在继续介绍基本流体求解器的核心之前,压强投影步骤使得流体不可压缩并满足其边界条件,我们需要转移到几何形状上.在上一章中,我们已经遇到了边界条件和相关的几何问题:

* 当粒子在固体内部时?(该点可能是我们在半拉格朗日期间回溯到的位置)
* 某些几何图体表面上最近的点是什么？
* 我们如何将一个区域的值外推到另一个区域?

前两个查询在流体求解器中极为常见.我们通常也希望几何图形的表示与网格上离散的方程配合使用,最实际的答案之一是水平集方法.

在本书中,我们将仅介绍所需的基础知识.读者可以参考Osher和Fedkiw的书[OF02],以更详细地研究水平集的数字和应用.

上面的第一个查询表明正确的方法是隐式曲面.虽然有很多方法可以将其概括,但我们将集中在连续的标量函数定义的几何体:

* 当时点在几何体之外,
* 当时点在几何体内部,且
* 当时点在几何体表面上.

在计算机图形学中的某些其他上下文中,约定可以颠倒(“外部”是负区域),也可以使用非零的阈值,但这通常是我们在流体模拟中采用的方法.

该表面也可以称为的零等值线，“等值线”是指等高线处的值均相同（“ iso”），而“零”表示该特殊值的轮廓。 在微积分（在一定水平上的点集）中也使用了“水平集”一词，但是在数值方法和计算机图形学中，它意味着更多.

如果在曲面某个点上是可微分的，向量微积分还告诉我们，梯度矢必须指向曲面的法向量.由于在内部为负,而外部则增加为正,因此,实际上梯度必须是曲面上朝外的法向量.(参考教材第十四章第六节内容”Tangent Planes to Level Surfaces”.)

如果我们要问梯度恰好是表面上的单位长度法线呢?换句话说,要求

至少在曲面附近? 由于梯度必须平行于法线，这将意味着的方向导数在法线方向上恰好为1,

这意味着对于一个小数值和表面上的点，

在这种情况下,是我们估计在朝外方向时距离表面的距离,并且这告诉我们值必须近似其本身距离.更精确的说,我们在寻找**符号距离[signed distance]**,当处于几何体内部时,值为负;这是外向距离,向内时该值为负.

实际上,我们可以解决这个问题:给定任意几何表示,将定义为有符号距离函数,有时缩写为SDF.然后,所得的将用作出色的隐式表面描述.

4.1 符号距离

给定任何封闭的点集,该点集的**距离函数[distance function]**为

也就是说,它是到中最近点的距离.如果将空间分为明确定义的内部和外部,则符号距离函数为

除梯度等于曲面上单位长度的向外法线外,符号距离还具有许多有用的属性.例如,在几何图形外的点处,令为曲面上距离的点.显然，如果我们直接远离曲面上最近的点，距离函数将增长最快,向量积分告诉我们,梯度必须指向“最陡的上升”方向.换句话说，在几何图形之外,始终指向曲面上最近的点.类似的推理表明,在几何图形内部，指向曲面上最近的点.

设是指向最近点的方向:

显然如果我们沿着该方向移动小距离到,距离最近的点依然是:曲面上的最近点不会因你移动而改变!因此距离函数在新点处必须等于.这给出了方向导数:

我们已经知道,几何体外部的梯度指向与相反的方向,方向导数只是梯度与方向的点积,,因此我们可以算出渐变在几何图形外部仅仅是-.类似的推理应用在几何内部.综上所述:

* 在几何体外部,是指向曲面最近点的单位向量,
* 在几何体内部,是指向曲面最近点的单位向量,
* 在几何体曲面上,是指向曲面法向量的单位向量.

