

# 臺灣二〇〇七年國際科學展覽會

科 別：地球與太空科學

作 品 名 稱：宇宙演化的黑手

得 獎 獎 項：第一名

加拿大正選代表：加拿大 2007 年科學展覽會

學 校 / 作 者：國立臺灣師範大學附屬高級中學 鍾岳霖

## 自傳



姓名：鍾岳霖

生日：1989/3/16

專長：數學、電腦、人際 EQ、合群、領導與組織能力

如果說我在復興學會如何走路，我在附中就學會如何昂然邁進；如果說復興教我如何在翻滾紅塵中孤然挺立，附中教我的就是如何在時代洪流中和群眾和而不同。

在復興畢業之後，依照學測成績而進入了附中就讀，在自由開放的附中，我雖不習慣但是很快的就在這裡找到自己的天地。延續著國中的能力，在附中也能利用自己原有的優勢繼續奮鬥。另一方面，附中也開啓了我的性中開朗樂觀的一面，讓我有了一個領導者的風範，也有做被領導的風度。

國中的短短三年讓我意識到自己在樹禮方面的潛能以及相當不錯的理解力；而昇上高中科教班後，經歷培養、競爭、鍛鍊琢磨，我更感謝附中闢了一條新路給我，讓我跳脫文科的框框，進入一個無限寬廣而更適合自己的世界。

# 目 錄

摘要	1
一、前言	2
(一)、研究動機	2
(二)、研究目的	2
二、實驗設備與器材	2
三、研究過程和方法	2
(一)、研究原理	2
(二)、研究方法	3
四、研究結果	6
五、討論	18
六、結論	19
七、參考資料及其它	19
附件	20

# Backstage Manipulator for Cosmic Evolution

## 宇宙演化的黑手

### 英文摘要(Abstract)

We study the effect of dark energy on the evolution of cosmic structure in a scenario where the dark energy is treated as free particles and thus can be localized. By theoretical derivation and numerical simulations, we found that:

1. The dark energy particles gain kinetic energy from a moving dark matter particle through gravitational interaction. Due to energy conservation, the dark matter particle will slow down with time

$$E_k(t) = E_{k0} - 9 \times 10^{-5} \left[ 1 + 3w \rho_{DE} \right]^{1.92} t$$

where  $E_k(t)$  is the kinetic energy of the dark matter particle,  $E_{k0}$  is its initial kinetic energy,  $w$  is the coefficient of equation of state for dark energy,  $\rho_{DE}$  is the mean energy density of dark energy, and  $t$  is the time.

2. The formation history and structure of galaxy clusters are different in the presence of localized dark energy. The more the localized dark energy, the earlier the formation of the cluster core. In addition, the kinetic energy  $E_k(R)$  as a function of  $R$  will be different if the  $\rho_{DE}$  is different. Thus we can compare the observed  $E_k(R)$  of clusters with our results to deduce the  $\rho_{DE}$  in our universe.

The results here can be applied to the observations in the near future.

### 中文摘要

我們探討宇宙結構演化受到可局部叢集之黑暗能量粒子的影響。藉由理論推導及電腦模擬，我們發現：

- 一、黑暗能量粒子會透過重力交互作用而從運動中的黑暗物質粒子獲得力學能。因力學能守恒，黑暗物質粒子的速率會減慢，滿足

$$E_k(t) = E_{k0} - 9 \times 10^{-5} \left[ 1 + 3w \rho_{DE} \right]^{1.92} t$$

其中  $E_k(t)$  為黑暗物質粒子的動能， $E_{k0}$  為其初始動能， $w$  為狀態方程式係數， $\rho_{DE}$  為黑暗能量的平均密度， $t$  為時間。

- 二、星系團的形成過程及結構，會因可局部叢集之黑暗能量的存在而改變。黑暗能量越多時，星系團的核心會越早形成。而且動能  $E_k(R)$  隨著至星系中心距離  $R$  的變化，會因  $\rho_{DE}$  的不同而不同，因此可以將量測到的  $E_k(R)$  和這裡的結果比對，推導出宇宙中的  $\rho_{DE}$ 。

這些研究成果將可直接應用在未來的觀測結果上。

### 一、前言：

#### (一)、研究動機

在展望系列演講以及科學人雜誌中得知，近年來在宇宙學領域中最熱門的話題之一，是占宇宙總能量約七成的**黑暗能量**和兩成半的**黑暗物質**，這些神秘的物質將是二十一世紀物理學家們的大挑戰。尤其是黑暗能量，更是一反課本中所學的傳統牛頓思想而提供「萬有斥力」，著實引人瑕思。這些新知引起我強烈的興趣，想要自己來看看它們之間的「萬有引力」和「萬有斥力」，到底會讓整個物理空間變成什麼樣子。

#### (二)、研究目的

利用課本所學，藉由理論的推導配合電腦模擬的手段，來探討黑暗物質和黑暗能量的作用力及相互關係，並進而了解一個運動中的質點所受這些黑暗力量的影響，以及宇宙中星系形成過程中所受這些黑暗力量的影響。一般的黑暗能量模型，都假設它在宇宙中是均勻分佈的，而此處我們假設它可以**自由移動**而造成**不均勻分佈**，看看這對宇宙結構演化有何影響。了解黑暗能量及黑暗物質的性質，是目前世界上宇宙學領域中最熱門的主題之一，所以我們的研究成果，將可以在短期的未來直接應用在這些期待中的觀測結果上，以揭開這些黑暗力量的神祕面紗。

### 二、實驗設備與器材

(一)、軟體：C 語言程式、Matlab 分析軟體

(二)、硬體：Sun Blade 2000 電腦工作站、Pentium 個人電腦

### 三、研究過程和方法

#### (一)、研究原理：

一般宇宙學中的黑暗能量模型，都假設它在宇宙中是均勻分佈的，但並沒有任何理論「規定」它一定要均勻分佈。因此我們在本研究中將允許黑暗能量在宇宙中**自由移動**，以使其分佈變成不均勻，並藉此探討此假設對宇宙結構演化所可能造成的影響，而宇宙中最大的結構莫過於星系團。

由於本研究將允許黑暗能量在宇宙中自由移動，所以我們無法使用一般學術界被廣泛使用的模擬軟體 Hydra 或 Gadget 來進行宇宙結構如星系團等的形成模擬，因為這些軟體都假設宇宙中的黑暗物質是均勻分佈的。因此我們必須自行撰寫程式來進行模擬。

本研究的主要原理是利用牛頓的反平方引力定理來探討黑暗物質和黑暗能量之間的萬有引力及萬有斥力。傳統的反平方定理是

$$F = \frac{GMm}{r^2} \quad (\text{公式 1})$$

其中  $F$  為力， $G$  為重力常數， $M$  及  $m$  為黑暗物質或黑暗能量的質量， $r$  為兩質點間的距離。但根據廣義相對論中的愛因斯坦方程式，式子中也必須考慮壓力，也就是  $m = \rho V$  中的  $\rho$  必須用  $(\rho + 3p)$  來取代，也就是

$$F = \frac{GMV(\rho + 3p)}{r^2} \quad (\text{公式 2})$$

對黑暗物質及一般物質而言  $p = 0$ ，所以得到（公式 1）的形式。但對黑暗能量而言

$$p = w\rho \quad (\text{公式 3})$$

其中  $w < -1/3$ 。（公式 3）稱為「狀態方程式」，而  $w$  稱為狀態方程式係數。最常見的黑暗能量為「宇宙常數」，由愛因斯坦於二十世紀初期提出，其  $w = -1$ ，所以（公式 2）變為

$$F = \frac{-2GMm}{r^2} \quad (\text{公式 4})$$

因此萬有引力變為萬有斥力！所以本研究的關鍵，在於利用黑暗能量的負壓力，以致使原本的萬有引力變為萬有斥力。

黑暗能量是組成宇宙最主要的部分（約占七成），也是科學上最大的謎。由於  $w < -1/3$ ，即  $p < -\rho/3$ ，所以  $\rho + 3p < 0$ ，根據（公式 2）它一定會提供一種斥力，因此可使宇宙得以加速膨脹。而一般物質的重力（引力）則會使膨脹減速。

## (二)、研究方法：

我們先探討一個黑暗物質粒子和眾多黑暗能量粒子的交互作用，掌握其初步的本質後，再進行星系團結構形成的模擬。研究過程乃依以下順序，所使用的程式語言為 C 語言。

1、在眾多黑暗能量粒子間放入一個黑暗物質粒子，觀察它的運動：

- (1) 在模擬的箱子中放入一個黑暗物質的粒子，並給它初速度。而黑暗能量粒子在初始時則靜止放置在箱子中的均勻格點上。
- (2) 依照（公式 1）及（公式 4），依序求出黑暗物質彼此間、黑暗能量和黑暗物質間、以及黑暗能量彼此間的交互作用力。
- (3) 求出每一個粒子所受的合力，利用

$$\mathbf{F} = m\mathbf{a} \quad (\text{公式 5})$$

的公式求出此粒子的加速度  $\mathbf{a}$ ，再利用

$$\mathbf{S} = \mathbf{v}dt + \frac{1}{2}\mathbf{a}dt^2 \quad (\text{公式 6})$$

求出此粒子在  $dt$  後的位移  $\mathbf{S}$ ，最後利用

$$\mathbf{v}(t + dt) = \mathbf{v}(t)dt + \frac{1}{2}\mathbf{a}dt^2 \quad (\text{公式 7})$$

出此粒子在  $dt$  後的速度  $\mathbf{v}$ 。

- (4) 每經過一個時間步  $dt$ ，輸出每個粒子的座標  $\mathbf{x}$  和速度  $\mathbf{v}$ 。
- (5) 利用迴圈來反覆執步驟(2)至(4)。
- (6) 經由 Matlab 處理所輸出之各個時間的資訊，以將粒子的運動進行分析，並做成動畫。

2、由純理論的角度切入，試圖簡化力學的計算及程式的演算。同樣考慮在眾多黑暗能量粒子中放入一個黑暗物質的粒子，觀察它的運動：

- (1) 在黑暗能量互相排斥的部分上，利用以下的推導結果簡化力學的計算。

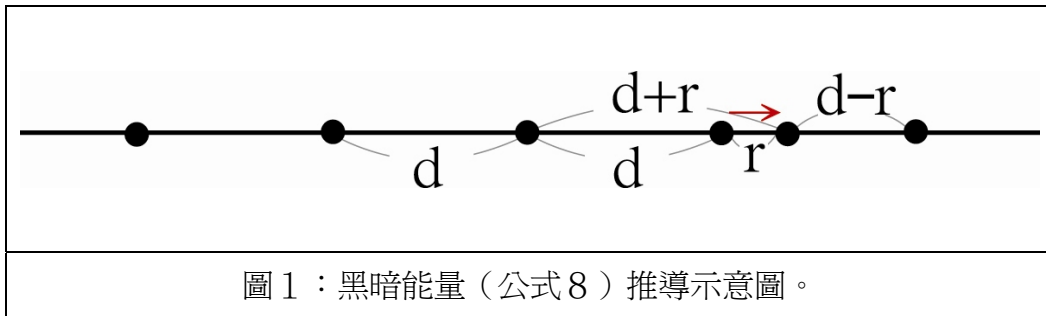


圖 1：黑暗能量（公式 8）推導示意圖。

利用（公式 4）及（公式 5），得知  $a = \frac{-2GM}{r^2}$ 。當某一個黑暗能量粒子相對於其原本的格點有一位移  $r$ （如上圖 1），則此質點所受到其左邊的各個粒子之加速度和為（方向為向右）

$$\frac{2GM}{(r+d)^2} + \frac{2GM}{(r+2d)^2} + \dots + \frac{2GM}{(r+nd)^2} + \dots = 2GM \sum_{n=1}^{\infty} \frac{1}{n^2 d^2 (\frac{r}{nd} + 1)^2}$$

同理其所受到其右邊的各個質點之加速度和為（方向為向左）

$$\frac{2GM}{(r-d)^2} + \frac{2GM}{(r-2d)^2} + \dots + \frac{2GM}{(r-nd)^2} + \dots = 2GM \sum_{n=1}^{\infty} \frac{1}{n^2 d^2 (\frac{r}{nd} - 1)^2}$$

兩式相減之後得加速度為

$$a \cong \frac{8rGM}{d^3} \sum_{n=1}^{\infty} \frac{1}{n^3} = \frac{8rGM}{d^3} * \text{Zeta}(3) \cong \frac{rGM}{d^3} * 8 * 1.20206 \quad (\text{公式 8})$$

