

# 第一章

# A-Life 是什麼

以人工方式重現生命的想法是如何產生的呢？

這一章將概述冀望藉由代表人工生命的「A-Life」運算，掌握生命機制的想法，在二十世紀到現在的資訊科學系譜，以及與人工智慧研究中，究竟是如何定位的。

## 1.1 科學上的生命定義

直至二十一世紀初，現代科學尚無法解開生命的全貌。儘管在整個二十世紀，地球上的生物分類急速增加，但是就我們所知，至今仍有大量未被發現的微生物。

即使我們逐漸理解生命系統的發生、成長、遺傳、自我複製、適應、演化、分化等各種機制，並透過克隆細胞、幹細胞等遺傳工程研究，在某種程度上，可以設計生命的發生，卻仍未達到整合所有生命特性，從零開始創造生命體的階段。

醫學、工學等領域追求的是製作出有益人類的東西。其中與生命有關的操作，主要是為了治療或預防人類的疾病、提高生活的便利性與效率等。人類的平均壽命持續延長，期望透過再生醫療的發展，治療不治之症，在開發新藥的過程中，運用人工智慧也提升了效果。

隨著遺傳工程的發展，由父母控制孩子的基因，製造出設計嬰兒（designer baby）也可望成真。此外，持續提升身障者的義肢技術，將讓身障者獲得超過正常人的身體能力。現在我們已經知道在生物體內，有無數微生物運作的微生物群系，會對身心健康帶來影響，而且有機栽培的有機食品或注入人工栽培菌類的食品，已大量出現在門市。

如上所示，當科技及研究工程與生命結合之後，人類對生命的認知，也從「與生俱來無法改變」逐漸轉變成「可以按照目的調整」。

可是，生命並非是為了某個目的而存在。

飛機的發明並不是為了重現鳥、昆蟲等生命，而是為了人類移動的目的，利用其他構造，重現鳥或昆蟲擁有的「飛行」特性。根據達爾文的演化論，我們瞭解到（假設理論成立時，大部分科學家相信的事情），所有生命遵循的物競天擇（Natural Selection）過程沒有目的。鳥類的翅膀發達，不是為了適應飛行，而是剛好讓翅膀發達的演化群體物適應了物競天擇的結果。

我們可以說，對受到社會合目的性（Zweckmässigkeit）驅動的工程學而言，無任何目的產生的生命現象，沒有真正全面理解的動機及存在的必要。

這恐怕是即使我們在軍事及經濟需求等強烈合目的性的推波助瀾下，創造出超先進的電腦，但是至今卻依舊無法觸及生命本質的原因之一。

我們可以說「ALife」（人工生命）面對的問題，就是生命是否能用運算這個典範來掌控。

## 1.2 人工生命是實驗數學

「人工生命」這個名詞是翻譯自 1986 年美國電腦科學家 Christopher Langton 提倡的「Artificial Life」，英文簡稱為「ALife」。

人工生命是指透過電腦、化學實驗、機器人實驗，探究「生命是什麼？」的領域。尤其 ALife 把現有生命（life-as-we-know-it）當作其中一部分，探索「可能存在的生命」（life-as-it-could-be）之可能性，取得研究更大型生命型態的方法。

現今，使用電腦模擬來思考生命究竟有何意義？是一個很重要的前提。

電腦的歷史是因為想釐清「計算究竟是什麼？」才開始發展的。

前言曾經提及，Alan Turing 建立了讓電腦概念成立的「可計算性理論」，而 John von Neumann 製作出實際的機器，讓該理論具體成形，他們最大的貢獻是擴大了數學這項工具的能力。這兩位有一個有趣的共通點，那就是他們都試圖用數學描述生命。Turing 在 1952 年發表了利用化學反應算式「反應擴散系統（reaction-diffusion system）」，創造生物圖紋的論文。von Neumann 在 1966 年發表的論文中，以數學方式描述生命最根本的特徵「自我增殖」<sup>[1]</sup>。

## 第二章

# 建立生命模式

- 13 現實世界裡的生命是內外交織多種模式，同時進行運作，即使沒有建立縝密的整體設計圖，結構也會隨著時間成長，按照具有整合性的「自我組織化」來運作。

本章將把重點擺在關鍵字「自我組織化」上，同時執行反應擴散系統的模擬、細胞自動機 (cellular automaton)、及生命遊戲等程式的程式碼，說明可以用何種理論描述生命現象建構出來的各種模式，最後介紹在現實世界裡，計算自我組織化的可能性。