如果我们结合值是离最近点的距离(在内部时为负值)这一事实,我们得到另一个有用的结果,对任意点,

是曲面上距离的最近点.符号距离函数很容易为我们提供本章开头列出的前两个查询的答案！

另一个优美的结果是是距离曲面最近点向外指向曲面法向量的单位向量.换句话说,它使我们将“法线”概念合理地扩展到了甚至不在表面上的点.例如,对于靠近但不完全在固体表面上的流体流动,我们使用,可以轻松地将速度分解为与表面垂直的分量,和与表面相切的余数.

深入研究水平集的数学方面，可以将梯度始终为单位长度这一事实表示为称为Eikonal方程的非线性PDE：

事实证明，有了适当的边界条件（即几何表面上的）和问题中信息流的技术条件，这也足以定义符号距离。一些论文和算法都从这个角度进行工作，但是本书中我们将大部分坚持几何推理。

可以证明，实际上除**中轴[medial axis]**以外,符号距离所有地方都是平滑的(即存在和更高的导数),所有距离值是由与表面的不同部分等距的点组成.中轴恰好是没有唯一最近点的地方，例如球体的中心和平板内部的中间平面.它伸展以接触尖锐几何形状的边缘和角落.在凹形几何体周围,中轴的某些部分在几何体之外.

对于典型的曲面，中间轴是一个较小的低维集，我们通常不必担心–即使在中间轴上，仍然是连续的，只是有一个纽结。 特别是在表面本身光滑的情况下，在表面上和表面附近的符号距离是光滑的。 但是，请务必记住，以上有关梯度的所有讨论都在中间轴处分解：该函数甚至在此处均不可微分，因此梯度不存在。 我们将很快讨论的梯度的数值近似值通常仍会给出一个矢量，但它可能比单位长度小得多，甚至可能为零。

符号距离函数的高阶导数在平滑的地方也具有几何意义.例如.以原点为中心的半径为r的球面的符号距离函数为

梯度为

对该函数取散度,即三维空间中原符号距离函数的拉普拉斯算子,

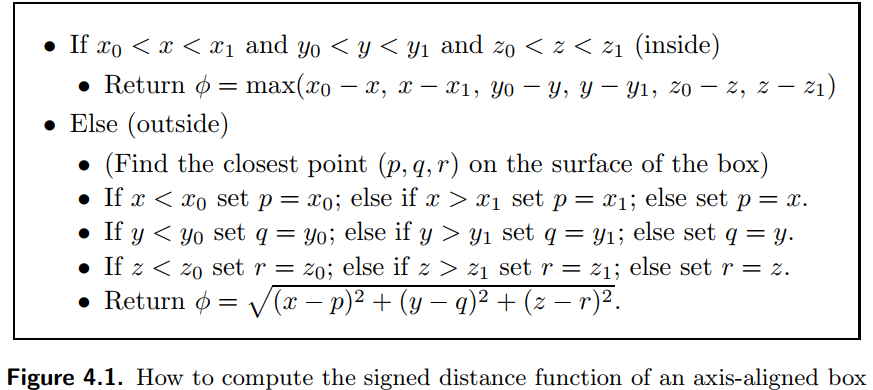
特别是,在的曲面上,拉普拉斯恰好等于球的平均曲率.这绝非偶然：对于任何光滑的曲面一般都成立,在曲面上求出的符号距离函数的拉普拉斯算术是平均曲率.

4.2 离散化符号距离函数

到目前为止,我们已经定义了精确的符号距离函数.对于一些几何对象，这很容易得出解析解.以为中心的半径r的球面的符号距离为

通过点且法线为的无穷平面的符号距离为

定义为[x0，x1]×[y0，y1]×[z0，z1]的轴对齐框的符号距离函数可以通过首先检查点在内部还是外部，然后使用最接近的距离来计算,参见图4.1.圆柱体和圆锥体也相当简单.