利用以上之（公式 8），將可大幅減少程式中的計算量。

- (2) 利用（公式 4）計算出黑暗物質粒子的淨加速度，利用（公式 8）計算出黑暗能量粒子間的作用力，然後利用（公式 6）及（公式 7）來算出  $dt$  後各粒子的位移及速度。
- (3) 將黑暗物質及黑暗能量等粒子的座標、速度和時間輸出至檔案。
- (4) 利用迴圈來反覆執步驟(2)及(3)。

有了以上的結果，將可大量減少計算量，以大幅加速電腦模擬的效率。

3、利用 2 的方法，在眾多黑暗能量粒子中放入一個黑暗物質粒子，進行以下探討：

- (1) 改變黑暗物質粒子的初速度  $v_0$ ，觀察其運動行為的改變。
- (2) 固定黑暗物質粒子的初速度  $v_0$ ，但改變黑暗能量粒子的密度，觀察黑暗物質粒子運動行為的改變。
- (3) 固定黑暗物質粒子的初速度  $v_0$ ，固定黑暗能量粒子的密度，但改變黑暗能量的狀態方程式係數  $w$ ，觀察黑暗物質粒子運動行為的改變。
- (4) 利用 Matlab 進行分析。

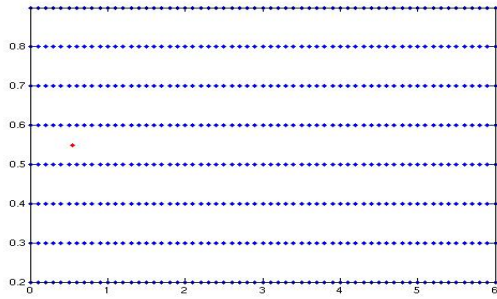
4、放置大量的黑暗物質粒子，模擬宇宙中星系團的形成。此部分分為兩部分，一個為具有黑暗能量，另一個為沒有黑暗能量：

- (1) 有黑暗能量：
  - A. 將黑暗能量皆放置於箱子中的格點上，其方法是利用三個迴圈來設定其黑暗能量的座標，並且利用畢氏定理來將 1000 顆黑暗物質放至於一中心座標為(0.5,0.5,0.5)，半徑為 0.2 的圓球範圍中。
  - B. 利用（公式 1）算出黑暗物質間的作用力，利用（公式 4）算出黑暗物質和黑暗能量間的作用力，利用（公式 8）算出黑暗能量間的作用力。
  - C. 求出各質點所受的和力，再利用（公式 6）及（公式 7）來算出它們在  $dt$  後的位移及速度。
  - D. 將黑暗物質及能量的位移、速度和時間輸出至檔案。
  - E. 利用迴圈來反覆執步驟 B 至 D。
  - F. 利用 Matlab 分析結果。
- (2) 沒有黑暗能量：
  - A. 比照 1 中的作法，但移除所有的黑暗能量，執步驟 A 至 E。
  - B. 利用 Matlab 分析結果。
- (3) 利用 Matlab 比較分析(1)及(2)之結果的異同。

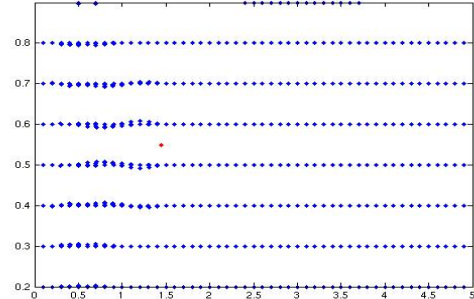


## 四、研究結果

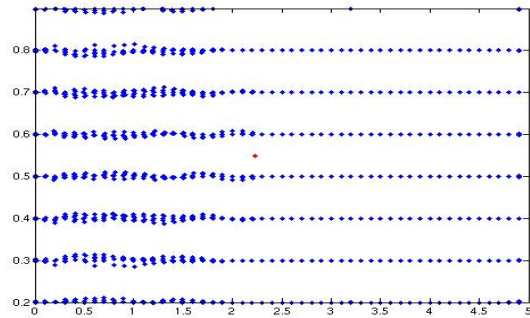
(一)、在眾多黑暗能量中放入一個黑暗物質的粒子，觀察所有粒子的運動：



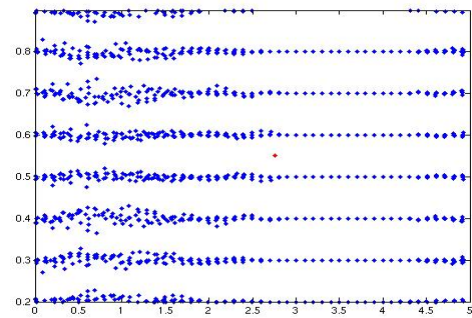
第 0 個時間步



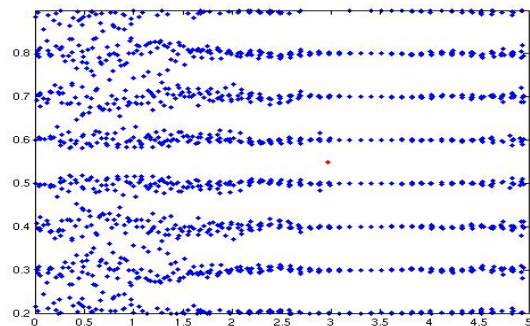
第 20 個時間步



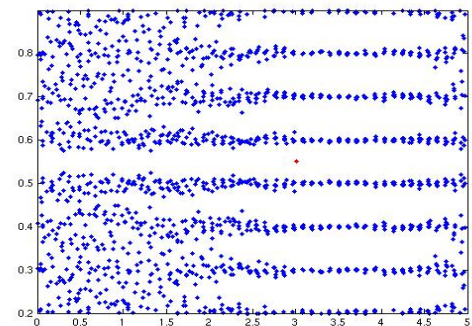
第 40 個時間步



第 60 個時間步



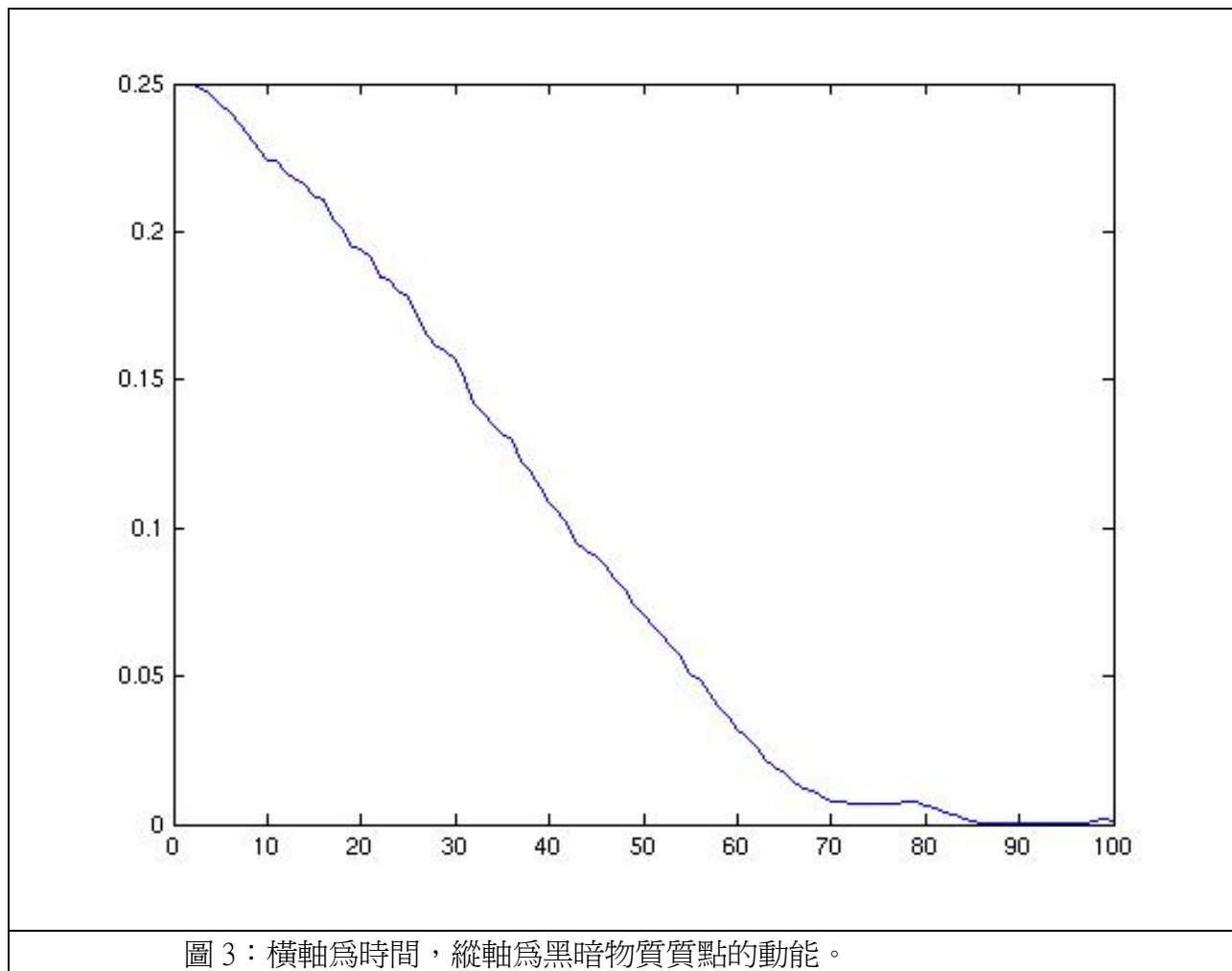
第 80 個時間步



第 100 個時間步

圖 2：橫軸及縱橫分別為質點的 x 及 y 座標。

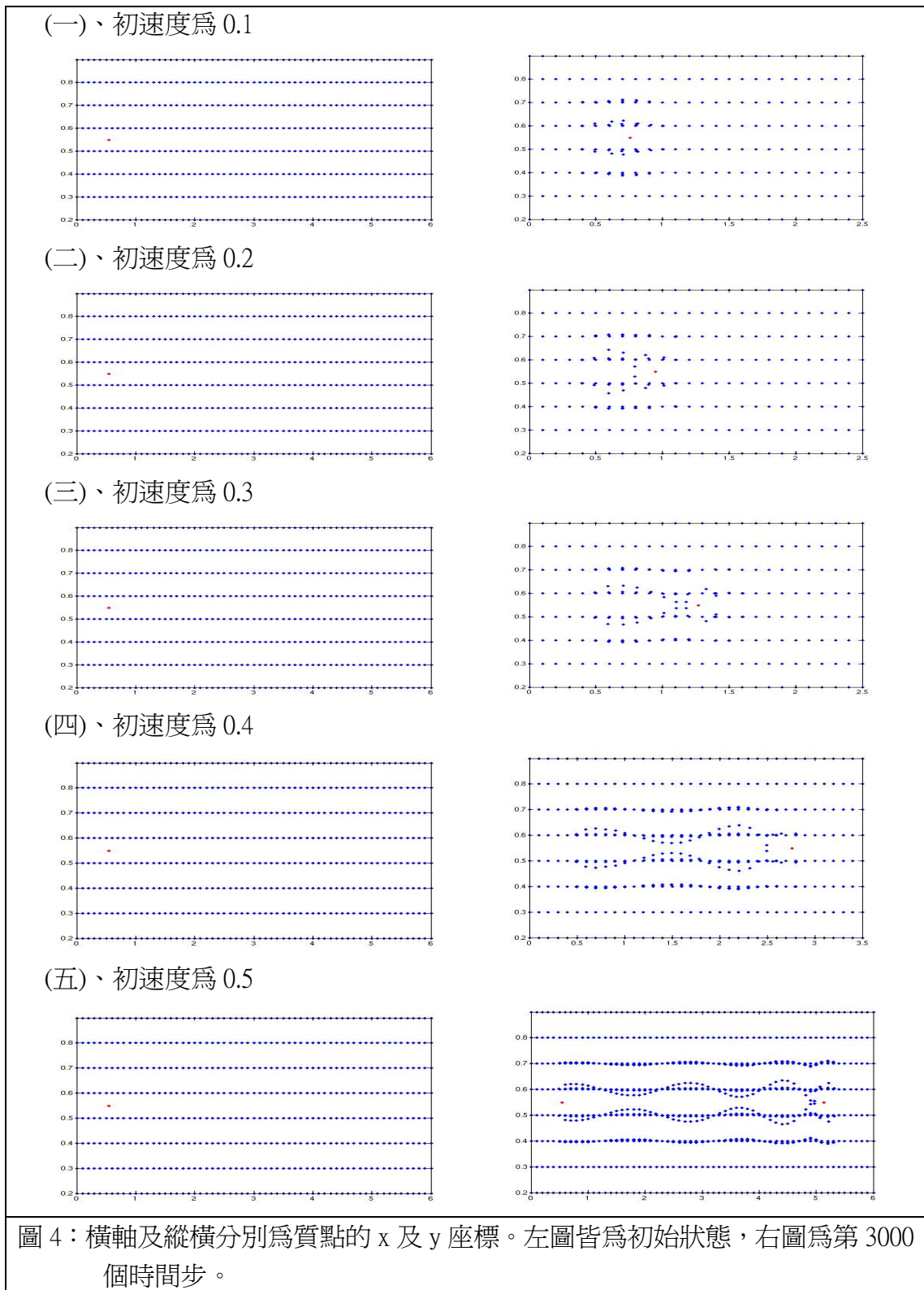
我們可以進一步分析黑暗物質粒子動能隨時間的變化，如圖 3 所示，橫軸為時間，縱軸為黑暗物質質點的動能。



