

Dreidimensionale Strukturen durch Selbstorganisation

Wolfhard Hövel
30.05.2010

In der belebten und unbelebten Natur entstehen vielfältige Strukturen durch das Phänomen Selbstorganisation. Muster, die sich durch lokale Bedingungen organisieren, weisen ganz typische Eigenschaften auf. Sie können spontan entstehen und vergehen. Bei genügender Vergrößerung zeigt sich ein Grundrauschen, die Bilder sind einer fortlaufenden Fluktuation unterworfen. Oft sind mehrere Attraktoren vorhanden, sodass sich die Formen nach einer gewissen Lebensdauer plötzlich in völlig andere umwandeln können. Durch dieses dynamische Verhalten sind die selbstorganisierten Systeme besonders interessant.

In der vorliegenden Arbeit stelle ich einen Algorithmus vor, der möglichst einfach ist und mit dem man Selbstorganisation im dreidimensionalen Raum untersuchen kann. Ausgangspunkt ist ein gerichteter Graph mit einer beliebigen Anzahl von Knoten und Kanten. Jede Kante wird bestimmten lokalen Regeln unterworfen, anschließend werden die Auswirkungen global superponiert.

Betrachten wir zunächst eine Kante mit ihren zwei Knoten. Der eine Knoten wird durch einen dreidimensionalen Verschiebungsvektor und der andere zu diesem genau entgegengesetzt bewegt. Damit bleibt der „Schwerpunkt“ der Kante konstant. Die beiden Knoten bewegen sich auf zwei Parallelen, deren Abstand ebenfalls gleichbleibend ist. Außerdem ist der Betrag des Verschiebungsvektors konstant. Damit verändert sich das „Drehmoment“ der Kante gleichfalls nicht. Diese Regeln werden auf jede Kante des Graphen angewandt. Bei den Knoten, an denen mehrere Kanten münden, werden die Verschiebungen der Kanten überlagert. Zum Schluss müssen noch Randbedingungen eingeführt werden, an den Grenzen werden die Knoten „reflektiert“. Sind sämtliche Knoten des Graphen um einen Schritt verschoben worden, wird das Rechenverfahren beliebig oft wiederholt. Damit bewegen sich die Knoten iterativ auf speziellen Bahnen.

Die Basis des Rechenverfahrens ist der Verschiebungsvektor \vec{h} . Der ganzzahlige Betrag von \vec{h} ist h und bis auf den Summanden ± 1 konstant. Seine Komponenten sind ebenfalls ganzzahlig und damit ergeben sich etwa $4 \cdot \pi \cdot h^2$ mögliche Verschiebungsvektoren. Die ganze Struktur wird also mit einer näherungsweise gleichbleibenden Schrittweite h aufgebaut.

Die gerichtete Kante des Graphen wird als Vektor aufgefasst. Die beiden Knoten 1 und 2 werden durch die Ortsvektoren \vec{r}_1 und \vec{r}_2 festgelegt. Damit ist die gerichtete Kante

$$\vec{r}_{12} = \vec{r}_2 - \vec{r}_1 \quad (\text{Bild 1}).$$

In Komponentenschreibweise gilt:

$$\vec{h}_{12} = \begin{pmatrix} h_{x12} \\ h_{y12} \\ h_{z12} \end{pmatrix}, \quad \vec{r}_{12} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}.$$

Die Verschiebung der Knoten erfolgt nach Bild 2. Der Abstand s der Parallelen sei der Einfachheit halber gleich $n \cdot h$ mit $n=1, 2, 3, \dots$. Damit die Knoten nicht beliebig auseinander streben, müssen noch Randbedingungen eingeführt werden. Hier hat man weitgehende Freiheiten. Zum Beispiel kann man eine maximale Summe der Längen der Kanten vorgeben.

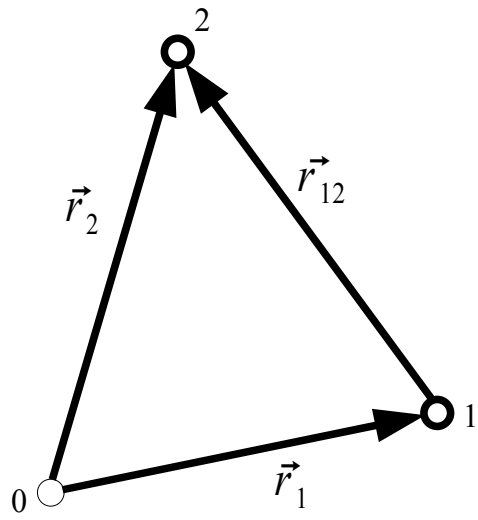


Bild 1

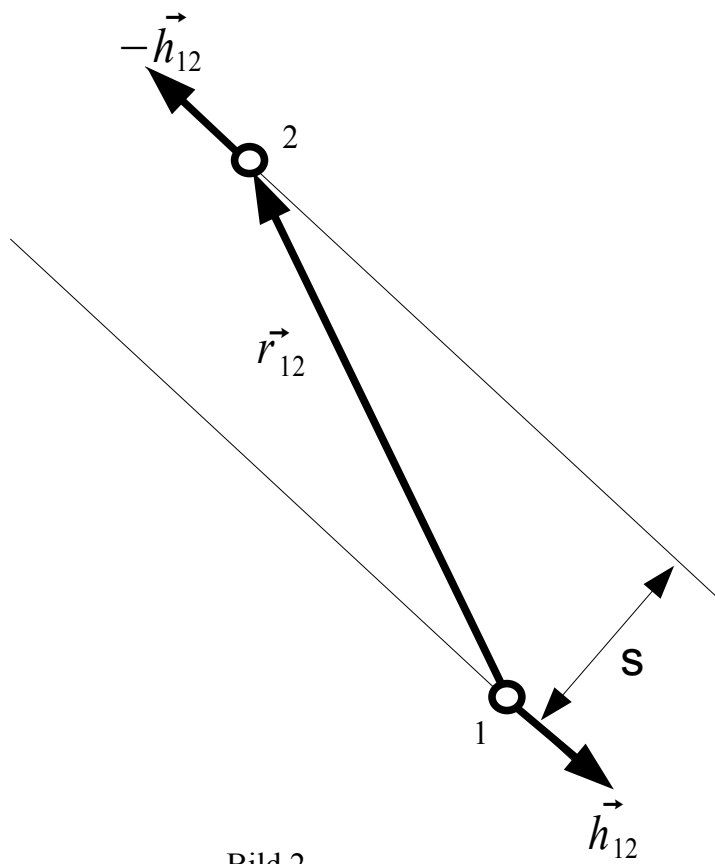


Bild 2

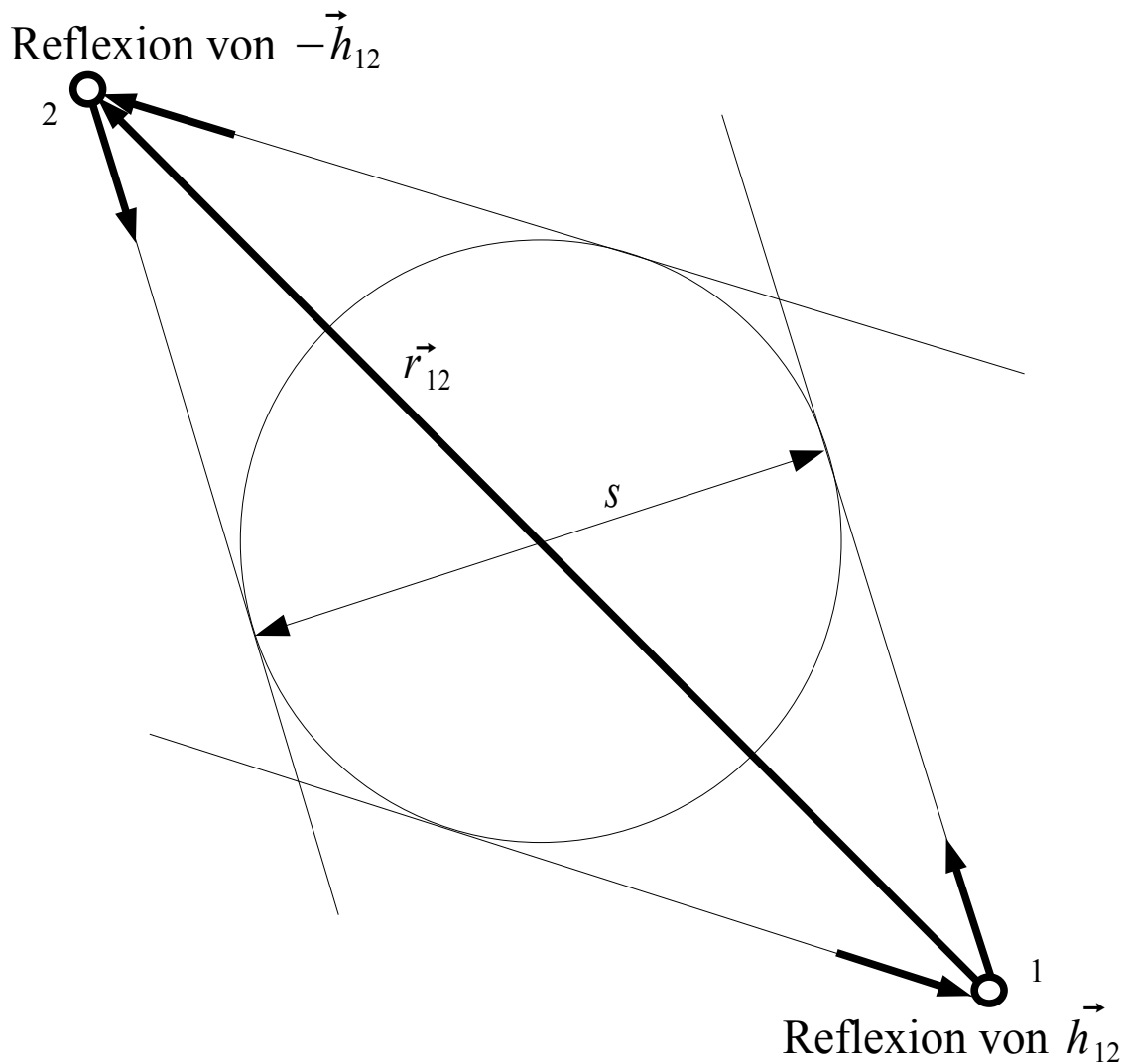


Bild 3

An der Grenze werden die Knoten „reflektiert“ (Bild 3). Hierbei muss darauf geachtet werden, dass wegen der späteren Superposition eine genauere Betrachtung nötig ist. Die Spur des Vektors $\vec{h}_{12\text{vor}}$ kann durch die Überlagerung parallel verschoben sein (Bild 4). Damit der Algorithmus so einfach wie möglich wird, genügt es, die Regel des konstanten Drehmomentes nur bei der Reflexion zu beachten.

Der neue Vektor $\vec{h}_{12\text{nach}}$ nach der Reflexion wird aus den Vektoren $\vec{h}_{12\text{vor}}$ und \vec{x} zusammengesetzt, wobei \vec{x} parallel zu \vec{r}_{12} ist (Bild 4).

$$q_1 * \vec{h}_{12\text{nach}} = \vec{h}_{12\text{vor}} + \vec{x} \quad ,$$

Der Faktor q_1 ist eine reelle Zahl und spielt nach einer Normierung keine Rolle.

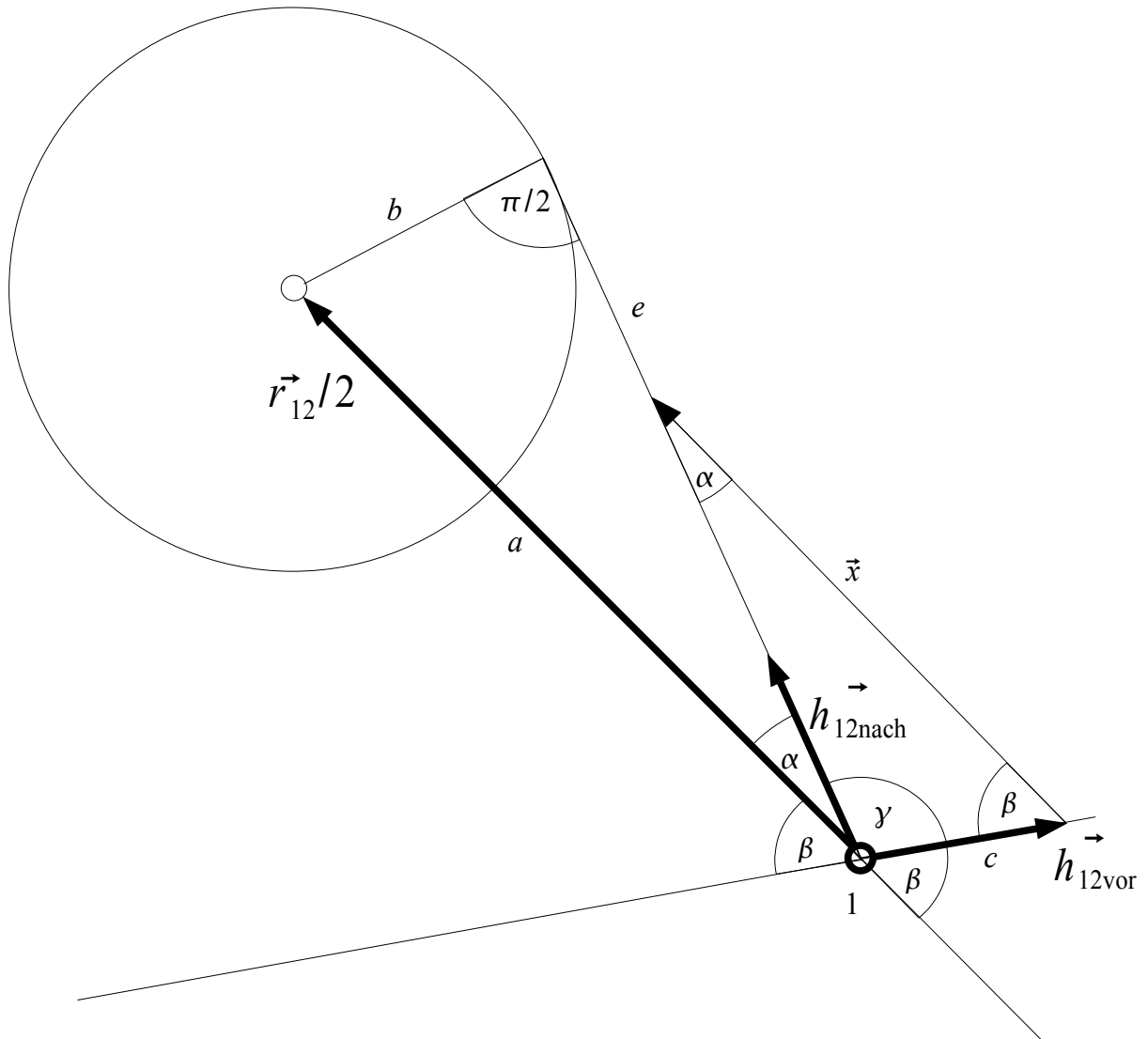


Bild 4

Weiterhin gilt mit den Abkürzungen $x = |\vec{x}|$, $skalar12 = h_{12vor} \cdot \vec{r}_{12}$,

$$skalar12 = h_{x12vor} \cdot (x_2 - x_1) + h_{y12vor} \cdot (y_2 - y_1) + h_{z12vor} \cdot (z_2 - z_1) ,$$

$$|\vec{r}_{12}| = r_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} ,$$

$$a = \frac{r_{12}}{2} , \quad c = h , \quad b = \frac{s}{2} , \quad k = |skalar12| = h \cdot r_{12} \cdot |\cos \beta| , \quad k = 2 \cdot c \cdot a \cdot |\cos \beta| ,$$

$$\sin \alpha = \frac{b}{a} , \quad \frac{c}{\sin \alpha} = \frac{x}{\sin \gamma} , \quad x = \frac{a \cdot c}{b} \cdot |\sin \gamma| , \quad \gamma = \pi - (\alpha + \beta) , \quad \sin \gamma = \sin(\alpha + \beta) ,$$

$$e = \sqrt{a^2 - b^2} \quad , \quad \cos \alpha = \frac{e}{a} = \sqrt{1 - \frac{b^2}{a^2}} \quad , \quad |\cos \beta| = \frac{k}{2 * c * a} \quad ,$$

$$\sin(\alpha + \beta) = \sin \alpha * \cos \beta + \cos \alpha * \sin \beta \quad , \quad \sin \beta = \sqrt{1 - \cos^2 \beta} \quad ,$$

$$x = \frac{k}{2 * a} \mp \sqrt{\left(\frac{a^2}{b^2} - 1\right) * \left(c^2 - \frac{k^2}{4 * a^2}\right)} \quad ,$$

$$x = \frac{k}{r_{12}} \mp \sqrt{\left(\frac{r_{12}^2}{s^2} - 1\right) * \left(h^2 - \frac{k^2}{r_{12}^2}\right)} \quad \text{mit } \vec{x} \text{ parallel zu } \vec{r}_{12} \quad ,$$

$$q_1 * h_{12\text{nach}}^{\rightarrow} = h_{12\text{vor}}^{\rightarrow} + \vec{x} \quad .$$

Nun muss der Vektor $h_{12\text{vor}}^{\rightarrow} + \vec{x}$ noch auf den Betrag h normiert werden:

$$h_{12\text{nach}}^{\rightarrow} = \frac{h_{12\text{vor}}^{\rightarrow} + \vec{x}}{|h_{12\text{vor}}^{\rightarrow} + \vec{x}|} * h \quad .$$

Insgesamt gibt es vier Lösungen, um das Drehmoment konstant zu halten. Gewählt wird der im Bild 4 gezeigte Weg. Es soll nur dann eine Reflexion erlaubt sein, wenn $\text{skalar}_{12} < 0$ und $r_{12} > s$ ist, das bedeutet, nur wenn die Knoten 1 und 2 auseinander streben.

Setzt man den Algorithmus in ein Programm um, muss man auf einige Dinge achten. Die zwei Knoten einer Kante müssen sich exakt entgegengesetzt bewegen. Dies ist hier leicht zu erreichen, da die Vektorkomponenten ganzzahlig sind und damit Rundungsfehler nicht auftreten. Der eine Knoten wird mit \vec{h} und der andere mit $-\vec{h}$ verschoben. Außerdem muss ein Rechendurchgang zunächst komplett mit den alten Daten gerechnet werden. Bevor der nächste Rechenzyklus begonnen wird, müssen die alten Daten durch die neu berechneten ersetzt werden.

Wie sehen die Ergebnisse des Rechenverfahrens aus? Beginnen wir mit dem einfachsten Fall. Der Graph besteht nur aus einer Kante und zwei Knoten. Die Spur der Knoten wird derart dargestellt, dass nach jedem Rechendurchgang ein Pixel mit der dem Knoten zugeordneten Farbe gesetzt wird (Bild 5). Vergrößert man das Bild nur hinreichend stark, entpuppen sich die scheinbar stetigen Kurven als diffuse Punktwolken, es zeigen sich immer die für Selbstorganisation so typischen Fluktuationen (Bild 6).

Sobald man den Graphen um einen Knoten und eine Kante erweitert, stellt sich ein wesentlich komplizierteres Verhalten heraus (Bild 7). Die Strukturen verhalten sich dynamisch und oft existieren mehrere Attraktoren. Die folgenden Bilder sind aus Graphen mit maximal 5 Knoten und 10 Kanten entstanden. Der Formenreichtum ist durch die beliebige Wahl der Graphen und Parameter praktisch unbegrenzt.

