

用python仿真带电粒子的运动

——计算物理期末论文

梁旭 物基一班 2015301110077

摘要：本文主要讨论了单粒子和多粒子在电磁场中运动的情况。对于单粒子体系，本文主要研究的是单个粒子在匀强磁场和匀强电场中的运动方式，并利用Python模拟了其运动轨迹。对于多粒子体系，本文主要研究的是多个粒子间的Lorentz力对其运动方式的影响，并同样利用Python模拟了其运动轨迹。在得到Python模拟的运动轨迹基础上，本文还对每种运动方式背后的隐含的物理意义进行了说明。最后，写了一些自己的感想与收获。

关键词：单粒子；多粒子；Lorentz力；Python

0 引言

在这里谈一谈自己选题的原因吧：

因为，上学期选修了缪庆元老师的MATLAB课程，学习了一些简单的MATLAB的应用。在最终的期末考核中，我就利用MATLAB写了一个关于粒子在电磁场中运动轨迹的研究有关的论文。这学期学习了Python以后，发现Python也可以做同样的事，而且在某种程度上可以做的更好。于是，就在百度上搜索了相关的内容后，庆幸的发现：竟然没有人这样模拟过！然后，又在github上搜索了相关的内容，发现也很少有人利用python进行相关的模拟，大多数都是用的matlab进行模拟。（可能是大神们觉得这个模拟太简单了吧。）所以自己就选了这个题目来做。

还有一方面的原因是，为了使自己的作业看起来很新颖，因为，在看了学长学姐的作业以后，发现大多数人都都在做“随机行走”有关的内容。也算是为了体现自己的所有水平吧，就选了这个题目，毕竟可以避免自己“抄袭”的嫌疑。

（注：本文讨论的均为非相对论粒子）

1 单个带电粒子在电磁场中的运动

1.1 带电粒子在电磁场中运动的原理

带电粒子在磁场中运动时会受到洛伦兹力的作用，且随着初始运动方向和磁场分布的不同，其运动轨迹会发生不同的变化。由洛伦兹力的推导公式可知，它垂直于粒子的运动速度，不对运动粒子做功，只改变其运动方向，其大小为：

$$F = qvB\sin\theta \quad (1)$$

其中， θ 为粒子速度与磁场方向的夹角因此，综合牛顿运动定律就可以精确确定带电粒子在磁场中的运动轨迹。

设带电粒子质量为 m ，带电量为 q ，电场强度 E 沿 y 方向，磁感应强度 B 沿 z 方向。则带电粒子在均匀电磁场中的运动微分方程为

$$\frac{dx}{dt} = v_x \quad (2)$$

$$\frac{dv_x}{dt} = \frac{d^2x}{dt^2} = \frac{qB}{m} v_y \quad (3)$$

$$\frac{dy}{dt} = v_y \quad (4)$$

$$\frac{dv_y}{dt} = \frac{d^2y}{dt^2} = \frac{qE}{m} - \frac{qB}{m} v_x \quad (5)$$

$$\frac{dz}{dt} = v_z \quad (6)$$

$$\frac{dv_z}{dt} = \frac{d^2z}{dt^2} = 0 \quad (7)$$

1.2 带电粒子在电磁场中运动的 Python 模拟

为了可以利用 Python 模拟粒子运动，利用教材中介绍的欧拉法，则上面微分方程可化作：

$$x_{i+1} = x_i + v_{x,i} \Delta t \quad (8)$$

$$v_{x,i+1} = v_{x,i} + \frac{qB}{m} v_{y,i} \Delta t \quad (9)$$

$$y_{i+1} = y_i + v_{y,i} \Delta t \quad (10)$$

$$v_{y,i+1} = v_{y,i} + \left(\frac{qE}{m} - \frac{qB}{m} v_{x,i} \right) \Delta t \quad (11)$$

$$z_{i+1} = z_i + v_{z,i} \Delta t \quad (12)$$

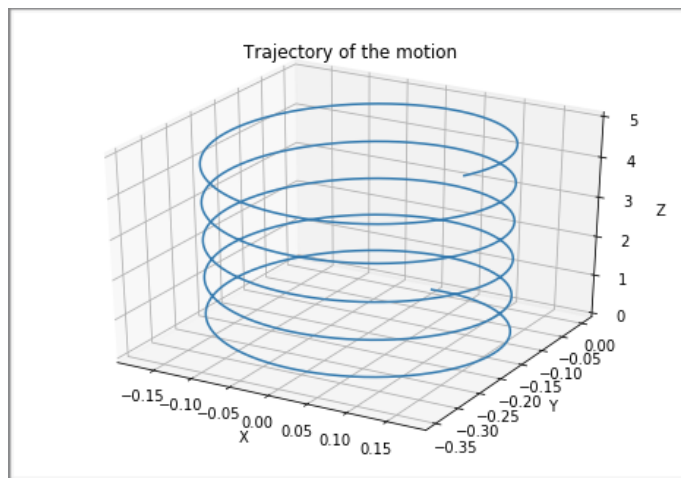
$$v_{z,i+1} = v_{z,i} \quad (13)$$

为了简化运算，将单粒子取为质子，即 $e = 1.6 \times 10^{-19} C$ ， $m_p = 1.6 \times 10^{-27} kg$ ，则其电荷质量比为： $h = \frac{q}{m} = 10^8 c/kg$ 。同时，把粒子的速度取为光速的1/10，即 $v = 3 \times 10^7 m/s$ ，设粒子初始时刻的速度在 x - z 平面内，其沿 x 轴的夹角取为 $\frac{\pi}{6}$ ，则 $v_x = 1.732 \times 10^7 m/s$ ， $v_y = 0$ ， $v_z = 1.5 \times 10^7 m/s$ 。取 $B=1T$ 后，不难看出（11）式中 $\frac{qB}{m} v_{x,i}$ 这一项将会变得特别大，为了得出精确的结果，就需要步长 Δt 特别小。测过我的不断测试，发现取步长 Δt 取0.0000000001或更小时，可以绘制出理想的图形。