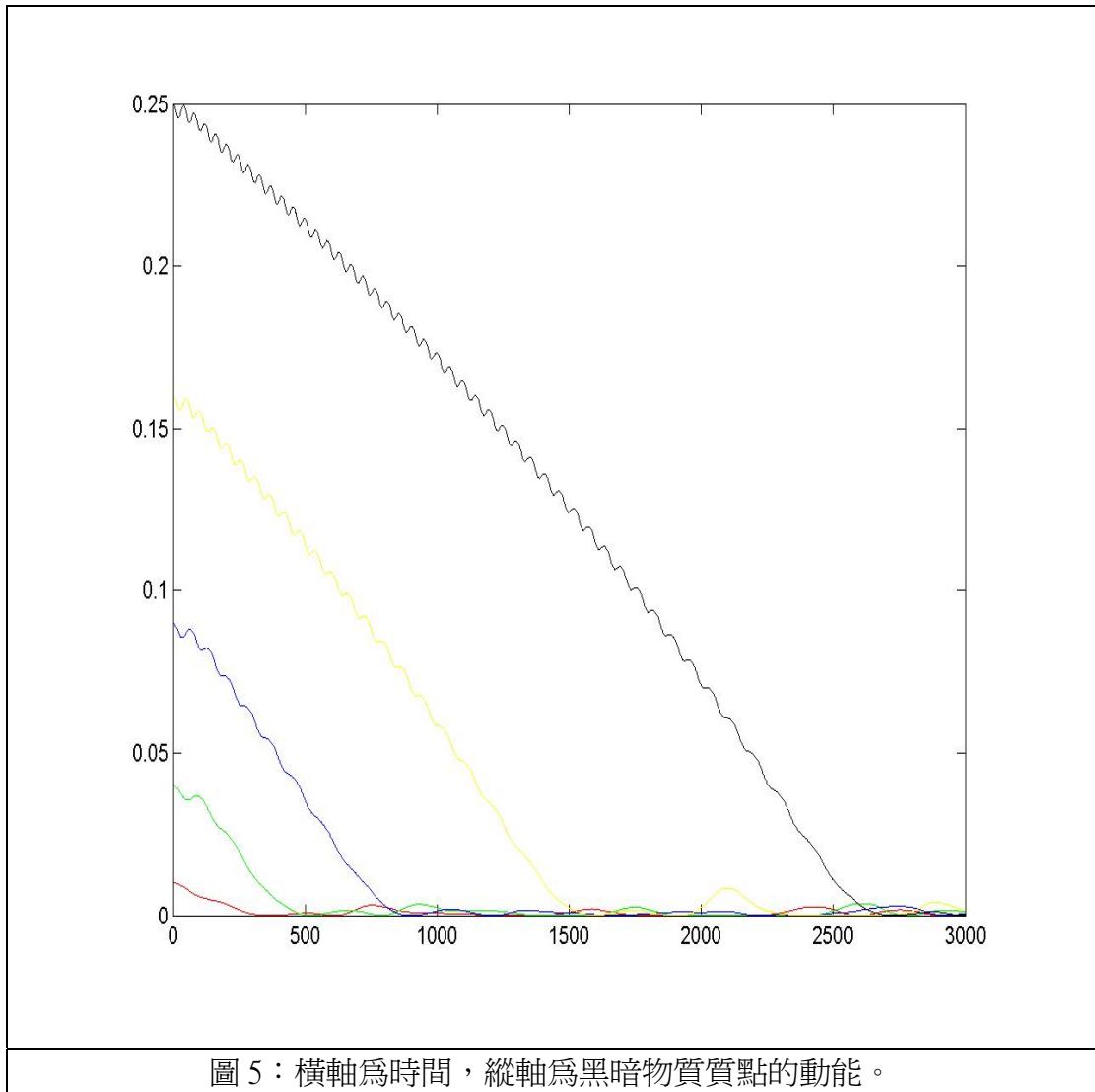
很明顯地，黑暗物質在黑暗能量中前進時，其動能  $E_k$  隨時間  $t$  作線性衰減，而且此衰減率（即圖中之線的斜率），經仔細分析為（當  $\rho_{DE} = 1$  時）

$$\frac{dE_k}{dt} = -9 \times 10^{-5}$$

(二)、利用簡化的力學演算，在眾多黑暗能量中放入一個黑暗物質的粒子，改變初速度，觀察它的運動：



由圖 4 可以很明顯地看出，黑暗物質質點的初速越大時，於同一時間內可移動得越遠，同時可造成黑暗能量質點的上下振盪。我們可以進一步分析黑暗物質質點動能隨時間的變化，如圖 5 所示，橫軸為時間，縱軸為黑暗物質質點的動能。

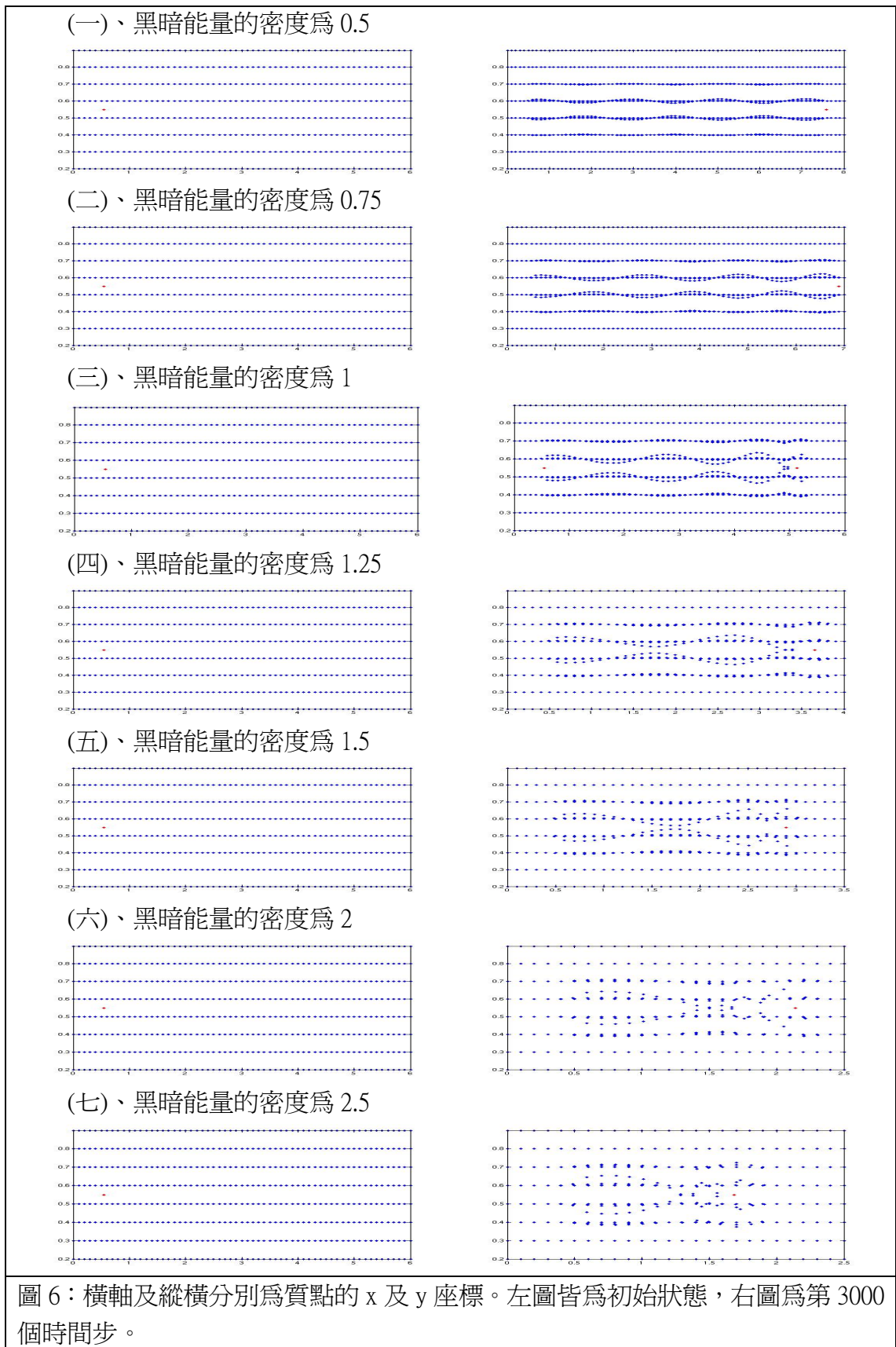


很明顯地和之前的結果相符合，黑暗物質在黑暗能量中前進時，其動能  $E_k$  隨時間  $t$  作線性衰減。而且，此衰減率（即圖中之線的斜率），並不會隨初速的改變而改變，經仔細分析為（當  $\rho_{DE} = 1$  時）

$$\frac{dE_k}{dt} = -9 \times 10^{-5} \quad (\text{公式 9})$$

圖中左下方的線之斜率和其它線不大一樣，而且圖中的每一條線上都有小幅的上下振盪，這都是因為在電腦模擬中，黑暗能量的分布是離散的質點，而不是連續的。其細節請詳見之後的討論。

(三)、利用簡化的力學演算，在眾多黑暗能量中放入一個黑暗物質的粒子，固定初速，但改變黑暗能量的密度  $\rho_{DE}$ ，觀察它的運動：



由圖 6 可以很明顯地看出，黑暗能量的密度  $\rho_{DE}$  越大時，黑暗物質質點於同一時間內可移動的距離就越短，且黑暗能量質點上下振盪的振幅就越大。我們可以進一步分析黑暗物質質點動能隨時間的變化，如圖 7 所示，橫軸為時間，縱軸為黑暗物質質點的動能。