我们将在下面介绍如何计算到三角形网格的有符号距离，但是它要复杂得多，并且如果没有优化的加速结构，可能会很慢。 显然，如果符号距离函数的评估比直接从网格计算内部/外部更昂贵，那么将内部/外部几何查询改写为仅评估有符号的距离函数并不会帮自己一个忙。 我们还没有真正更改底层的几何表示：我们只是将其隐藏了。

这就是水平集法起作用的地方.与其从几何信息中解析地计算出符号距离,不如将其与其他流体变量一样直接存储在网格上.然后,当我们需要评估时,从周围的网格点内插一个近似值.结果通常是近似值,但也足够好:无论如何,所有流体模拟都处理近似结果,因此也可以接受几何近似.这是人们通常用“水平集”一词来理解的核心:距离函数可以在网格上采样.

一旦使用网格值,就可以通过插值快速而轻松地评估.使用有限差分,对梯度进行近似也很容易.例如,可以在任意两个网格点之间沿x轴的中途获得的精确有限差分估计：

同样，我们可以在网格值之间的其他中点获得y导数和z导数的估计值。然后，我们可以在这些值之间进行插值，以在网格中的任何点获得近似的梯度向量。通常，这比直接对φ的插值求导更可取，因为我们通常使用的插值（分段三线性或分段三三次）在网格单元之间的导数中具有跳跃间断.

4.3 计算符号距离

如果我们没有给定几何图形的简单解析公式，但是想要生成它的水平集，则需要一种算法来计算网格上的有符号距离.一个特别有帮助的脚注是算法不一定是精确的:插值时,我们已经引入了误差,因此可以容忍网格值本身中的小误差,尤其是远离曲面本身的误差.

有两种主要的计算水平集的方法：从几何（找到最近的点并测量到它们的距离）或从PDE（求解Eikonal方程）.两者都有其用途：几何方法通常更准确，更容易理解，但是PDE方法适用于几何图形不明确的情况.

4.3.1 点之间的距离

我们将从一种特殊情况开始，尽管这是一种常见操作：通过几何方法计算到一组有限点的距离。 请注意，没有“内部”，因此该距离是否带符号是没有意义的。 该算法在图4.2中给出，其中一个关键要素未指定：我们在网格上循环传播距离信息的顺序。 它基于Tsai文章[Tsa02]中的算法。 有关其他许多算法的综述，请参阅Jones等人的论文[JBS06]。

在第一阶段，我们直接在紧邻输入点的网格单元中计算精确的距离和最接近的点信息，而无需任何花哨的几何数据结构。 第二阶段可以再次通过网格将其有效地从邻居传播到邻居，而无需额外的数据结构。 但是，这并不精确：最接近给定网格单元的点可能与最接近其任何邻居的点不同。 话虽如此，到真正最接近点的距离永远不会与通过这种方法计算得出的到最接近点的距离有很大的不同：在实践中，这非常有效。

