

# 流体模拟（理论篇）

## 波动方程

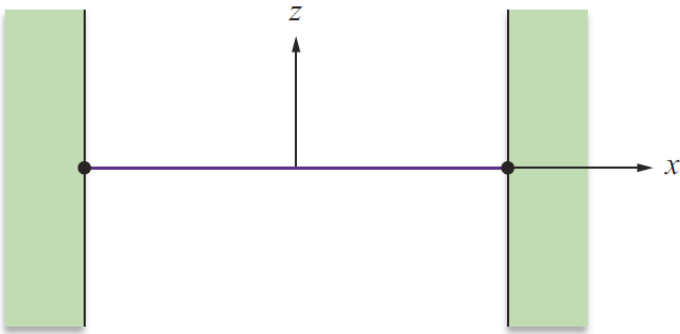
波动方程是用于描述一维弦上或者二维平面上点的运动的偏微分方程，其最简表示如下，是关于位置 $x$ 和时间 $t$ 的标量函数 $u$ （代表各点偏离平衡位置的距离）。本质上，这个公式左边其实是加速度。

$$\frac{\partial^2 \mu}{\partial t^2} = c^2 \Delta^2 \mu (c \text{ 通常是一个固定常数，代表波的传播速率})$$

下面来详细推导一个这个公式。

现在假设在一个一维的场景中，有一根密度线性均匀的弦，被固定在两个固定点位，如下图。

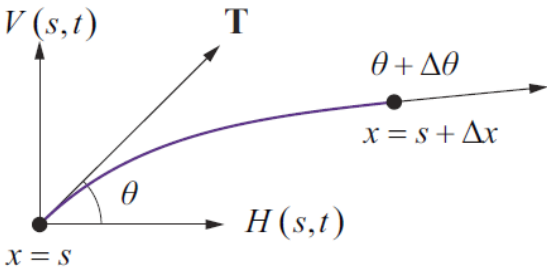
$\rho$ 为密度， $T$ 为恒定的张力



**Figure 15.1.** A string having linear density  $\rho$  is tightly bound between two endpoints and experiences a constant tension  $T$ .

令  $z(x,t)$  表示弦上一点 $x$ 在时间点 $t$ 下，垂直方向的位移。当弦的 $z$ 方向上位移时，张力会在弦上的每一点产生加速度。根据牛顿第二定律，力 $F(x, t)$ 可以表示为弦上某一小段的质量和加速度的乘积。

$$a(x,t) = \frac{F(x,t)}{\rho \Delta x}$$



如上图所示，我们可以把受的里分解为垂直跟水平两个方向，并且让 $\theta$ 表示张力 $T$ 跟水平轴的夹角。就可以得到水平跟竖直的力分别为

$$\begin{aligned} H(s,t) &= T \cos \theta \\ V(s,t) &= T \sin \theta \end{aligned}$$

现在让 $\theta + \Delta\theta$ 表示在 $s + \Delta x$ 点与x轴的正切角度，则水平跟竖直的力，在 $s + \Delta x$ 的位置为

$$H(s + \Delta x, t) = T \cos(\theta + \Delta\theta)$$

$$V(s + \Delta x, t) = T \sin(\theta + \Delta\theta)$$

对于比较微小的运动，可以假设水平方向的加速度是0，且跟x的位置无关了。

$$H(t) = a, (a \text{ 为常数，此处与原文有出入，包含个人理解})$$

因此，对于加速度的计算，可以得出一下公式，偏导只需考虑垂直方向了

$$a_z(s, t) = \frac{\partial^2}{\partial t^2} z(s, t) = \frac{V(s + \Delta x, t) - V(s, t)}{\rho \Delta x}$$

两边同时乘以密度，并且对 $\Delta x$ 求极限，可以得出

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = \lim_{\Delta x \rightarrow 0} \frac{V(s + \Delta x, t) - V(s, t)}{\Delta x}$$

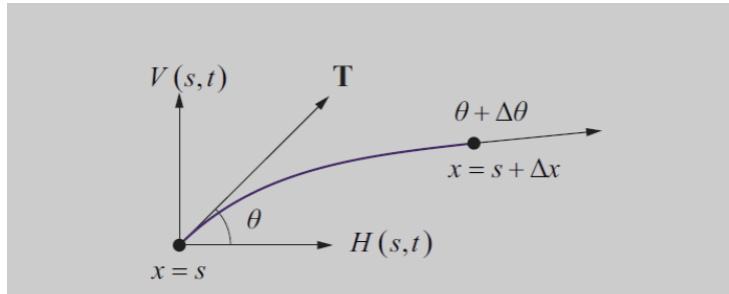
上式的右边，正好等于V方程对x的偏导，所以上式也可以改写成如下

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = \frac{\partial}{\partial x} V(s, t)$$