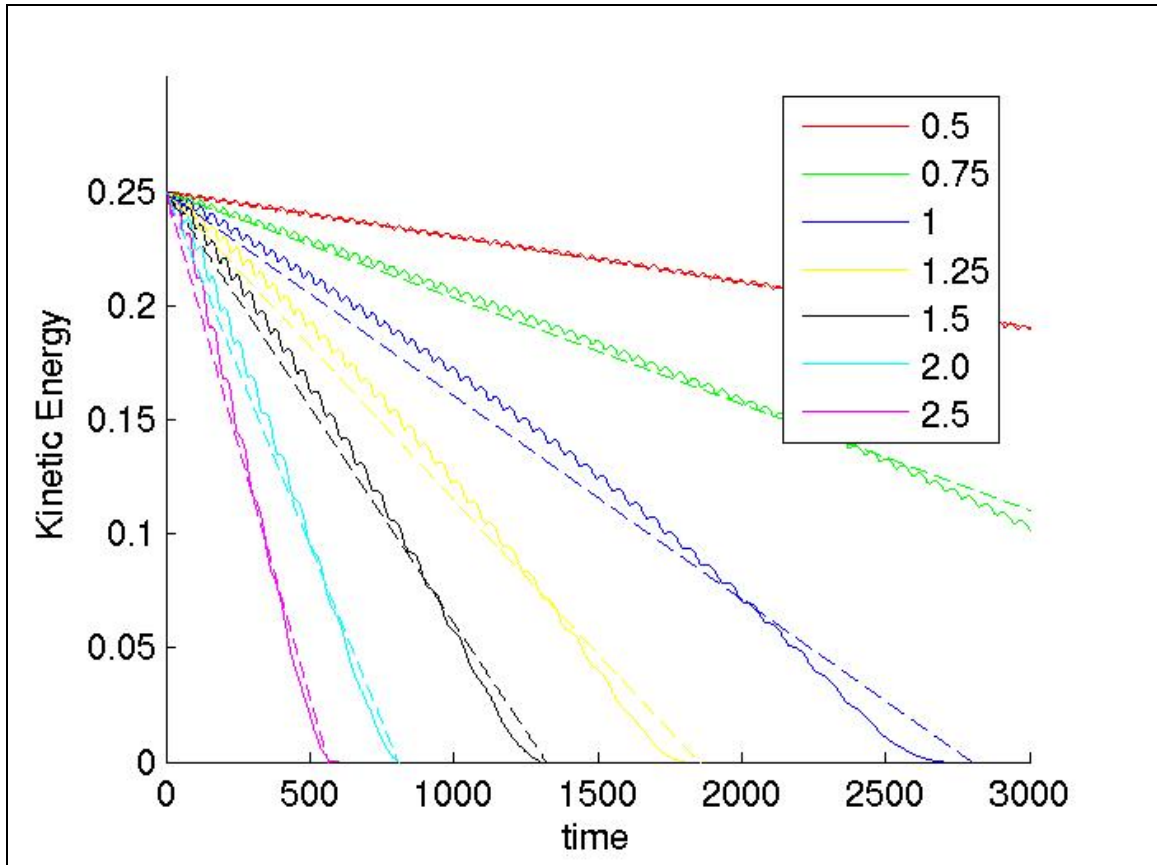
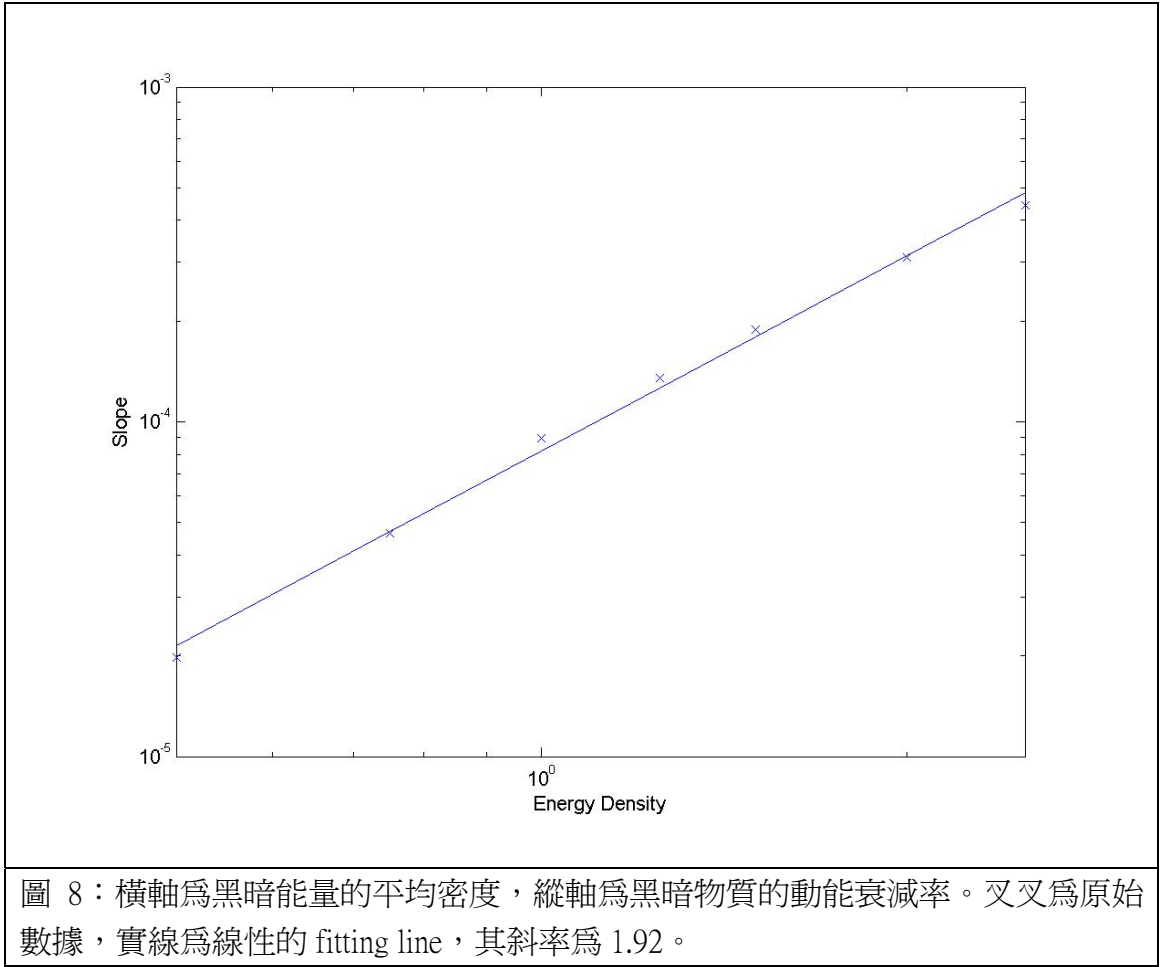


圖 7：橫軸為時間，縱軸為黑暗物質質點的動能。實線為原始數據，虛線為線性的 fitting line。

再一次明顯地，黑暗物質在黑暗能量中前進時，其動能  $E_k$  隨時間  $t$  作線性衰減。然而此衰減率（即圖中之線的斜率之絕對值），會隨黑暗能量密度  $\rho_{DE}$  的增加而增加。若將斜率的絕對值對黑暗能量密度作圖，可得圖 8。圖中的直線斜率為 1.92，故可得

$$\frac{dE_k}{dt} \propto \rho_{DE}^{1.92} \quad (\text{公式 10})$$



(四)、利用簡化的力學演算，在眾多黑暗能量中放入一個黑暗物質的粒子，固定初速，固定黑暗能量的密度  $\rho_{DE}$ ，但改變黑暗能量的狀態方程式係數  $w$ ，觀察它的運動。我們發現

$$\frac{dE_K}{dt} \propto (1 + 3w)^{1.92} \quad (\text{公式 1 1})$$

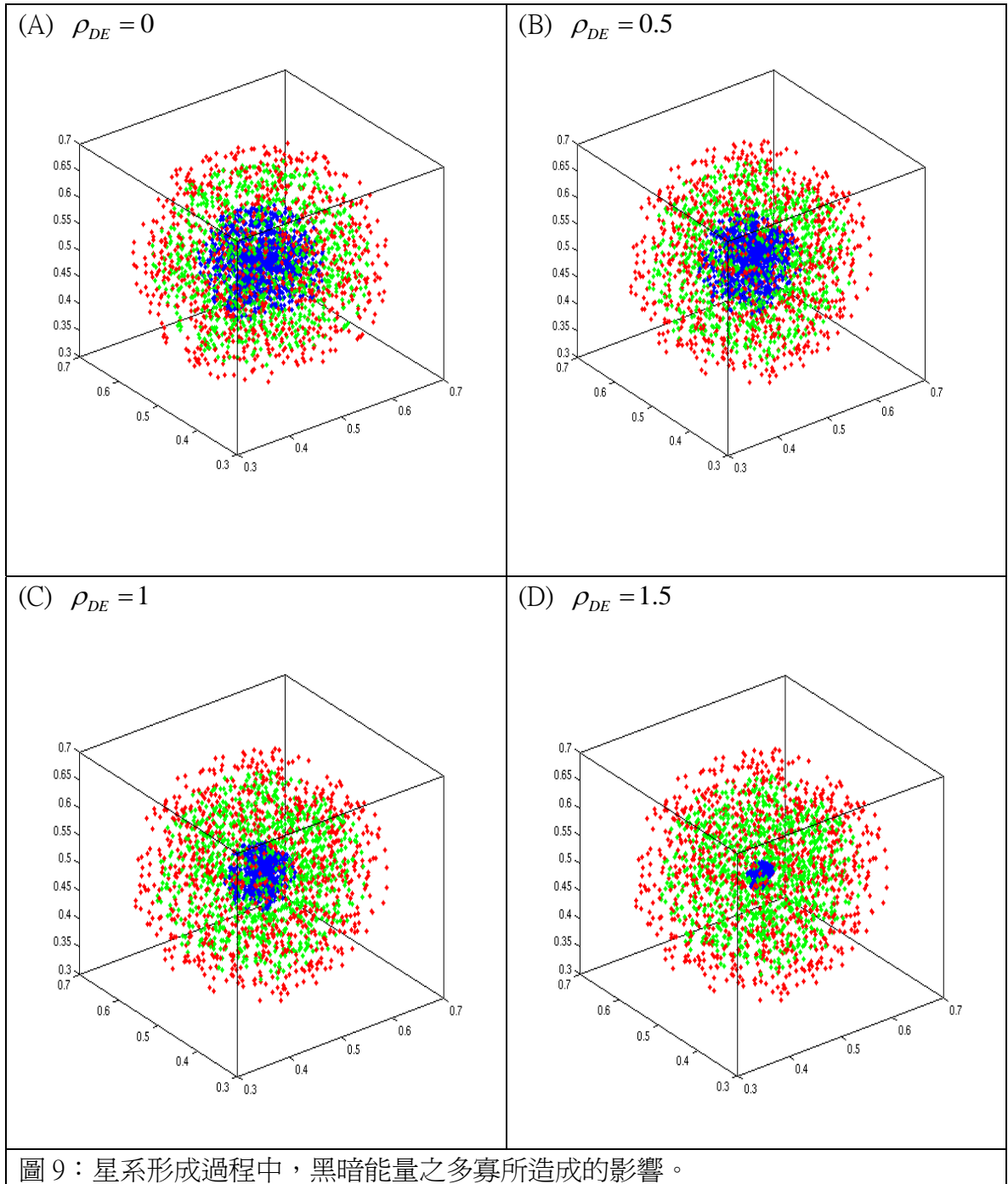
最後我們可結合（公式 9）、（公式 1 0）及（公式 1 1）的結果，推得公式：

$$E_k(t) = E_{k0} - 9 \times 10^{-5} \left[ 1 + 3w \rho_{DE} \right]^{1.92} t \quad (\text{公式 1 2})$$

因此，只要給定黑暗能量的密度  $\rho_{DE}$ ，狀態方程式係數  $w$ ，及黑暗物質質點的初始動能  $E_{k0}$ ，我們便可預測此質點的運動方式。

(五)、放置大量的黑暗物質，模擬宇宙中星系團的形成：

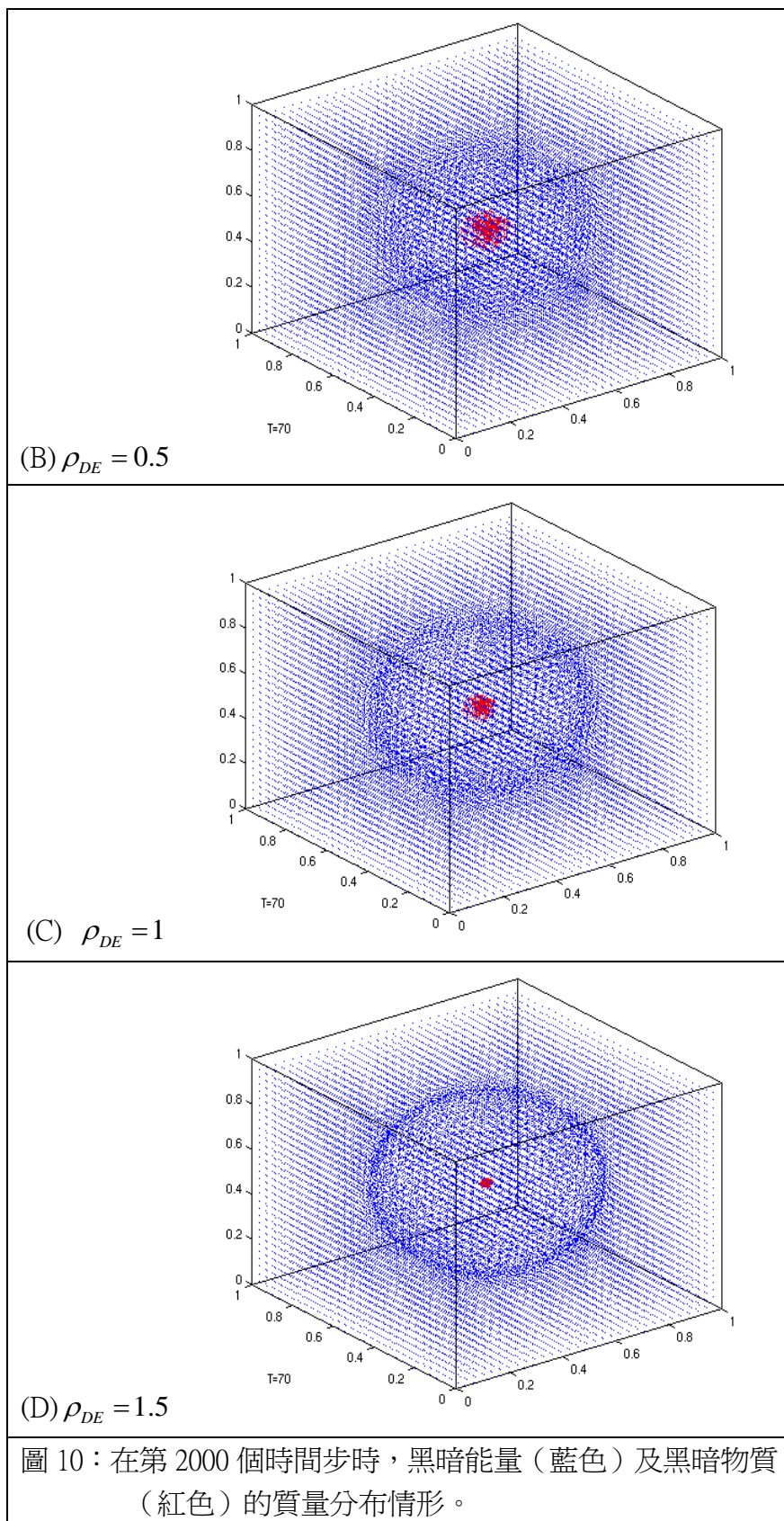
此部分分爲兩種情形，一個爲具有黑暗能量，另一個爲沒有黑暗能量。圖 9 爲此兩種情形的比較結果。圖中的紅色爲起始時黑暗物質的位置，綠色爲經過 1000 個時間步時黑暗物質的位置，藍色爲經過 2000 個時間步時黑暗物質的位置。很顯然地，當黑暗能量密度  $\rho_{DE}$  越大時，會加快星系的質量往中心集中。



