

Workreport_(15th)

2020.3.6—3.8

1. 黑洞生成筛选程序的调试与运行结果及分析

以下展示了某次的运行结果（定义粒子质量为 5, 临界判据值为 50, 即单位体积内质量达到 50 或数量达到 10 即符合，这里为方便测试，均采取无量纲值）

100 个在长度为 10 个单位的立方体内随机分布的粒子的坐标

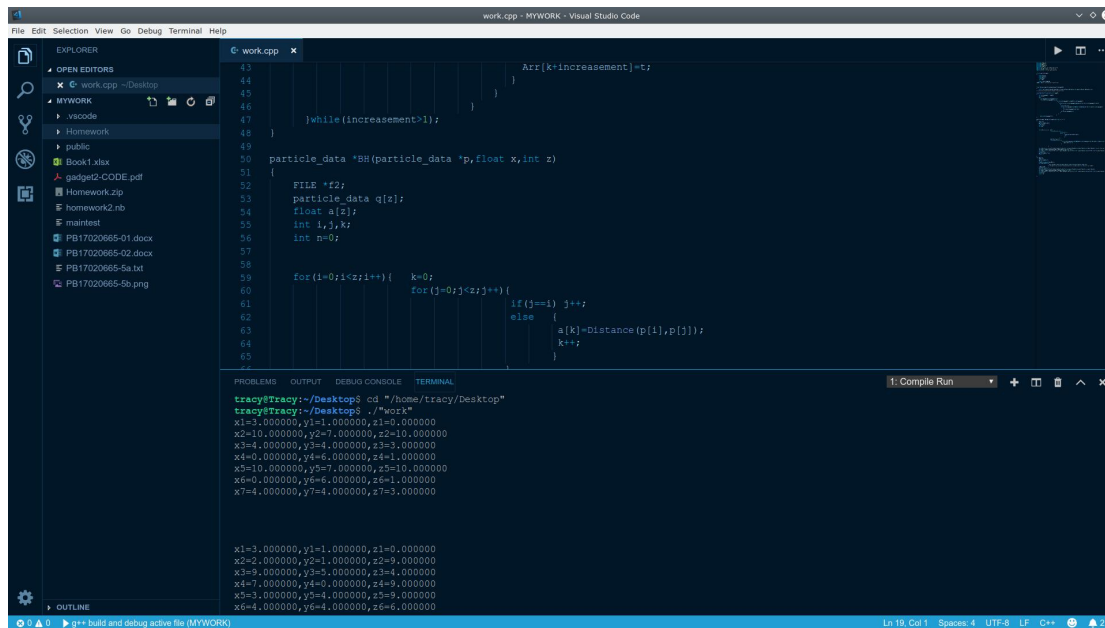
```
x1=3.000000,y1=1.000000,z1=0.000000
x2=2.000000,y2=1.000000,z2=9.000000
x3=9.000000,y3=5.000000,z3=4.000000
x4=7.000000,y4=0.000000,z4=9.000000
x5=3.000000,y5=4.000000,z5=9.000000
x6=4.000000,y6=4.000000,z6=6.000000
x7=1.000000,y7=4.000000,z7=10.000000
x8=4.000000,y8=9.000000,z8=7.000000
x9=6.000000,y9=9.000000,z9=9.000000
x10=2.000000,y10=10.000000,z10=4.000000
x11=7.000000,y11=2.000000,z11=6.000000
x12=7.000000,y12=2.000000,z12=5.000000
x13=4.000000,y13=10.000000,z13=10.000000
x14=6.000000,y14=6.000000,z14=10.000000
x15=5.000000,y15=9.000000,z15=1.000000
x16=3.000000,y16=2.000000,z16=6.000000
x17=8.000000,y17=4.000000,z17=10.000000
x18=7.000000,y18=8.000000,z18=7.000000
x19=3.000000,y19=1.000000,z19=5.000000
x20=10.000000,y20=1.000000,z20=3.000000
x21=4.000000,y21=8.000000,z21=5.000000
x22=10.000000,y22=3.000000,z22=8.000000
x23=4.000000,y23=7.000000,z23=7.000000
x24=1.000000,y24=0.000000,z24=2.000000
x25=9.000000,y25=5.000000,z25=9.000000
x26=10.000000,y26=7.000000,z26=10.000000
x27=3.000000,y27=4.000000,z27=1.000000
x28=1.000000,y28=9.000000,z28=7.000000
x29=8.000000,y29=0.000000,z29=8.000000
x30=0.000000,y30=10.000000,z30=7.000000
x31=3.000000,y31=1.000000,z31=2.000000
x32=7.000000,y32=9.000000,z32=5.000000
x33=2.000000,y33=2.000000,z33=10.000000
x34=7.000000,y34=1.000000,z34=0.000000
x35=7.000000,y35=10.000000,z35=5.000000
x36=5.000000,y36=8.000000,z36=1.000000
x37=4.000000,y37=9.000000,z37=3.000000
x38=5.000000,y38=8.000000,z38=0.000000
x39=1.000000,y39=3.000000,z39=9.000000
x40=7.000000,y40=4.000000,z40=6.000000
x41=3.000000,y41=5.000000,z41=8.000000
x42=6.000000,y42=10.000000,z42=6.000000
x43=9.000000,y43=1.000000,z43=7.000000
x44=9.000000,y44=6.000000,z44=6.000000
x45=9.000000,y45=2.000000,z45=6.000000
x46=3.000000,y46=6.000000,z46=1.000000
x47=3.000000,y47=8.000000,z47=8.000000
x48=4.000000,y48=3.000000,z48=6.000000
