

# MATLAB大作业——运动电荷电场模拟

刘畅 基应01 2010012176

## 1 摘要

静止电荷的电场较为简单且很容易画出，但运动电荷的电场就要复杂得多。本次大作业模拟运动电荷的电场，将电荷在不同运动状态下每一时刻的电场线画出，并利用电影动画展示一个电场随时间变化的动态效果。目前支持的电荷运动方式有匀速直线、圆周和简谐振动。

## 2 数学模型

根据电动力学理论，由于电磁场在空间中的传播的速度是有限的，所以当电荷运动时，在空间中某一点的电场并不是由此时刻的电荷产生的，而是由之前某一时刻的电荷产生的，即推迟效应。这个对应于这一点的之前的某一时刻称为这一点的推迟时刻，推迟时刻电荷产生的电场在当前时刻刚好传播到对应推迟时刻的那一点处。考虑推迟效应的运动电荷的场的势称为李纳-维谢尔势。

## 3 算法

根据电动力学理论，要计算当前时刻某一点的电场，首先要确定对应这点的推迟时间。以下，令 $\vec{r}$ 为场点位置， $t$ 为当前时刻， $\vec{E}(\vec{r}, t)$ 为当前时刻场点处的电场， $\vec{r}_0(t)$ 为 $t$ 时刻电荷的位置， $t^*$ 为对应于场点 $\vec{r}$ 和当前时刻 $t$ 的推迟时刻， $\vec{r}^*$ 、 $\vec{v}^*$ 、 $\vec{a}^*$ 分别表示在推迟时刻 $t^*$ 电荷的位置、速度和加速度， $c$ 为电场传播速度。则：

$$|\vec{r} - \vec{r}^*| - c(t - t^*) = |\vec{r} - \vec{r}_0(t^*)| - c(t - t^*) = 0 \quad (1)$$

在确定了 $t$ 和 $\vec{r}$ 以及函数 $\vec{r}_0(t)$ 之后，即可通过式1来确定推迟时刻 $t^*$ 。在实际计算中，由于时刻 $t$ 是离散的，且作为推迟时刻即过去的时刻， $t^*$ 应为 $t$ 中的某一个元素。所以在确定 $t^*$ 时，计算估计出来的 $t^*$ 所在的区间中的每一个时刻点的 $|\vec{r} - \vec{r}_0(t^*)| - c(t - t^*)$ 的值，挑选其中最小的对应的 $t^*$ 即为推迟时刻。继而可以计算 $\vec{r}^*$ 、 $\vec{v}^*$ 、 $\vec{a}^*$ 等。

有了上面的准备，可以根据电动力学理论给出运动电荷的电场为（ $\epsilon_0$ 为真空中介电常数）：

$$\vec{E}(\vec{r}, t) = \frac{q}{4\pi\epsilon_0 S^{*3}} \left[ \left( 1 + \frac{\vec{R}^* \cdot \vec{a}^* - v^{*2}}{c^2} \right) \vec{Q}^* - \left( \frac{\vec{R}^* \cdot \vec{Q}^*}{c^2} \right) \vec{a}^* \right] \quad (2)$$

其中

$$\begin{aligned}\vec{R}^* &= \vec{r} - \vec{r}^* \\ \vec{Q} &= \vec{R}^* - \frac{R^*}{c} \vec{v}^* \\ S^* &= R^* - \frac{\vec{R}^* \cdot \vec{v}^*}{c}\end{aligned}\quad (3)$$

在得到推迟时刻之后，由上式即可计算出运动电荷的电场。编程时， $c$ 取为1，即传播时间的数值等于电场传播距离的数值。

以上为基本算法，在进行实际计算时，还有几点要说明：

产生场的时刻不是运动开始的时刻，早在电荷开始运动之前场边早已产生。所以在进行时间分划时，应在开始时刻之前给出一段时间的静电场，电荷的位置就在运动一开始时的位置，产生静电场的时间（代码中的**Tpre**）应大于电场传遍整个观测空间的时间。

由于电荷运动的形式由一个函数给出，即由一个符号型变量给出，所以在计算每一时刻的 $r_0$ 、 $\vec{v}$ 、 $\vec{a}$ 时都需要用到**subs(fun,old,new)**函数。当**new**是一个数组时，返回值也应该为一个同阶的数组，对应位置上的元素是自变量与函数值的关系。但若**fun**中不含**old**，即函数为常函数，应该返回一个与**new**同阶的元素均为**fun**的数组，但**subs**函数给出的却只是一个等于**fun**的数而不是数组。为解决这个问题，另外编写了函数**NewSubs(fun,old,new)**，在**fun**为常数时返回一个与**new**同阶的数组。生成 $r_0$ 、 $\vec{v}$ 、 $\vec{a}$ 时均用**NewSubs**函数。