1.3 Python 模拟结果的讨论

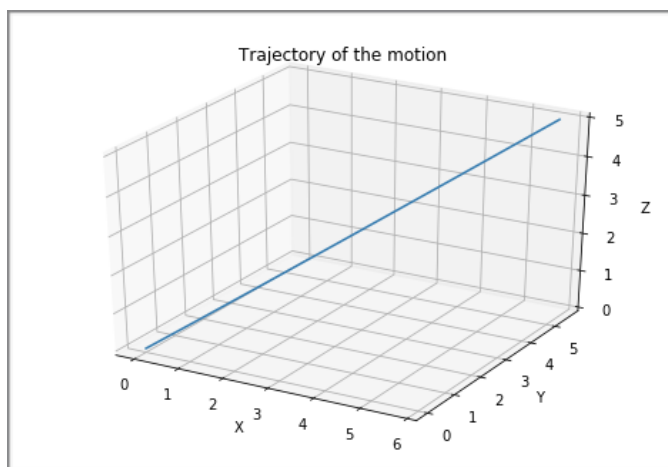
a. 取 $E=0V/m$ ， $B=1T$ 时，利用Python所得结果如图（1）所示的结果：

图（1）是带电粒子在 $E=0\text{ V/m}$ ， $B=1\text{ T}$ 的电磁场中运动时的轨迹，此时带电粒子只受到洛伦兹力的作用。而由于洛伦兹力的作用效果是只改变粒子运动的方向而不改变其大小。故带电粒子的运动轨道是半径不变的圆轨道。而又由于粒子沿 z 轴方向有一个速度，这个速度导致粒子不断向上运动。因而，粒子总的运动趋势是一个沿 z 轴向上的螺旋运动。



图（1）

b. 取 $E=1 \times 10^6\text{ V/m}$ ， $B=0\text{ T}$ 时，利用Python所得结果如图（2）所示的结果：

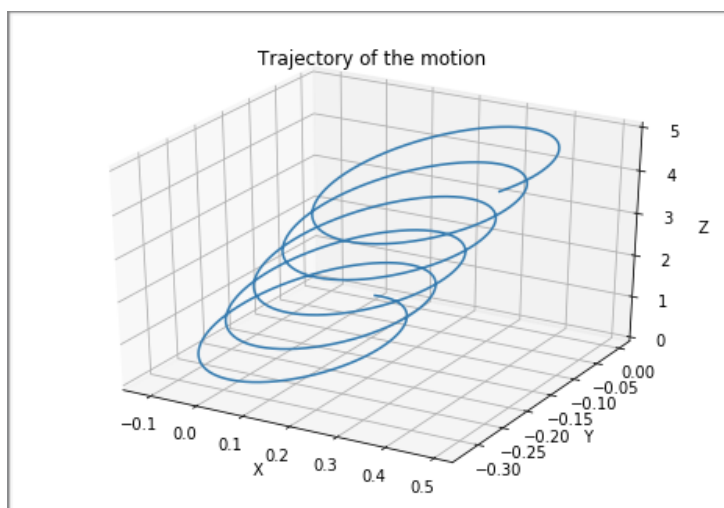


图（2）

图（2）是带电粒子在 $E=1 \times 10^6\text{ V/m}$ ， $B=0\text{ T}$ 的电磁场中运动时的轨迹，此时带电粒子只受到电场力的作用。此时，粒子运动是三个分运动的叠加：沿 x 轴方向的匀速直线运动，沿 y 轴方向的匀加速直线运动和沿 z 轴方向的匀速直线运动。此时，带电粒子的运动轨迹是 是一条曲线

c. 取 $E=1 \times 10^6\text{ V/m}$ ， $B=1\text{ T}$ 时，利用Python所得结果如图（3）所示的结果：

图（3）是带电粒子在 $E=1 \times 10^6\text{ V/m}$ ， $B=1\text{ T}$ 的电磁场中运动时的轨迹。此时，粒子要受到电场力和洛伦兹力的作用综合作用。电场力会改变粒子的速度大小和方向，洛伦兹力只能改变速度方向。由图不难看出，粒子的总的运动轨迹是一个 向右向上加速螺旋运动。本质来说，图（3）是图（1）和图（2）的叠加，图（2）其实描述的是图（3）圆心运动的趋势。