x49=4.000000,y49=2.000000,z49=9.000000
x50=0.000000,y50=10.000000,z50=0.000000
x51=7.000000,y51=0.000000,z51=4.000000
x52=4.000000,y52=4.000000,z52=3.000000
x53=8.000000,y53=3.000000,z53=3.000000
x54=4.000000,y54=1.000000,z54=7.000000
x55=9.000000,y55=10.000000,z55=10.000000
x56=2.000000,y56=0.000000,z56=3.000000
x57=3.000000,y57=3.000000,z57=0.000000
x58=0.000000,y58=6.000000,z58=1.000000
x59=4.000000,y59=8.000000,z59=4.000000
x60=3.000000,y60=9.000000,z60=1.000000
```

```
x61=1.000000,y61=5.000000,z61=1.000000
x62=5.000000,y62=9.000000,z62=6.000000
x63=7.000000,y63=4.000000,z63=9.000000
x64=10.000000,y64=7.000000,z64=10.000000
x65=6.000000,y65=5.000000,z65=7.000000
x66=3.000000,y66=7.000000,z66=7.000000
x67=4.000000,y67=8.000000,z67=0.000000
x68=5.000000,y68=6.000000,z68=4.000000
x69=6.000000,y69=0.000000,z69=1.000000
x70=8.000000,y70=1.000000,z70=10.000000
x71=9.000000,y71=2.000000,z71=4.000000
x72=9.000000,y72=6.000000,z72=0.000000
x73=4.000000,y73=2.000000,z73=5.000000
x74=0.000000,y74=10.000000,z74=1.000000
x75=8.000000,y75=3.000000,z75=4.000000
x76=4.000000,y76=5.000000,z76=0.000000
x77=9.000000,y77=9.000000,z77=8.000000
x78=7.000000,y78=1.000000,z78=3.000000
x79=0.000000,y79=6.000000,z79=1.000000
x80=2.000000,y80=3.000000,z80=2.000000
x81=1.000000,y81=0.000000,z81=3.000000
x82=4.000000,y82=9.000000,z82=9.000000
x83=4.000000,y83=0.000000,z83=9.000000
x84=7.000000,y84=0.000000,z84=8.000000
x85=6.000000,y85=6.000000,z85=9.000000
x86=10.000000,y86=8.000000,z86=3.000000
x87=8.000000,y87=6.000000,z87=0.000000
x88=5.000000,y88=3.000000,z88=1.000000
x89=7.000000,y89=1.000000,z89=5.000000
x90=8.000000,y90=3.000000,z90=9.000000
x91=9.000000,y91=3.000000,z91=9.000000
x92=1.000000,y92=7.000000,z92=5.000000
x93=8.000000,y93=9.000000,z93=5.000000
x94=4.000000,y94=4.000000,z94=3.000000
x95=1.000000,y95=10.000000,z95=7.000000
x96=8.000000,y96=8.000000,z96=4.000000
x97=10.000000,y97=5.000000,z97=8.000000
x98=10.000000,y98=9.000000,z98=0.000000
x99=9.000000,y99=3.000000,z99=2.000000
x100=2.000000,y100=9.000000,z100=3.000000
```