绘制某个时间点的直接命令是调用**streamline**函数，调用此函数需要指定场线的起始点。起始点取为电荷周围以5倍空间间隔为半径的圆周上均匀分布的点。若半径太小，由于靠近电荷的场的离散性较强，产生的电场线会出现较为明显的摆动。而若半径太大，靠近电荷附近的电场又无法显示。所以采用了5倍空间间隔为半径。

为提高计算效率，在计算某一时刻整个观测空间中的推迟时间场和电场时，应尽量采用MATLAB擅长的矩阵运算，而不是像C等语言中的元素运算（即使用二重循环为场赋值）。经过测试，在运行**ElecField(8,'harmonic',0.8)**时，计算每个时间点的电场平均耗时为0.4秒，而在各个参数均相同的情况下，用采用元素运算方式的程序（**ElecField**的最初版本）计算每个时间点的电场平均耗时为2.7秒，二者差距约7倍。可见采用矩阵运算是十分必要的。

在采用矩阵运算时遇到的最大问题是，无法通过普通的矩阵乘法生成三维矩阵。为解决这一问题，可以充分利用**reshape**函数，将二维矩阵先转化成列向量，然后右乘待乘的行向量，得到一个二维矩阵，再利用**reshape**函数将这个二维矩阵转化为一个三维矩阵。第三维的维数即为待乘的行向量的维数。由于**reshape**函数将矩阵元按列排列，所以得到的二维矩阵的每一列即为一个原来的二维矩阵的展开。在利用上面叙述的算法计算推迟时刻时用到了**min**函数，但求最小是对时间进行的，也就是对三维矩阵的第三维进行的，所以用到了**min(3Dmatrix,[],3)**，返回值将是一个和观测空间同维的指标场，对于一个一维向量，将得到的指标场作为下标即可得到一个这个量对应的场（二维向量）。

以上只是一些想法上的叙述，具体实现方法请参看程序代码。代码中有足够的注释。

## 4 结果分析

通过上述算法，可以得到演示运动电荷电场的函数`ElecField(T,type,k)`，其中`T`是运动（场）持续的时间，`type`是电荷运动的形式，目前有`'harmonic'`（简谐振动）、`'line'`（直线运动）和`'circle'`（圆周运动），`k`是电荷运动的最大速度与光速（已取为1）的比值。运行此函数可以得到较好的模拟结果。之后可考虑将此程序利用GUI来实现。

下图为运行`ElecField(10,'harmonic',0.8)`时某时刻的图像。可以看出效果还是很好的。

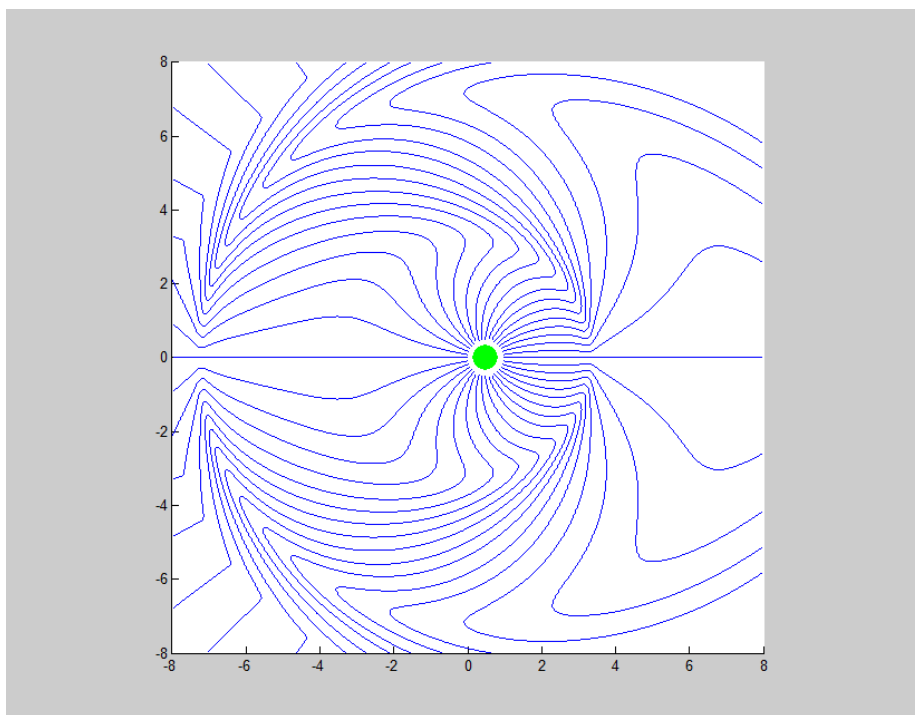


Figure 1: 模拟简谐运动时某一时刻的电场