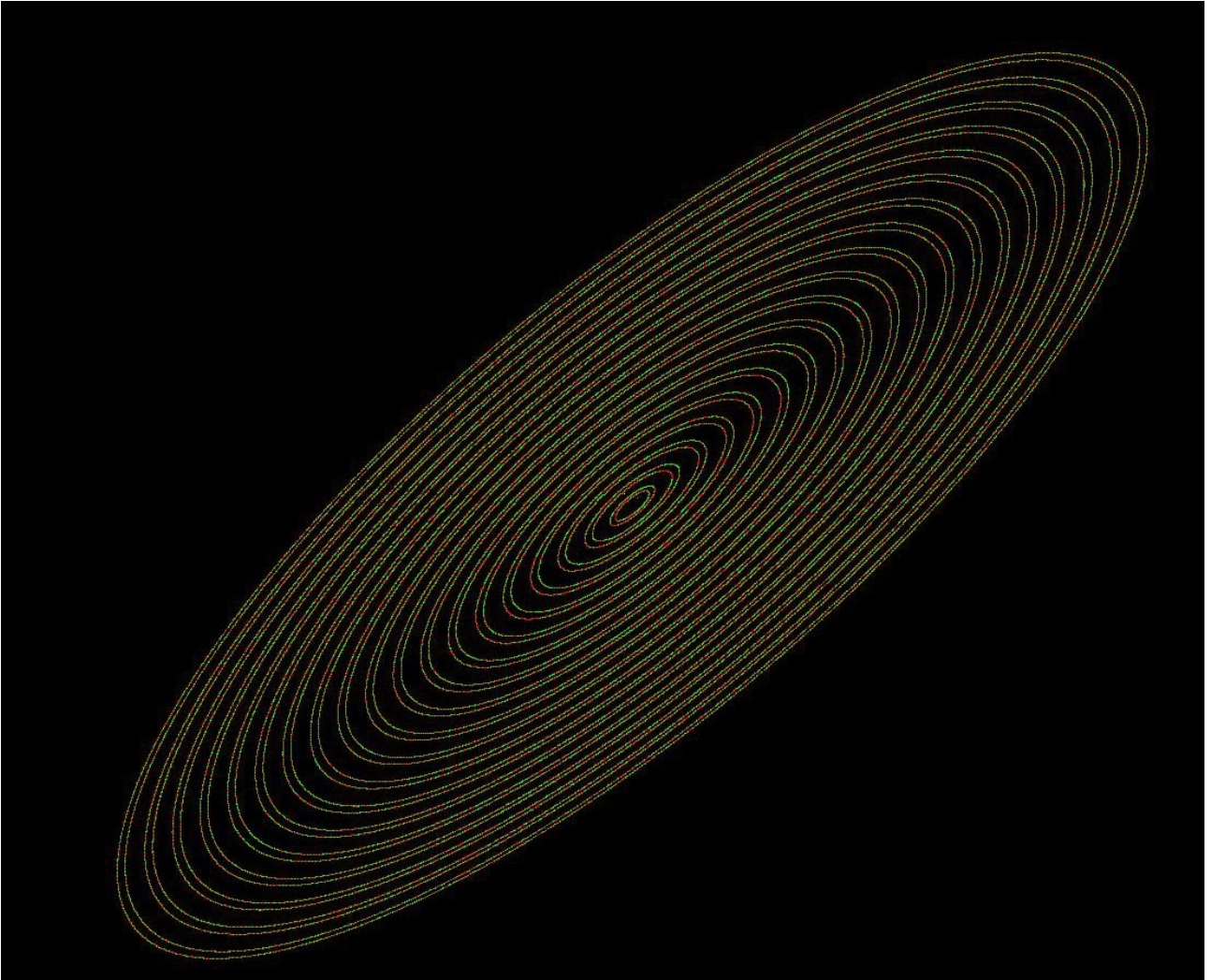
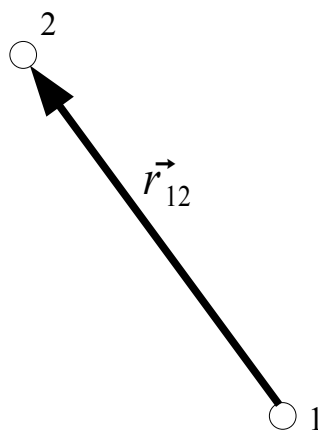


Bild 5

Der dazugehörige Graph:



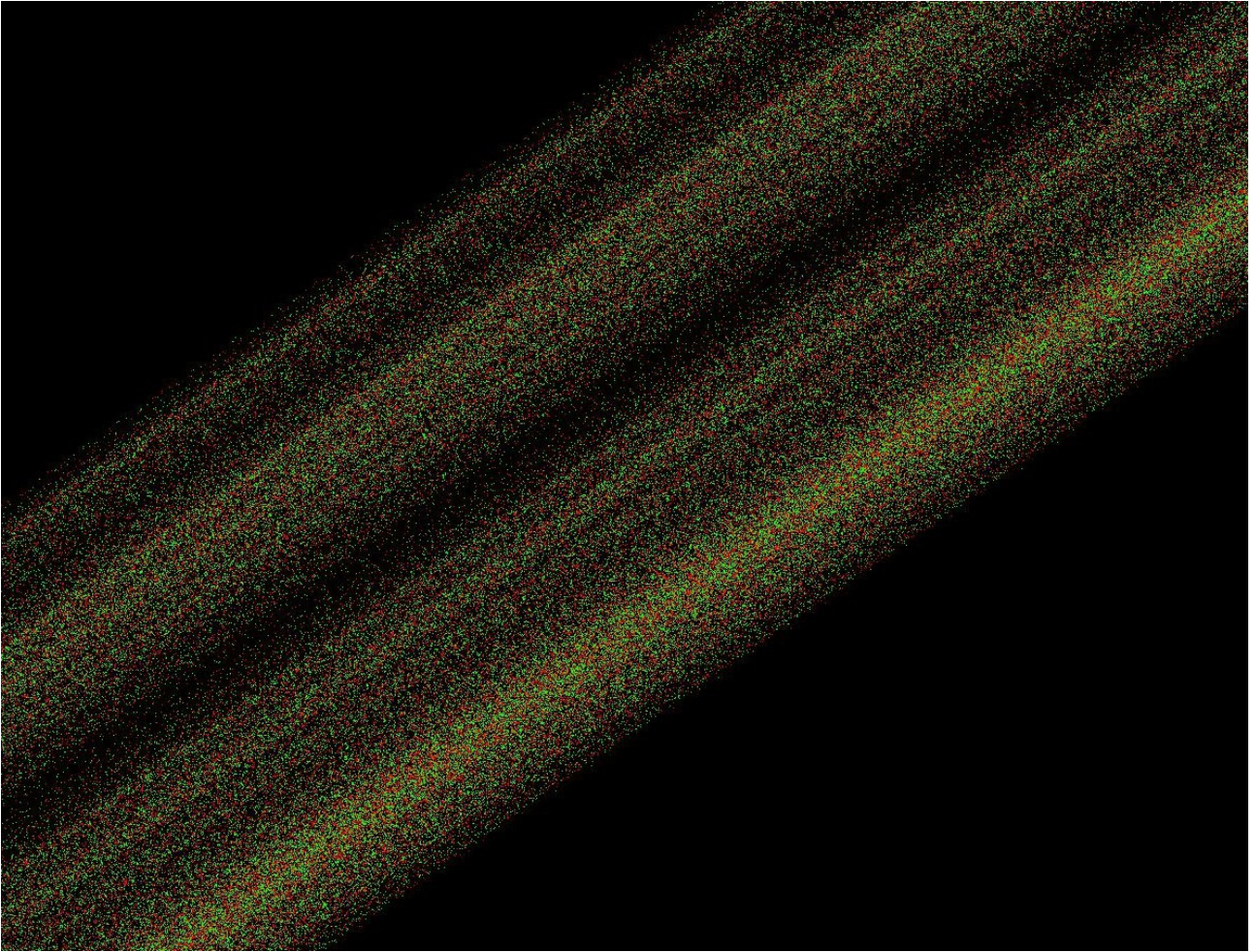
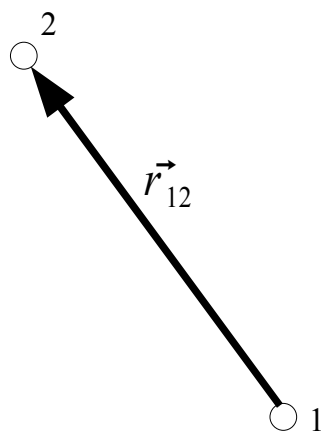


Bild 6

Der dazugehörige Graph:



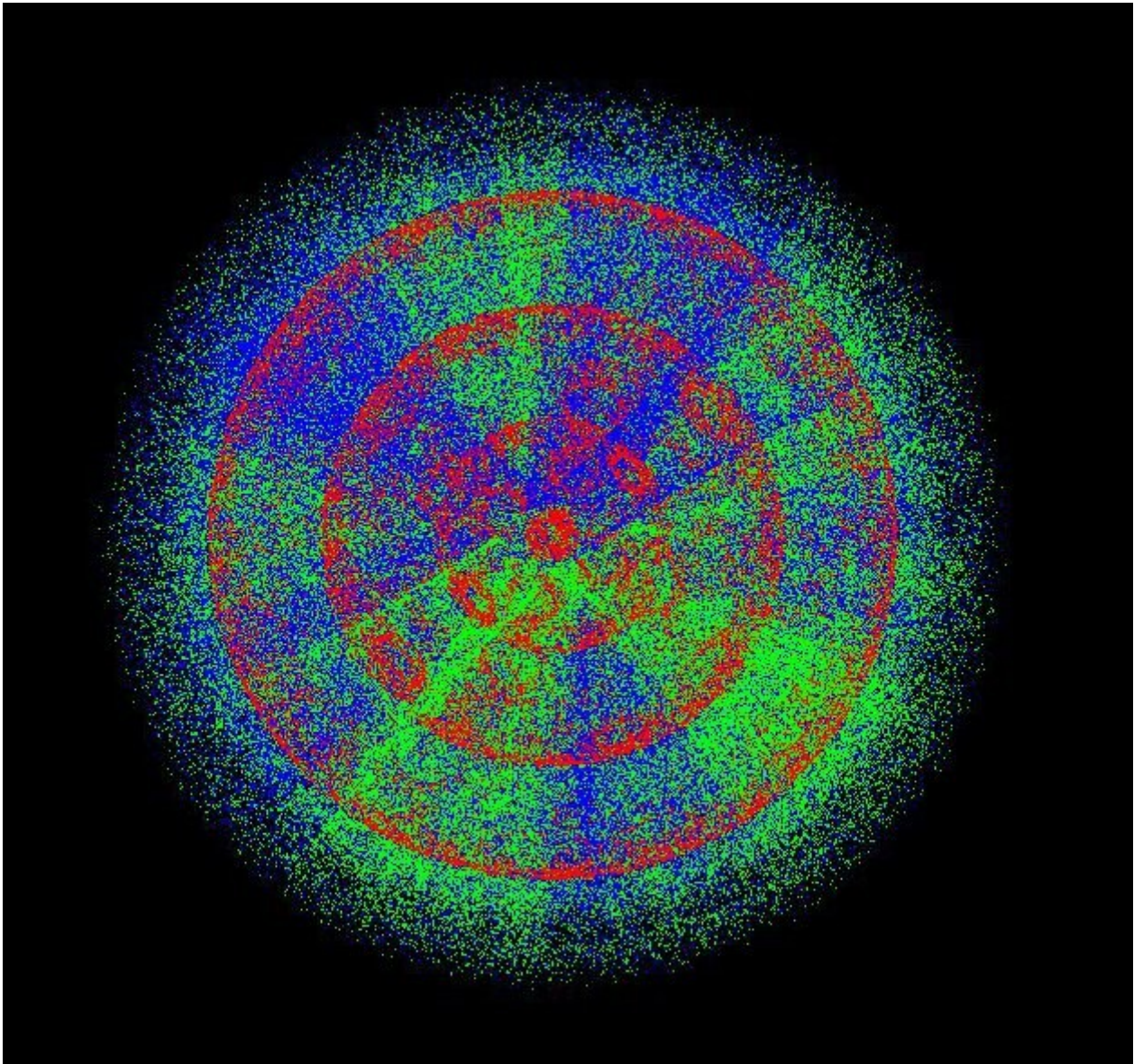
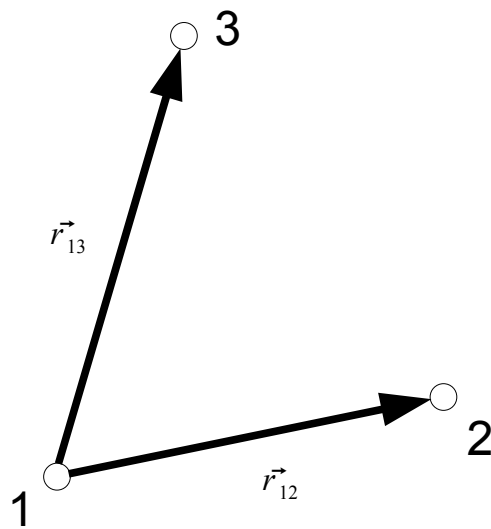


Bild 7

Der dazugehörige Graph:



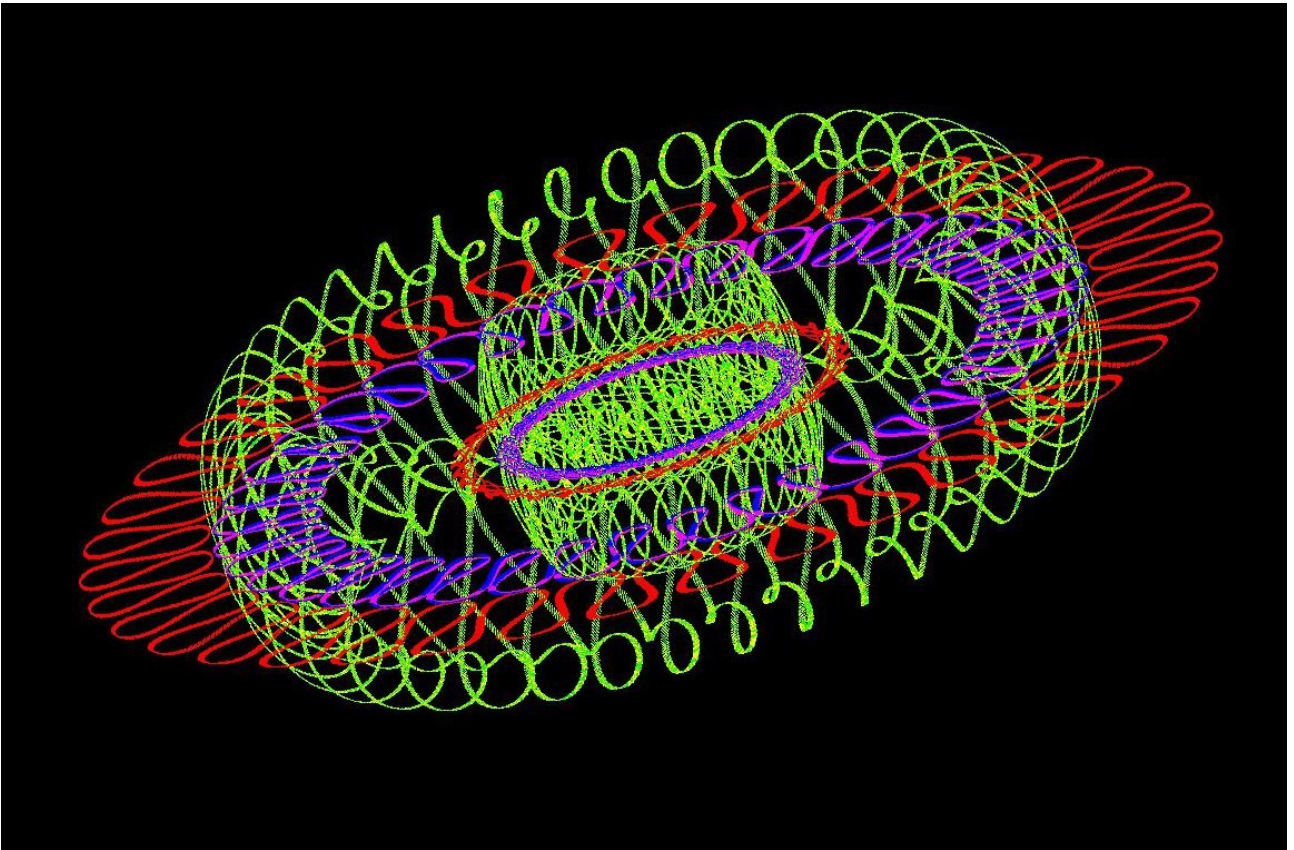
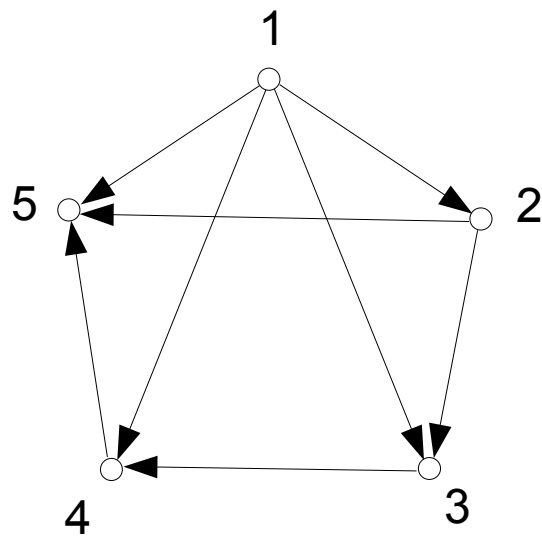


Bild 8

Der dazugehörige Graph:



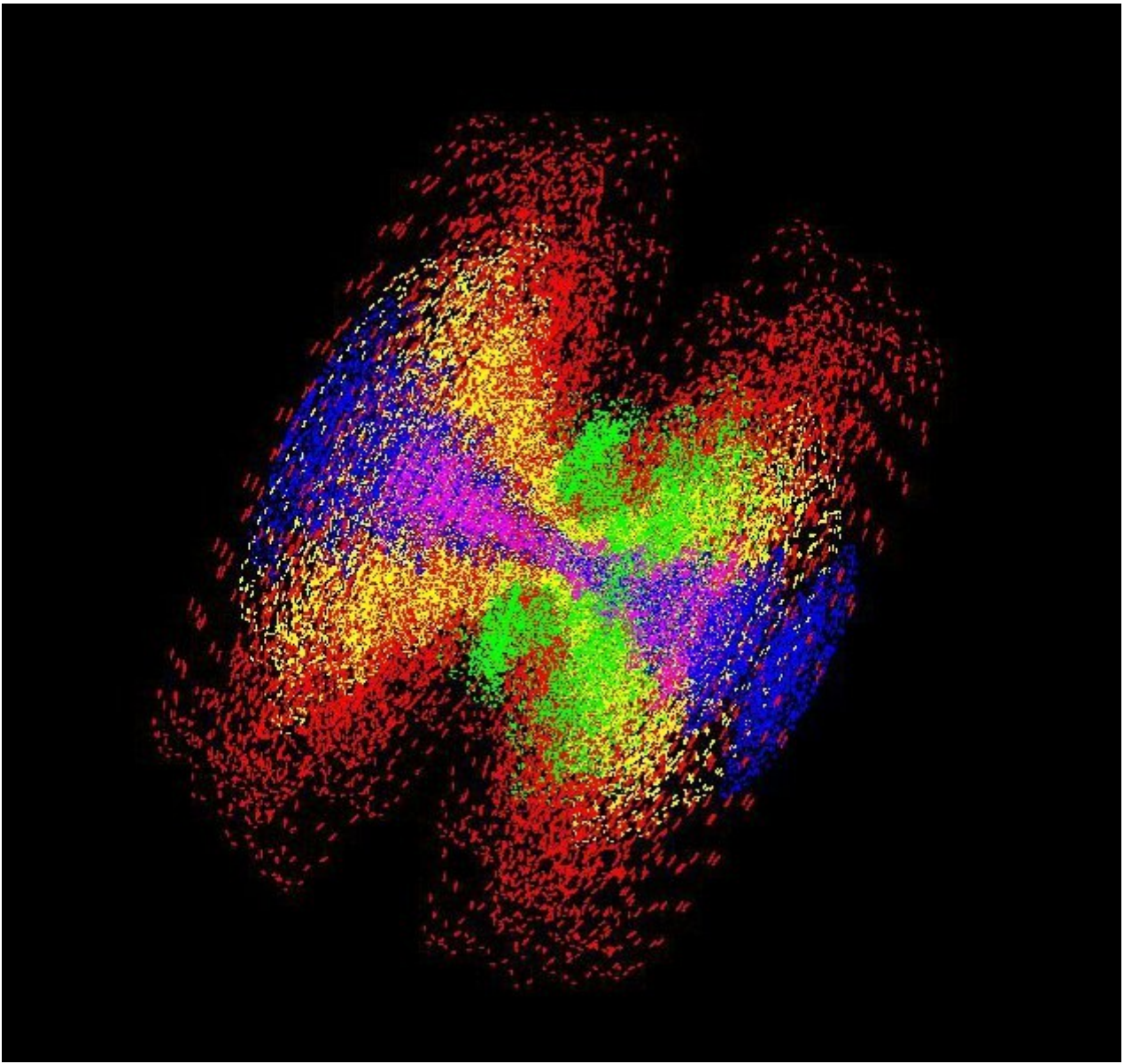
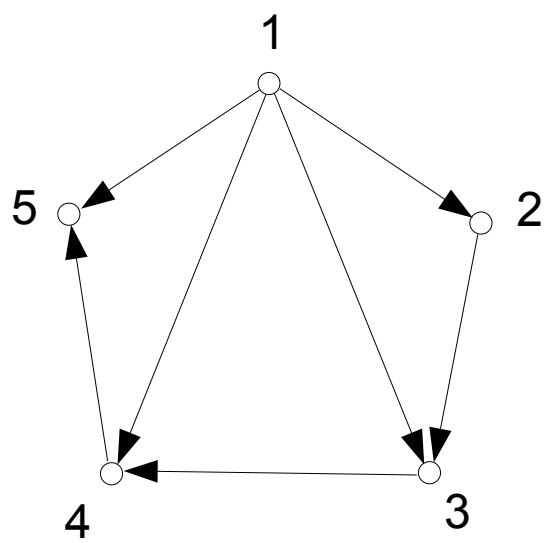