在具有黑暗能量的情況下（圖 9 中的 B、C、D），我們也將第 2000 個時間步時，黑暗能量及黑暗物質的質量分布，同時畫在一個圖中，如圖 10 所示。非常有趣



地，黑暗能量由於和黑暗物質推擠，而當後者往中央集中時，前者卻往四周散開。當黑暗能量越多時，此現象越嚴重。



爲了進一步定量地探討黑暗能量之多寡，對加速黑暗物質集中的效應，我們進行了以下的分析。在初始時間時，共有 1000 個黑暗物質質點均勻地分布在半徑爲 0.2 的球形區域內，故我們可考慮，在半徑爲 0.1 的球形區域內之黑暗物質質點個數，在不同的  $\rho_{DE}$  下，會各自如何隨時間增加。結果如圖 11 所示。橫軸爲時間步，縱軸爲在半徑爲 0.1 的球形區域內之黑暗物質質點個數，而紅、綠、黑、紫分別表示圖 9 中之(A)、(B)、(C)、(D)的情況（即  $\rho_{DE}$  分別等於 0、0.5、1、1.5）。我們可以很明顯地看出，在黑暗能量越多時，星系的核心就會越早形成。

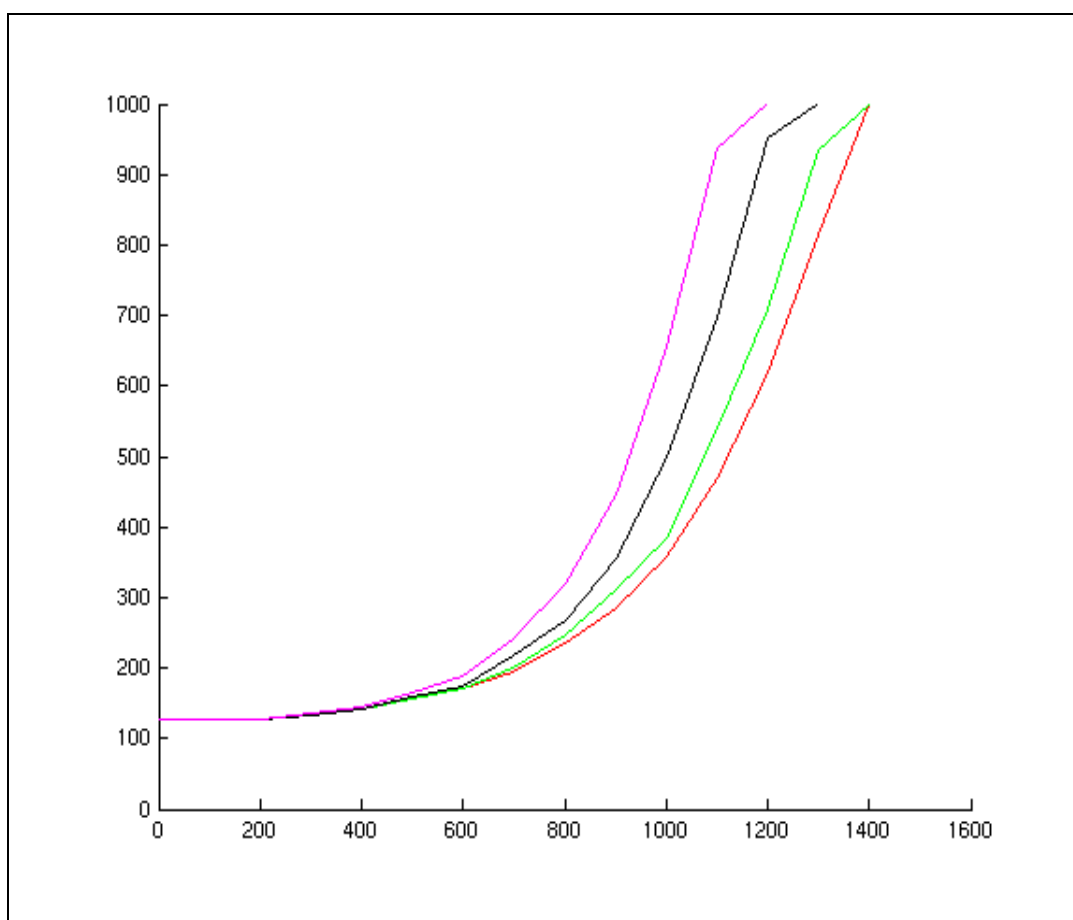


圖 11：橫軸爲時間步，縱軸爲在半徑爲 0.1 的球形區域內之黑暗物質質點個數，而紅、綠、黑、紫分別表示圖 9 中之(A)、(B)、(C)、(D)的情況（即  $\rho_{DE}$  分別等於 0、0.5、1、1.5）。

接著，我們要探討黑暗能量之多寡，對星系結構的影響。我們可以考慮於同一時間上，在圖 9 中之(A)、(B)、(C)、(D)的情況下（即  $\rho_{DE}$  分別等於 0、0.5、1、1.5），星系中距離中心  $R$  的球殼上之平均動能隨  $R$  的變化，如圖 12 所示。其中橫軸為至星系中心的距離  $R$ ，而縱軸為以  $R$  為半徑之球殼上，所有黑暗物質質點的動能  $E_k$ 。非常令人驚訝地，在黑暗能量密度  $\rho_{DE}$  不同時，此  $E_k(R)$  函數具有完全不同的結構，因此在觀測上，我們可以試圖去量測此  $E_k(R)$  函數，然後和這裡的電腦模擬結果比對，我們就可以推出宇宙中黑暗能量的密度  $\rho_{DE}$ 。

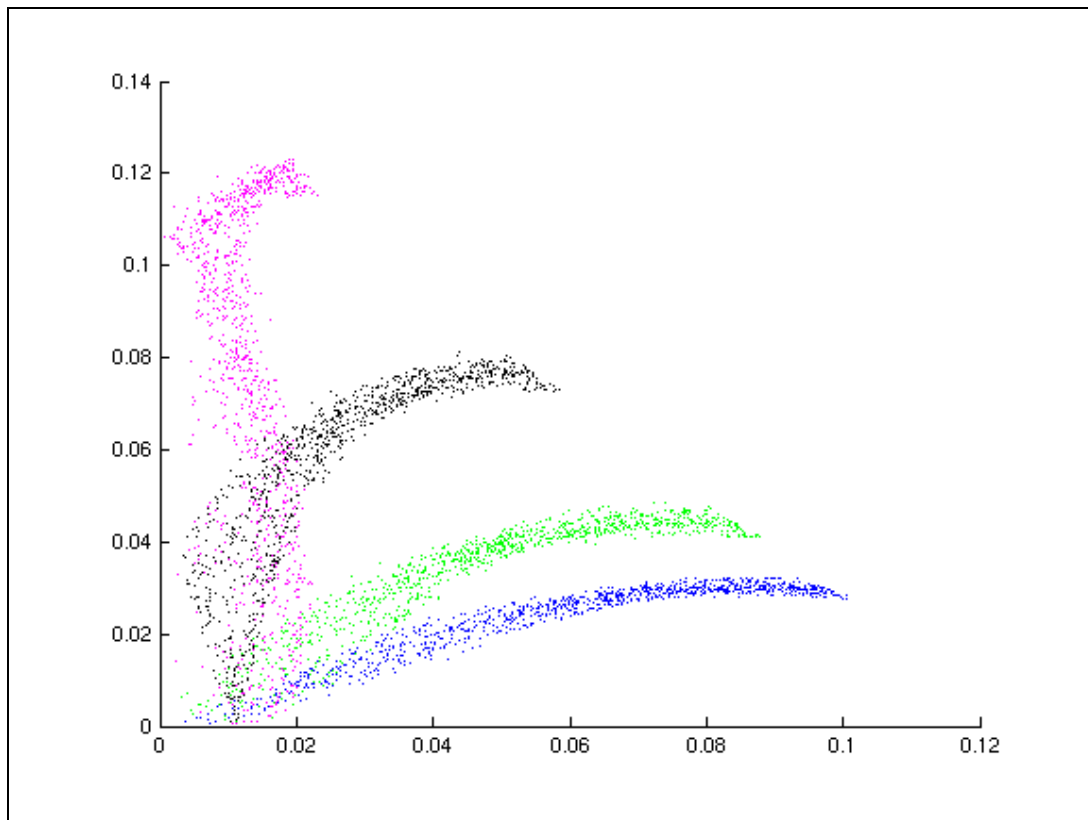


圖 12：橫軸為至星系中心的距離  $R$ ，而縱軸為以  $R$  為半徑之球殼上，所有黑暗物質質點的動能。而紅、綠、黑、紫分別表示圖 9 中之(A)、(B)、(C)、(D) 的情況（即  $\rho_{DE}$  分別等於 0、0.5、1、1.5）。

此外，我們也探討黑暗能量粒子對星系團形成過程中之旋轉的影響。圖 13 的橫軸是到星系團中心的距離，縱軸是以此距為半徑之球殼上所有質點的平均旋轉角速率。我們發現，雖然黑暗能量的存在會影響星系團核心的形成情形，但對其旋轉的性質似乎沒有太大的影響。也就是不管黑暗能量存在與否，星系團的旋轉性質都很相近。

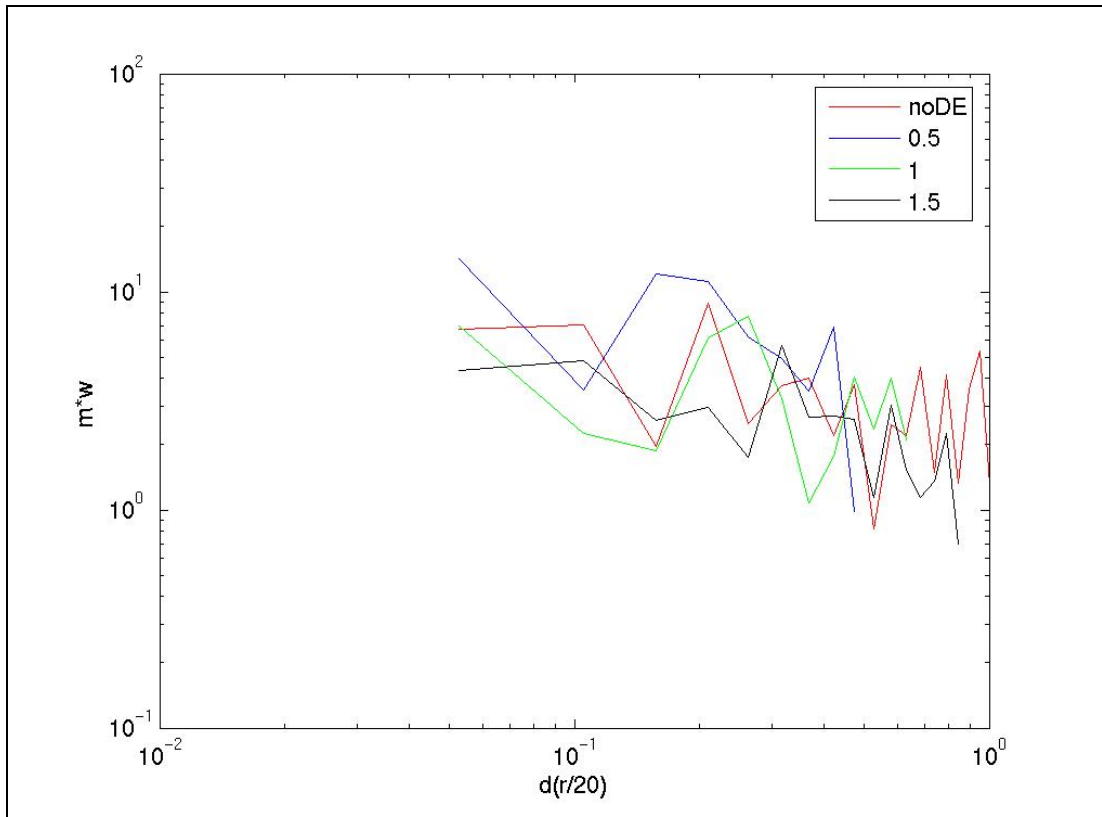


圖 13：橫軸為至星系中心的距離  $R$ ，縱軸是以此距為半徑之球殼上所有質點的平均旋轉角速率。而紅、藍、綠、黑分別表示圖 9 中之(A)、(B)、(C)、(D) 的情況（即  $\rho_{DE}$  分別等於 0、0.5、1、1.5）。