### 2.1 自我組織化的自然界模式

自然界充滿著非人工設計出來的美麗模式。

例如，裂縫、雲層、漩渦、都市的塞車狀態、網際網路的資料傳輸等呈現出來的模式 (圖 2-1)，乍看之下形成了連結生命與非生命現象的「有機模式」。



圖 2-1 自然界的模式

自左圖起) Fissures BY thephotographymuse (CC:BY-SA 2.0) <https://www.flickr.com/photos/marcygallery/2535052057/> Clouds BY Daniel Spiess (CC:BY-SA 2.0) <https://www.flickr.com/photos/deegephotos/3714844027/> Spiral Wake BY Andrew E. Larsen (CC:BY-SA 2.0) <https://www.flickr.com/photos/papalars/987545947/in/album-72157594554255900/>

自然界的各種例子顯示出，不用依賴建立模式的構成元素，就可以產生普遍性的模式與構造。這個意思是指，即使元素本身沒有變化，只要元素之間存在著非線性的相互作用，就會產生模式。這是非線性物理學研究出來的結果。另一方面，隨著元素變化也可能產生模式。

元素沒有變化時，能輕易當作人類操作的對象來處理。人類把從自然界中看到，簡單創造模式的方法，運用在流行時尚、建築、或日用品的設計上。例如在建築界，如果要建造複雜的結構，會根據顯示出完成結果的設計圖，亦即完成圖來打造建築物。

一般而言，我們無法建造不知道完成結果的東西。可是，自然界沒有像建築界一樣的完成圖，而是產生「自發性的組織化」，構造隨著時間而成長。

這種沒有完成圖，卻能建構出結構的情況，稱作「自我組織化」。自我組織化除了生物之外，也存在於颱風、金平糖、白雪結晶、卡門渦街\*等各種自然現象中，這種自我組織化的原理是支撐所有生命形成的特徵之一。ALife 的研究也是以自我組織化為核心。

可是，問題在於，在生物系統中，元素也會出現變化。例如，我們知道，取得基因資料、細胞分化與發展過程、大腦的記憶等，根本上與化學反應有關。

例如，神仙魚的 64 個細胞分裂了 100 萬次，最後發生原腸內陷（形成身體基本構造的胚胎初期，出現的細胞運動），經過細部分化，長出眼睛及尾巴。這意味著，如果眼睛及尾巴的細胞種類是基因表現模式的差異造成的，那麼在分裂次數前後，模式會不一樣。

這種完成一個「個體」的發育過程很特殊，不會出現在結晶或一般魚、貝殼表面的圖案上。如果要產生一個個體，微型物體得製作出巨型物體，而這個部分必須讓微型物體的行為產生變化。

在生命形成的過程中，比較容易瞭解的自我組織化模式範例包括，蛇的表皮、蝴蝶的翅膀圖案、熱帶魚的條紋等（圖 2-2）由化學反應產生的多元化現象。



圖 2-2 生物的圖案

自圖左起) Pomacanthus imperator BY Jordi Pay (CC:BY-ND 2.0) <https://www.flickr.com/photos/arg0s/14162350619/> Peacock Butterfly BY Tony Hisgett (CC:BY 2.0) <https://www.flickr.com/photos/hisgett/7822792836/> Lacertae skin BY Gruzd (CC:BY-SA 3.0) <https://commons.wikimedia.org/>

\* 譯註：在一定條件下，氣流繞過某些物體時，物體兩側會周期性地產生旋轉方向相反的兩排旋渦，如街道兩邊的街燈，是流體力學上的現象。

首先，讓我們用 Python 執行 Gray-Scott 模型的程式，創造出反應擴散系統的模式吧！

#### 範例程式的執行方法

範例程式位於 chap02 目錄，請切換至儲存該檔案的目錄再執行。

```
$ cd chap02
$ python gray_scott.py
```

執行之後，如果描繪出以下模式，代表成功了。

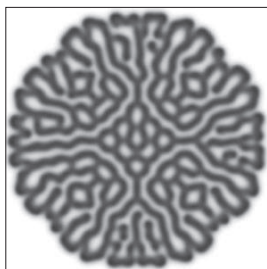


圖 2-3 Gray-Scott 模型的模式

### ● Gray-Scott 模型的結構

Gray-Scott 模型顧名思義是化學家 Gray 與 Scott 在 1983 年發表的論文中，提出來的概念<sup>[10]</sup>。

這個模型是描述代表 U 和 V 等兩個物質的濃度變數  $u$  及  $v$  的變化。「反應」與「擴散」就是物質 U 與 V 互相「反應」，透過媒介「擴散」。結果，空間中的 U 與 V 濃度會隨著時間產生變化，利用濃度高低，繪製出各式各樣的模式。