又通过正切函数，可以得到

$$V(s, t) = H(t) \tan \theta$$

这里的 $\theta$ 角度，正好表示的是弦的切线与水平轴的夹角，所以 $\tan \theta$ 正好就是斜率，可以再回顾一下这个图



因此，又可以把上式写为

$$V(s, t) = H(t) \frac{\partial}{\partial x} z(s, t)$$

可以得出

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = \frac{\partial}{\partial x} V(s, t) = \frac{\partial}{\partial x} [H(t) \frac{\partial}{\partial x} z(s, t)]$$

又因为H(t)是独立于x的值，因此可以把H(t)提出来，最终变成下式

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = H(t) \frac{\partial^2}{\partial x^2} z(s, t)$$

对于比较微小的运动，弦会比较贴近于水平， $\cos \theta$ 接近于1，这里可以把H(t)近似于张力T。让 $c^2$ 等于 $T/\rho$ ，可以得到以下式子

$$\frac{\partial^2 z}{\partial t^2} = c^2 \frac{\partial^2 z}{\partial x^2}$$

对于多维的情况，则是将各个分量上的偏导加起来就可以了，二维情况下可以得到

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left[ \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right]$$

常数c可以认为是速度，这里不需要证明，因为波的速度正比于T，且反比于密度。

上面的式子是不考虑除了表面张力以外的力的。因此表面波的振幅不会减少。我们可以添加一个与运动方向相反的阻尼力，如下

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left[ \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right] - \mu \frac{\partial z}{\partial t}$$

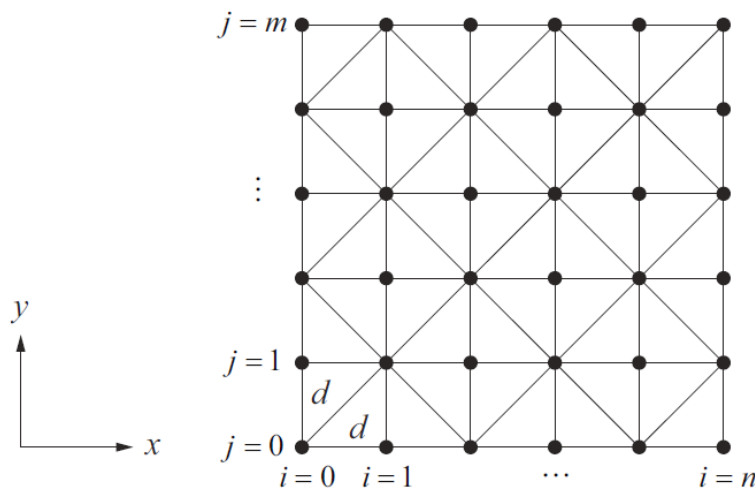
其中 $\mu$ 代表流体的粘性（粘稠度），用于控制波的衰减距离。

（这里其实不知道为啥阻尼力是用一阶导啊啊啊啊）

## 导数近似

上面带阻尼力的公式，是可以求出解析解的，但是由于计算量巨大，不适合实时模拟，因此我们选择使用数值的方式去建模波在流体表面的传播。

假如用一个Mesh去代表我们的水面，如下图



让d代表每个格子的距离(x轴方向跟y轴方向)，让t作为计算的时间间隔。z(i,j,k)代表水面的位移，ij分别是水平面的下标向量，k是时间轴向量。

z(i, j, k)代表点<i,d,j,d>在时间kt下，顶点的位移。

我们强制认为在边缘处的顶点始终是固定于高度0的位置。内部顶点的位置可以用上文的公式来计算，且导数可以使用相邻顶点的位移差来近似。

我们可以假设，x轴就是在表面的切线方向，然后计算前后相邻点的Delta位移的斜率均值（有点像插值的感觉），可以定义偏导数如下(d=Delta)