## 五、討論

- (一)、在圖 5 及圖 7 中的線，皆呈小幅度的振盪，這些上下振盪是由於電腦模擬中，黑暗能量的分佈是離散的，而不是呈現連續的分佈，因此，當一顆黑暗物質的粒子在接近一顆黑暗能量粒子時，會因推力而減速，反之，當它通過此黑暗能量粒子後而遠離時，它會因推力而加速。所以，黑暗物質的粒子在行進的過程中，會不停地遭遇到這樣的過程，以致不停地小幅減及加速，而造成圖 5 及圖 7 中的曲線上下振盪。因此，只要不斷的將電腦模擬的解析度提高（也就是將黑暗能量粒子的單位質量及彼此距離都不斷縮小），這些振盪的現象就會發生在越高頻且振幅減小，但這將會需要更多的時間作電腦計算。此處我們已經過不斷的嘗試，找出不使結果太失真，又可在合理的時間內計算出結果的解析度，約為箱子每邊長各有 31 顆黑暗能量粒子，故共有  $31^3 = 29791$  顆黑暗能量粒子。黑暗物質的粒子數則為 1000 顆。
- (二)、在圖 5、圖 7 及（公式 1 2）中，我們觀察到，黑暗物質粒子的動能，會隨時間的增加而減少，這個現象可用**能量守恆定理**來解釋。在圖 4 及圖 6 中，我們看到黑暗物質粒子在向右方前進時，會造成黑暗能量粒子沿垂直於行進的方向上振盪，由原本的靜止而變成具有振盪的力學能，故依力學能守恆定律，由於黑暗能量獲得能量，黑暗物必然會損失能量，而致使進行速率減緩。
- (三)、黑暗物質粒子在眾多黑暗能量粒子中行進時，其動能隨時間作線性衰減。至於為何是線性，而不是其它的函數關係，值得再作進一步的理論探討。
- (四)、黑暗物質粒子在眾多黑暗能量粒子中行進時，其動能隨黑暗能量平均密度的 1.92 次方作衰減。為何是 1.92，而不是剛好 2 或其它數字，值得再作進一步的理論探討。
- (五)、如果能在太空中釋放一顆測試粒子，則可利用（公式 1 2）來推導出黑暗能量的密度  $\rho_{DE}$  及狀態方程式係數  $w$  等重要性質。
- (六)、在圖 10 中，當黑暗物質往中央聚集形成星系時，黑暗能量會被往四週推開。這是很自然的**作用力與反作用力定律**（牛頓第三運動定律）之結果，也就是說，當黑暗能量被黑暗物質往四週推開時，黑暗物質所受到的反作用力，會致使黑暗物質更往中心集中，因此造成圖 9 及 10 中，黑暗能量越多、黑暗物質就越集中的現象。
- (七)、在計算圖 9 及 10 的過程中，需要計算兩粒子間的交互作用力。我們只考慮距離為 0.25 以內的粒子間之交互作用，而捨棄更長距離之粒子間的交互作用力，這是為了模擬宇宙是無窮大的事實。因為如果不這麼做，所有的粒子都會受到一個來自它們總質心（位於箱子中心）的淨力，一旦黑暗能量較大時，所有的黑暗物質都會被向外推擠出箱子外。在真實的宇宙中，較長距離的粒子間之作用力，通常一方面由於宇宙大結構上的均勻性，另一方面由於重力的反平方定理，會變得合力幾乎為零，故可忽略不計。
- (八)、在計算反平方力的過程中，我們加入了所謂的 Softening length，也就是防止在兩顆粒子距離為零時，發生力為無窮大的不合理現象。這是一般多體運動模擬中所廣泛使用的技巧。

## 六、結論

我們藉由理論的推導，配合電腦模擬的手段，來探討宇宙中黑暗物質和黑暗能量的神祕力量。我們發現：

- (一)、黑暗能量的存在，會透過交互作用而從運動中的黑暗物質獲得力學能，而且因力學能守恆致使黑暗物質減慢速率，損失動能，滿足

$$E_k(t) = E_{k0} - 9 \times 10^{-5} [1 + 3w|\rho_{DE}]^{1.92} t$$

其中  $E_k$  為黑暗物質的動能， $E_{k0}$  為其初始動能， $w$  為狀態方程式係數， $\rho_{DE}$  為黑暗能量的密度， $t$  為時間。

- (二)、星系的形成過程及結構，會因黑暗能量的存在而改變。  
(三)、在黑暗能量越多時，星系的核心就會越早形成。  
(四)、在距離星系中心  $R$  上的動能  $E_k(R)$ ，會因黑暗能量密度  $\rho_{DE}$  的不同，而呈現不同的線形結構，因此在觀測上，我們可以試圖去量測此  $E_k(R)$  函數，然後和這裡的電腦模擬結果比對，就可以推得出宇宙中黑暗能量的密度  $\rho_{DE}$ 。  
(五)、不管黑暗能量存在與否，星系團的旋轉性質都很相近。

這些研究成果，將可以在短期的未來，直接應用在觀測結果上。

## 七、參考資料及其他

- (一) 高中課本《物質科學物理篇（上）》，龍騰文化事業公司。  
(二) Scientific American, NO.25（科學人，2004年3月號；中文版），遠流出版公司。  
(三) “A first course in general relativity”，B. F. Schutz, Cambridge University Press.

# 附件

## 黑暗物質和黑暗能量密度相同

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define mn 10
#define r0 1.0e-10
#define mas 2.0e-4
#define turn 200

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
    int N,M,R;
    double x,y,z,deltaT,m1[mn][3],m2[mn][3],F1[mn][3],F2[mn][3],a1[mn][3],a2[mn][3],v1[mn][3],v2[mn][3];
    char buf[32],filename[64];

    deltaT=1.0e-3;
    N=0;
    while(N<mn)
    {
        x=((double)rand())/32768.0;
        y=((double)rand())/32768.0;
        z=((double)rand())/32768.0;

        if(sqrt(((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)+(z-0.5)*(z-0.5)))<=0.4)
        {
            m1[N][0]=x;
            m1[N][1]=y;
            m1[N][2]=z;
            F1[N][0]=0.0;
            F1[N][1]=0.0;
            F1[N][2]=0.0;
            a1[N][0]=0.0;
            a1[N][1]=0.0;
            a1[N][2]=0.0;
            v1[N][0]=0.0;
            v1[N][1]=0.0;
            v1[N][2]=0.0;
            N++;
        }
    }

    N=0;
    while(N<mn)
    {
        x=((double)rand())/32768.0;
        y=((double)rand())/32768.0;
        z=((double)rand())/32768.0;

        if(sqrt(((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)+(z-0.5)*(z-0.5)))<=0.4)
        {
            m2[N][0]=x;
```



```

        m2[N][1]=y;
        m2[N][2]=z;
        F2[N][0]=0.0;
        F2[N][1]=0.0;
        F2[N][2]=0.0;
        a2[N][0]=0.0;
        a2[N][1]=0.0;
        a2[N][2]=0.0;
        v2[N][0]=0.0;
        v2[N][1]=0.0;
        v2[N][2]=0.0;
        N++;
    }
}

FILE *output;
output=fopen("output0.txt","w");
for(N=0;N<mn;N++)
    fprintf(output, "%e %e %e %e %e %e\n", m1[N][0],m1[N][1],m1[N][2], v1[N][0],v1[N][1],v1[N][2]);

fprintf(output,"\n");

for(N=0;N<mn;N++)
    fprintf(output, "%e %e %e %e %e %e\n", m2[N][0],m2[N][1],m2[N][2], v2[N][0],v2[N][1],v2[N][2]);

fclose(output);

for(R=1;R<turn+1;R++)
{
    for(N=0;N<mn;N++)
    {
        for(M=N+1;M<mn;M++)
        {

            tempd1=m1[M][0]-m1[N][0];
            tempd2=m1[M][1]-m1[N][1];
            tempd3=m1[M][2]-m1[N][2];
            tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
            tempd5=sqrt(tempd4);
            tempd6=1.0*mas / ((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            F1[N][0]+= tempd1;
            F1[N][1]+= tempd2;
            F1[N][2]+= tempd3;

            F1[M][0]-= tempd1;
            F1[M][1]-= tempd2;
            F1[M][2]-= tempd3;

        }
    }

    for(N=0;N<mn;N++)
    {
        for(M=0;M<mn;M++)

```

```

{
tempd1=m2[M][0]-m1[N][0];
tempd2=m2[M][1]-m1[N][1];
tempd3=m2[M][2]-m1[N][2];
tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
tempd5=sqrt(tempd4);
tempd6=-2.0*mas / ((tempd5+r0)*(tempd5+r0));
tempd1 *= tempd6/(tempd5+r0);
tempd2 *= tempd6/(tempd5+r0);
tempd3 *= tempd6/(tempd5+r0);

F1[N][0]+= tempd1;
F1[N][1]+= tempd2;
F1[N][2]+= tempd3;

F2[N][0]-= tempd1;
F2[N][1]-= tempd2;
F2[N][2]-= tempd3;
}

a1[N][0]=F1[N][0]/mas;
a1[N][1]=F1[N][1]/mas;
a1[N][2]=F1[N][2]/mas;

m1[N][0]+=v1[N][0]*deltaT+0.5*a1[N][0]*deltaT*deltaT;
m1[N][1]+=v1[N][1]*deltaT+0.5*a1[N][1]*deltaT*deltaT;
m1[N][2]+=v1[N][2]*deltaT+0.5*a1[N][2]*deltaT*deltaT;

v1[N][0]+=a1[N][0]*deltaT;
v1[N][1]+=a1[N][1]*deltaT;
v1[N][2]+=a1[N][2]*deltaT;
}

for(N=0;N<mn;N++)
{
for(M=N+1;M<mn;M++)
{

tempd1=m2[M][0]-m2[N][0];
tempd2=m2[M][1]-m2[N][1];
tempd3=m2[M][2]-m2[N][2];
tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
tempd5=sqrt(tempd4);
tempd6=-2.0*mas / ((tempd5+r0)*(tempd5+r0));
tempd1 *= tempd6/(tempd5+r0);
tempd2 *= tempd6/(tempd5+r0);
tempd3 *= tempd6/(tempd5+r0);

F2[N][0]+= tempd1;
F2[N][1]+= tempd2;
F2[N][2]+= tempd3;

F2[M][0]-= tempd1;
F2[M][1]-= tempd2;
F2[M][2]-= tempd3;