Bild 9

Der dazugehörige Graph:



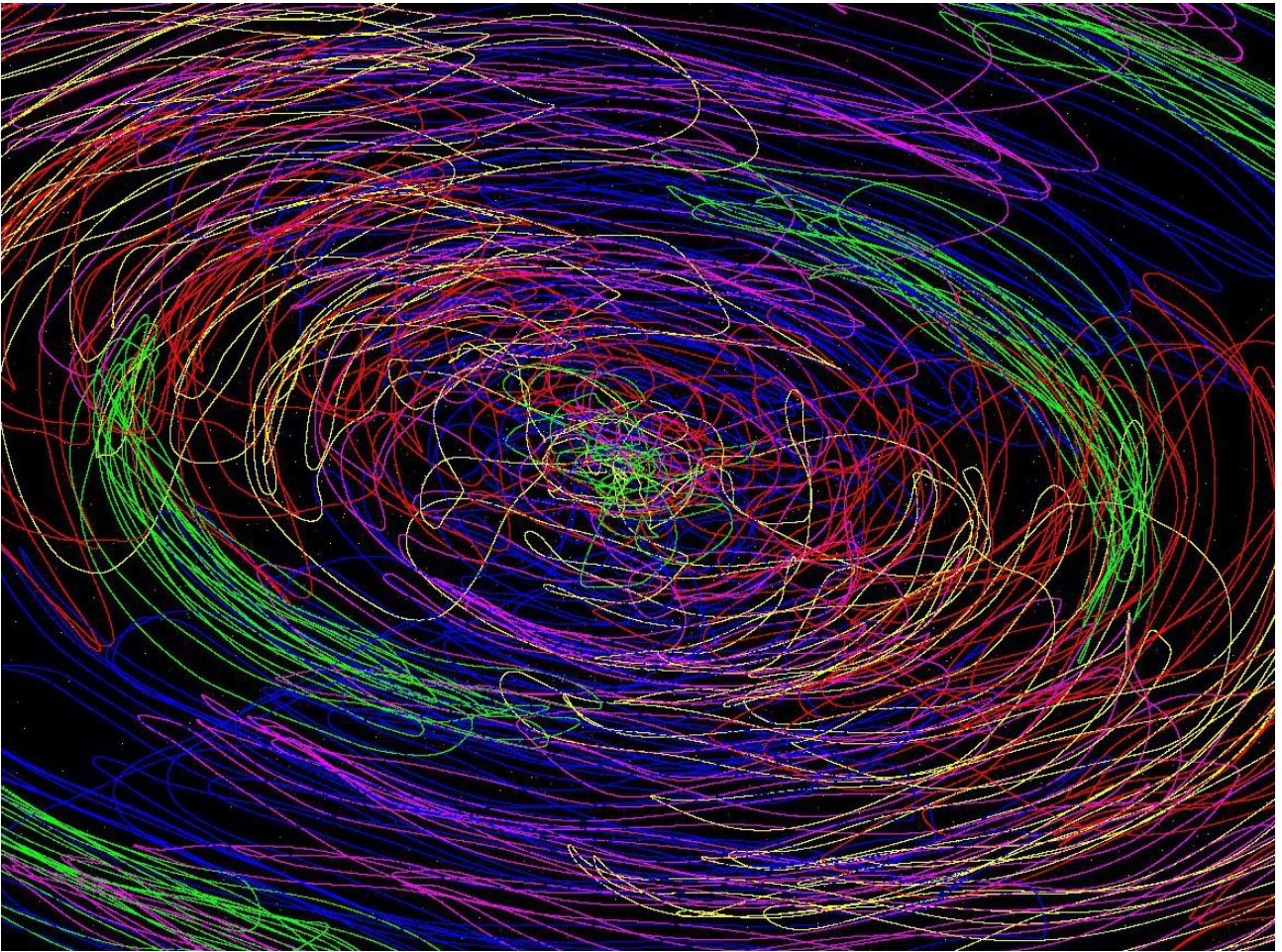
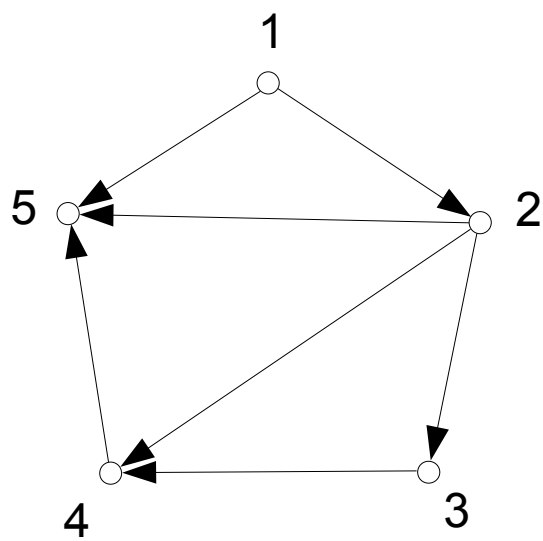
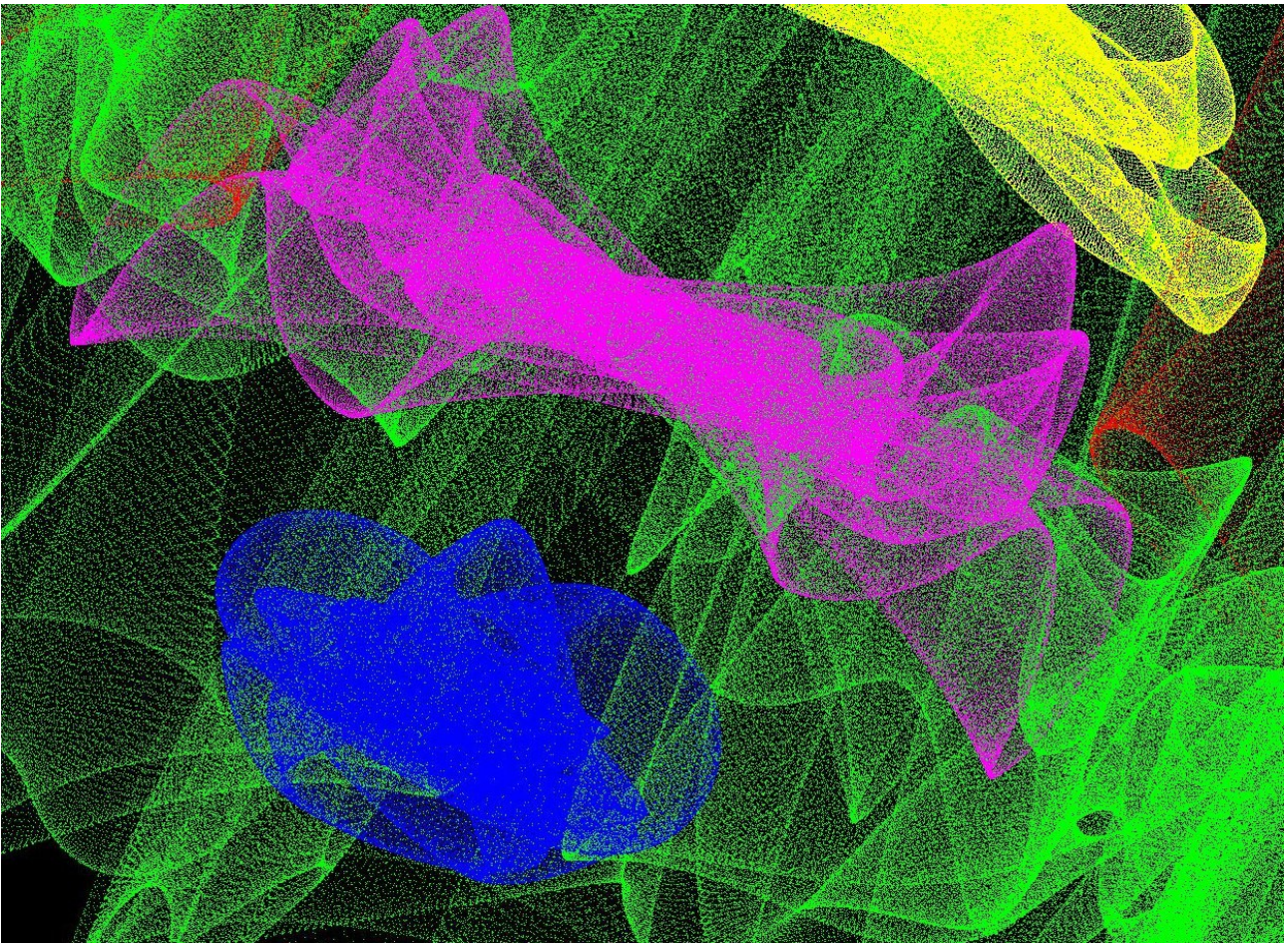


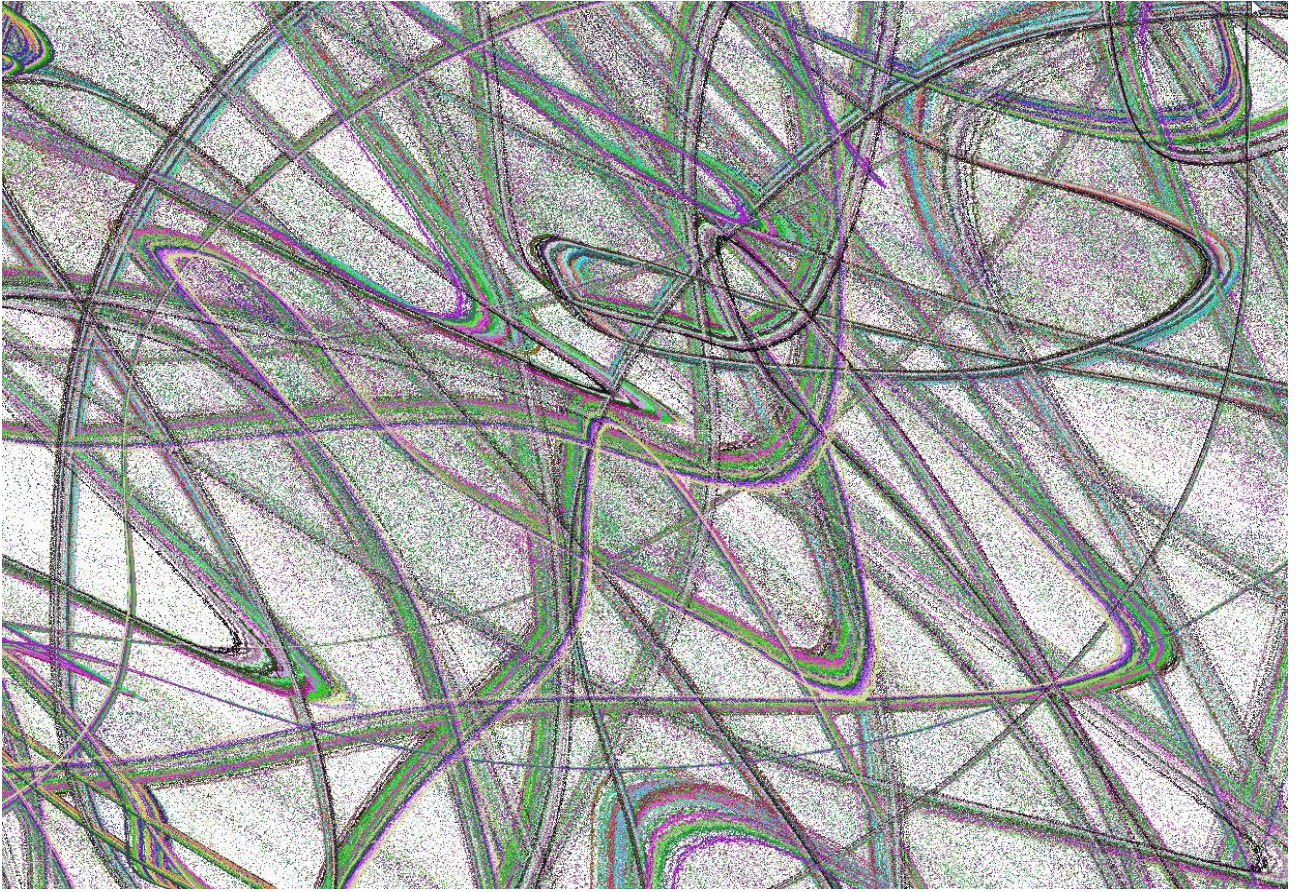
Bild 10

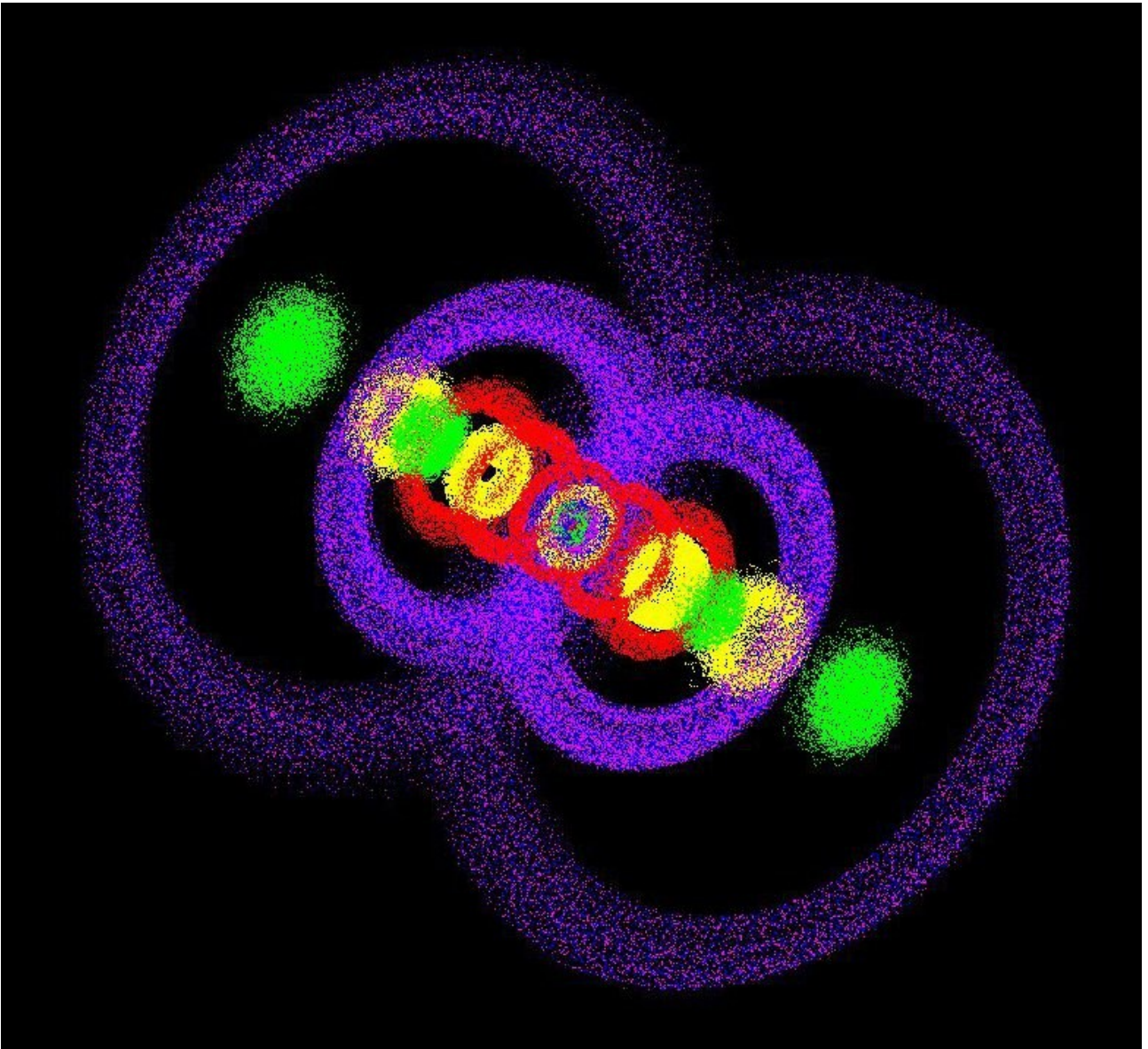
Der dazugehörige Graph:

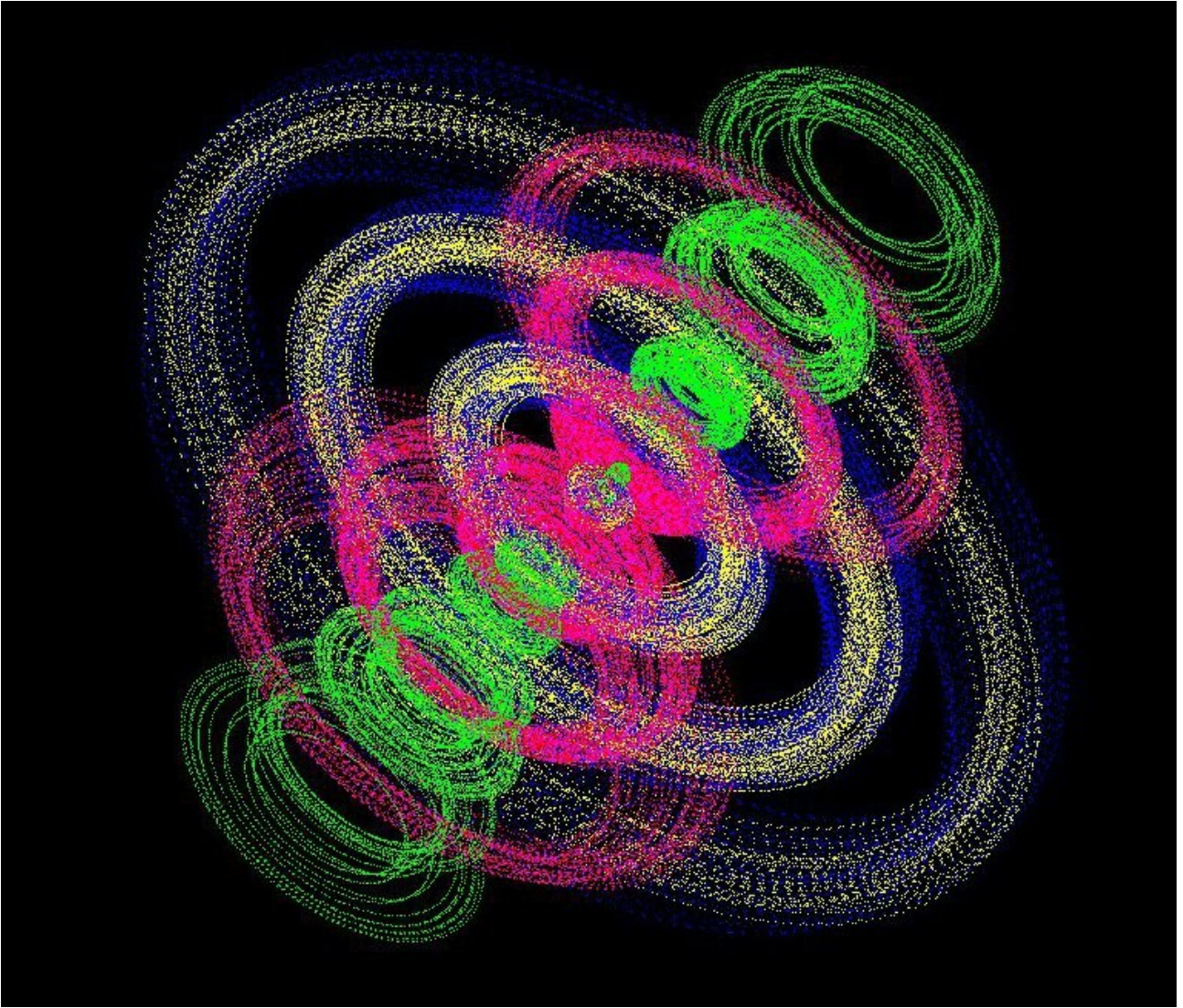


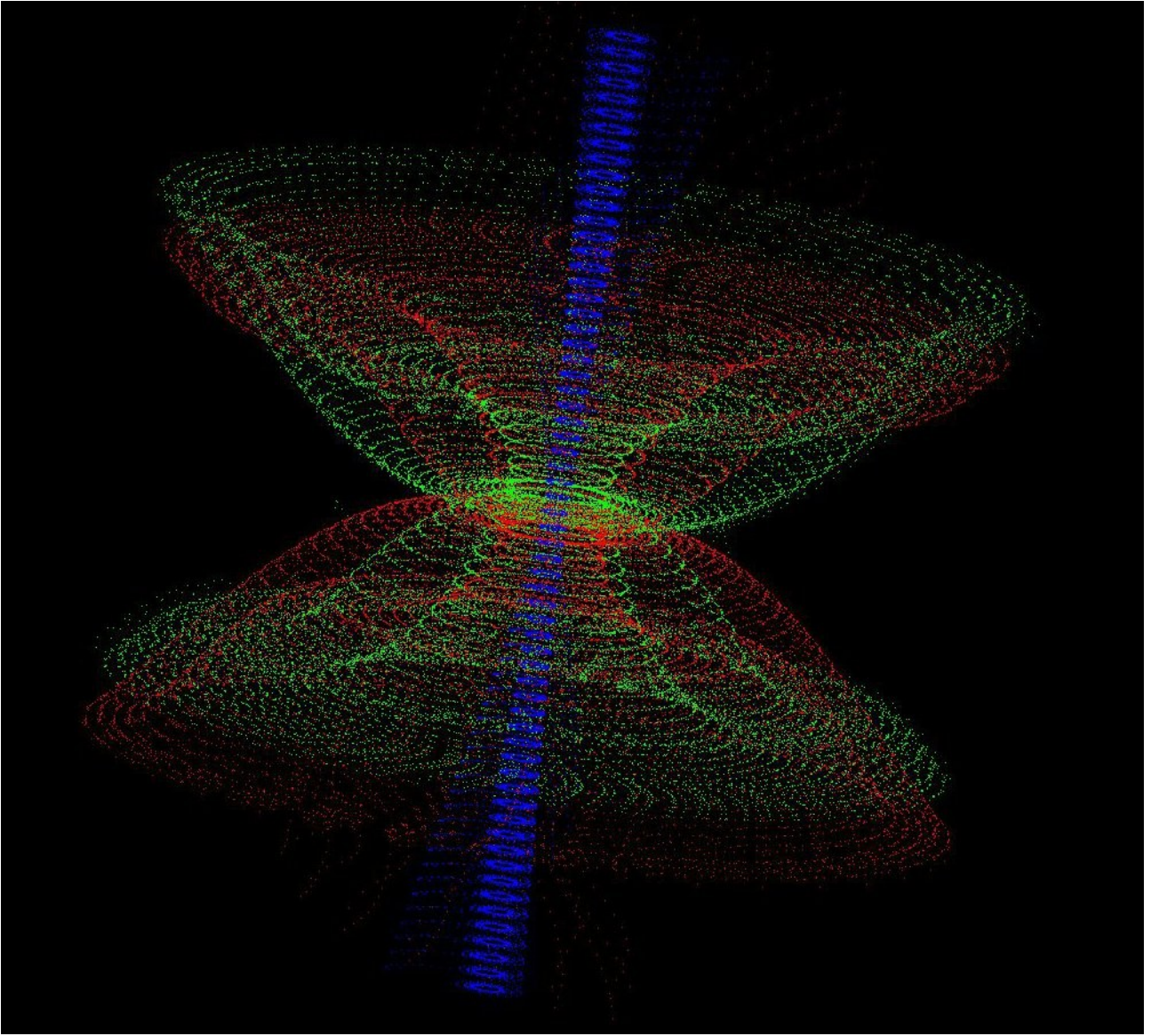
Es folgen einige weitere Beispiele von Strukturen, die der Algorithmus erzeugt:

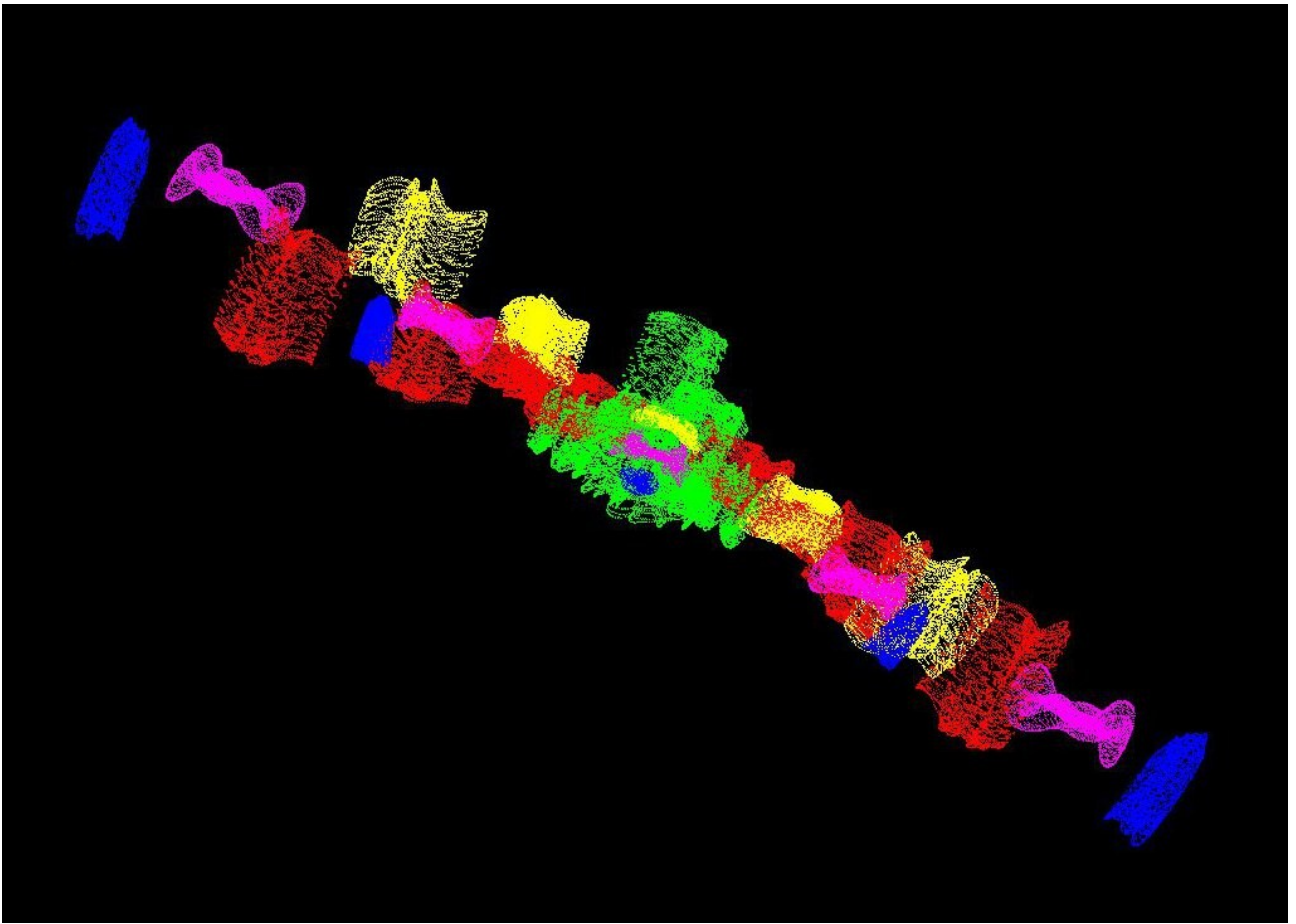


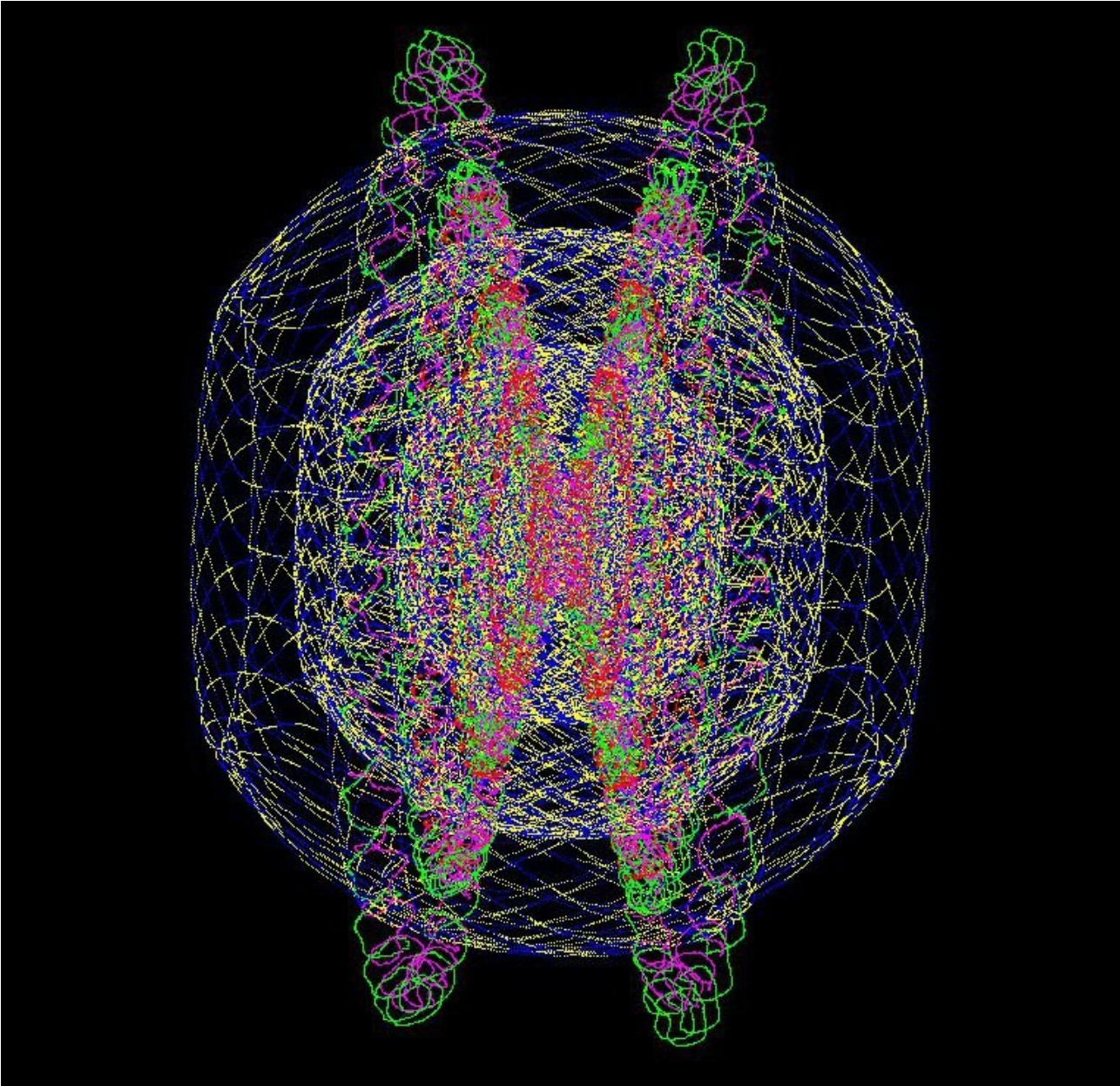


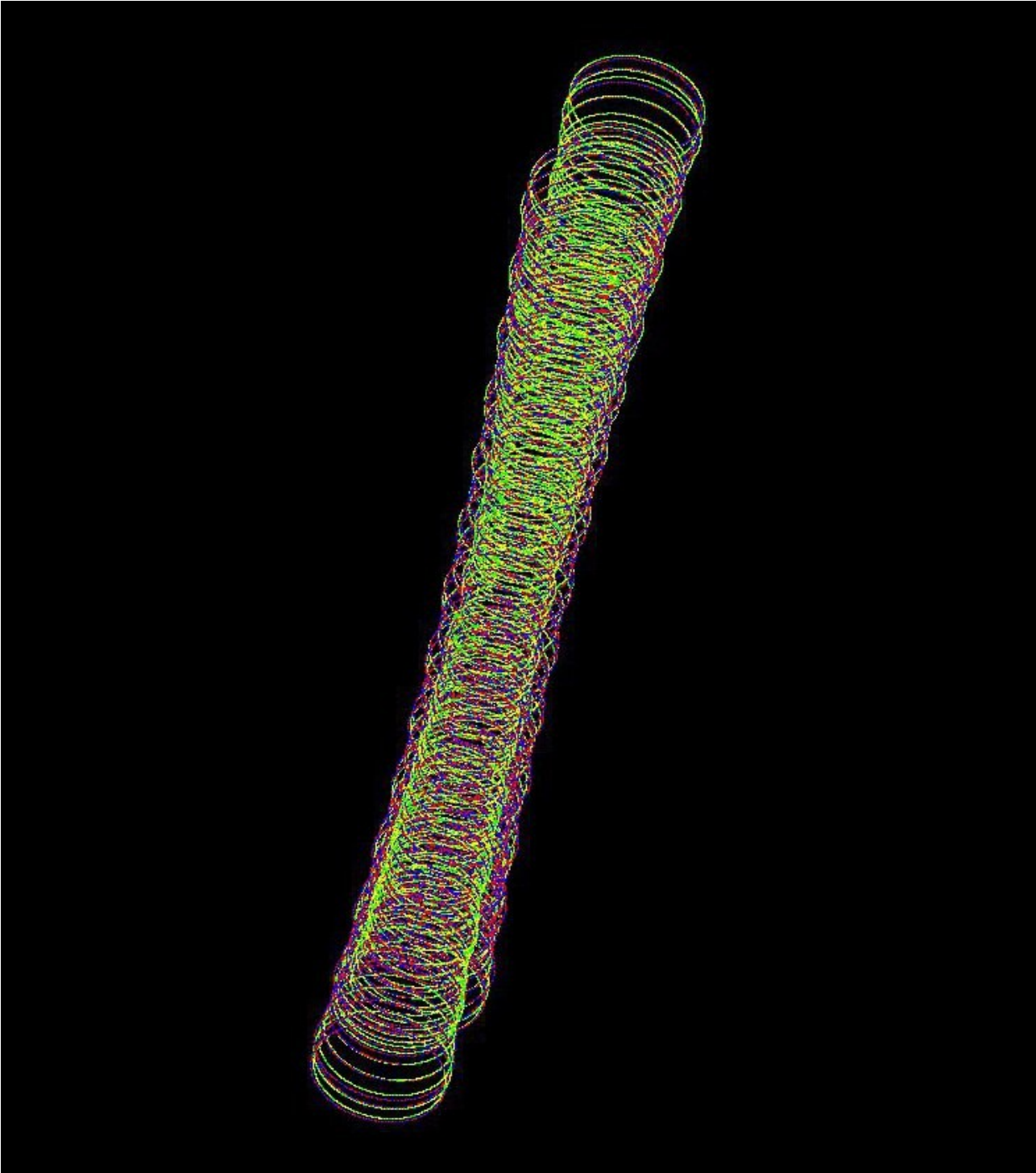


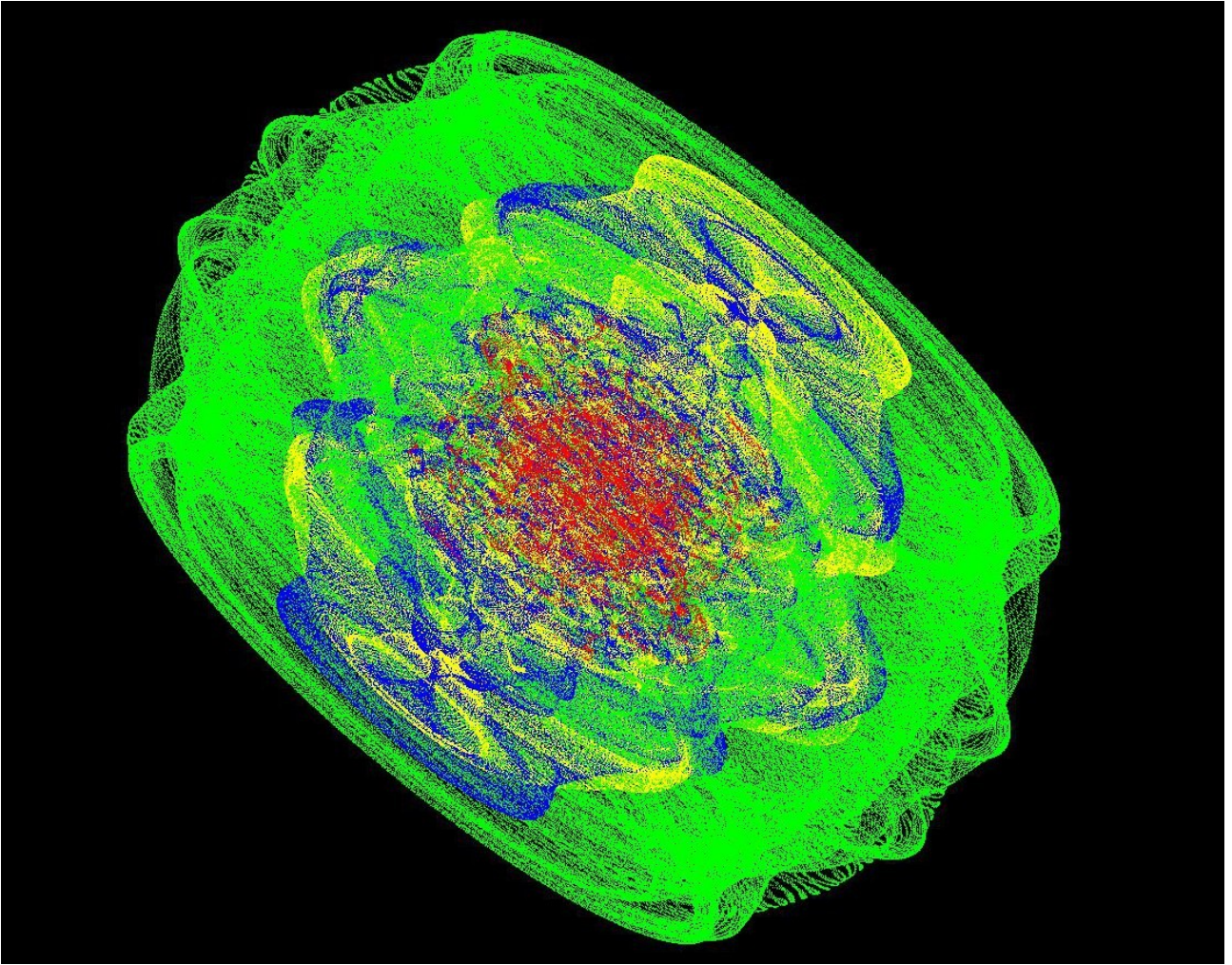


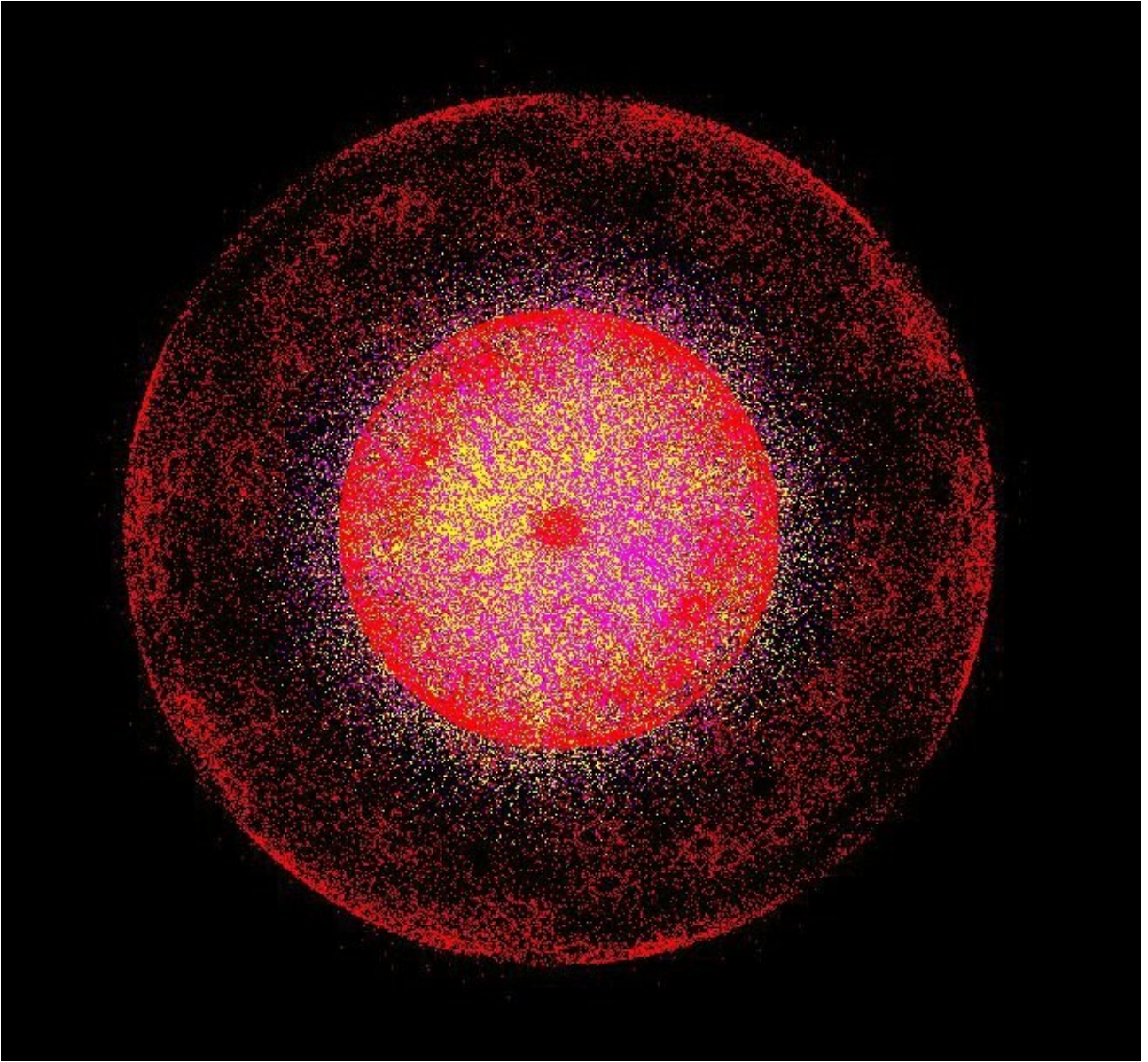


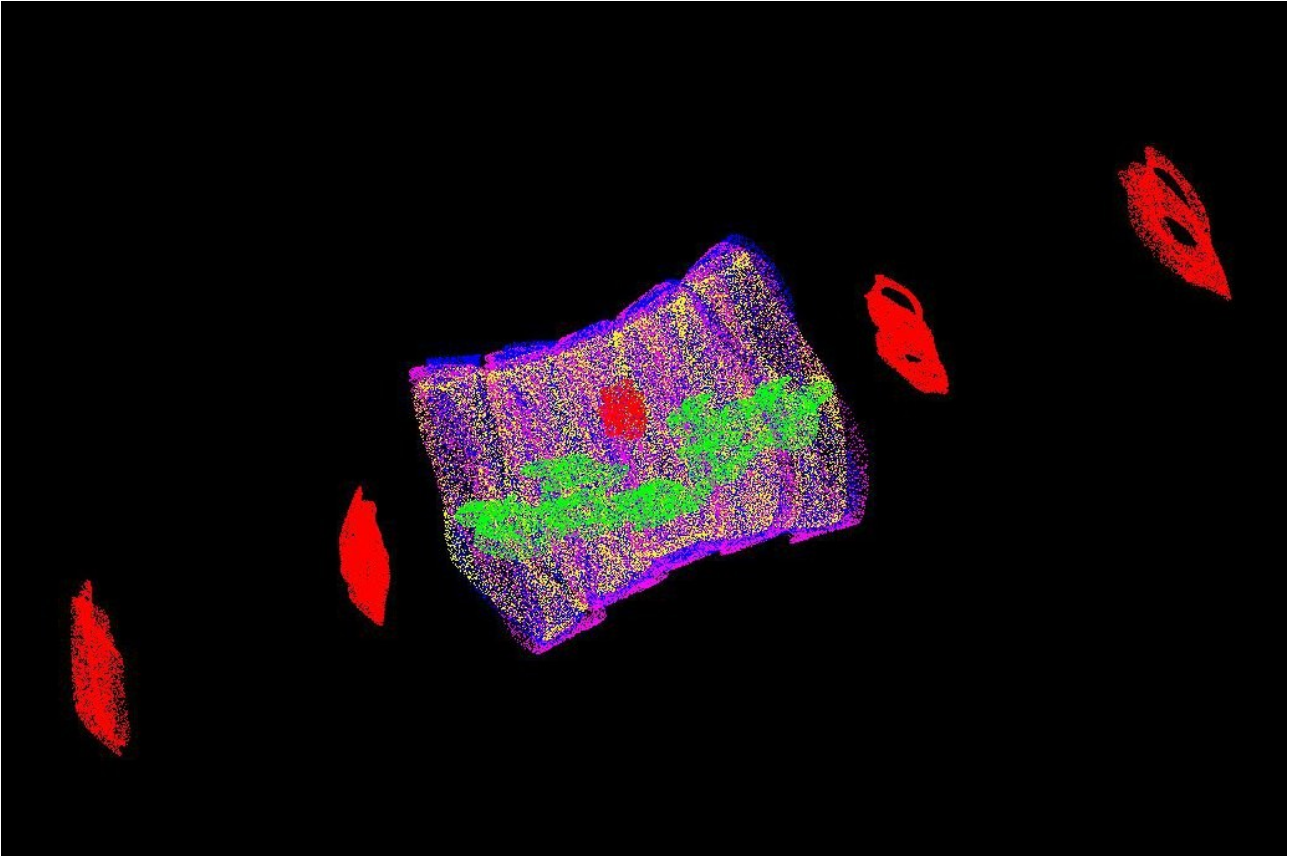


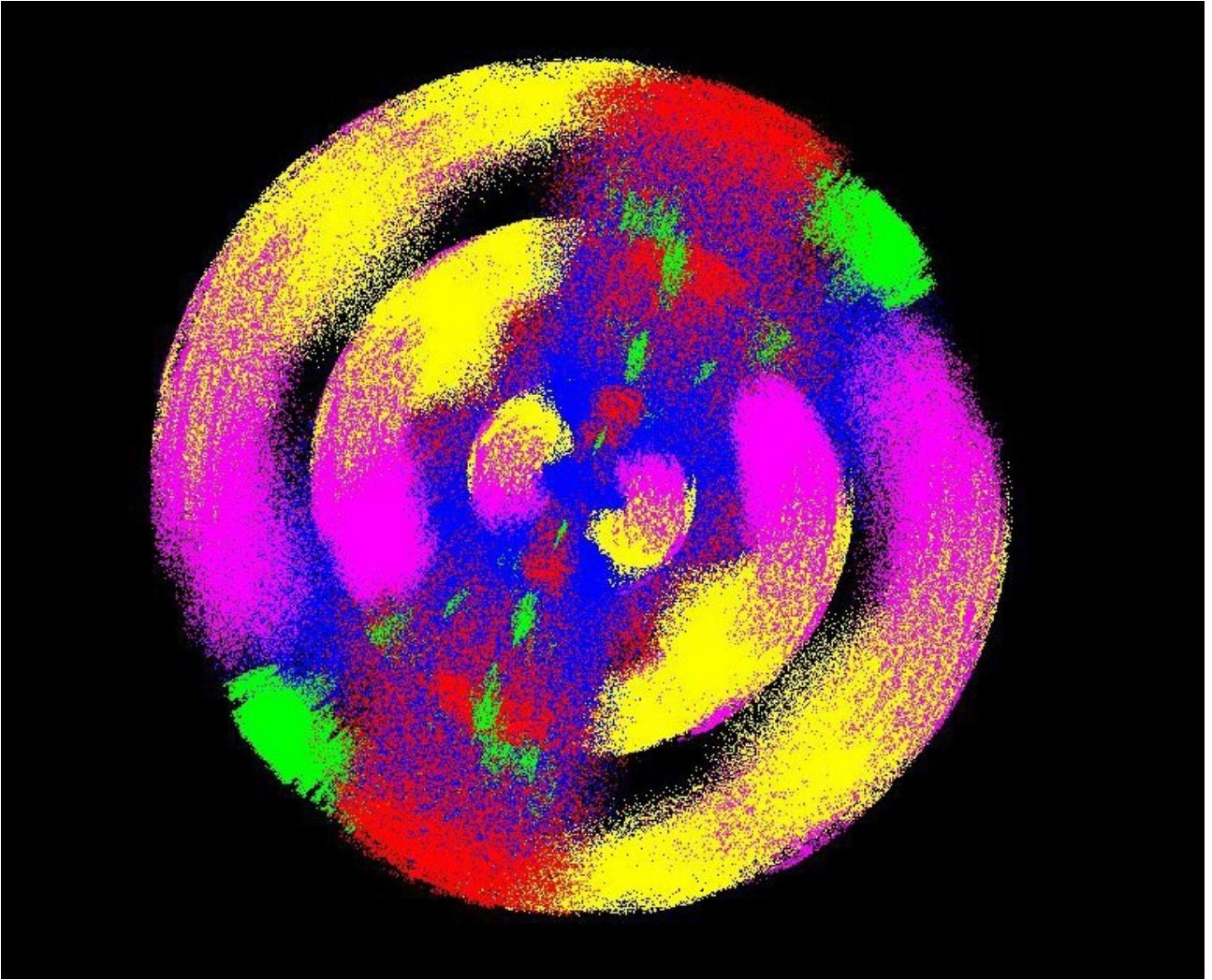


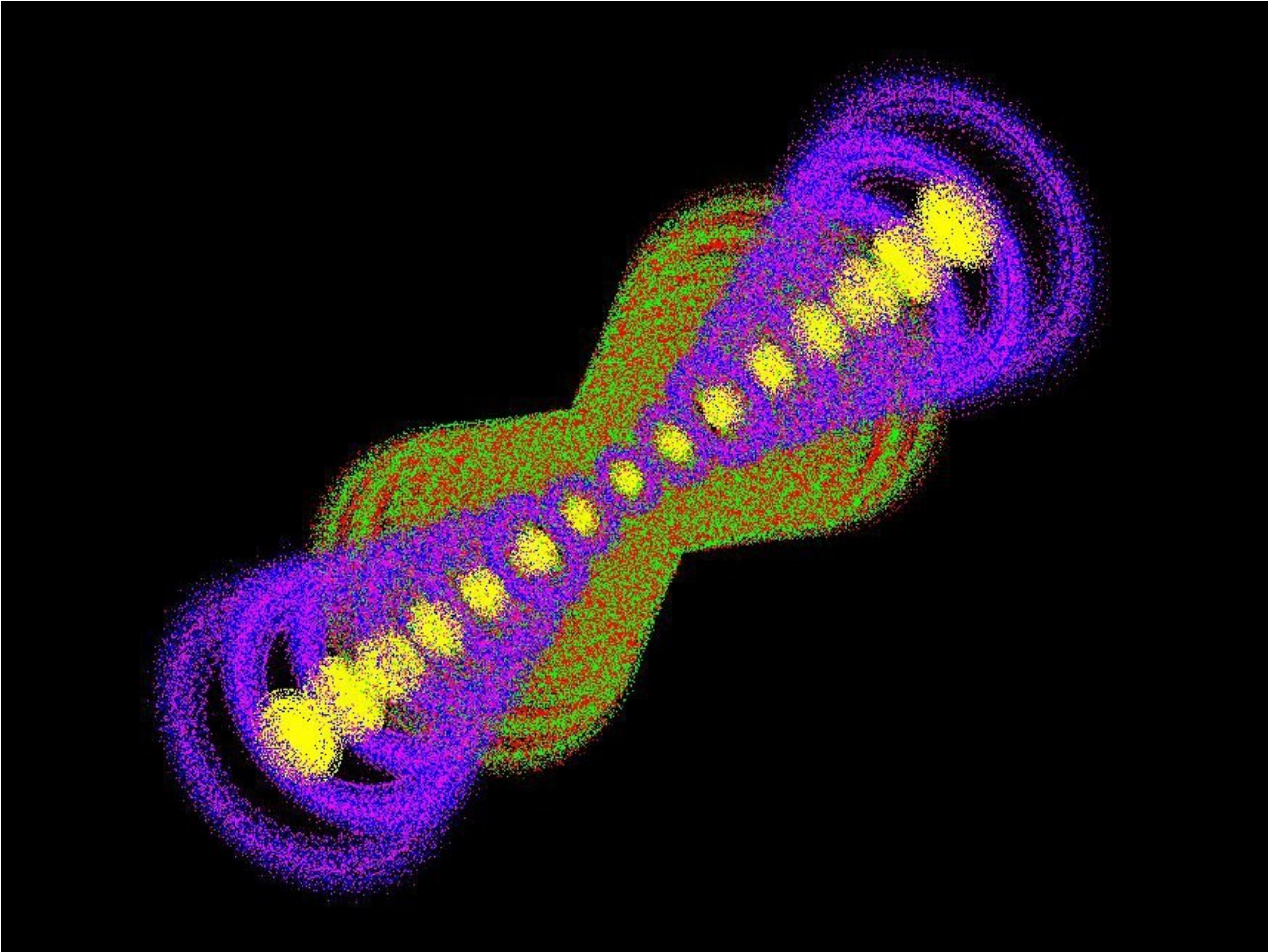