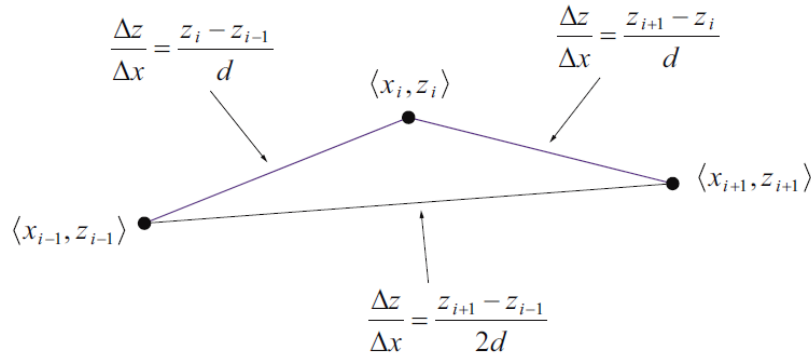
$$\frac{\partial}{\partial x} z(i, j, k) = \frac{\frac{z(i, j, k) - z(i-1, j, k)}{\Delta x} + \frac{z(i+1, j, k) - z(i, j, k)}{\Delta x}}{2} = \frac{z(i+1, j, k) - z(i-1, j, k)}{2\Delta x}$$

同理可以得出y轴偏导如下

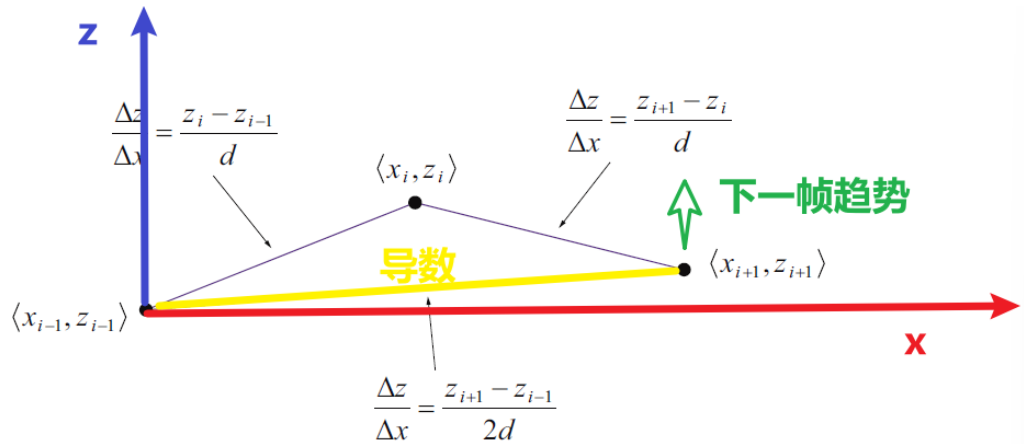
$$\frac{\partial}{\partial y} z(i, j, k) = \frac{z(i, j+1, k) - z(i, j-1, k)}{2\Delta y}$$

对于时间轴的偏导，也可以类似的计算

$$\frac{\partial}{\partial t} z(i, j, k) = \frac{z(i, j, k+1) - z(i, j, k-1)}{2\Delta t}$$



上图展示的计算过程，可以看出此时波的传播方向是沿着x轴，而当前时间点波峰是中间的点。



对于二阶偏导，也可以用同样的方法去近似。首先求一下前后一阶导的均值，如下

$$\Delta \frac{\partial}{\partial x} z(i, j, k) = \frac{\frac{\partial}{\partial x} z(i+1, j, k) - \frac{\partial}{\partial x} z(i-1, j, k)}{2}$$

然后将一阶导的近似计算公式带入可以得到

$$\Delta \frac{\partial}{\partial x} z(i, j, k) = \frac{\frac{z(i+2, j, k) - z(i, j, k) - z(i, j, k) + z(i-2, j, k)}{2\Delta x}}{2} = \frac{z(i+2, j, k) - 2z(i, j, k) + z(i-2, j, k)}{4\Delta x}$$

这时候，我们算出了Delta一阶导，需要求一阶导的变化率，则除以Delta x就可以了，得到

$$\frac{\partial^2}{\partial x^2} z(i, j, k) = \frac{z(i+2, j, k) - 2z(i, j, k) + z(i-2, j, k)}{4(\Delta x)^2}$$

上个公式的计算，需要采样到i+2跟i-2的位置，不过幸运的是，可以做一个优化，可以将Delta x缩小为原来的一半，那么i+1位置，其实对应的就是正常Delta下的i+1位置，于是式子可以变成

$$\frac{\partial^2}{\partial x^2} z(i, j, k) = \frac{z(i+1, j, k) - 2z(i, j, k) + z(i-1, j, k)}{(\Delta x)^2}$$

同理，可以分别算出y和t的二阶偏导

$$\frac{\partial^2}{\partial y^2} z(i, j, k) = \frac{z(i, j+1, k) - 2z(i, j, k) + z(i, j-1, k)}{(\Delta y)^2}$$

$$\frac{\partial^2}{\partial t^2} z(i, j, k) = \frac{z(i, j, k+1) - 2z(i, j, k) + z(i, j, k-1)}{(\Delta t)^2}$$