4.3.2 循环顺序

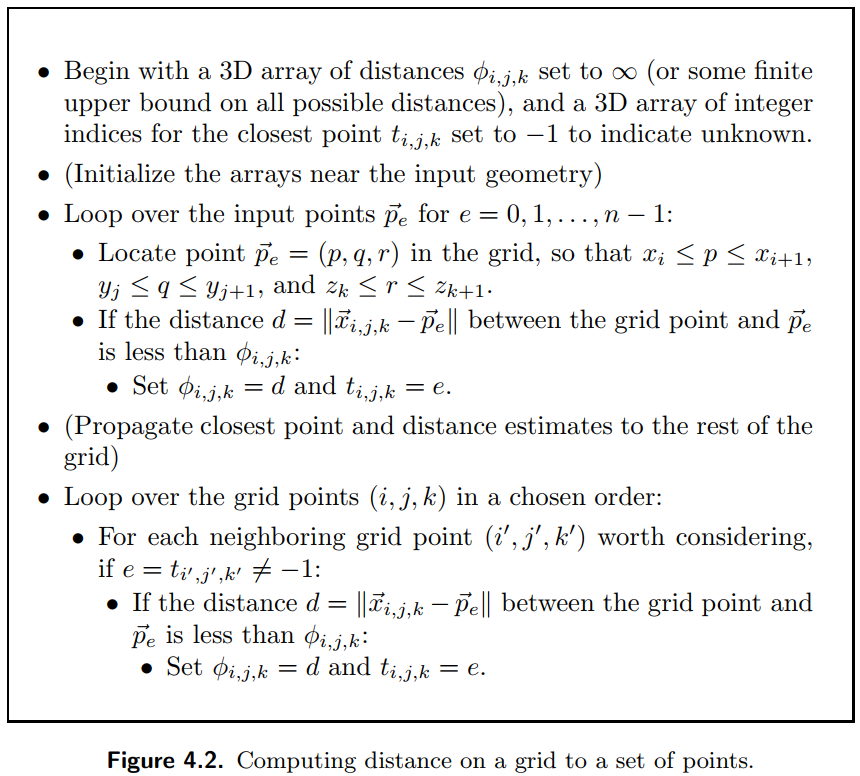
对于循环顺序，Tsai等人[Tsa02]给出两个建议。一是基于快速行进方法[Set96，Tsi95]，另一种基于快速扫描方法[Zha05].

快速行进方法基于以下认识：网格点应从较近的点而不是相反的点获取有关到几何的距离的信息。 因此，我们希望遍历从最近到最远的网格点。 通过将未知网格点存储在以当前距离的当前估计作为键的优先级队列（通常实现为堆）中，可以简化此操作。 我们使用已知网格点的邻居，其距离值和从那些已知网格点估计的最近点来初始化堆。 我们选择最小值并将其从优先级队列中删除，将其设置为已知值，然后更新其未知邻居的距离和最近点信息（可能将它们添加到优先级队列中或在优先级队列中向上移动）。 然后一次又一次地执行此操作，直到优先级队列为空。 它以O（n log n）的时间运行n个未知的网格点。

快速扫描方法方法从信息从更近的点传播到更远的点这一事实得出了不同的策略。 对于任何网格点，最终，它的最接近点信息将从网格中的一个特定方向（例如，从（i + 1，j，k），或者从（i，j-1，k） ）， 等等。 为了确保信息可以沿正确的方向传播，我们以所有可能的循环顺序扫过网格点：i递增或递减，j递增或递减，k递增或递减。 三维有八种组合，二维有四种。 为了获得更高的准确性，我们可以再次重复扫描。 在实践中，两次扫描可以提供出色的结果，尽管可以进行更多次迭代。

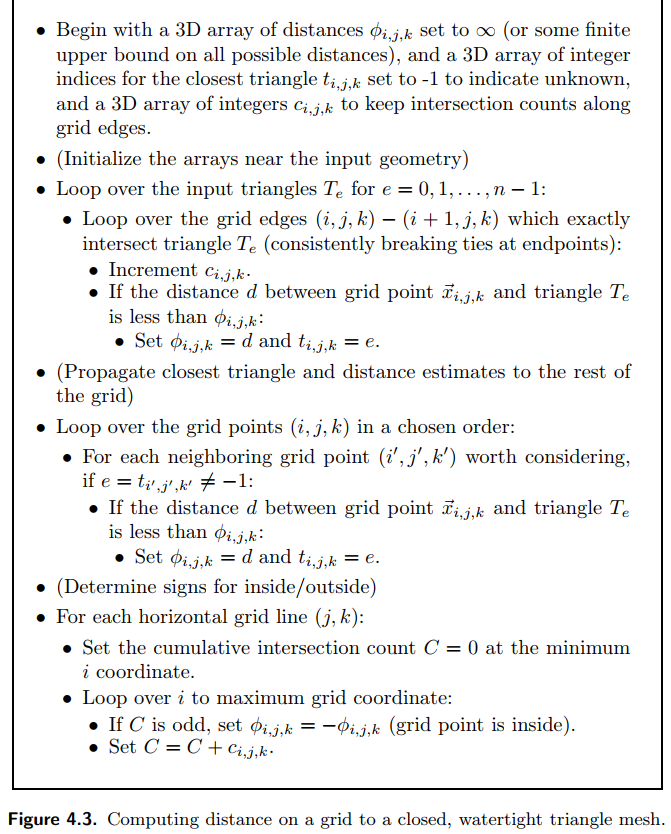
快速扫描优于快速行进的好处在于它是O（n），不需要网格以外的其他数据结构，因此实现起来非常简单。 当计算整个网格上的距离时，快速扫描可能是最好的选择。

