comparatively few coarse scale wavelet vectors, where efficiency improvement is achieved, and many fine scale wavelet vectors, where there is little gain.

Adaptivity is usually an important feature of front tracking algorithms. Since the length or area of the interface can grow quickly and the number of marker points used initially may not be enough to resolve it, an adaptive mechanism which adds and removes marker points as the resolution of the interface changes becomes necessary. For multiresolution methods, an advantage to define adaptive techniques is an efficient data representation with an accurate estimation of the local approximation error. Based on the details, or wavelet coefficients, between two consecutive grid-refinement levels, multiresolution methods provide a rigorous regularity analysis [14], while for adaptive mesh refinement methods rigorous error estimators are quite difficult to be derived. In the past, adaptive wavelet-based multiresolution methods have been introduced to improve the computational efficiency and to reduce the memory requirement of the algorithms, e.g., [9–13]. According to error estimates from different resolution levels, numerical schemes have been developed for adjusting grid resolution locally and dynamically. To obtain additional speed-up, space-time adaptive methods [15] are introduced, where the size of