## 位移评估

算了这么多公式，现在我们需要用这些公式来做位移的评估。

首先，结合一阶偏导跟二阶偏导的近似公式，以及带有阻尼力的二维波动方程

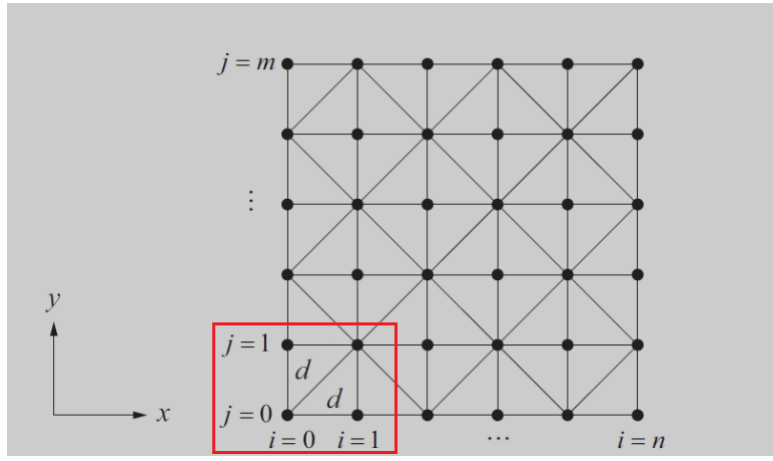
$$\frac{\partial^2 z}{\partial t^2} = c^2 \left[ \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right] - \mu \frac{\partial z}{\partial t}$$

把偏导近似公式代入上面的式子，可以得出

$$\begin{aligned} & \frac{z(i, j, k+1) - 2z(i, j, k) + z(i, j, k-1)}{t^2} = \\ & c^2 \frac{z(i+1, j, k) - 2z(i, j, k) + z(i-1, j, k)}{(\Delta x)^2} \\ & + c^2 \frac{z(i, j+1, k) - 2z(i, j, k) + z(i, j-1, k)}{(\Delta y)^2} \\ & - \mu \frac{z(i, j, k+1) - z(i, j, k-1)}{2t} \end{aligned}$$

现在，我们可以直接预测出t=k+1时的位移了，可以从上述式子知道，可以由t=k跟t=k-1的位移得到。

$$\begin{aligned} z(i, j, k+1) &= \left[ c^2 \frac{z(i+1, j, k) - 2z(i, j, k) + z(i-1, j, k)}{(\Delta x)^2} \right. \\ & \quad + c^2 \frac{z(i, j+1, k) - 2z(i, j, k) + z(i, j-1, k)}{(\Delta y)^2} \\ & \quad \left. - \mu \frac{z(i, j, k+1) - z(i, j, k-1)}{2t} \right] t^2 + 2z(i, j, k) - z(i, j, k-1) \\ &= \left[ c^2 t^2 \frac{z(i+1, j, k) - 2z(i, j, k) + z(i-1, j, k)}{(\Delta x)^2} \right. \\ & \quad + c^2 t^2 \frac{z(i, j+1, k) - 2z(i, j, k) + z(i, j-1, k)}{(\Delta y)^2} \\ & \quad \left. - \mu t \frac{z(i, j, k+1) - z(i, j, k-1)}{2} \right] + 2z(i, j, k) - z(i, j, k-1) \end{aligned}$$



如上图，由于之前我们已经假设了  $\Delta x = \Delta y = d$ , 现在代入可以得到

$$z(i, j, k+1) = \frac{4 - 8c^2t^2/d^2}{\mu t + 2} z(i, j, k) + \frac{\mu t - 2}{\mu t + 2} z(i, j, k-1) + \frac{2c^2t^2/d^2}{\mu t + 2} [z(i+1, j, k) + z(i-1, j, k) + z(i, j+1, k) + z(i, j-1, k)]$$

从上面的式子可以看出，如果  $c$  或者  $t$  特别大，则会导致方程的迭代变得越来越大。因此需要设定一个条件让方程稳定。

现在我们假设在这个网格中， $t=0$  是，所有顶点高度为 0，除了坐标为  $\langle i_0, j_0 \rangle$  的点。让这个点在  $t=0$  跟  $t=1$  时，高度都是  $h$ 。现在假设这个点在  $t=2$  时被释放了，这时候，我们代入式子评估  $t=2$  时的高度值，得到如下等式

$$\begin{aligned} z(i_0, j_0, 2) &= \frac{4 - 8c^2t^2/d^2}{\mu t + 2} z(i_0, j_0, 1) + \frac{\mu t - 2}{\mu t + 2} z(i_0, j_0, 0) \\ &= \frac{2 - 8c^2t^2/d^2 + \mu t}{\mu t + 2} h \end{aligned}$$