图（3）

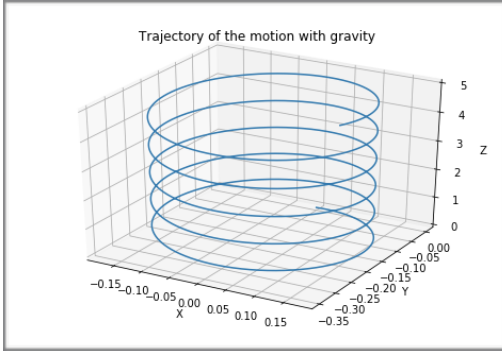
d. 由于粒子在实际电场中均要受到重力的影响，在这里我们考虑粒子收到的重力：则上述（7）式将变为

$$\frac{dv_z}{dt} = \frac{d^2z}{dt^2} = -g \quad (14)$$

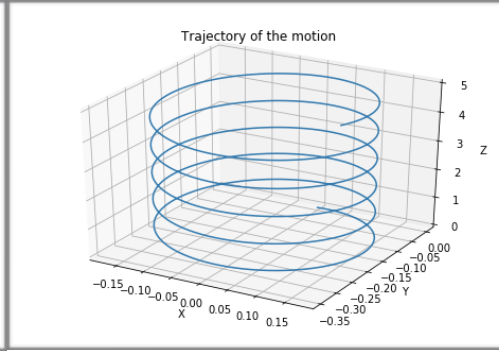
上述第（13）式变为：

$$v_{z,i+1} = v_{z,i} - g\Delta t \quad (15)$$

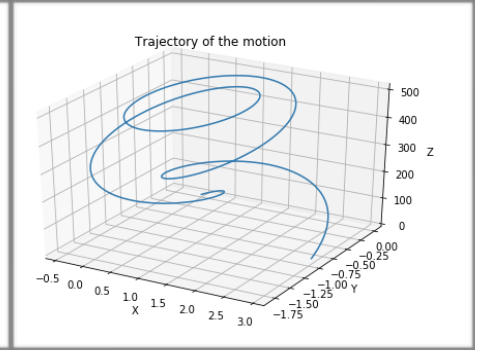
取 $E=0V/m$, $B=1T$ 时，利用Python所得结果如图（4）所示的结果：



图（4）



图（1）



图（5）

通过对比图（4）与图（1）中的结果，我们可以得出一个结论：在粒子处于高速运动状态时，其自身的重力对其运动轨迹的影响可以忽略不计。图（5）为粒子的速度取为 $200m/s$ 时，考虑粒子重力时绘制出的曲线实际。不难看出，对于低速运动的粒子，重力对其运动轨迹的影响很大。

因而，在实际的回旋粒子加速器中，由于粒子的运动速度极大，实际的运动时间较短，所以不必考虑粒子的重力因素，因而加速器管道不必建的很粗。

2 多个带电粒子在电磁场中的运动

2.1 多个带电粒子在电磁场中运动的原理

在二维条件下，我们考虑由于一个带点粒子的电场对其他粒子运动的影响：

由库仑定律，真空中两个点电荷之间的相互作用力：

$$\vec{F}_{21} = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{r_{21}^3} \vec{r}_{21} \quad (16)$$

由此可以计算出点电荷 q_1 在 \vec{r}_{21} 处激发的电场为：

$$\vec{E}_{21} = \frac{1}{4\pi\epsilon_0} \frac{q_1}{r_{21}^3} \vec{r}_{21} \quad (17)$$

而当粒子运动时，由电动力学相关知识可知，它还将激发磁场：

当一个点电荷以速度 \vec{v} 运动时，由电流密度矢量 \vec{j} 的定义：

$$\vec{j} = \rho \vec{v} = q \vec{v} \quad (18)$$

根据Biot-Savart 定律， 磁感应强度定义为：

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \int_{V'} \frac{\vec{j}(\vec{r}') \times \vec{r}_{12}}{r_{12}^3} d\tau' \quad (19)$$

于是，对于一个做直线运动的粒子，由于其运动产生的磁感应强度为：

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \frac{q \vec{v} \times \vec{r}_{12}}{r_{12}^3} d\tau' \quad (20)$$

再根据洛伦兹力密度的定义：

$$\vec{f} = \rho \vec{E} + \rho \vec{v} \times \vec{B} \quad (21)$$

而对于一个做直线运动的粒子，上式将变为：

$$\vec{f} = q \vec{E} + q \vec{v} \times \vec{B} \quad (22)$$

2.2 多个带电粒子在电磁场中运动的 Python 模拟

为了可以利用在具体模拟的过程中可以更好的调试程序，我将程序分为了四个模块：

a. 二维欧拉法模块：此模块的目的是利用欧拉法来模拟粒子的运动：

$$x_{i+1} = x_i + v_{x,i} \Delta t \quad (23)$$

$$v_{x,i+1} = v_{x,i} + a \Delta t \quad (24)$$

$$y_{i+1} = y_i + v_{y,i} \Delta t \quad (25)$$

$$v_{y,i+1} = v_{y,i} + a \Delta t \quad (26)$$

b. 场模块：此模块的目的是根据粒子的间的相对位置，利用公式（17）和（20）计算出电场强度和磁感应强度。

由（17）和（18）式，可以看出，当粒子间的距离为 $\vec{r}_{12}=0$ 时， \vec{B}, \vec{E} 将变为 ∞ ，为了避免这种情况，在此模块中要预先定义好，当 $\vec{r}_{12}=0$ 时 $\vec{B}, \vec{E}=0$ 。（具体这样定义的原因将在本文下面的讨论中给出）在将公式转换为具体的计算代码时要用到 Numpy 数据包的一些基本知识，这里就不过多介绍。

c. 粒子间洛伦兹力的计算模块：此模块的目的是根据场模块中计算出来的电磁场来计算的电场强度和磁感应强度，带入到（22）式中计算出粒子间的相互作用力。

d. 运动轨迹绘制模块：此模块的目的是根据 粒子间洛伦兹力的计算模块计算出的粒子间的洛伦兹力，然后就可以计算出a，带入到二维欧拉法模块就可以绘制出粒子的运动轨迹。