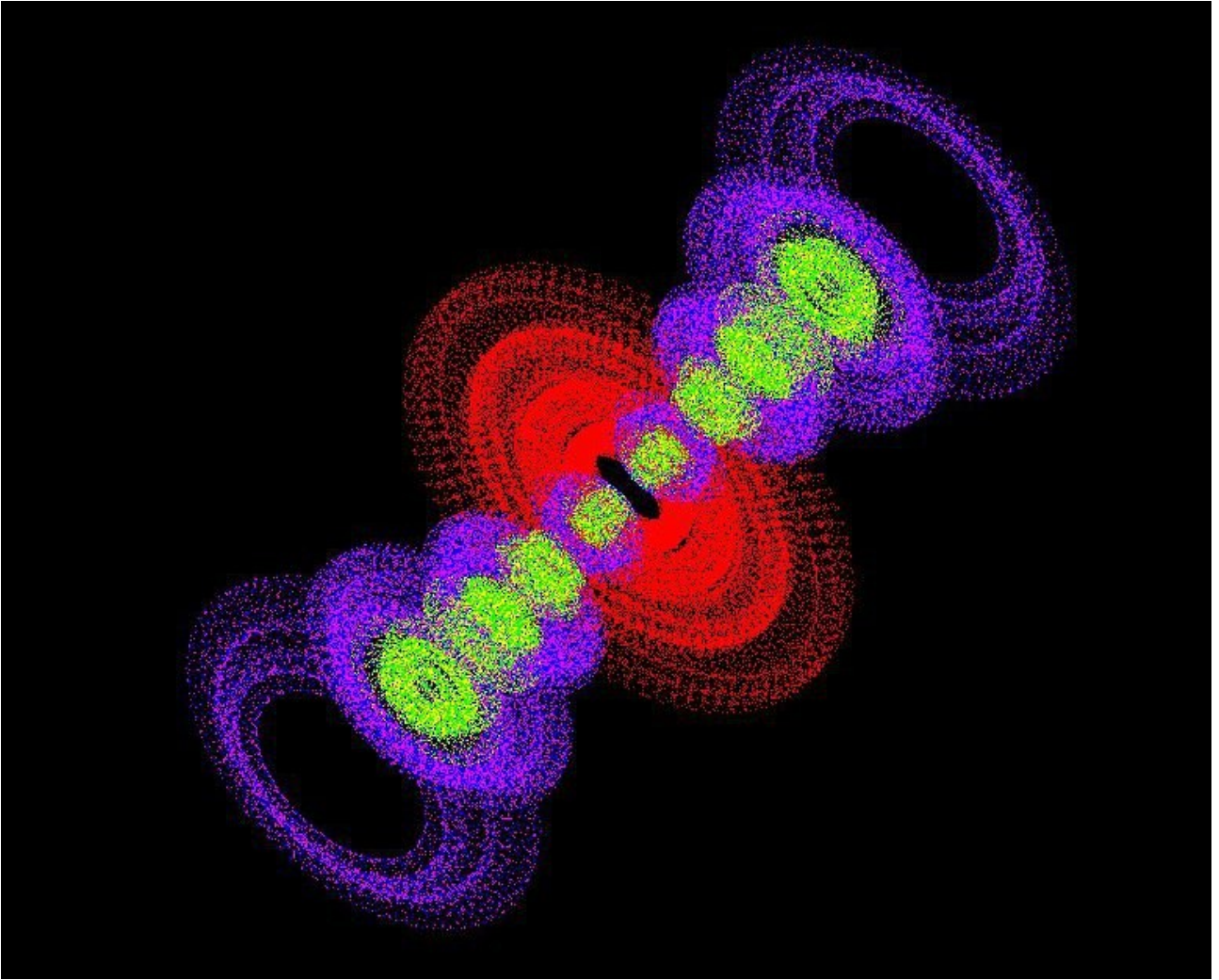


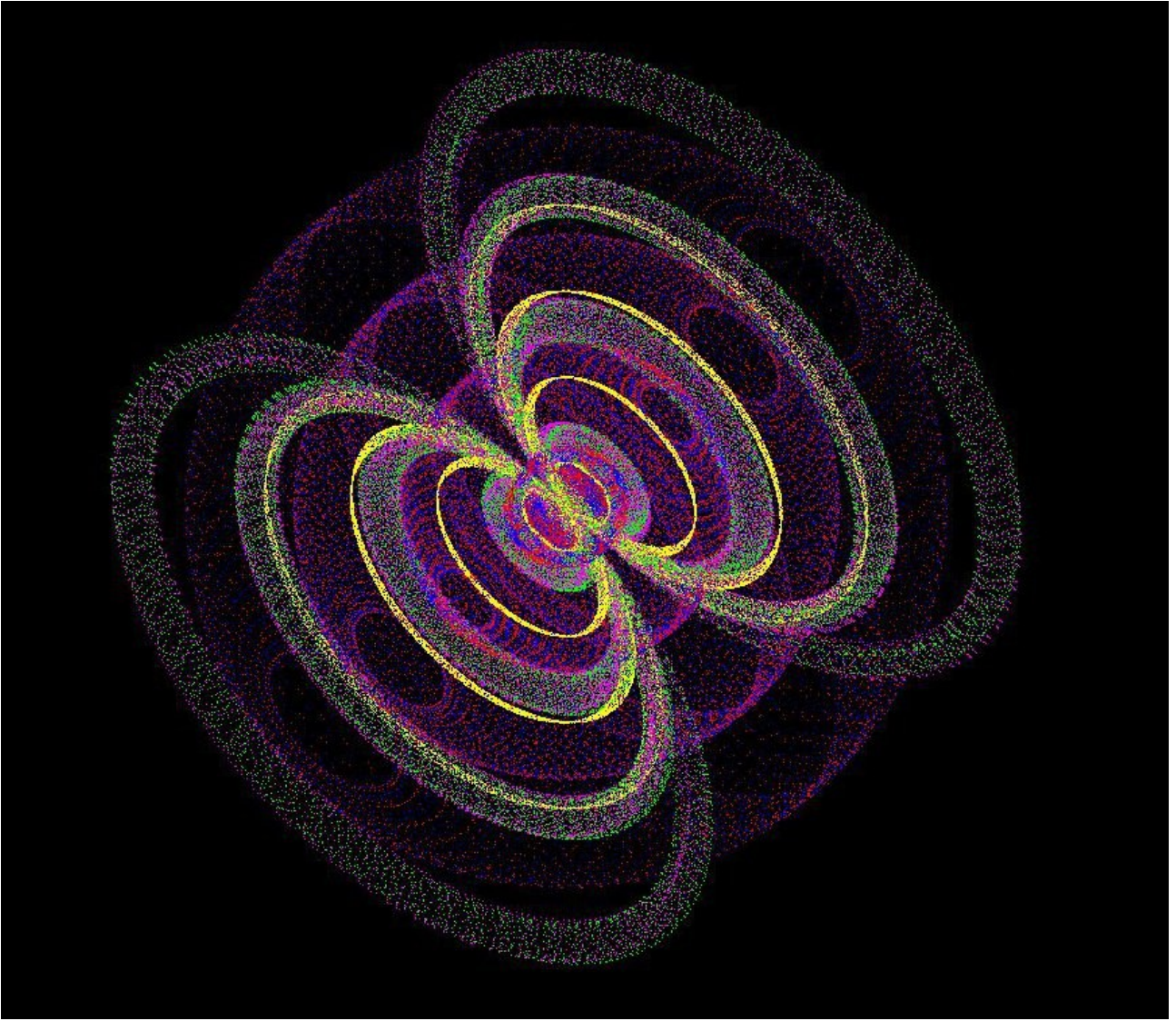


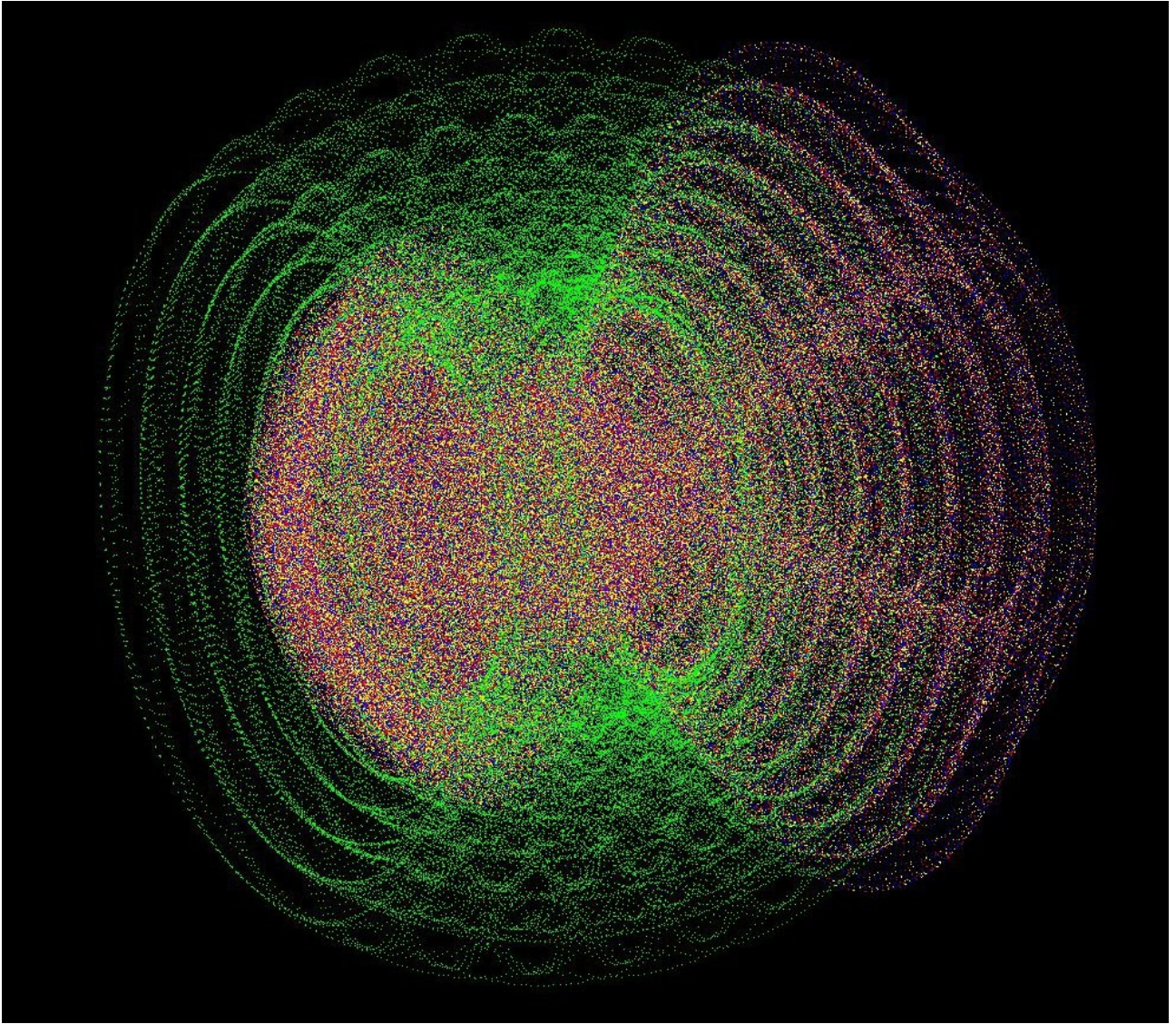


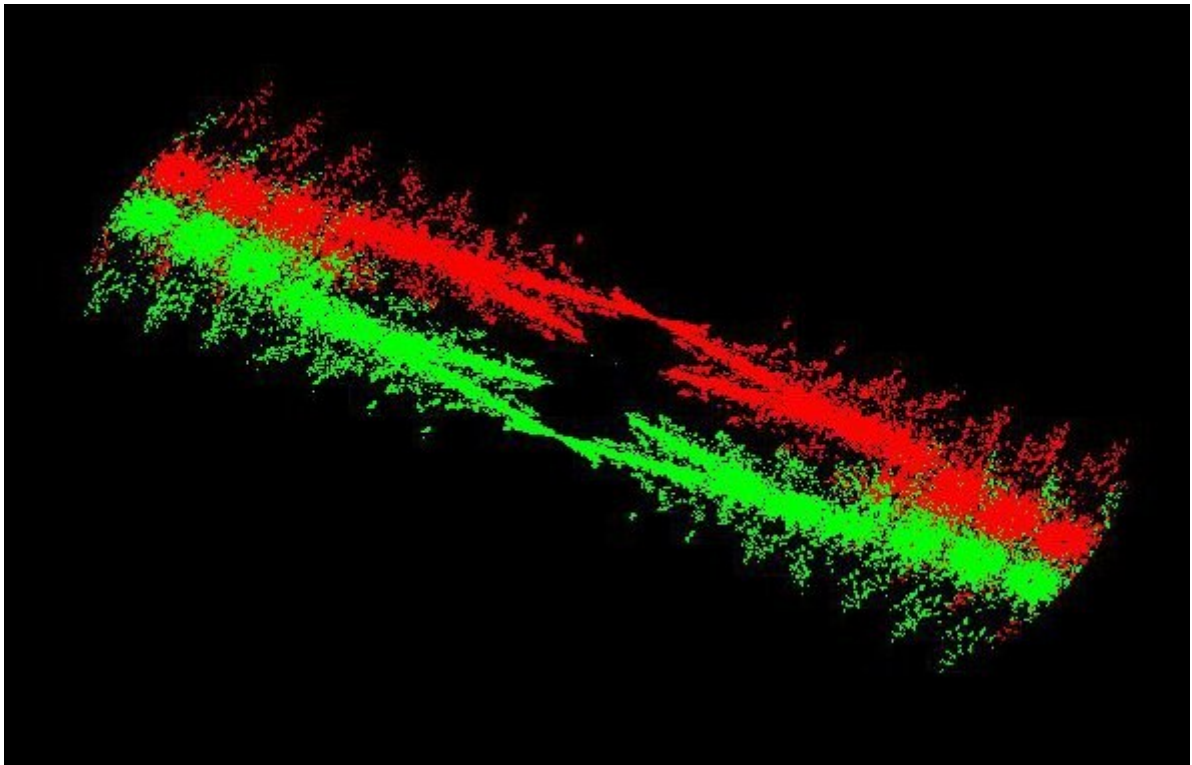


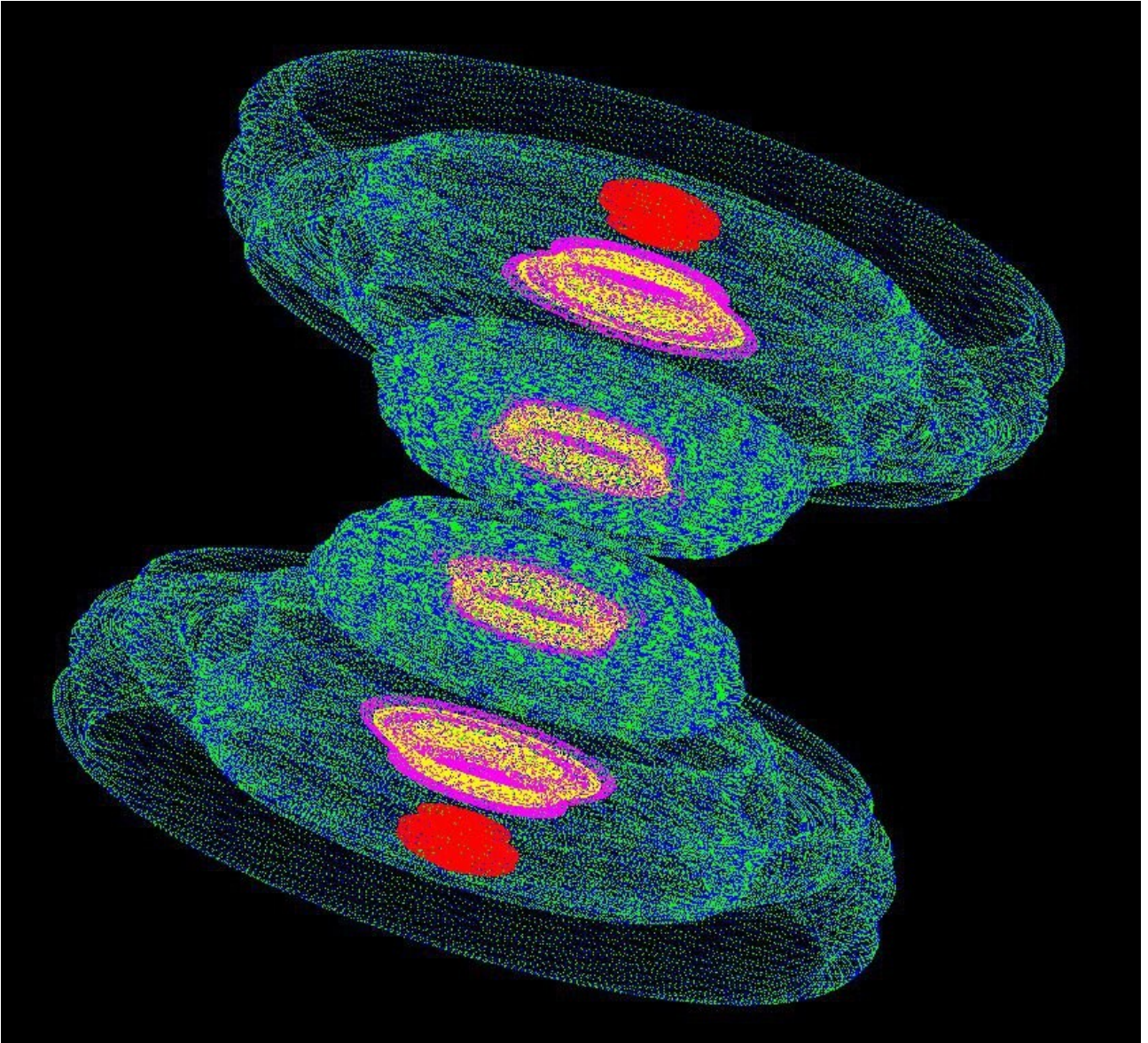


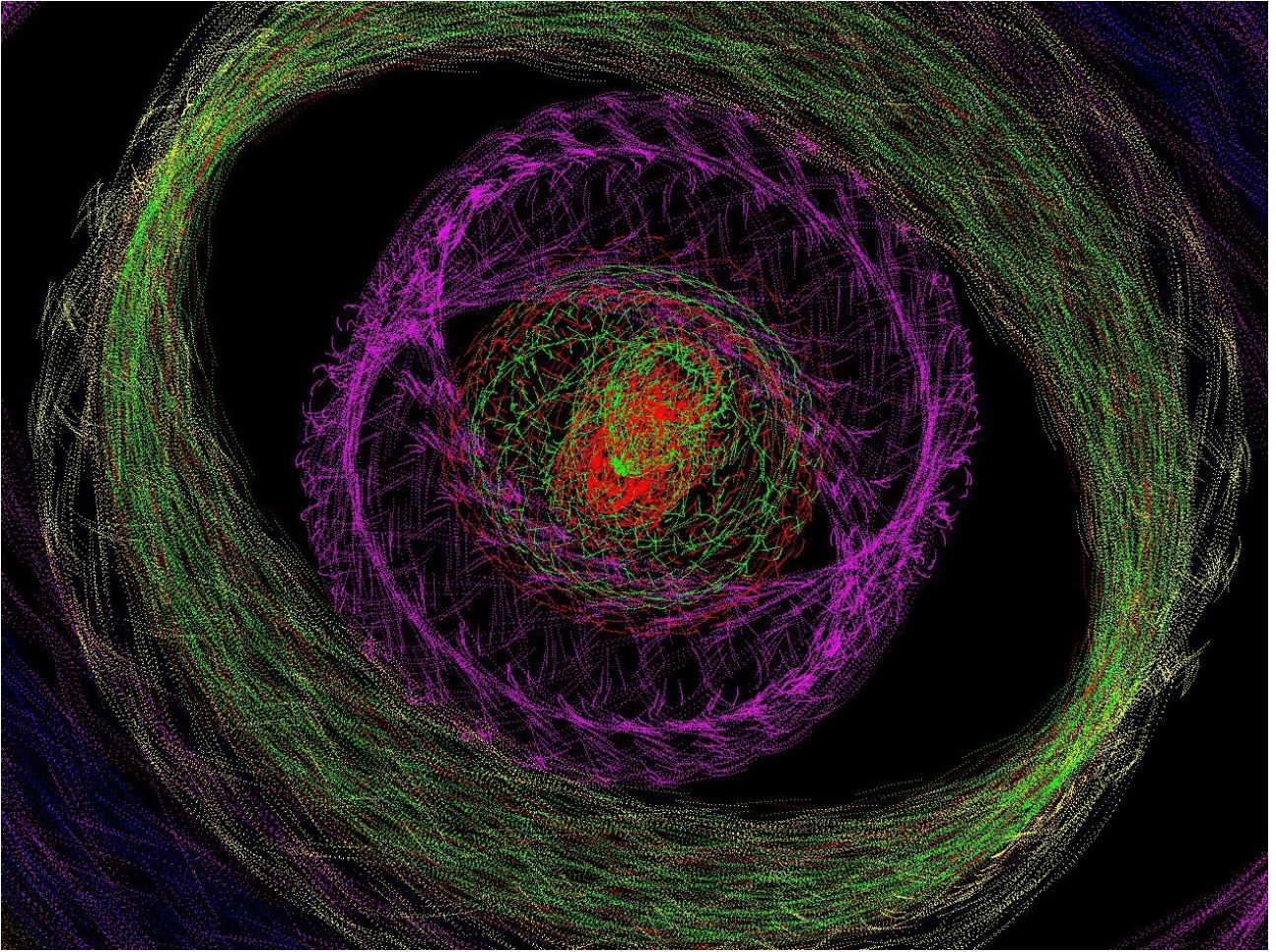


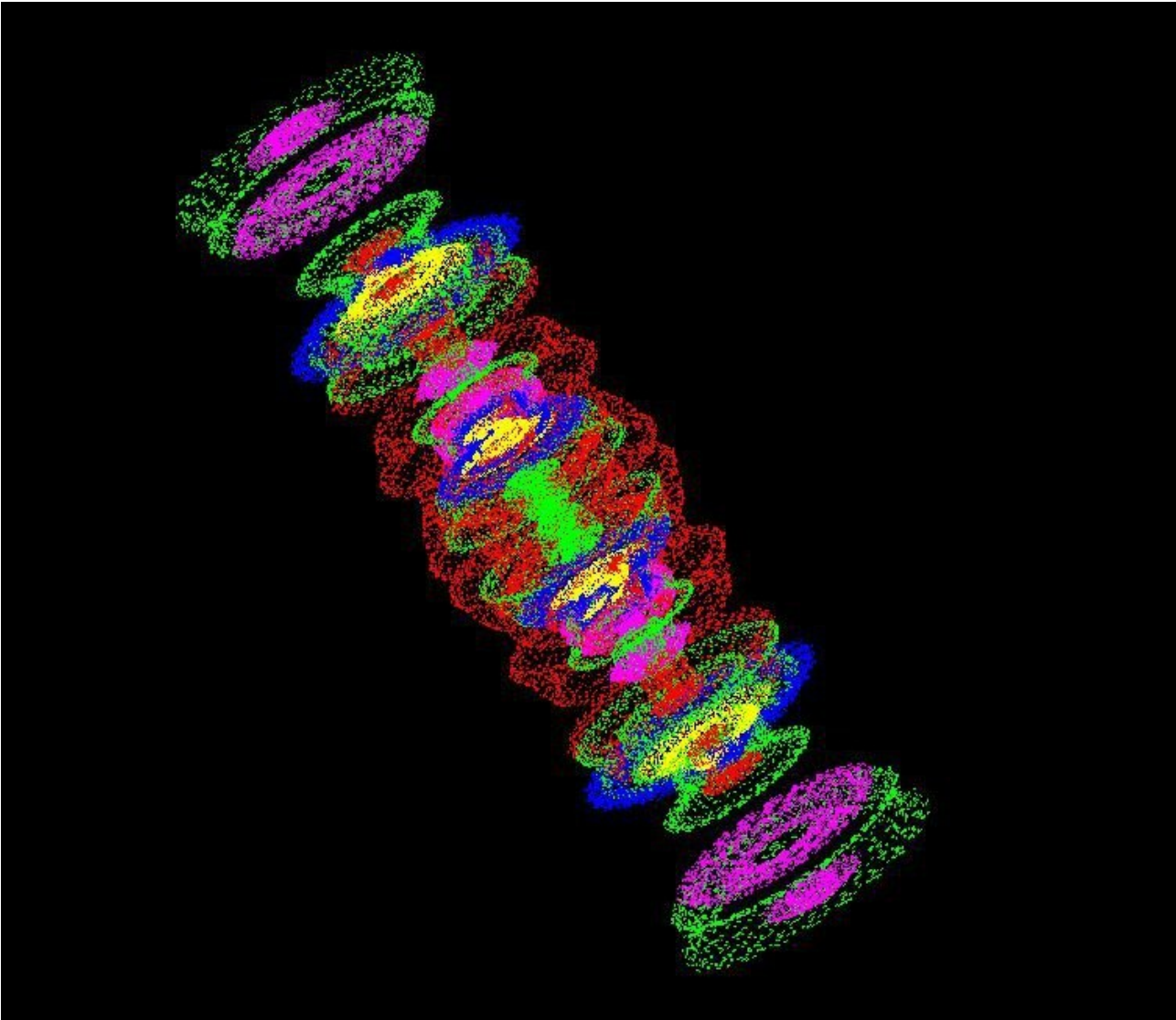
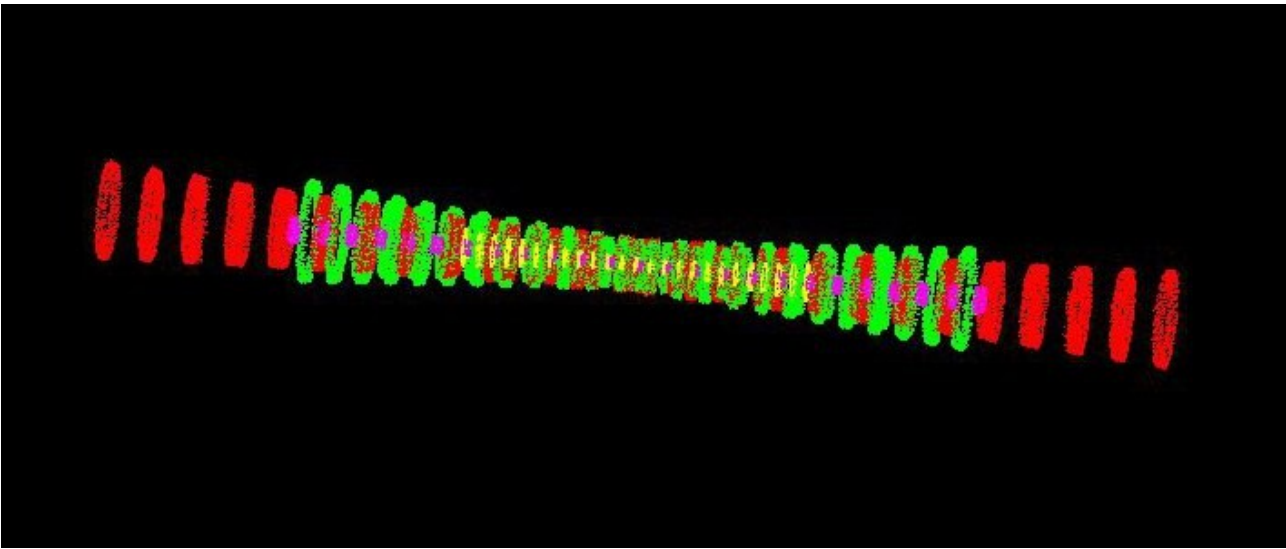