}

a2[N][0]=F2[N][0]/mas;

```

```

a2[N][1]=F2[N][1]/mas;
a2[N][2]=F2[N][2]/mas;

m2[N][0]+=v2[N][0]*deltaT+0.5*a2[N][0]*deltaT*deltaT;
m2[N][1]+=v2[N][1]*deltaT+0.5*a2[N][1]*deltaT*deltaT;
m2[N][2]+=v2[N][2]*deltaT+0.5*a2[N][2]*deltaT*deltaT;

v2[N][0]+=a2[N][0]*deltaT;
v2[N][1]+=a2[N][1]*deltaT;
v2[N][2]+=a2[N][2]*deltaT;
}

strcpy(filename,"output");
strcat(filename,gcvt((float)R,4,buf));
strcat(filename, ".txt");
FILE *output1;
output1=fopen(filename,"w");
for(N=0;N<mn;N++)
    fprintf(output1, "%e %e %e %e %e %e\n", m1[N][0],m1[N][1],m1[N][2],
        v1[N][0],v1[N][1],v1[N][2]);

fprintf(output1, "\n");

for(N=0;N<mn;N++)
    fprintf(output1, "%e %e %e %e %e %e\n", m2[N][0],m2[N][1],m2[N][2],
        v2[N][0],v2[N][1],v2[N][2]);

fclose(output1);

for(N=0;N<mn;N++)
{
    F1[N][0]=0.0;
    F1[N][1]=0.0;
    F1[N][2]=0.0;
    a1[N][0]=0.0;
    a1[N][1]=0.0;
    a1[N][2]=0.0;
    F2[N][0]=0.0;
    F2[N][1]=0.0;
    F2[N][2]=0.0;
    a2[N][0]=0.0;
    a2[N][1]=0.0;
    a2[N][2]=0.0;
}
}
}

```

## 黑暗物質和黑暗能量密度爲 3:7

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define mn 50
#define r0 1.0e-10
#define mas 1.0e-5
#define turn 200

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
    int N,M,R;
    double x,y,z,deltaT,m1[mn][3],m2[mn][3],F1[mn][3],F2[mn][3],a1[mn][3],a2[mn][3],v1[mn][3],v2[mn][3];
    char buf[32],filename[64];

    deltaT=1.0e-3;
    N=0;
    while(N<mn)
    {
        x=((double)rand())/32768.0;
        y=((double)rand())/32768.0;
        z=((double)rand())/32768.0;

        if(sqrt(((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)+(z-0.5)*(z-0.5))<=0.4))
        {
            m1[N][0]=x;
            m1[N][1]=y;
            m1[N][2]=z;
            F1[N][0]=0.0;
            F1[N][1]=0.0;
            F1[N][2]=0.0;
            a1[N][0]=0.0;
            a1[N][1]=0.0;
            a1[N][2]=0.0;
            v1[N][0]=0.0;
            v1[N][1]=0.0;
            v1[N][2]=0.0;
            N++;
        }
    }

    N=0;
    while(N<mn)
    {
        x=((double)rand())/32768.0;
        y=((double)rand())/32768.0;
        z=((double)rand())/32768.0;

        if(sqrt(((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)+(z-0.5)*(z-0.5))<=0.4))
        {
            m2[N][0]=x;
            m2[N][1]=y;
```

```

        m2[N][2]=z;
        F2[N][0]=0.0;
        F2[N][1]=0.0;
        F2[N][2]=0.0;
        a2[N][0]=0.0;
        a2[N][1]=0.0;
        a2[N][2]=0.0;
        v2[N][0]=0.0;
        v2[N][1]=0.0;
        v2[N][2]=0.0;
        N++;
    }
}

FILE *output;
output=fopen("output0.txt","w");
for(N=0;N<mn;N++)
    fprintf(output, "%e %e %e %e %e %e\n", m1[N][0],m1[N][1],m1[N][2], v1[N][0],v1[N][1],v1[N][2]);

fprintf(output,"\n");

for(N=0;N<mn;N++)
    fprintf(output, "%e %e %e %e %e %e\n", m2[N][0],m2[N][1],m2[N][2], v2[N][0],v2[N][1],v2[N][2]);

fclose(output);

for(R=1;R<turn+1;R++)
{
    for(N=0;N<mn;N++)
    {
        for(M=N+1;M<mn;M++)
        {

            tempd1=m1[M][0]-m1[N][0];
            tempd2=m1[M][1]-m1[N][1];
            tempd3=m1[M][2]-m1[N][2];
            tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
            tempd5=sqrt(tempd4);
            tempd6=1.0*mas*3 / ((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            F1[N][0]+= tempd1;
            F1[N][1]+= tempd2;
            F1[N][2]+= tempd3;

            F1[M][0]-= tempd1;
            F1[M][1]-= tempd2;
            F1[M][2]-= tempd3;

        }
    }

    for(N=0;N<mn;N++)
    {
        for(M=0;M<mn;M++)
        {

```

```

tempd1=m2[M][0]-m1[N][0];
tempd2=m2[M][1]-m1[N][1];
tempd3=m2[M][2]-m1[N][2];
tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
tempd5=sqrt(tempd4);
tempd6=-2.0*mas*7 / ((tempd5+r0)*(tempd5+r0));
tempd1 *= tempd6/(tempd5+r0);
tempd2 *= tempd6/(tempd5+r0);
tempd3 *= tempd6/(tempd5+r0);

F1[N][0]+= tempd1;
F1[N][1]+= tempd2;
F1[N][2]+= tempd3;

F2[N][0]-= tempd1;
F2[N][1]-= tempd2;
F2[N][2]-= tempd3;
}

a1[N][0]=F1[N][0]/mas;
a1[N][1]=F1[N][1]/mas;
a1[N][2]=F1[N][2]/mas;

m1[N][0]+=v1[N][0]*deltaT+0.5*a1[N][0]*deltaT*deltaT;
m1[N][1]+=v1[N][1]*deltaT+0.5*a1[N][1]*deltaT*deltaT;
m1[N][2]+=v1[N][2]*deltaT+0.5*a1[N][2]*deltaT*deltaT;

v1[N][0]+=a1[N][0]*deltaT;
v1[N][1]+=a1[N][1]*deltaT;
v1[N][2]+=a1[N][2]*deltaT;
}

for(N=0;N<mn;N++)
{
    for(M=N+1;M<mn;M++)
    {

tempd1=m2[M][0]-m2[N][0];
tempd2=m2[M][1]-m2[N][1];
tempd3=m2[M][2]-m2[N][2];
tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
tempd5=sqrt(tempd4);
tempd6=-2.0*mas*7 / ((tempd5+r0)*(tempd5+r0));
tempd1 *= tempd6/(tempd5+r0);
tempd2 *= tempd6/(tempd5+r0);
tempd3 *= tempd6/(tempd5+r0);

F2[N][0]+= tempd1;
F2[N][1]+= tempd2;
F2[N][2]+= tempd3;

F2[M][0]-= tempd1;
F2[M][1]-= tempd2;
F2[M][2]-= tempd3;

}
}

```

```

a2[N][0]=F2[N][0]/mas;
a2[N][1]=F2[N][1]/mas;
a2[N][2]=F2[N][2]/mas;

m2[N][0]+=v2[N][0]*deltaT+0.5*a2[N][0]*deltaT*deltaT;
m2[N][1]+=v2[N][1]*deltaT+0.5*a2[N][1]*deltaT*deltaT;
m2[N][2]+=v2[N][2]*deltaT+0.5*a2[N][2]*deltaT*deltaT;

v2[N][0] +=a2[N][0]*deltaT;
v2[N][1]+=a2[N][1]*deltaT;
v2[N][2]+=a2[N][2]*deltaT;
}

strcpy(filename,"output");
strcat(filename,gcvt((float)R,4,buf));
strcat(filename,".txt");
FILE *output1;
output1=fopen(filename,"w");
for(N=0;N<mn;N++)
    fprintf(output1, "%e %e %e %e %e %e\n", m1[N][0],m1[N][1],m1[N][2],
        v1[N][0],v1[N][1],v1[N][2]);

fprintf(output1,"\n");

for(N=0;N<mn;N++)
    fprintf(output1, "%e %e %e %e %e %e\n", m2[N][0],m2[N][1],m2[N][2],
        v2[N][0],v2[N][1],v2[N][2]);

fclose(output1);

for(N=0;N<mn;N++)
{
    F1[N][0]=0.0;
    F1[N][1]=0.0;
    F1[N][2]=0.0;
    a1[N][0]=0.0;
    a1[N][1]=0.0;
    a1[N][2]=0.0;
    F2[N][0]=0.0;
    F2[N][1]=0.0;
    F2[N][2]=0.0;
    a2[N][0]=0.0;
    a2[N][1]=0.0;
    a2[N][2]=0.0;
}
}
}

```

## 在眾多黑暗能量中放入一個黑暗物質的質點

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define mn 10
#define r0 1.0e-2
#define mas 5.0e-4
#define turn 100

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
int N,M,R,a;
double
x,y,z,deltaT,m1[1][3],m2[mn*mn*mn*5][3],a1[mn*mn*mn][3],a2[mn*mn*mn*5][3],v1[mn*mn*mn][3],v2[mn*mn*
mn*5][3],r[mn*mn*mn*5][3];
char buf[32],filename[64];

deltaT=1.0e-1;

a=0;
for(N=0;N<mn*5;N++)
{
for(M=0;M<mn;M++)
{
for(R=0;R<mn;R++)
{
x=((double)N)/10.0;
y=((double)M)/10.0;
z=((double)R)/10.0;

m2[a][0]=x;
m2[a][1]=y;
m2[a][2]=z;
a2[a][0]=0.0;
a2[a][1]=0.0;
a2[a][2]=0.0;
v2[a][0]=0.0;
v2[a][1]=0.0;
v2[a][2]=0.0;
r[a][0]=0.0;
r[a][1]=0.0;
r[a][2]=0.0;

a=a+1;
}
}
}
printf("a=%d\n", a);
fflush(stdout);
N=0;
```



```

m1[0][0]=0.5;
m1[0][1]=0.55;
m1[0][2]=0.55;
v1[0][0]=0.5;
v1[0][1]=0.0;
v1[0][2]=0.0;
a1[0][0]=0.0;
a1[0][1]=0.0;
a1[0][2]=0.0;
printf("1\n");
fflush(stdout);

FILE *output, *fid;

fid=fopen("vt0.txt","w");
fprintf(fid, "0 %e %e %e %e %e %e\n",v1[0][0],v1[0][1],v1[0][2],m1[0][0],m1[0][1],m1[0][2]);

output=fopen("output0.txt","wb");
fwrite(&m1[0][0],sizeof(double), 3, output);
fwrite(&v1[0][0],sizeof(double), 3, output);
for(N=0;N<a;N++){
    fwrite(&m2[N][0],sizeof(double), 3, output);
    fwrite(&v2[N][0],sizeof(double), 3, output);
}
fclose(output);
printf("2\n");
fflush(stdout);

M=0;
N=0;
R=0;
for(R=1;R<turn+1;R++)
{
    for(M=0;M<a;M++)
    {
        tempd1=m2[M][0]-m1[0][0];
        tempd2=m2[M][1]-m1[0][1];
        tempd3=m2[M][2]-m1[0][2];
        tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
        tempd5=sqrt(tempd4);
        if(tempd5<=0.25)
        {
            tempd6=-2.0*mas/((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a1[0][0]+= tempd1;
            a1[0][1]+= tempd2;
            a1[0][2]+= tempd3;

            a2[M][0]-= tempd1;
            a2[M][1]-= tempd2;
            a2[M][2]-= tempd3;
        }
    }
    m1[0][0]+=v1[0][0]*deltaT+0.5*a1[0][0]*deltaT*deltaT;
    m1[0][1]+=v1[0][1]*deltaT+0.5*a1[0][1]*deltaT*deltaT;
    m1[0][2]+=v1[0][2]*deltaT+0.5*a1[0][2]*deltaT*deltaT;
}

```

```

v1[0][0]+=a1[0][0]*deltaT;
v1[0][1]+=a1[0][1]*deltaT;
v1[0][2]+=a1[0][2]*deltaT;

for(N=0;N<a;N++)
{
    for(M=N+1;M<a;M++)
    {
        tempd1=m2[M][0]-m2[N][0];
        if (tempd1>4.0) tempd1 -= 5.0;
        if (tempd1<-4.0) tempd1 += 5.0;
        tempd2=m2[M][1]-m2[N][1];
        if (tempd2>0.7) tempd2 -= 1.0;
        if (tempd2<-0.7) tempd2 += 1.0;
        tempd3=m2[M][2]-m2[N][2];
        if (tempd3>0.7) tempd3 -= 1.0;
        if (tempd3<-0.7) tempd3 += 1.0;
        tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
        tempd5=sqrt(tempd4);
        if(tempd5<=0.25)
        {
            tempd6=-2.0*mas / ((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a2[N][0]+= tempd1;
            a2[N][1]+= tempd2;
            a2[N][2]+= tempd3;

            a2[M][0]-= tempd1;
            a2[M][1]-= tempd2;
            a2[M][2]-= tempd3;
        }
    }
    m2[N][0]+=v2[N][0]*deltaT+0.5*a2[N][0]*deltaT*deltaT;
    m2[N][1]+=v2[N][1]*deltaT+0.5*a2[N][1]*deltaT*deltaT;
    m2[N][2]+=v2[N][2]*deltaT+0.5*a2[N][2]*deltaT*deltaT;

    v2[N][0]+=a2[N][0]*deltaT;
    v2[N][1]+=a2[N][1]*deltaT;
    v2[N][2]+=a2[N][2]*deltaT;
}
if(R%10==0){
    strcpy(filename,"output");
    strcat(filename,gcvt((float)R,4,buf));
    strcat(filename,".txt");
    FILE *output1;
    output1=fopen(filename,"wb");
    fwrite(&m1[0][0],sizeof(double), 3, output1);
    fwrite(&v1[0][0],sizeof(double), 3, output1);
    for(N=0;N<a;N++){
        fwrite(&m2[N][0],sizeof(double), 3, output1);
        fwrite(&v2[N][0],sizeof(double), 3, output1);
    }
    fclose(output1);
}

```