2.3 Python 模拟结果的讨论

2.3.1 两个带等量电荷粒子运动的模拟

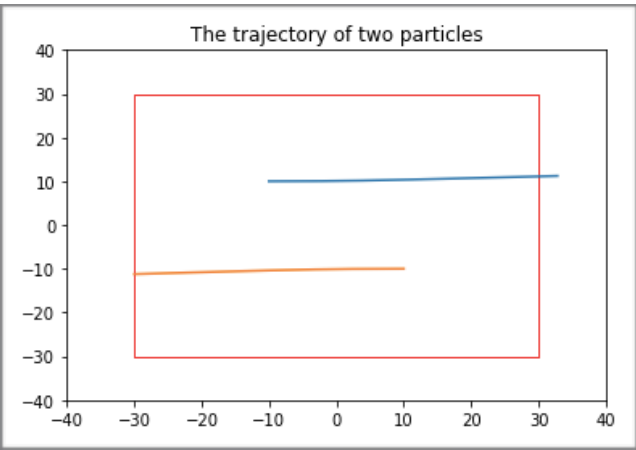


图 (6)

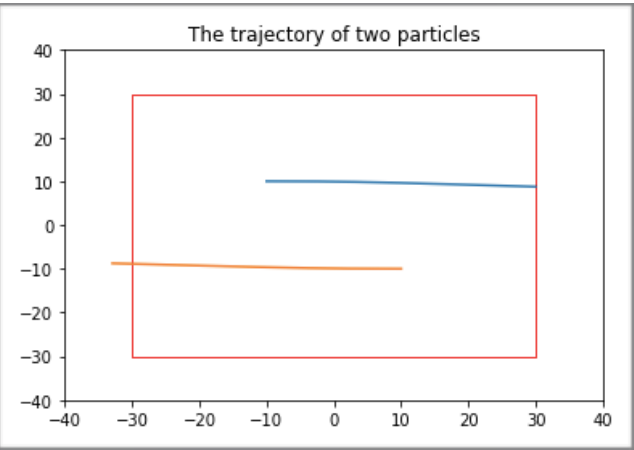


图 (7)

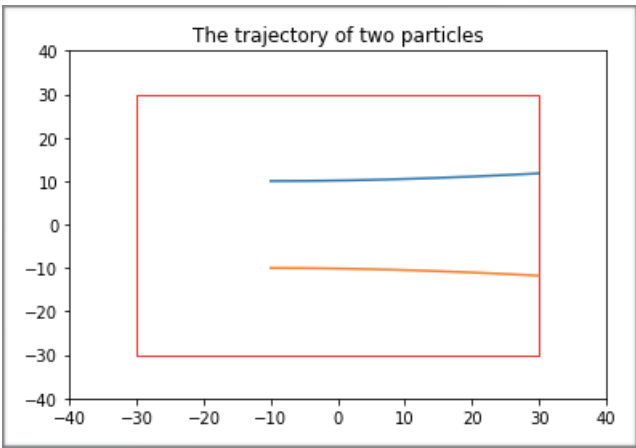


图 (8)

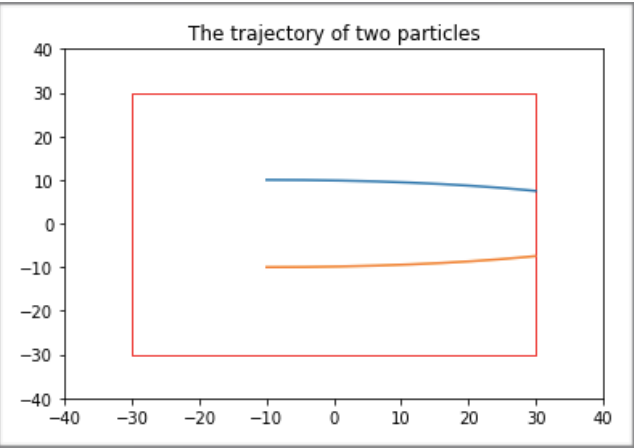


图 (9)

如图 (6) 所示，在0时刻，处于 $(-10, 10)$ 和 $(10, -10)$ 的两个带正电的粒子沿 x 轴正向运动，两个粒子的运动轨迹将由于它们之间的洛伦兹力的作用而相互偏离。

如图 (7) 所示，在0时刻，处于 $(-10, -10)$ 和 $(10, -10)$ 的一个带正电，一个带负电的粒子沿 x 轴正向入射，两个粒子的运动轨迹将由于它们之间的洛伦兹力的作用而相互靠近。