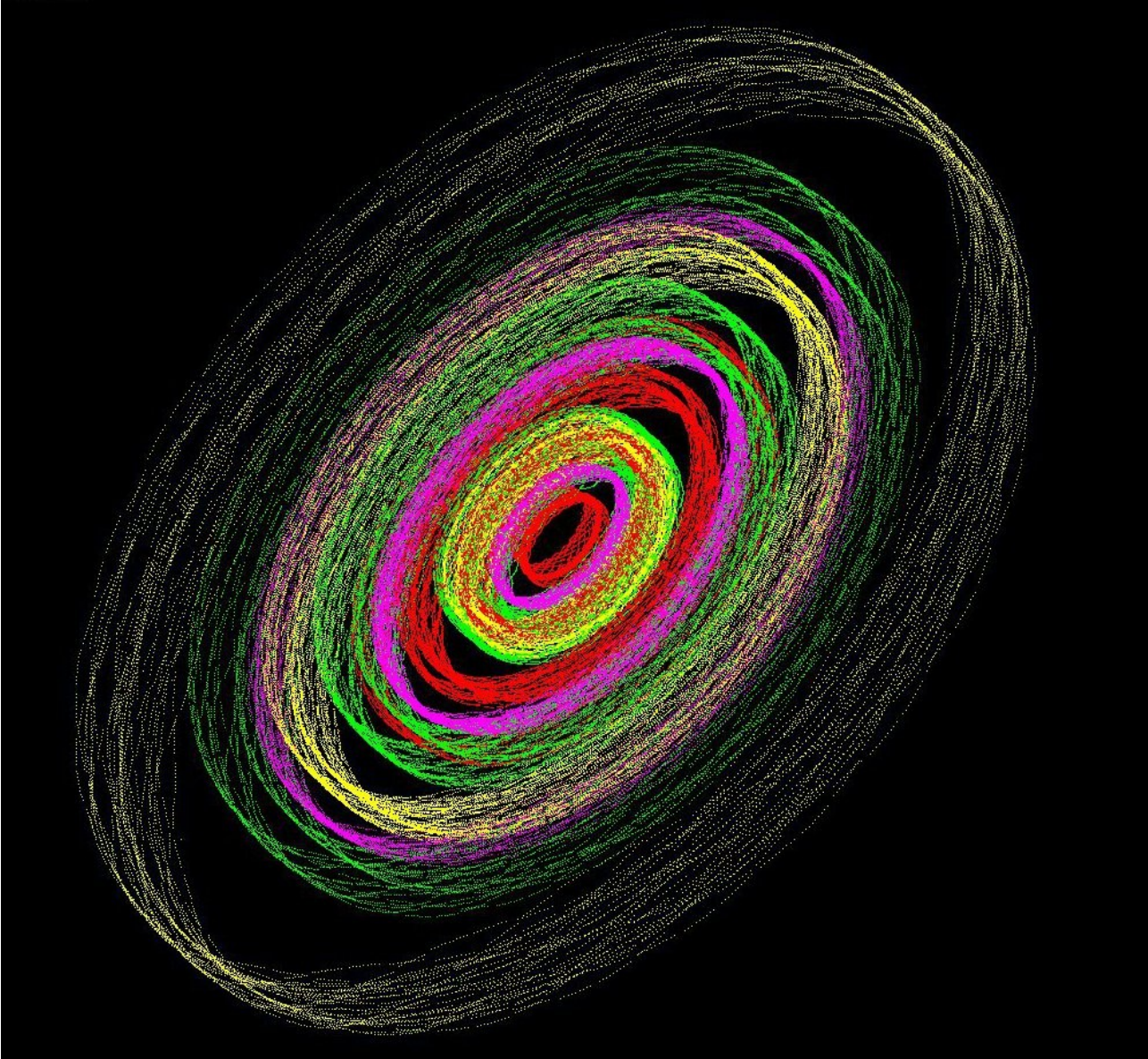


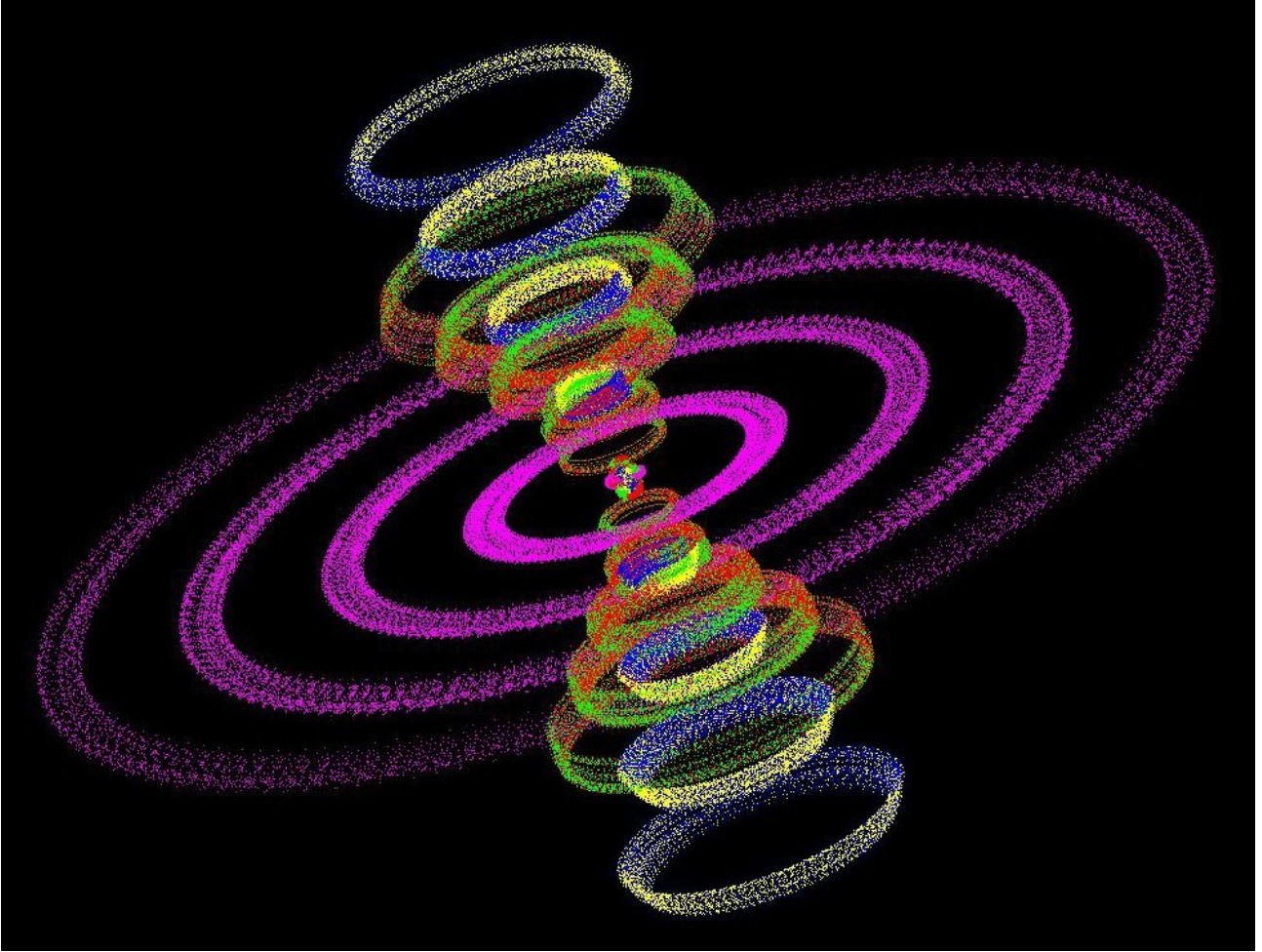


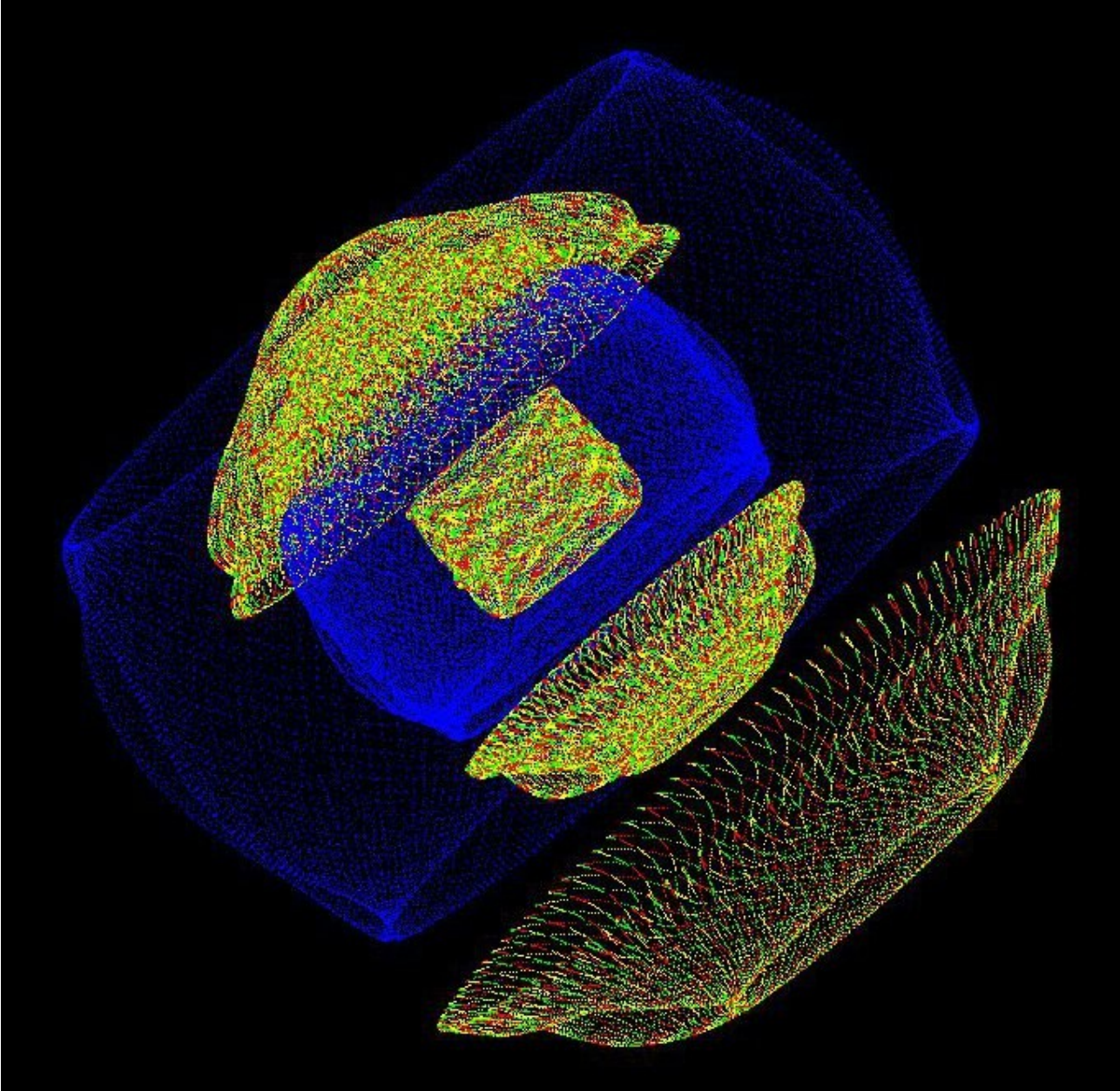


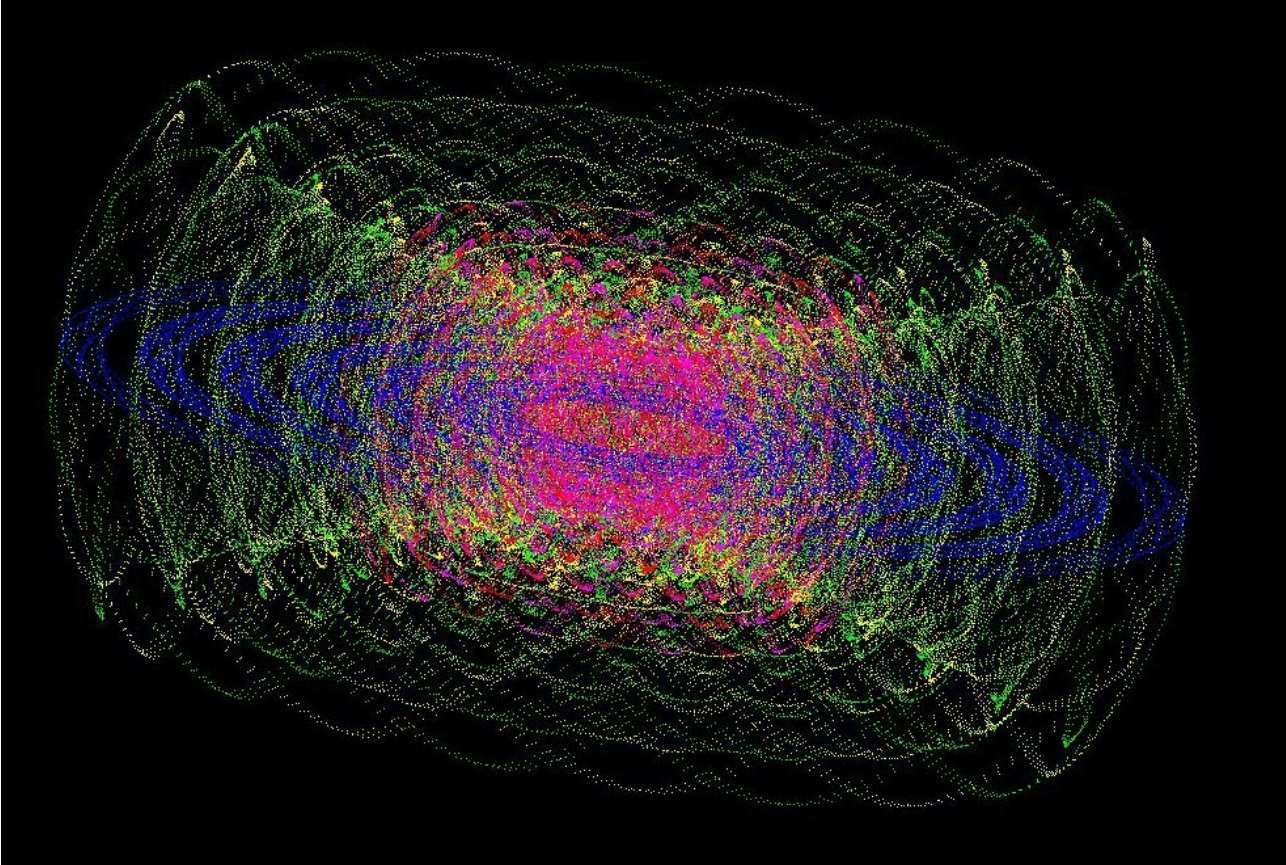


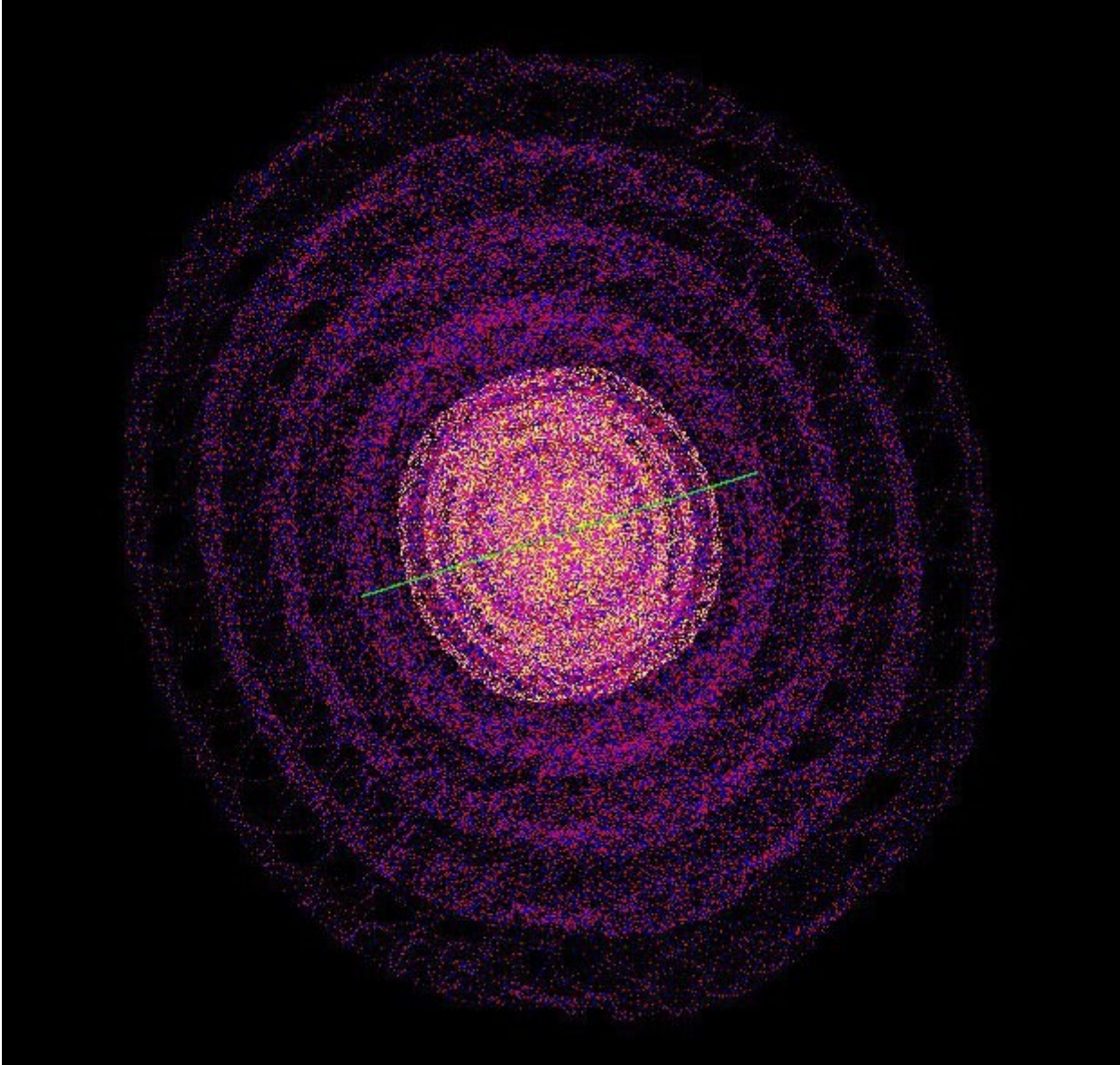


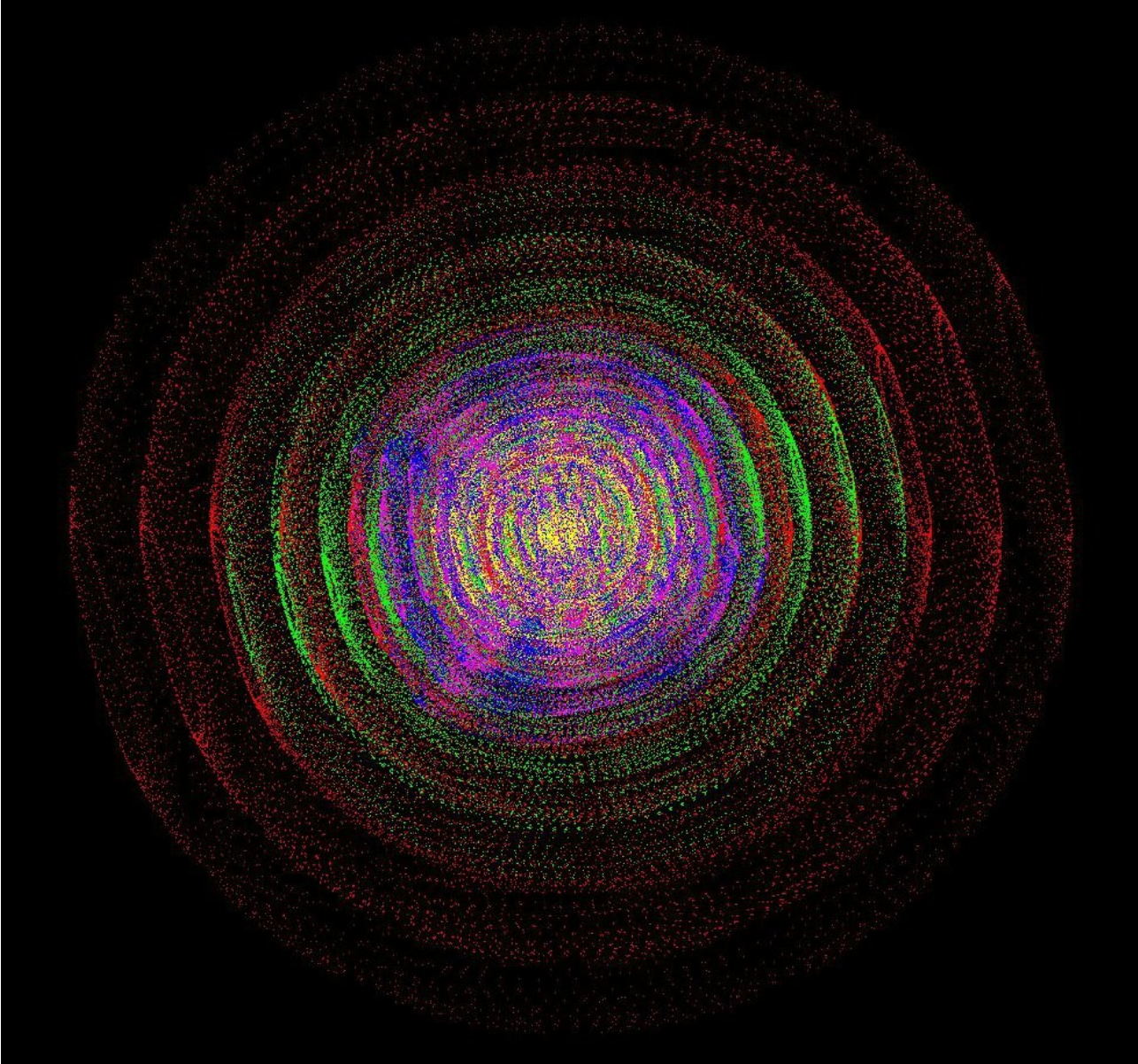


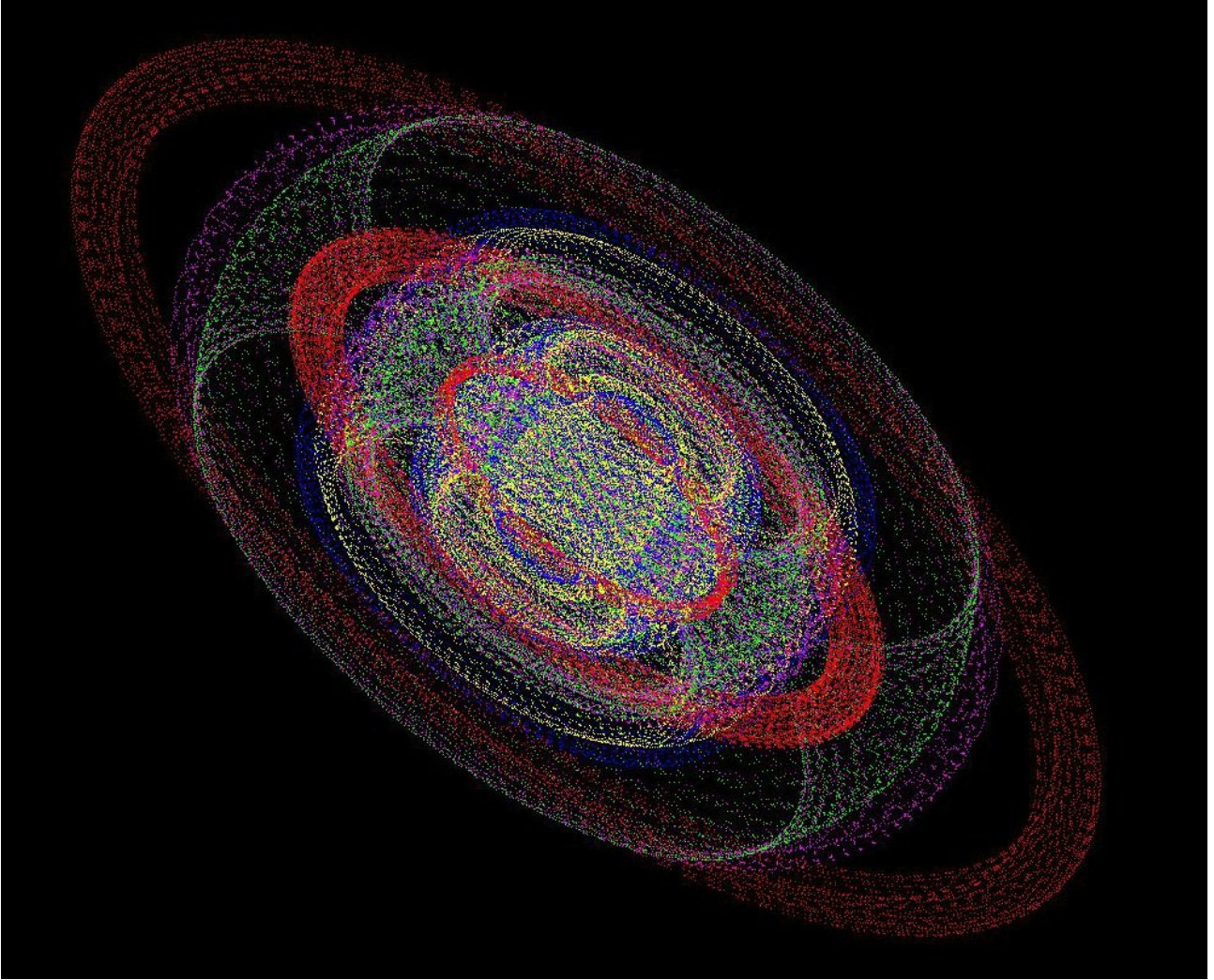


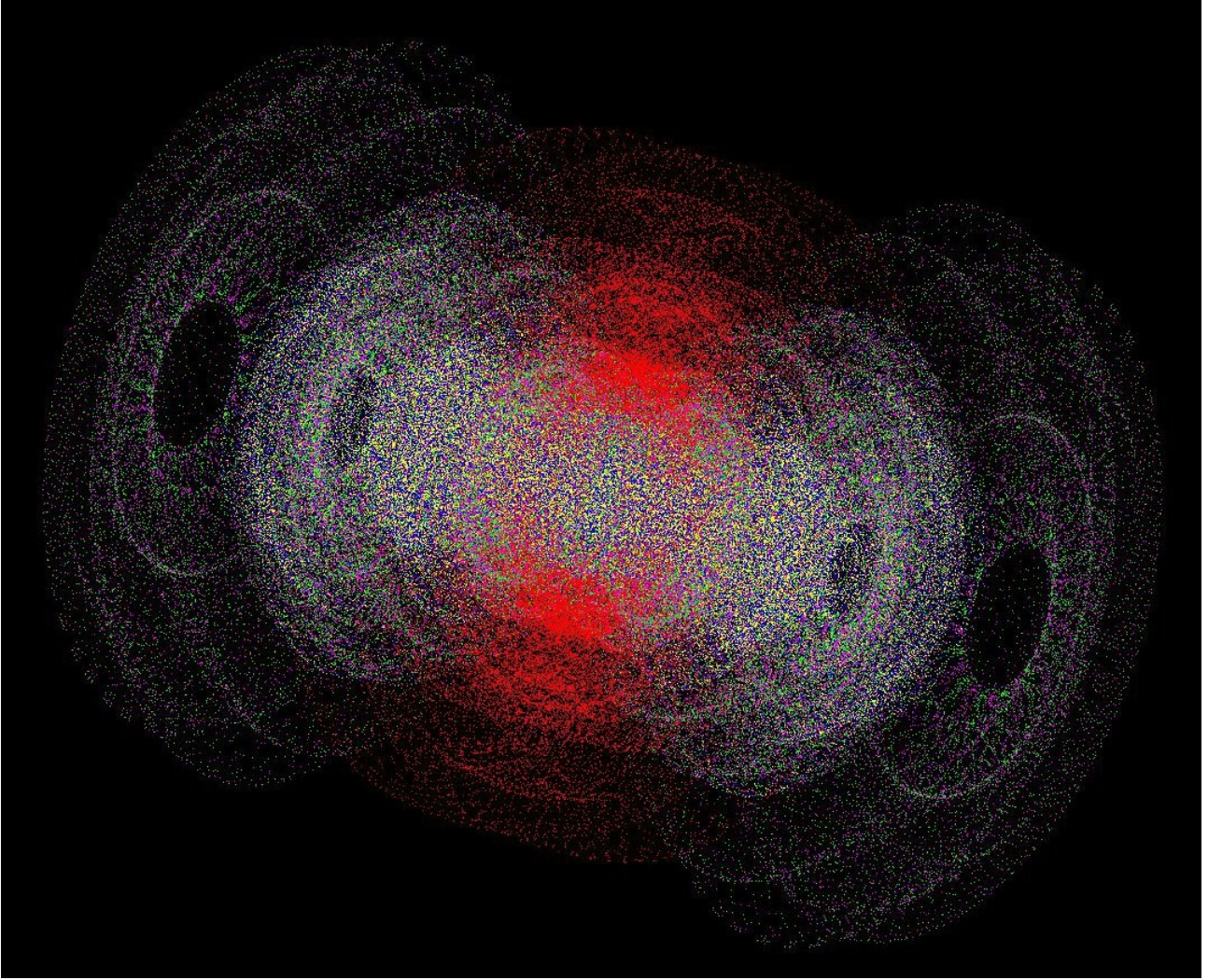


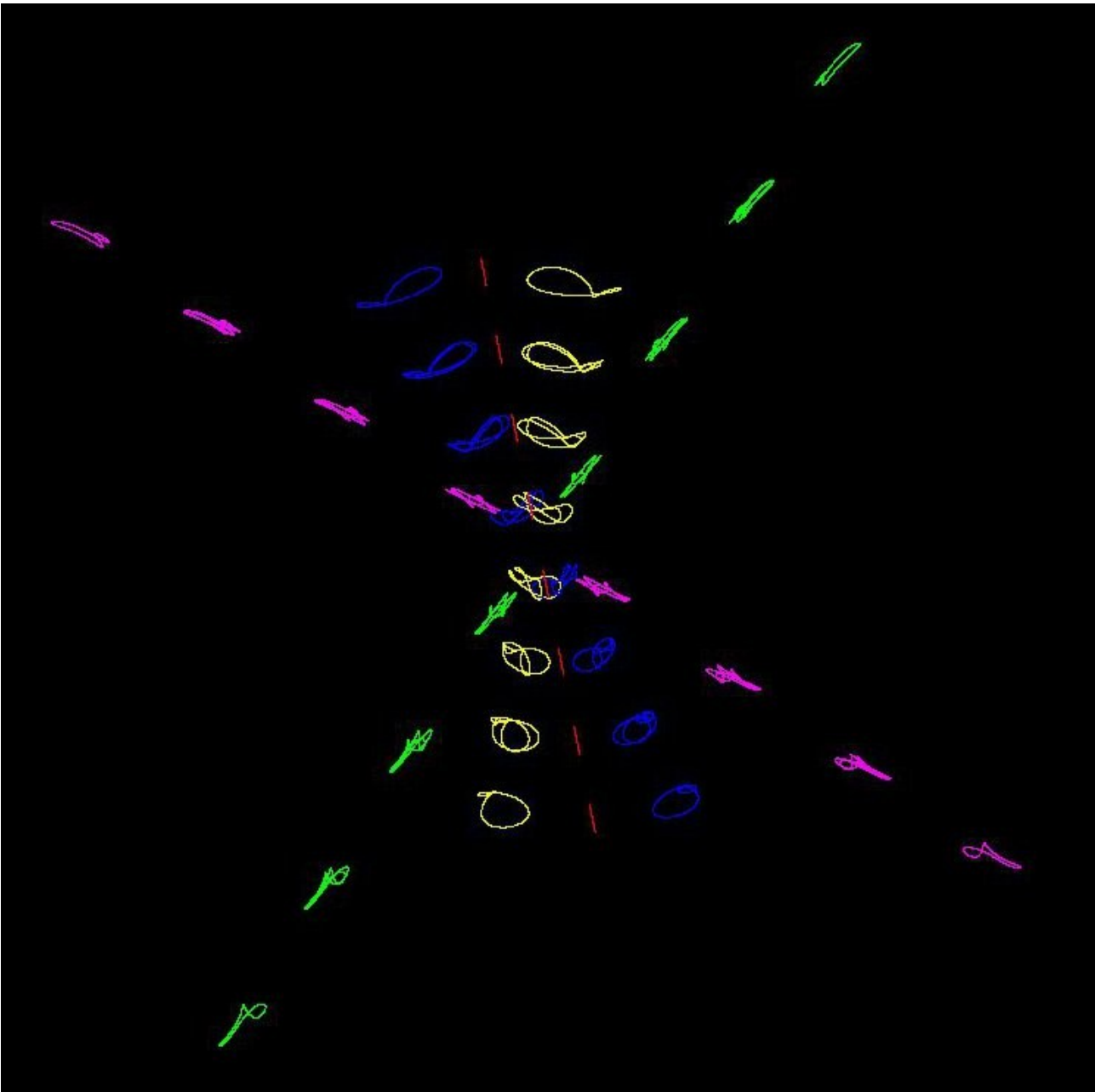