但是，如前所述，现代的流体求解器可能会使用稀疏的平铺网格，这会使清扫变得复杂。 在这种情况下，可以采用混合方法。 我们可以在图块内部高效运行快速扫掠，以基于图块及其邻居的信息来更新距离，但是我们可以选择以快速行进样式求解图块（并在更新邻居时重新求解）的顺序。 从包含输入点的图块开始，将其设置为“ redistance”。 每当通过快速扫掠对瓷砖进行重新分配时，请检查任何人脸邻居中的距离值是否大于此瓷砖中存储的距离大于：如果是，则将相邻的瓷砖添加到需要重新分配的集合中。



4.3.3 计算三角形的符号距离

接下来让我们处理更有趣的几何图形：封闭的三角形网格。 我们将采用相同的通用方法，但不是跟踪最近的点，而是跟踪最近的三角形以提高准确性。 我们还将采取额外的步骤来找出内部/外部信息，首先计算原始绝对距离，然后在最后一个循环中固定符号。 图4.3给出了伪代码。 同样，距离传播的循环顺序也未指定：快速扫描，快速行进或混合平铺组合在这里同样适用。



计算点和三角形之间的距离是此算法的核心操作.琼斯在技术报告中提供了两种技术的可能详细信息，值得阅读[Jon95].更直接但可能更慢的方法是计算平面上点通过三角形顶点和最近点的重心坐标,使用下列线性系统:

其中是的缩写,其它一样.如果重心坐标均为非负值,则平面中最近的点在三角形内,您可以使用和之间的距离.否则,最接近的点必须位于三角形的边缘之一上. 如果重心坐标为正,则不需要检查对边（与等相对）,但是在某些情况下,必须同时考虑两条边,并且将两条边的距离最小.

计算点与边之间的距离就是求点到线段的距离(这个过程很简单,资料也很容找到):

如果,最近点为，如果,最近点为，否则最近点为.

该算法中的另一个棘手点，实际上是一个更加棘手的点，是内部/外部的确定。 它本质上是通过沿负x轴将射线投射到负无穷大，计算与三角形的相交数并确定相交数是否为奇数来确定“内部”，从而确定网格点是否在网格内。 假设在负无穷大处，我们在网格外部，并且如果要沿轴向后移动，则每次穿过三角形时，我们都必须从内部切换到外部，反之亦然。 对于封闭的“水密”三角形网格确实如此。

为了使该算法更健壮，必须非常仔细地编写相交例程。 如果错过了一个交叉路口，计数将被取消，结果将不正确。 如果两个三角形恰好沿着x对齐的网格线相交，并且我们对零个或两个相交进行计数，则该计数将关闭并且结果不正确：在这种情况下，我们只能对一个和一个相交进行计数。 如果三角形平行于x轴并恰好在网格线上，并且如果几何体恰好包含一个与轴对齐的平面，这种情况经常发生，那么我们应确保不计算任何交点。 每次都使所有这些情况正确，这超出了本书的范围。 我的代码依赖于Shewchuk的可靠谓词来计算符号精确的方向行列式[She97]，以及遵循简单模拟（SOS）[EM90]策略时，小心的符号扰动，以在三角形准确地与网格对齐时始终“断开联系”。 ]。

当然，有时输入三角形网格并不是完美的水密结构.例如,它们可能有裂纹或重叠的部分不太吻合,或者只指定了一侧（例如，海洋的上表面没有底部）.我们仍然可以使用上述算法为三角形计算距离场,但是交点计数和内部/外部符号都不可靠.