如图 (8) 所示，在0时刻，处于 $(-10, 10)$ 和 $(10, -10)$ 的两个带正电的粒子，一个沿 x 轴正向入射，一个沿 x 轴反向入射，两粒子将由于它们之间的洛伦兹力作用效果而相互偏离。通过与图 (6) 的对比可以看出，此时两粒子的偏离程度小于两个粒子并行同向入射的偏离程度。

如图 (9) 所示, 在0时刻, 处于 $(-10, 10)$ 和 $(10, -10)$ 的两个带电粒子, 一个带正电沿 x 轴正向入射, 一个带负电沿 x 轴反向入射, 两个粒子的运动轨迹将由于它们之间的洛伦兹力的作用而相互靠近。通过与图 (7) 的对比可以看出, 此时两粒子的偏离程度小于两个粒子并行同向入射的偏离程度。

2.3.2 四个带等量电荷的粒子运动的模拟

为了便于讨论, 将处于 $(-20, 20)$ 位置的粒子标记为粒子1, 粒子1沿 y 轴反向入射; 处于 $(-20, -20)$ 位置的粒子标记为粒子2, 粒子2沿 x 正向入射; 处于 $(20, -20)$ 位置的粒子标记为粒子3, 粒子3沿 y 轴正向入射; 处于 $(20, 20)$ 位置的粒子标记为粒子4, 粒子4沿 x 轴反向入射。

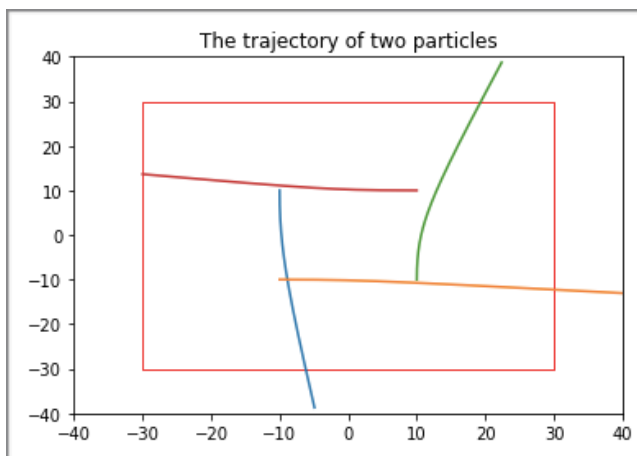


图 (10)

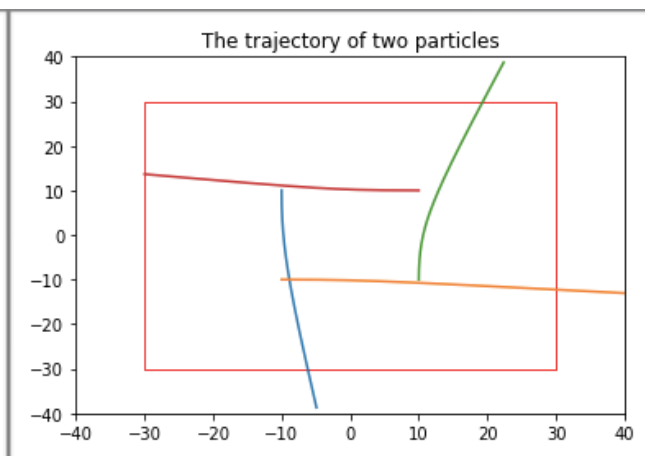


图 (11)

图 (10) 为, 4个粒子均带正电时的模拟结果; 图 (11) 是将粒子全部换成带负电荷后的模拟结果。我们发现粒子的运动轨迹模拟图完全相同。这是因为, 如果粒子都是同种电性, 不管是其同为正电还是同为负电, 粒子间的洛伦兹力作用效果相同。

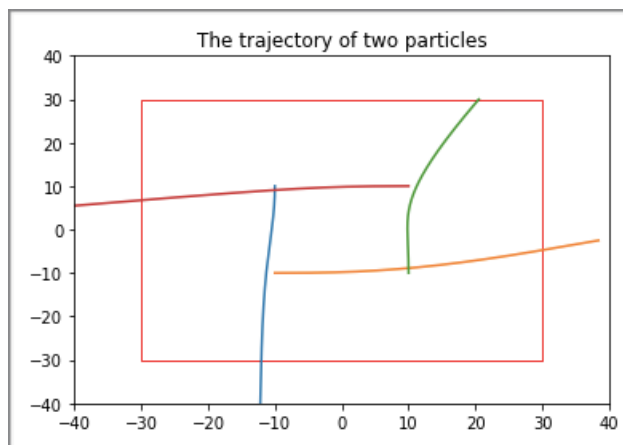


图 (12)