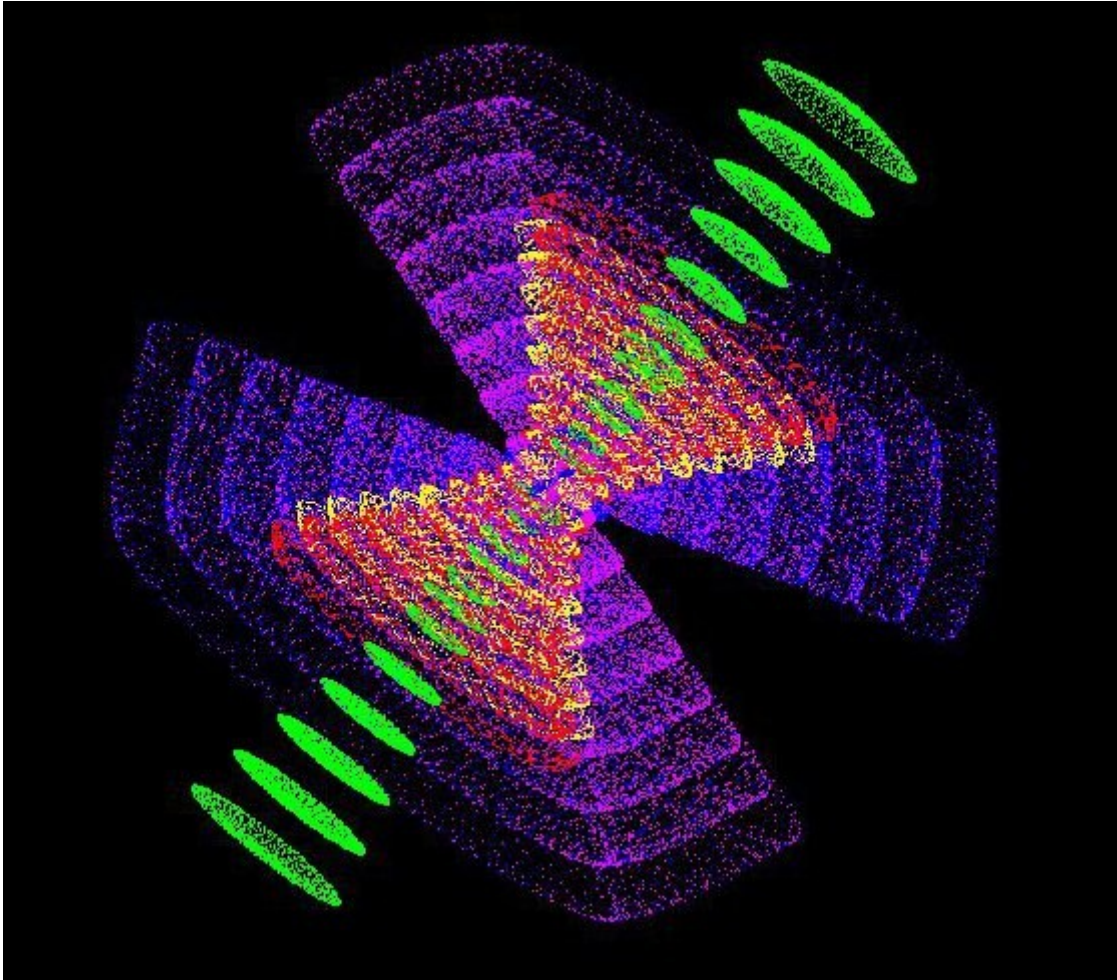


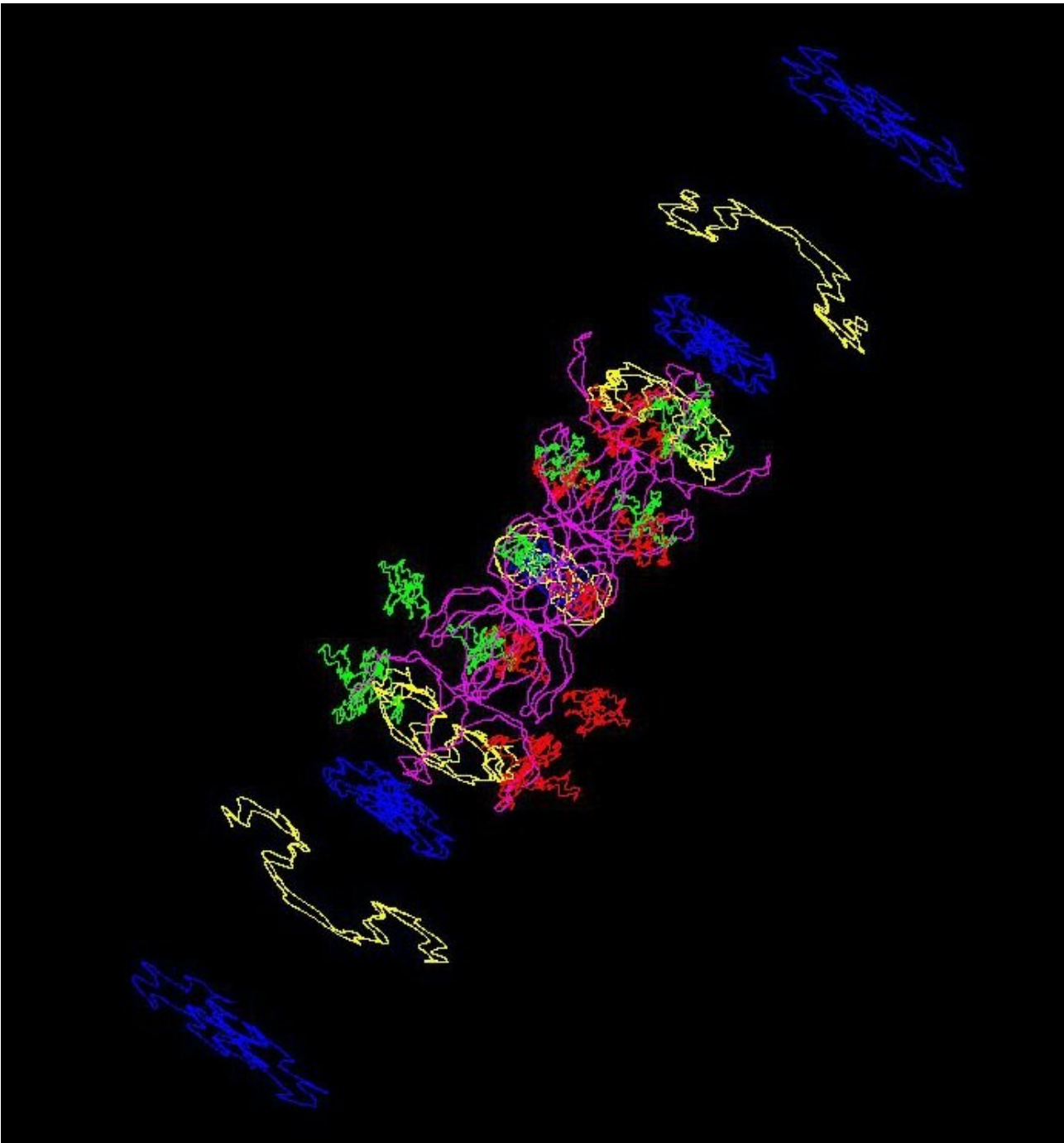














Ein Programm „Graph.exe“ steht unter folgender Adresse zum Download bereit:

<http://www.ohm-hochschule.de/bib/ohmdok/hoewel/Graph.zip>

Alternativ kann das Programm von Wolfhard.Hoevel@ohm-hochschule angefordert werden.

Mit diesem Programm können interaktiv beliebig viele Strukturen im dreidimensionalen Raum generiert werden.