接下來，讓我們先說明反應。反應的概念圖如圖 2-4 所示。

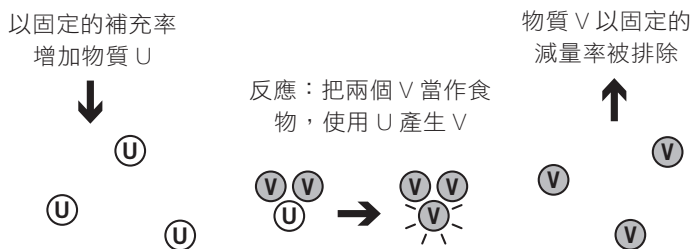


圖 2-4 Gray-Scott 模型的反應概念圖

參考「Reaction-Diffusion Tutorial」(<http://www.karlsims.com/rd.html>)

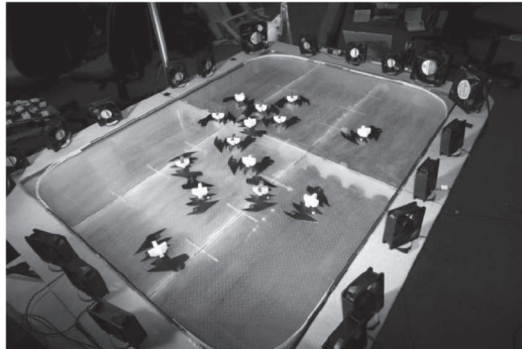


圖 3-17 Nathanael 與 Chrysantha 製作的實機

引用自文獻 [23]

看起來像蝙蝠的物體是使用轉軸當作動力，再加上磁鐵與橡皮製作而成。如圖 3-17 所示，有黑白兩種，和 Penrose 的鉤子一樣，會一邊旋轉，一邊整合成特殊形狀，結果就產生類似 Penrose 的複製單元。

這裡的重點是，出現特殊合體的蝙蝠形狀設計。這個形狀是為了用遺傳演算法演化，產生特殊性而設計出來的。因此，和 von Neumann 的通用複製子不同，也沒有磁帶。利用遺傳演算法，準備各種類型的形狀，隨機選擇，反覆嘗試錯誤之後，使得某個形狀出現反覆自我複製的狀態。

如果是單純的形狀，會與任何物體連結，形成一大塊。因此，讓「反應的特異性」演化，只允許某個已經決定的形狀組合、自我增殖。這一點和 Langton 的複製子一樣，雖然不是通用複製子，卻有非平凡性！這裡所謂的非平凡性是指，無法輕易知道答案，卻有某個重要部分。

此外，在這種來回晃動的環境下自我複製，需要能因應雜訊的穩健性。這是指，即使想利用雜訊組合物件，如果沒有成對，就無法合體，這種形狀的特異性確保了穩健性。

可是相對來說，演化的可能性會出現問題。過於穩健的自我複製接近晶體生長，完全不會演化，這一點請見下一節的說明。

### 3.3 自我複製與演化

自我複製只是生命系統的其中一半，更重要的另一半是演化。

四十億年前，地球誕生，當水開始形成地球表面之後，就產生了可以自我複製的分子，讓演化露出一線曙光。出現能自我複製的個體，開始物競天擇。換句話說，複製出來的子孫，只有環境適應力最優秀的個體才會存活。

上述的機器與磁帶可以模擬許多生態系。各個機器和磁帶會按照其數量成等比例增加，舊的機器與磁帶有一定的比例會被消滅。

你可以從中觀察到哪種磁帶會建立編碼機器，哪種機器會讀取磁帶，是否發生自我複製。

### 3.3.2 核心網路

「磁帶與機器」模型在幾乎沒有外部雜訊，亦即很少出現被動突變的情況下，只有最低限度的自我複製網路（亦即一個磁帶編碼一台機器，該機器可以完全複製磁帶）加上側鏈（磁帶有時會產生雜訊）的簡單網路存在。