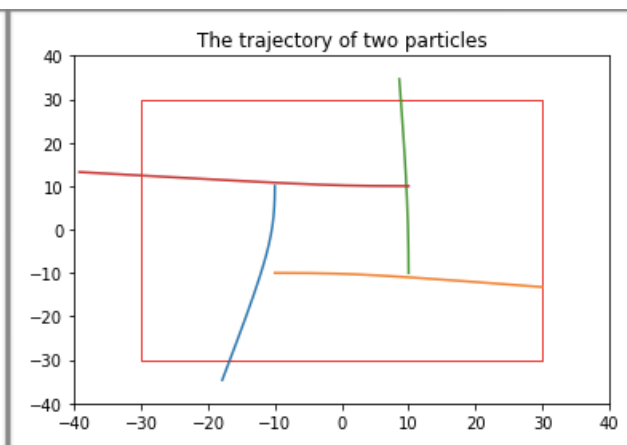
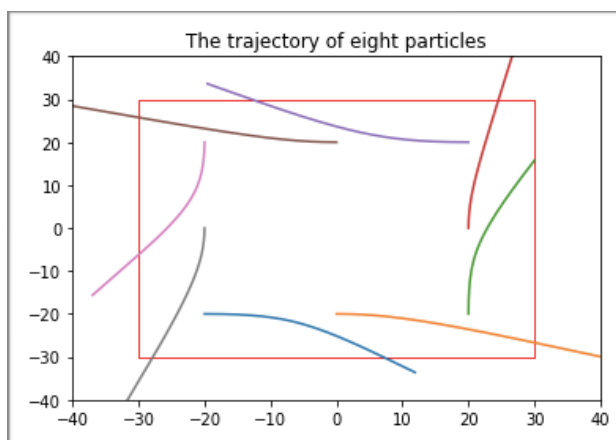


图 (13)

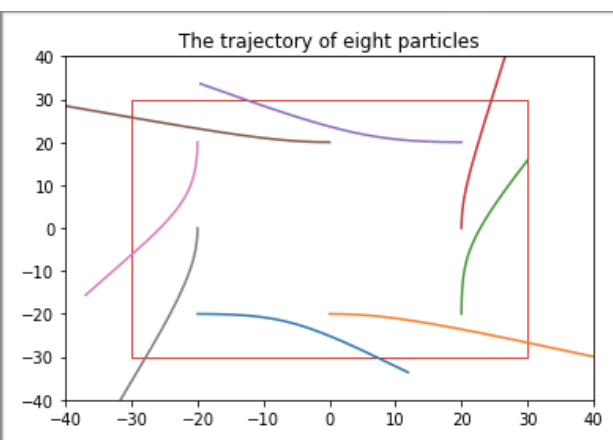
图（12）是将粒子2换为带等量负电后的模拟结果，图（13）是将粒子2和粒子4换成是带等量负电荷时的模拟结果。可以发现，一旦改变其中一个或几个粒子的电性，粒子的运动轨迹将发生很大的变化。

2.3.3 八个带等量电荷的粒子运动的模拟结果：

为了便于讨论，将处于 $(-20, 20)$ 位置的粒子标记为粒子1，粒子1沿 y 轴反向入射；处于 $(-20, 0)$ 位置的粒子标记为粒子2，粒子2沿 y 轴反向入射；处于 $(-20, -20)$ 位置的粒子标记为粒子3，粒子3沿 x 正向入射；处于 $(0, -20)$ 位置的粒子标记为粒子4，粒子4沿 x 轴正向入射；处于 $(20, -20)$ 位置的粒子标记为粒子5，粒子5沿 y 轴正向入射，处于 $(20, 0)$ 位置的粒子标记为粒子6，粒子6沿 y 轴正向入射；处于 $(20, 20)$ 位置的粒子标记为粒子7，粒子7沿 x 轴反向入射；处于 $(0, 20)$ 位置的粒子标记为粒子8，粒子8沿 x 轴反向入射。

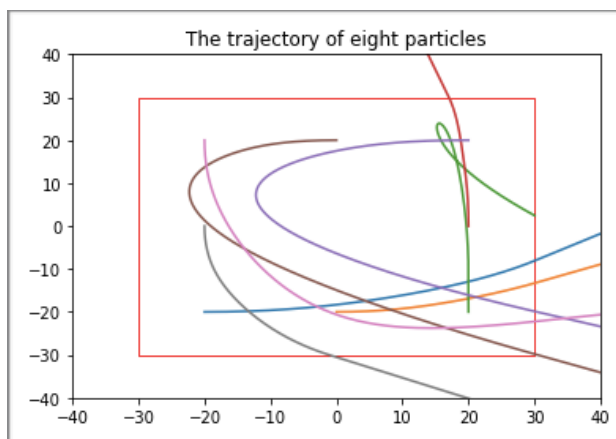


图（14）

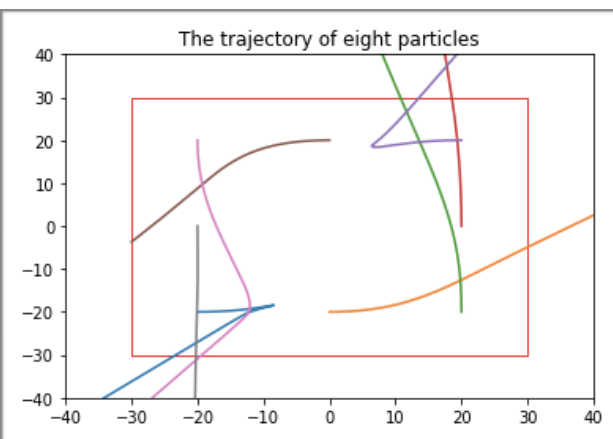


图（15）

图（14）为，4个奇数号粒子带正电，4个偶数号粒子带负电时的模拟结果；图（15）是4个奇数号粒子带负电，4个偶数号粒子带正电时的模拟结果。同样，我们发现粒子的运动轨迹模拟图完全相同。这是因为该体系高度对称，粒子间的lorentz力作用效果相同。