经过筛选，满足条件的所有粒子坐标

```
x1=3.000000,y1=1.000000,z1=0.000000
x2=10.000000,y2=7.000000,z2=10.000000
x3=4.000000,y3=4.000000,z3=3.000000
x4=0.000000,y4=6.000000,z4=1.000000
x5=10.000000,y5=7.000000,z5=10.000000
x6=0.000000,y6=6.000000,z6=1.000000
x7=4.000000,y7=4.000000,z7=3.000000
```

可以看到，100 个粒子中，最终筛选出 7 个粒子，而调整不同质量密度临界值，越大，则给出满足要求的例子数越少，越小则粒子数越大，这一结果和我们所期待的是相符合的。将质量密度设置为 10 时，输出的结果是 53,即单位体积内要求数量达到 2 时，有 53 个粒子满足要求，可见程序运行结果非常不错，就不粘贴在这里了（以上结果来自 VSCODE 输出的结果）



The screenshot shows the Visual Studio Code interface with a C++ file named `work.cpp` open. The code defines a `particle_data` struct and a function `particle_data *BH(particle_data *p, float x, int z)`. The function calculates distances between particles and updates their positions. The terminal output shows the results of the function, listing coordinates for 10 particles (x1 to x10, y1 to y10, z1 to z10).

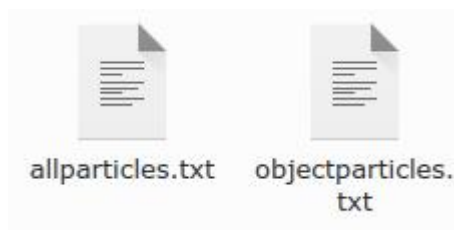
```
43 Arr[k+increasement]=t;
44 }
45 }
46 }
47 }
48 }
49 }
50 particle_data *BH(particle_data *p,float x,int z)
51 {
52     FILE *f2;
53     particle_data q[z];
54     float a[z];
55     int i,j,k;
56     int n=0;
57
58     for(i=0;i<z;i++){
59         k=0;
60         for(j=0;j<z;j++){
61             if(j==i) j++;
62             else {
63                 a[k]=Distance(p[i],p[j]);
64                 k++;
65             }
66         }
67     }
68 }
```

```
tracy@Tracy:~/Desktop$ cd ~/home/tracy/Desktop
tracy@Tracy:~/Desktop$ ./"work"
x1=3.000000,y1=1.000000,z1=0.000000
x2=10.000000,y2=7.000000,z2=10.000000
x3=4.000000,y3=4.000000,z3=3.000000
x4=0.000000,y4=6.000000,z4=1.000000
x5=10.000000,y5=7.000000,z5=10.000000
x6=0.000000,y6=6.000000,z6=1.000000
x7=4.000000,y7=4.000000,z7=3.000000

x1=3.000000,y1=1.000000,z1=0.000000
x2=2.000000,y2=1.000000,z2=9.000000
x3=9.000000,y3=5.000000,z3=4.000000
x4=7.000000,y4=0.000000,z4=9.000000
x5=3.000000,y5=4.000000,z5=9.000000
x6=4.000000,y6=4.000000,z6=6.000000
```