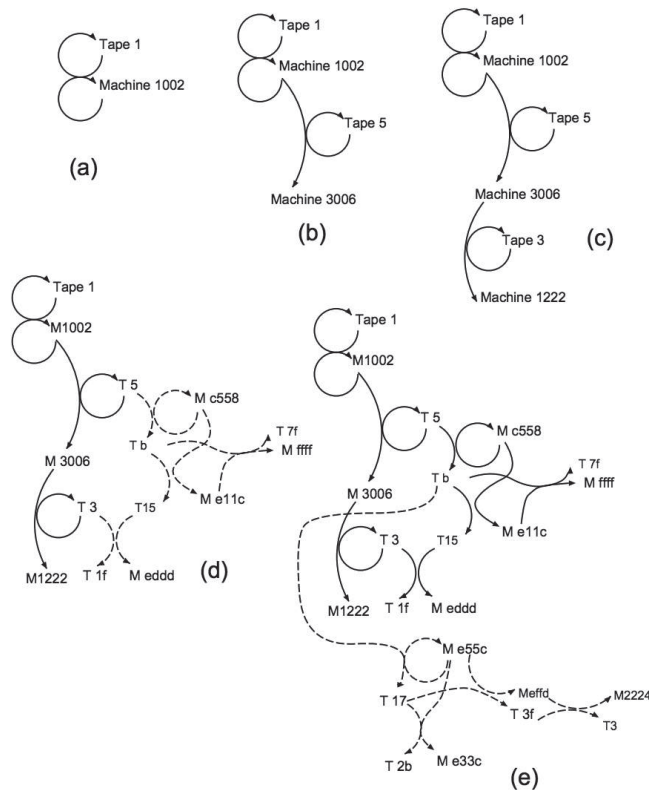


圖 3-18 最低限度的自我複製

如圖 3-18 所示，當外部突變（被動突變）小，會反覆執行 (a) (b) (c)，但是基本上 Tape1 與 Machine1002 是成對的，M1002 複製 Tape1，Tape1 將 M1002 編碼。可是，當外部雜訊增加超過一定程度，網路將會變複雜 (d)(e)。

因為略微增加外部雜訊時，在簡單的小型自我複製網路中，會根據其他機器製作的磁帶，建立出別的機器。這種突變會讓主機器的數量減少，而且按照主現象產生的機器數量也會減少，還會反覆發生增加主機器個數的振動。

此外，當外部的雜訊稍微增加，自體無法單獨生存時，就會寄生在其他個體，自我複製出網路。有趣的是，這種寄生物種的被動突變率容易變高。被動突變代表機器複製磁帶失敗。換句話說，因為失敗，才開始能複製整體。

即使整體穩定之後，去除外在雜訊，只有主動突變，亦即僅以決定論型替換，也能維持整體的自我複製（相同的磁帶與機器群）。換句話說，這是指程式模仿以隨機雜訊產生的磁帶替換及機器複製，並進化成演化演算法式的錯誤。這種完全用程式替換來維持的網路，稱作「核心網路」。

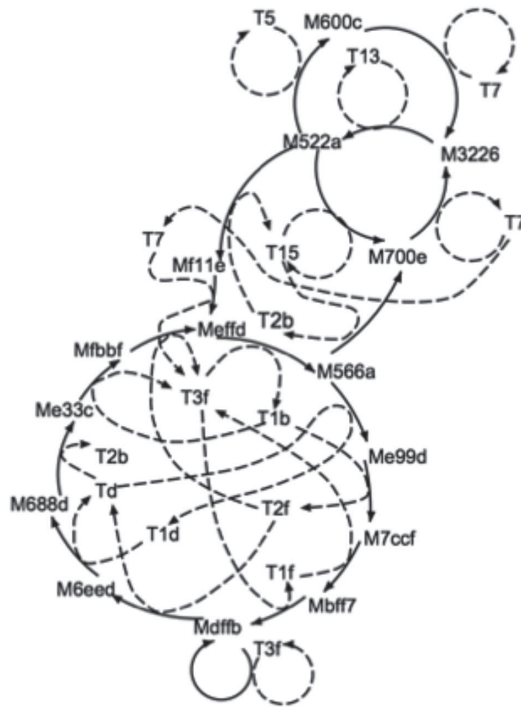


圖 3-19 核心網路的範例

引用自文獻<sup>[24]</sup>



# 第四章

## 生命群體

當個體產生，並能自我複製時，該集團會形成群體，不過此時的群體會像是一個個體來動作。在 ALife 領域，自主移動的個體之間，彼此協調並產生動作時，會當作「突現」機制來探究群體本身的行為類型或結構。