图（16）



图（17）

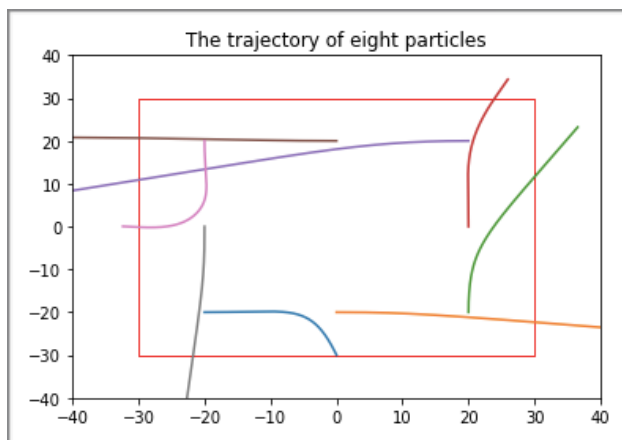


图 (18)

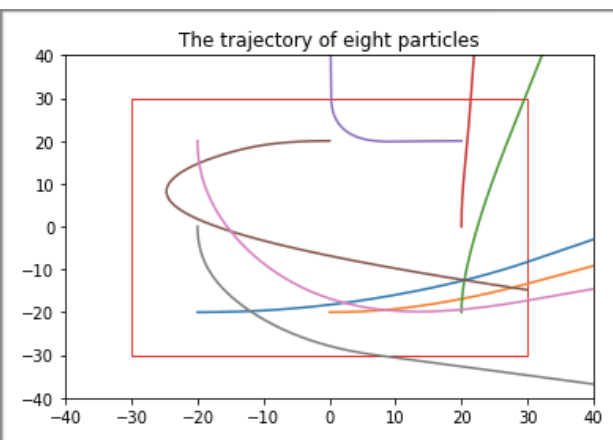


图 (19)

图 (16) 为，全部粒子都带正电时的模拟结果；图 (17) 前4个粒子带负电，后4个粒子带正电时的模拟结果；图 (18) 为，斜对角线上的粒子都带正电时，即 (3) (7) 两粒子带正电的模拟结果，图 (19) 是将其中一个粒子，粒子8变为带带负电时的模拟结果。我们发现粒子的运动轨迹极为复杂。如果利用数值计算的方法来计算时，我们可能只能得到粒子大概的运动趋势，但是利用python模拟，我们可以得到一个直观的轨迹图。

2.3.4 部分粒子不带电时粒子运动的模拟结果：

图 (20) 是在图 (14) 的基础上，在原点放置一个正电荷时的模拟结果；图 (21) 是在图 (14) 的基础上，在原点放置一个负电荷时的模拟结果；

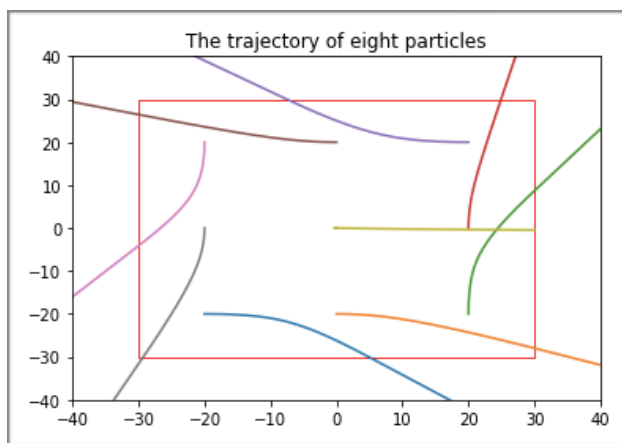


图 (21)

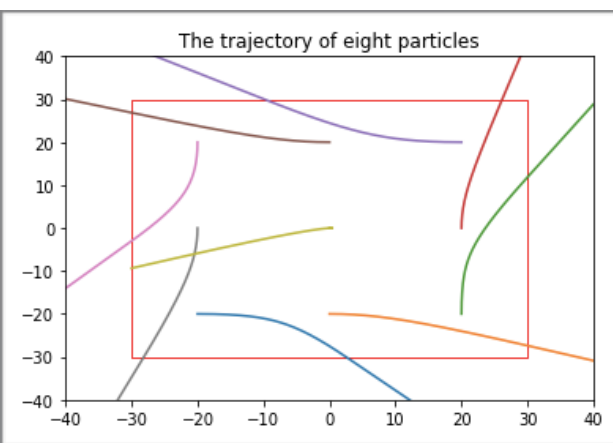


图 (22)

我们可以看到，即使粒子不带电，且粒子受力为0，由于其它粒子的lorentz力的作用效果，粒子也将发生运动。通过对比（21）和（22）两图，不难看出，不带粒子的运动轨迹与各个粒子的初速度方向有关，还取决于自己本身的电性。

2.3.5 关于此模拟失效的讨论：

在前面的讨论过程，其实，我都选择了一些粒子位置特殊的情况下来讨论。在本文（2.2）节中，我们规定了粒子，当 $\vec{r}_{ij}=0$ 时 $\vec{B}, \vec{E}=0$ 。现在给出解释：