如果所有输入三角形均具有一致的朝外的法线，可以根据的点积来设置初始循环中附近网格点的符号（内部/外部）,其中表示三角形上距离网格点最近的点,表示三角形的法向量.然后,在距离传播循环中,我们使用而不是来比较距离，我们从邻居处取符号（内部/外部）.

如果使用更粗糙的模型，则需要更复杂的方案来估算内部/外部（可能一点也不清楚）。 最近最完整的例子是Jacobson等人[JKSH13]，他本质上计算了向任意方向发射的随机射线穿过奇数个三角形的概率，以决定内部/外部的“原始”估计，然后进一步处理所有点的结果以获得清晰的分割。

4.4 重新计算符号距离

上一节使用基于几何的算法直接从显式给定的几何计算符号距离。我们遇到的另一种常见情况是，我们有一个由网格上的值定义的水平集，该水平集符号距离很远，我们希望准确地重新计算符号距离或重新分配该水平集。这通常是由于几何处理算法无法完全保留距离而导致的。

在这里，我们没有可用于提供方便的最接近点的显式几何，因此，我们采用PDE方法并直接求解Eikonal方程。 这遵循原始的快速行进方法[Set96，Tsi95]和快速扫描方法[Zha05]。

我们的第一步是准确地估计紧挨着表面的距离，即在有符号变化的网格单元中（某些角落为负,另一些为正）.如果我们确信它们已经足够接近距离了，我们可以保持它们不变.否则，我们需要估计零等值线在网格单元中的位置，并为此使用带符号的距离。假设旧函数为,并且在网格点上,我们发现具有与k不同的符号.我们可以沿这两点之间的边缘线性插值，并求解线性插值为零的沿边缘的分数：

我们可以临时设置，即到从F继承符号（内/外）的那个交叉点的距离。 当然，我们应该对F中具有不同符号的所有邻居进行相同的计算，并采用最接近零的符号距离值。 为了获得更准确的结果，我们甚至可以通过这些交点的集合来拟合线段或三角形（实际上，是局部生成Marching Cubes [WMW86，LC87]样式的网格），并采用与其最接近的点。

为紧邻曲面的网格点设置值之后,我们将剩余的网格点全部设为,或者根据输入的设置为所有可能距离的上界或下界值.然后和之前一样将网格距离使用扫描法、前进法或混合法传播下去.然而,与其使用几何计算来确定距离值,不如直接求解Eikonal方程.回想一下,符号距离函数的梯度必须为单位长度:

网格点处的符号距离可以从邻近值计算,我们使用有限差分替换偏导数,然后求解二次方程的未知量.关键点是我们只能从附近离曲面比较近的点计算距离.

例如,我们从邻近点和处估算.将上述方程离散化,得到

为简单起见,假设所有值均为正(如果不是,则将符号翻转,计算新值,然后再次翻转符号).令这三个相邻值按排序.然后,我们尝试使用一个,两个或三个相邻的值来估计距离,仅当计算的距离小于我们对的现有估计时,才采用最小的计算距离.参见图4.4.