本章將執行模擬鳥群的 Boids 模型，探討「群體」的本質，檢視集體的突現現象在網際網路中產生群體智慧，進而衍生整個社會系統的溝通問題流程。

### 4.1 突現現象

歷史學家 David Christian 在他的隨筆中寫到，「突現 (emergence)」是最美的科學概念。

突現現象是自主移動的個體之間，彼此合作的結果、產生自我組織性的模式或結構的現象。在自然界的生命中，存在著許多突現現象。例如形成生物形狀或模樣的自我組織化，或鳥群、昆蟲群體的個體之間局部作用，呈現出高層次群體行為的現象等。

在 ALife 的研究歷史中，也有許多產生突現現象的研究。

前面幾章介紹過的 von Neumann 細胞自動機，Alan Turing 的反應擴散系統模型，都是把這種微小相互作用的結果，以模式或結構呈現突現現象的範例。其他還有模仿蟻群找出食物到蟻穴最短路徑的計算方法（蟻群演算法 Ant Colony Optimization）、模仿魚群或鳥群的其中一個個體，發現較好的路徑時，不論在哪裡，群體中的其他個體都會立刻學習的計算方法（粒子群最佳化 Particle Swarm Optimization）等應用。

在 ALife 的突現現象模型中，Craig Reynolds 於 1986 年提出的「Boids」模型堪稱是劃時代的研究結果<sup>[25]</sup>。Boids 是從 bird-oid，亦即「類似鳥群」衍生出來的單字。

Boids 模型是利用三個簡單的規則，建立群體的演算法。Reynolds 的想法讓當時使用於電腦動畫上的電腦圖像（CG）技術，出現大幅的進步。過去，要表現大部分物體的動作時，必須由創意師逐一設定各個物體，是非常困難的工作。

Reynolds 的 Boids 模型可以有效率地自動控制形成或移動群體。Boids 模型出現的隔年 1987 年，這個模型運用在 CG 短篇動畫《Stanley and Stella in: Breaking the Ice》，受到各界矚目。之後，由 Tim Burton 執導的好萊塢電影《蝙蝠俠大顯神威》，也在蝙蝠群及企鵝群的場景使用了這個模型。

Reynolds 提出 Boids 模型之後，為了產生更複雜、更逼真的生物動作，在 Boids 模型加入了各式各樣的改良。例如，提出了建立群體時，結合恐懼情緒影響的模型，或像真實鳥群般，察覺外在危險的鳥兒成為首領，導入帶領鳥群避開危險能力的模型。

整合這種群體的電腦動畫技術，最知名的軟體就是「MASSIVE」。MASSIVE 是由學過 ALife 的天才工程師 Stephen Regelous 製作，並且廣泛運用在《魔戒首部曲：魔戒現身》，半獸人族群等大規模戰鬥場景的電影或電玩作品中。