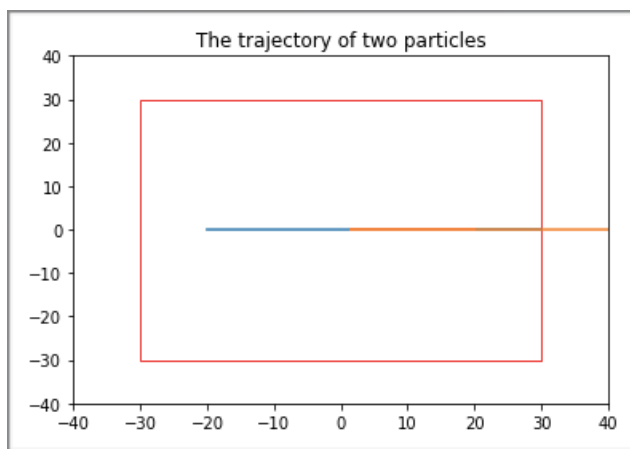


图 (23)

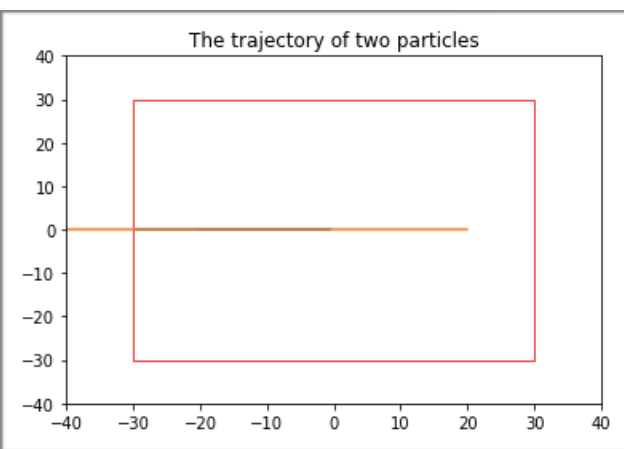


图 (24)

图（23），（24）乍看上去像是两个完全对称的图，但是两图都是失效的。因为，上面两图模拟的是粒子沿对向同速度运动，图（23）是左边带正电右边带负电，图（24）是右边带正电左边带负电时的模拟结果。由于两粒子发生了碰撞，电荷中和，但是本文用到的python算法本身并不可以体现出电荷中和这一特性。图（25）是一个典型的说明，一旦出现这种情况，本算法将不适用。

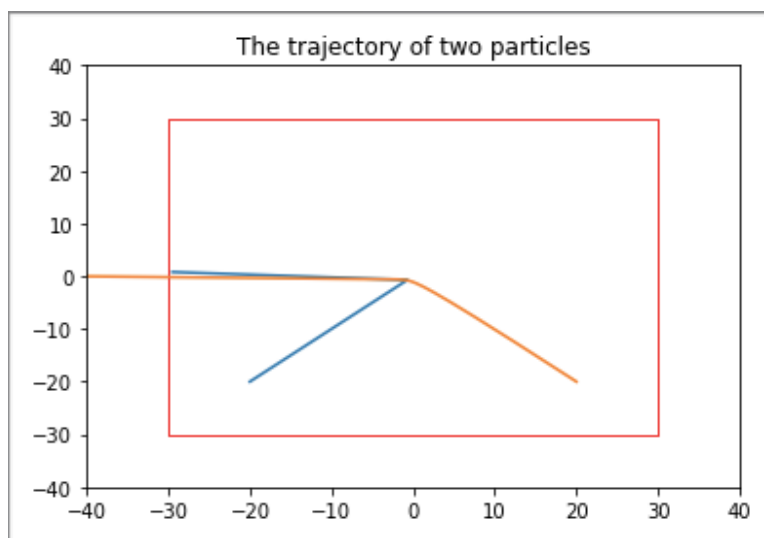


图 (25)

3 结束语

以上内容，就是我的全部期末展示了。其实，我认为这次作业已经把我这学期计算物理这门课学到的知识全部用到了。如包括用 latex 打公式，利用`matplotlib`绘制3维和2维图，利用`Numpy`数据包来转化方程，利用欧拉法来绘制曲线等（本来是想做一个欧拉法与Euler-Carmen法的对比的，但是我发现其实在这种情况下，两张方法的模拟结果基本一致，对比本身完全没有必要）。具体自己写并修改代码是一件极其痛苦的事，如，自己在利用欧拉法模拟单粒子的运动轨迹时，始终得不到正确的图形，也导致自己多次怀疑自己的算法本身是不是有问题，后来不断尝试，发现竟然是步长取的不合适。但是，当自己用自己的代码绘制出理想的图形时，又是极其欣喜的。其中的苦与乐，只有切身感受才能懂得。

我想再次说明的是：本文第一小节用到的代码是自己原创，第二小节用到的代码是根据 MATLAB 模拟多电子运动体系的代码改的，也许改的并不完美，但是至少可以得出一个像模像样的结果。

(本文用到的所有代码已上传至github)

参考文献：

- [1] Nicolas J. Giordano, Hisao Nkanishi, Computational Physics, second edition.
- [2] 刘觉平, 电动力学[M], 武汉大学出版社, 2002
- [3] NumPy(<http://www.numpy.org>)
- [4] matplotlib(<https://matplotlib.org/index.html>)