此时，因为外力释放了，所以  $t=2$  时的高度，必定是要小于  $t=1$  时的高度

$$|z(i_0, j_0, 2)| < |z(i_0, j_0, 1)| = |h|$$

把公式代入得到

$$-1 < \frac{2 - 8c^2t^2/d^2 + \mu t}{\mu t + 2} < 1$$

然后求解  $c$  的不等式，可以得到

$$0 < c < \frac{d}{2t} \sqrt{\mu t + 2}$$

另外，我们可以根据给定的  $d$  和  $c$  来计算最大的时间间隔  $t$ 。

$$0 < \frac{4c^2}{d^2}t^2 < \mu t + 2$$

$$\Rightarrow \frac{4c^2}{d^2}t^2 - \mu t - 2 < 0$$

$$\text{求解}t\text{可以得出}t = \frac{\mu \pm \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2}$$

由于关于t一次二次方程中的a是正的，所以抛物线开口朝上，因此当t位于两个解之间时，方程的值为负值。同时，当t的解析式中根号内的值大于u时，t是负值，可以直接舍去，可以得到t的范围不等式

$$0 < t < \frac{\mu + \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2}$$

## 应用

现在来介绍一下实际的应用场景。在渲染流体表面的时候，可以用一个buffer存储当前的顶点位置，另一个buffer存储上一帧的顶点位置。当计算新的位置是，直接把最新的位置存储到上一帧的buffer中，这样当前帧的会变成上一帧，类似ABbuffer的操作。

在计算光照的时候，我们需要知道正确的法线和切线。在<i, j>顶点处，Tangent跟Bitangent可以表示成如下公式

$$T = \langle 1, 0, \frac{\partial}{\partial x} z(i, j, k) \rangle$$

$$B = \langle 0, 1, \frac{\partial}{\partial y} z(i, j, k) \rangle$$

把偏导公式代入得

$$T = \langle 1, 0, \frac{z(i+1, j, k) - z(i-1, j, k)}{2d} \rangle$$

$$B = \langle 0, 1, \frac{z(i, j+1, k) - z(i, j-1, k)}{2d} \rangle$$

法线N则可以由切线T跟副切线B叉乘得出， $N = T \times B$

$$N = \begin{vmatrix} i & j & k \\ 1 & 0 & \langle 1, 0, \frac{z(i+1, j, k) - z(i-1, j, k)}{2d} \rangle \\ 0 & 1 & \langle 0, 1, \frac{z(i, j+1, k) - z(i, j-1, k)}{2d} \rangle \end{vmatrix}$$

$$= \langle -\frac{z(i+1, j, k) - z(i-1, j, k)}{2d}, -\frac{z(i, j+1, k) - z(i, j-1, k)}{2d}, 1 \rangle$$

把TBN分别都放大2d倍，不会改变它的方向性，可以得到

$$T = \langle 2d, 0, z(i+1, j, k) - z(i-1, j, k) \rangle$$

$$B = \langle 0, 2d, z(i, j+1, k) - z(i, j-1, k) \rangle$$

$$N = \langle z(i-1, j, k) - z(i+1, j, k), z(i, j-1, k) - z(i, j+1, k), 2d \rangle$$

需要注意的是，流体表面的模拟，需要相对恒定的时间间隔，因此在游戏实际实现中，应该使用某种机制来避免时间间隔受帧率的影响（譬如当游戏帧率特别高的情况下，需要限制达到一定间隔后，才去更新参数）。

总结，在实际计算的时候，我们需要先计算出高度信息，来位移顶点，然后再重建法线信息去进行光照着色，因此主要用到的公式有以下

$$z(i, j, k+1) = \frac{4 - 8c^2t^2/d^2}{\mu t + 2} z(i, j, k) + \frac{\mu t - 2}{\mu t + 2} z(i, j, k-1) \\ + \frac{2c^2t^2/d^2}{\mu t + 2} [z(i+1, j, k) + z(i-1, j, k) + z(i, j+1, k) + z(i, j-1, k)]$$

$$T = \langle 2d, 0, z(i+1, j, k) - z(i-1, j, k) \rangle$$

$$B = \langle 0, 2d, z(i, j+1, k) - z(i, j-1, k) \rangle$$

$$N = \langle z(i-1, j, k) - z(i+1, j, k), z(i, j-1, k) - z(i, j+1, k), 2d \rangle$$

## 工程实现