## 4.2 Boids 模型

接下來，讓我們立刻執行程式碼 chap04/boids.py，瞭解什麼是 Boids 模型。

### 範例程式的執行方法

範例程式位於 chap04 目錄中，請切換至儲存該檔案的目錄再執行。

```
$ cd chap04
$ python boids.py
```

在 Boids 模型中，個體有各自的位置與方向，按照「分離」、「對齊」、「結合」等三種規則來移動。

- 「分離」（SEPARATION）  
移開以避免與周圍的個體碰撞。
- 「對齊」（ALIGNMENT）  
朝著與周圍個體相同方向飛行。

- 「結合」(COHESION)  
統一朝周圍個體的中心方向移動。

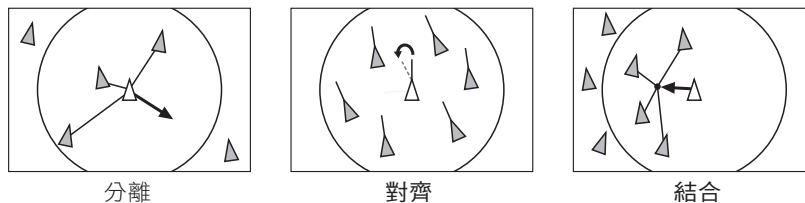


圖 4-1 Boids 模型的規則

這三種規則分別含有決定力量大小、與互相作用對象的距離、可視角度的參數。改變這些參數，能產生各種群體模式。

試著執行程式，如果出現如圖 4-2 的影像並且開始移動，就代表成功了。剛開始隨意飛翔的個體到形成群體為止，可能需要花一點時間。此外，請根據你的電腦效能，調整變數 N 設定的個體數量。



圖 4-2 Boids 模型

參數：

```
# 力量強弱
COHESION_FORCE = 0.008
SEPARATION_FORCE = 0.4
ALIGNMENT_FORCE = 0.06
# 力量作用的距離
COHESION_DISTANCE = 0.5
SEPARATION_DISTANCE = 0.05
ALIGNMENT_DISTANCE = 0.1
# 力量作用的角度
COHESION_ANGLE = np.pi / 2
SEPARATION_ANGLE = np.pi / 2
ALIGNMENT_ANGLE = np.pi / 3
```

以三角形表示的各個個體，含有位置與速度等資料。

位置儲存在陣列  $x$ ，速度儲存在陣列  $v$ 。模擬的是三維空間，所以在陣列  $x$  會準備個體數  $\times 3$  (長、寬、高) 的二維陣列。陣列  $v$  是三維空間中的各個個體速度。在  $x$  位置的 Boids，從結合 (COHESION)、分離 (SEPARATION)、對齊 (ALIGNMENT) 等三種力量算出速度  $v$ ，並且儲存在該處。為了簡化位置與速度的預設值，而讓  $x$ 、 $y$ 、 $z$  成分帶有隨機值。

```
# 位置與速度
x = np.random.rand(N, 3) * 2 - 1
v = (np.random.rand(N, 3) * 2 - 1) * MIN_VEL
```

截至這裡為止，Boids 模型本體的執行過程就結束了，實際進行模擬時，邊界條件也會成為重要因素。

換句話說，只有上述條件，空間會無限延伸，個體幾乎都會分散飛行。在範例程式中，為了避免這種問題，針對原點到半徑 1 的球形內出現的個體，套用與距離成等比的中心力。

因此，可以先將個體群限制在模擬空間的原點附近，有助於產生群體。

```
dist_center = np.linalg.norm(x_this) # 與原點的距離
dv_boundary[i] = - BOUNDARY_FORCE * x_this * (dist_center - 1) / dist_center
if (dist_center > 1) else 0
```

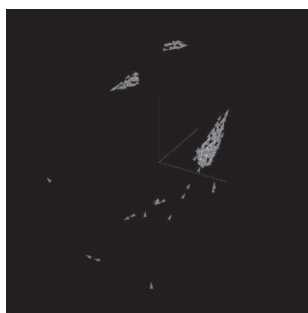
這個部分是用來計算中心力。中心力與上述三種力一樣，最後會加上速度。

週期邊界條件（來自右邊的個體出現在左邊）很常用，而且執行方式也很簡單。

Boids 模型建立的群體模式看起來複雜且多元，但是有人嘗試將其行為歸納成 sawrm、torus、dynamic parallel group、highly parallel group 等四大類<sup>[26]</sup>。

sawrm 是建立一個固定的群體，群體本身幾乎不動，內部的個體會朝各種方向移動。torus 是個體聚集成環狀，形成會旋轉的群體。dynamic parallel group 和 swarm 一樣，會聚集在一起，但是群體本身會動態移動。highly parallel group 是群體的個體統一朝著相同方向，直線前進的模式。

讓我們一起來檢視改變九種參數，群體的模式會產生何種變化。



```
# 力量的強弱
COHESION_FORCE = 0.2
SEPARATION_FORCE = 0.10
ALIGNMENT_FORCE = 0.03
# 力量作用的距離
COHESION_DISTANCE = 0.5
SEPARATION_DISTANCE = 0.08
ALIGNMENT_DISTANCE = 0.1
# 力量作用的角度
COHESION_ANGLE = np.pi / 2
SEPARATION_ANGLE = np.pi / 2
ALIGNMENT_ANGLE = np.pi / 3
```