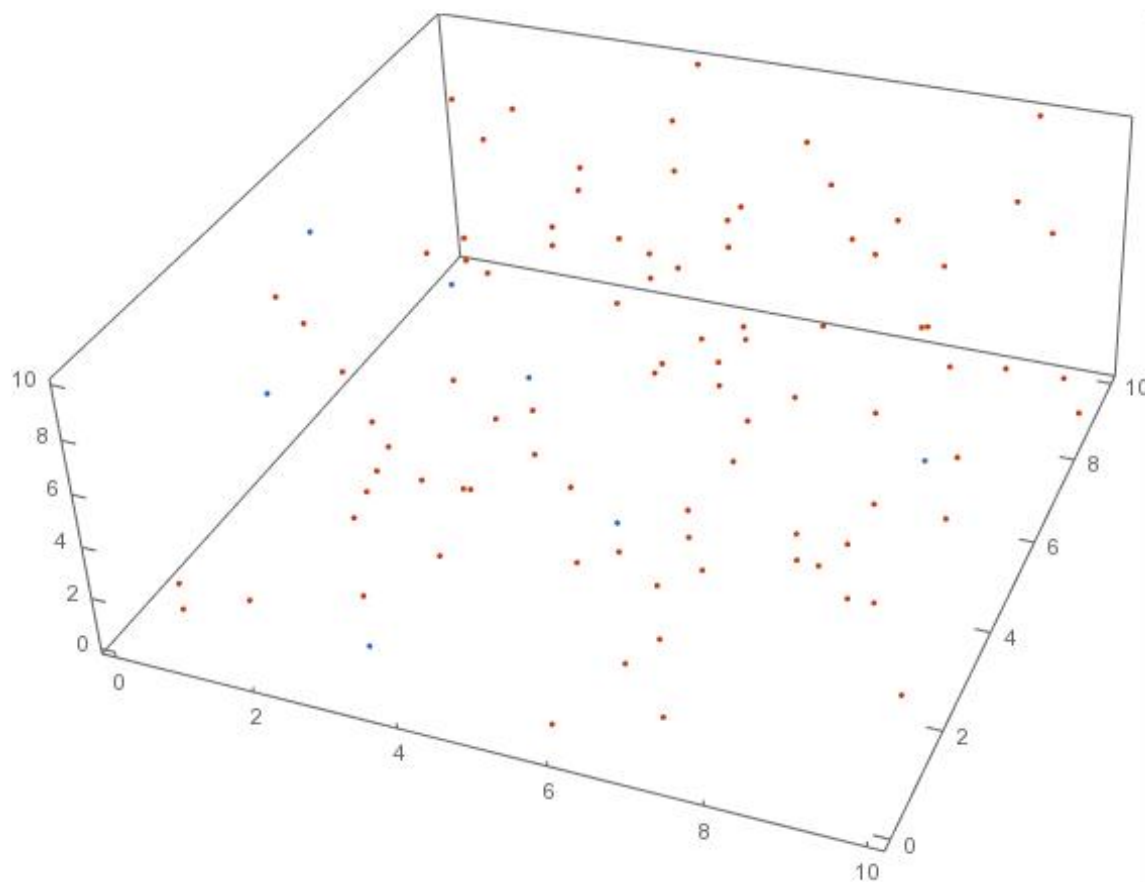
2.Mathematica（以下简称 MMA）对输出结果的可视化

将以上的输出结果以 MMA 可读取的格式输出到了两个.txt 文件中：allparticles.txt 和 objectparticles.txt



然后用 MMA 绘制出 3 维散点图

```
ListPointPlot3D[{data1, data2}, PlotTheme -> "Web"]
```

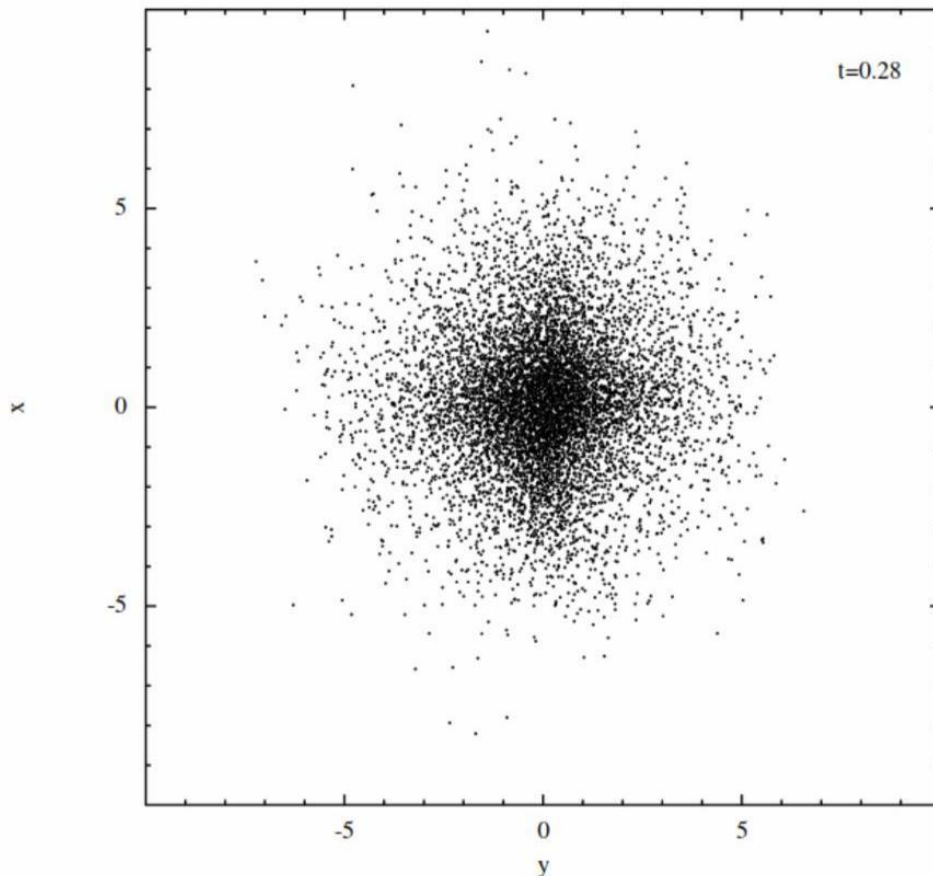


红色+蓝色为所有的点，蓝色为筛选出的点

由于粒子数较少，采取的分布是均匀分布，所以效果看起来可能不是太好，肉眼上貌似分辨不出筛选出的地方密度是否较大一点

PLANS:

1. 本打算采取更好的分布更多的粒子重复进行测试，如指数分布，但是这种分布容纳不下过多粒子，只能 15 个左右，20 个左右就已经溢出了，所以效果还不是很好，既然已经可以运行了而且 YYH 已经做出了一定的初始条件并已经可以跑出来了，所以最新的打算是用 YYH 自己初始条件模拟的第四张 Snapshot 的粒子位置数据再来测试这个代码，理想的结果将是中间的粒子团被标记出，即这张图：



(d) $t = 0.28 \text{ Gyr}$

