7

Constructing Signed Distance Functions

7.1 Introduction

As we have seen, a number of simplifications can be made when $\phi$ is a signed distance function. For this reason, we dedicate this chapter to numerical techniques for constructing approximate signed distance functions. These techniques can be applied to the initial data in order to initialize $\phi$ to a signed distance function.

As the interface evolves, $\phi$ will generally drift away from its initialized value as signed distance. Thus, the techniques presented in this chapter need to be applied periodically in order to keep $\phi$ approximately equal to signed distance. For a particular application, one has to decide how sensitive the relevant techniques are to $\phi$'s approximation of a signed distance function. If they are very sensitive, $\phi$ needs to be reinitialized to signed distance both accurately and often. If they are not sensitive, one can reinitialize with a lower-order accurate method on an occasional basis. However, even if a particular numerical approach doesn't seem to depend on how accurately $\phi$ approximates a signed distance function, one needs to remember that $\phi$ can develop noisy features and steep gradients that are not amenable to finite-difference approximations. For this reason, it is always advisable to reinitialize occasionally so that $\phi$ stays smooth enough to approximate its spatial derivatives with some degree of accuracy.
7.2 Reinitialization

In their seminal level set paper, Osher and Sethian [126] initialized their numerical calculations with \( \phi = 1 \pm d^2 \), where \( d \) is the distance function and the "±" sign is negative in \( \Omega^- \) and positive in \( \Omega^+ \). Later, it became clear that the signed distance function \( \phi = \pm d \), was a better choice for initializing \( \phi \). Mulder, Osher, and Sethian [115] demonstrated that initializing \( \phi \) to a signed distance function results in more accurate numerical solutions than initializing \( \phi \) to a Heaviside function. While it is obvious that better results can be obtained with smooth functions than nonsmooth functions, there are those who insist on using (slightly smeared out) Heaviside functions, or color functions, to track interfaces.

In [48], Chopp considered an application where certain regions of the flow had level sets piling up on each other, increasing the local gradient, while other regions of the flow had level sets separating from each other, flattening out \( \phi \). In order to reduce the numerical errors caused by both steepening and flattening effects, [48] introduced the notion that one should reinitialize the level set function periodically throughout the calculation. Since only the \( \phi = 0 \) isocontour has any meaning, one can stop the calculation at any point in time and reset the other isocontours so that \( \phi \) is again initialized to a signed distance function. The most straightforward way of implementing this is to use a contour plotting algorithm to locate and discretize the \( \phi = 0 \) isocontour and then explicitly measure distances from it. Unfortunately, this straightforward reinitialization routine can be slow, especially if it needs to be done at every time step. In order to obtain reasonable run times, [48] restricted the calculations of the interface motion and the reinitialization to a small band of points near the \( \phi = 0 \) isocontour, producing the first version of the local level set method. We refer those interested in local level set methods to the more recent works of Adalsteinsson and Sethian [2] and Peng, Merriman, Osher, Zhao, and Kang [130].

The concept of frequent reinitialization is a powerful numerical tool. In a standard numerical method, one starts with initial data and proceeds forward in time, assuming that the numerical solution stays well behaved until the final solution is computed. With reinitialization, we have a less-stringent assumption, since only our \( \phi = 0 \) isocontour needs to stay well behaved. Any problems that creep up elsewhere are wiped out when the level set is reinitialized. For example, Merriman, Bence, and Osher [114] proposed numerical techniques that destroy the nice properties of the level set function and show that poor numerical solutions are obtained using these degraded level set functions. Then they show that periodic reinitialization to a signed distance function repairs the damage, producing high-quality numerical results. Numerical techniques need to be effective only for the \( \phi = 0 \) isocontour, since the rest of the implicit surface can be repaired by reinitializing \( \phi \) to a signed distance function. This greatly in-