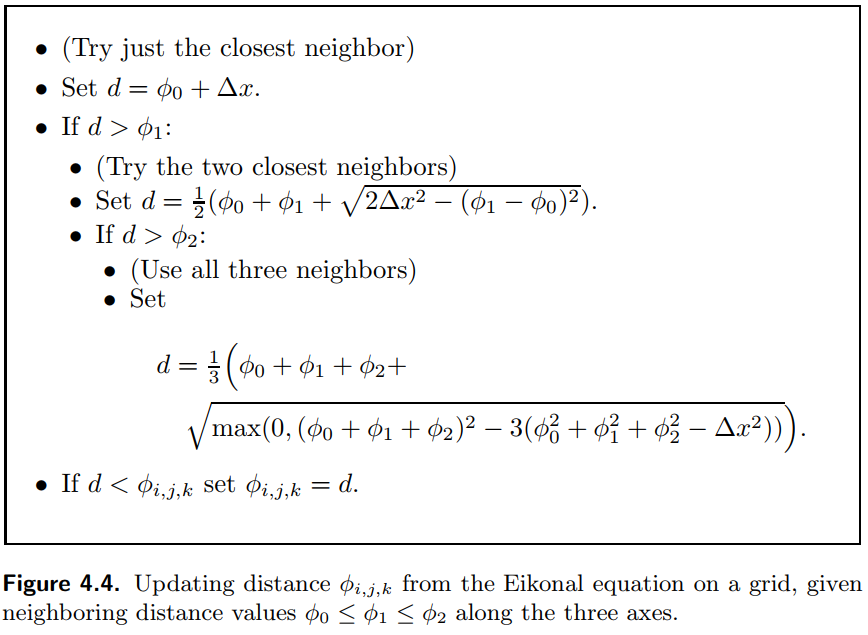
4.5 水平集操作

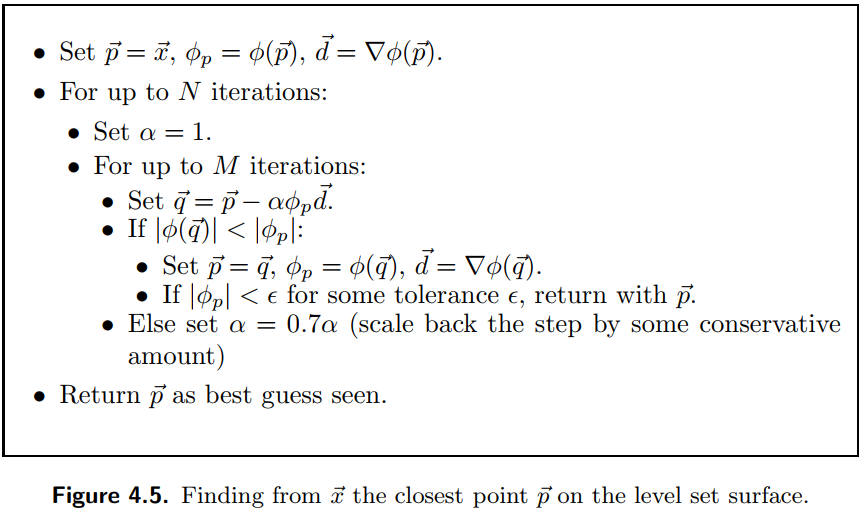
当涉及到几何的有效性时,几何的水平集方法会带来额外的好处.首先让我们看一下几何查询.我们已经看到,确定内部与外部之间的关系很简单,即从附近的网格点内插并检查其符号.估计曲面最近点的法线或方向就像获取一些有限差以近似一样简单.

实际上，从给定点找到曲面上最近点并不像我们最初确定的那样简单.利用精确的符号距离函数,.但是,如果值存在误差,并且由于插值和有限差分而产生误差，则不能保证该公式会在曲面上准确给出一个点.沿着梯度进行迭代操作,只要值趋近于零,就可以给出更可靠的结果,请参见图4.5.