YYH 会发给我它对应的 Snapshot 文件，我将先从其中读取出来所有粒子的位置信息并放入一个.txt 文件中，然后运行代码

2. 考虑像 YYH 一样写一个 readme

3. 着手思考有关吸积率的代码：首先要承认这样一个问题，有时候可能粒子密度大的地方并不是集中在一个区域，而是分别属于两个或者多个粒子团，要发展一个代码将这些例子团区分出来，即多个独立的黑洞

参考在天文学实验中阅读文献了解到的，星系团归类的系列代码，即 Clustering 的 K-means 方法：

- (1) 随机投 k 个点并以此代表数据团 (cluster) 的质心，并进行几次 (2) (3) 中的迭代
- (2) 跟据各个数据点到质心的距离，将各个数据和某个数据团联系起来 (某个数据点距离哪个质心最近，则可理解为该点与该质心所属的数据团相联系，即该点属于以上述质心为质心的数据团)
- (3) 根据这些被联系到某个数据团的数据点重新计算数据团的质心
- (4) 当每个数据团里面的数据点不再发生变化时，停止迭代

这是天文观测中用来划分星系团的方法，觉得可以抛砖引玉，用在不同黑洞的划分

而此后，关于黑洞的合并，则可看成在 t_0 本来有两个分离的黑洞，拥有不同的质心，到 t_1 这两个数据团有了同一个质心而黑洞的吸积，则可看成从 t_0 到 t_1 ，以某个点为质心的数据团不断又有更多数据点的过程