可以直接跟踪级别集以进行渲染，并且将射线与级别集相交也可以在其他地方使用。一种可能是进入网格，将其视为常规的光线跟踪加速结构：如果所有拐角处的φ值都具有相同的符号（因此零等高线不通过网格），则可以跳过网格细胞）。可以用八叉树或粗网格进一步加速。加快跟踪速度的另一种可能性是自己使用带符号的距离值：如果它们足够准确，它们将在不接触曲面的情况下沿光线传播的距离提供一个保守的下限（因为到物体上最近点的距离）表面是沿光线到表面的距离的下限）。一旦足够接近表面，并且特别是一旦在射线上找到两个具有不同符号的点，就可以使用数字二分法搜索（例如二等分搜索或割线搜索）来获取交点。

应该注意的是，用于水平集的插值方法在光线追踪时会起到很大的作用。 简单的分段三线性插值法可在网格单元内提供平滑的三线性面片，但曲面的法线可以从一个网格单元不连续地跳到下一个网格单元：即使使用漫反射着色器，这也非常明显。 通常需要更平滑的C1方法，例如使用二次B样条。





像其他任何场一样,水平集也可以通过速度场平移.我们将在第8章中回到这一点,这是模拟像水这样的液体的重要组成部分.特别是,如果要在速度场为的情况下移动曲面，则的点应遵循，同时保持为零.更一般地说,我们可以说域中的每个点都应随速度场移动,并保持其值移动,给出以下等式:

根据需要移动曲面,直至出现数值误差,从而产生新的水平集.通常,新的水平集将不再是完全有符号的距离,因此我们决定重新计算每帧的距离,但是在大多数情况下,紧邻曲面的值将“足够好”并且不需要重新计算(这可能会导致伪像).水平集对流对数值扩散非常敏感,因此至少必须使用我们在第3章末尾讨论的尖锐三次插值.即使如此,随着时间的流逝,尖锐特征势必会被平滑掉甚至消失:小孔被填充,薄的结构消失.如何更好地处理此问题将留在本书的后面.

对流还可以将位移纹理或自由形式的变形应用于水平集，将体积位移场视为基本整合了速度的一个步骤以使原始水平集几何变形.或者，可以将经过精心设计的“凹凸贴图”场直接添加到水平集,以在法线方向上稍微扰动几何形状,但是如果步长太大，这可能不是可靠的纹理几何方法.

将常量添加到级别集将扩大或缩小几何体.再次考虑半径为r的球的水平集:

如果将加到的值上,很明显,我们得到了半径为的球面.通常适用:如果将一个正值添加到水平集，它将沿法线向内收缩该水平集距离;如果我们添加一个负值,则它将使该水平集沿“法线”向外增长或“扩张”距离.但是,请注意:除了在像球这样的非常简单的情况下,所得的水平集可能不再是真实的符号距离.进行这样的操作后,可能需要重新计算距离.

平滑和锐化滤镜也可以直接应用于级别集，就好像它们是任何其他3D数据一样。 这些通常具有预期的结果，除非过滤器内核与中间轴重叠（表面的两个或多个不同部分之间的“中途点”）：在这种情况下，来自表面另一部分的信息开始污染计算。 半径大于表面一部分厚度的平滑滤镜会导致该部分表面完全消失。 如果两个不相交的表面分量比过滤器内核的半径近，则它们可能会相互影响。

布尔或构造实体几何（CSG）运算可以非常容易地（至少近似地）使用水平集进行计算. 一个对象的补码（从内到外，反之亦然）可以简单地通过将符号反转为来实现.可以将具有水平集和的两个对象的并集计算为

交集为

在前者中，当且仅当该点在至少一个对象中（两个值的最小值为负）时，该点才在联合中，后者的逻辑类似。 请注意，采用两个内插值中的最小值可能与对最小值进行插值有很大不同：如果通过在其他级别集之间取最小值或最大值来计算在网格上采样的新级别集，然后使用标准插值来处理新级别集 ，您会找到较早的几何形状相交的“接缝”。 根据您的需求，这可能是好事也可能是坏事。 诸如联合和相交之类的布尔运算的结果在局部上仍然仍然满足Eikonal方程，但可能与真实的符号距离相去甚远：同样，可能需要重新分配距离。

4.6 等高线