```
fprintf(fid, "%d %e %e %e %e %e %e\n",R,v1[0][0],v1[0][1],v1[0][2],m1[0][0],m1[0][1],m1[0][2]);
fflush(fid);

a1[0][0]=0.0;
a1[0][1]=0.0;
a1[0][2]=0.0;
for(N=0;N<a;N++)
{
    a2[N][0]=0.0;
    a2[N][1]=0.0;
    a2[N][2]=0.0;
}
}
printf("a1=%e\n", a1[0][0]);
printf("a1=%e\n", a1[0][1]);
printf("a1=%e\n", a1[0][2]);
fflush(stdout);
fclose(fid);
}
```

## 放置一顆黑暗物質並改變初速度

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define mn 10
#define r0 1.0e-10
#define mas 2.0e-4
#define turn 3000

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
    int N,M,R,a;
    double
    x,y,z,deltaT,m1[1][3],m2[mn*mn*mn*50][3],a1[mn*mn*mn][3],a2[mn*mn*mn*50][3],v1[mn*mn*mn][3],v2[
    mn*mn*mn*50][3],r[mn*mn*mn*50][3];
    char buf[32],filename[64];

    deltaT=5.0e-3;
    a=0;
    for(N=0;N<mn*50;N++)
    {
        for(M=0;M<mn;M++)
        {
            for(R=0;R<mn;R++)
            {
                x=((double)N)/10.0;
                y=((double)M)/10.0;
                z=((double)R)/10.0;

                m2[a][0]=x;
                m2[a][1]=y;
                m2[a][2]=z;
                a2[a][0]=0.0;
                a2[a][1]=0.0;
                a2[a][2]=0.0;
                v2[a][0]=0.0;
                v2[a][1]=0.0;
                v2[a][2]=0.0;
                r[a][0]=0.0;
                r[a][1]=0.0;
                r[a][2]=0.0;

                a=a+1;
            }
        }
    }
    N=0;

    m1[0][0]=0.55;
    m1[0][1]=0.55;
    m1[0][2]=0.5;
    v1[0][0]=初速度;
    v1[0][1]=0.0;
```

```

v1[0][2]=0.0;

FILE *output, *fid;
fid=fopen("vt0.txt","w");
fprintf(fid, "%e %e %e %e %e %e\n",v1[0][0],v1[0][1],v1[0][2],m1[0][0],m1[0][1],m1[0][2]);

output=fopen("output0.txt","wb");
fwrite(&m1[0][0],sizeof(double), 3, output);
fwrite(&v1[0][0],sizeof(double), 3, output);
for(N=0;N<a;N++){
    fwrite(&m2[N][0],sizeof(double), 3, output);
    fwrite(&v2[N][0],sizeof(double), 3, output);
}
fclose(output);
M=0;
N=0;
R=0;
for(R=1;R<turn+1;R++)
{
    for(M=0;M<a;M++)
    {

        tempd1=m2[M][0]-m1[0][0];
        tempd2=m2[M][1]-m1[0][1];
        tempd3=m2[M][2]-m1[0][2];
        tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
        tempd5=sqrt(tempd4);
        if(tempd5<=0.25)
        {
            tempd6=-2.0*mas/((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a1[0][0]+= tempd1;
            a1[0][1]+= tempd2;
            a1[0][2]+= tempd3;

            a2[M][0]-= tempd1;
            a2[M][1]-= tempd2;
            a2[M][2]-= tempd3;
        }
    }

    m1[0][0]+=v1[0][0]*deltaT+0.5*a1[0][0]*deltaT*deltaT;
    m1[0][1]+=v1[0][1]*deltaT+0.5*a1[0][1]*deltaT*deltaT;
    m1[0][2]+=v1[0][2]*deltaT+0.5*a1[0][2]*deltaT*deltaT;

    v1[0][0]+=a1[0][0]*deltaT;
    v1[0][1]+=a1[0][1]*deltaT;
    v1[0][2]+=a1[0][2]*deltaT;

    for(N=0;N<a;N++)
    {
        a2[N][0]-=r[N][0]*(8.0*1.20206)*mas/0.001;
        a2[N][1]-=r[N][1]*(8.0*1.20206)*mas/0.001;
        a2[N][2]-=r[N][2]*(8.0*1.20206)*mas/0.001;

        v2[N][0]+=a2[N][0]*deltaT;
    }
}

```

```

v2[N][1]+=a2[N][1]*deltaT;
v2[N][2]+=a2[N][2]*deltaT;

tempd1=v2[N][0]*deltaT+0.5*a2[N][0]*deltaT*deltaT;
tempd2=v2[N][1]*deltaT+0.5*a2[N][1]*deltaT*deltaT;
tempd3=v2[N][2]*deltaT+0.5*a2[N][2]*deltaT*deltaT;

r[N][0]+=tempd1;
r[N][1]+=tempd2;
r[N][2]+=tempd3;

m2[N][0]+=tempd1;
m2[N][1]+=tempd2;
m2[N][2]+=tempd3;
}
if(R%50==0){
strcpy(filename,"output");
strcat(filename,gcvt((float)R,4,buf));
strcat(filename,".txt");
FILE *output1;
output1=fopen(filename,"wb");
fwrite(&m1[0][0],sizeof(double), 3, output1);
fwrite(&v1[0][0],sizeof(double), 3, output1);
for(N=0;N<a;N++){
    fwrite(&m2[N][0],sizeof(double), 3, output1);
    fwrite(&v2[N][0],sizeof(double), 3, output1);
}
fclose(output1);
}

fprintf(fid, "%d %e %e %e %e %e %e\n",R,v1[0][0],v1[0][1],v1[0][2],m1[0][0],m1[0][1],m1[0][2]);

a1[0][0]=0.0;
a1[0][1]=0.0;
a1[0][2]=0.0;
for(N=0;N<a;N++)
{
    a2[N][0]=0.0;
    a2[N][1]=0.0;
    a2[N][2]=0.0;
}
}
fclose(fid);
}

```

## 放置一顆黑暗物質並改變黑暗能量密度

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define mn 10
#define r0 1.0e-10
#define mas 2.0e-4
#define turn 3000

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
    int N,M,R,a;
    double
x,y,z,deltaT,m1[1][3],m2[mn*mn*mn*50][3],a1[mn*mn*mn][3],a2[mn*mn*mn*50][3],v1[mn*mn*mn][3],v2[mn*mn
*mn*50][3],r[mn*mn*mn*50][3];
    char buf[32],filename[64];

    deltaT=5.0e-3;
    a=0;
    for(N=0;N<mn*50;N++)
    {
        for(M=0;M<mn;M++)
        {
            for(R=0;R<mn;R++)
            {
                x=((double)N)/10.0;
                y=((double)M)/10.0;
                z=((double)R)/10.0;

                m2[a][0]=x;
                m2[a][1]=y;
                m2[a][2]=z;
                a2[a][0]=0.0;
                a2[a][1]=0.0;
                a2[a][2]=0.0;
                v2[a][0]=0.0;
                v2[a][1]=0.0;
                v2[a][2]=0.0;
                r[a][0]=0.0;
                r[a][1]=0.0;
                r[a][2]=0.0;

                a=a+1;
            }
        }
    }
    printf("a=%d\n", a);
    fflush(stdout);
    N=0;
    m1[0][0]=0.55;
    m1[0][1]=0.55;
    m1[0][2]=0.5;
    v1[0][0]=0.5;
```

```

v1[0][1]=0.0;
v1[0][2]=0.0;
a1[0][0]=0.0;
a1[0][1]=0.0;
a1[0][2]=0.0;

FILE *output, *fid;
fid=fopen("vt0.txt","w");
fprintf(fid, "0 %e %e %e %e %e %e\n",v1[0][0],v1[0][1],v1[0][2],m1[0][0],m1[0][1],m1[0][2]);

output=fopen("output0.txt","wb");
fwrite(&m1[0][0],sizeof(double), 3, output);
fwrite(&v1[0][0],sizeof(double), 3, output);
for(N=0;N<a;N++){
    fwrite(&m2[N][0],sizeof(double), 3, output);
    fwrite(&v2[N][0],sizeof(double), 3, output);
}
fclose(output);

M=0;
N=0;
R=0;
for(R=1;R<turn+1;R++)
{
    for(M=0;M<a;M++)
    {
        tempd1=m2[M][0]-m1[0][0];
        tempd2=m2[M][1]-m1[0][1];
        tempd3=m2[M][2]-m1[0][2];
        tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
        tempd5=sqrt(tempd4);
        if(tempd5<=0.25)
        {
            tempd6=黑暗能量的密度*mas/((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a1[0][0]+= tempd1;
            a1[0][1]+= tempd2;
            a1[0][2]+= tempd3;

            a2[M][0]-= tempd1;
            a2[M][1]-= tempd2;
            a2[M][2]-= tempd3;
        }
    }

    m1[0][0]+=v1[0][0]*deltaT+0.5*a1[0][0]*deltaT*deltaT;
    m1[0][1]+=v1[0][1]*deltaT+0.5*a1[0][1]*deltaT*deltaT;
    m1[0][2]+=v1[0][2]*deltaT+0.5*a1[0][2]*deltaT*deltaT;

    v1[0][0]+=a1[0][0]*deltaT;
    v1[0][1]+=a1[0][1]*deltaT;
    v1[0][2]+=a1[0][2]*deltaT;

    for(N=0;N<a;N++)
    {
        a2[N][0]-=r[N][0]*(8.0*1.20206)*mas/0.001;
    }
}

```



```

a2[N][1]-=r[N][1]*(8.0*1.20206)*mas/0.001;
a2[N][2]-=r[N][2]*(8.0*1.20206)*mas/0.001;

v2[N][0]+=a2[N][0]*deltaT;
v2[N][1]+=a2[N][1]*deltaT;
v2[N][2]+=a2[N][2]*deltaT;

tempd1=v2[N][0]*deltaT+0.5*a2[N][0]*deltaT*deltaT;
tempd2=v2[N][1]*deltaT+0.5*a2[N][1]*deltaT*deltaT;
tempd3=v2[N][2]*deltaT+0.5*a2[N][2]*deltaT*deltaT;

r[N][0]+=tempd1;
r[N][1]+=tempd2;
r[N][2]+=tempd3;

m2[N][0]+=tempd1;
m2[N][1]+=tempd2;
m2[N][2]+=tempd3;

}
if(R%50==0){
strcpy(filename,"output");
strcat(filename,gcvt((float)R,4,buf));
strcat(filename,".txt");
FILE *output1;
output1=fopen(filename,"wb");
fwrite(&m1[0][0],sizeof(double), 3, output1);
fwrite(&v1[0][0],sizeof(double), 3, output1);
for(N=0;N<a;N++){
    fwrite(&m2[N][0],sizeof(double), 3, output1);
    fwrite(&v2[N][0],sizeof(double), 3, output1);
}
fclose(output1);
}

fprintf(fid, "%d %e %e %e %e %e %e %e\n",R,v1[0][0],v1[0][1],v1[0][2],m1[0][0],m1[0][1],m1[0][2]);

a1[0][0]=0.0;
a1[0][1]=0.0;
a1[0][2]=0.0;
for(N=0;N<a;N++)
{
    a2[N][0]=0.0;
    a2[N][1]=0.0;
    a2[N][2]=0.0;
}
}
fclose(fid);
}

```

# 放置大量的黑暗物質，模擬宇宙中星系的形成

## 1. 有黑暗能量

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define dm 1000
#define mn 31
#define r0 5.0e-2
#define mas 1.0e-5
#define turn 100

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
int N,M,R,a;
double
x,y,z,deltaT,m1[dm][3],m2[mn*mn*mn][3],a1[dm][3],a2[mn*mn*mn][3],v1[dm][3],v2[mn*mn*mn][3],r[mn*mn*mn]
[3];
char buf[32],filename[64];

deltaT=2.0e-2;

a=0;
for(N=0;N<mn;N++)
{
for(M=0;M<mn;M++)
{
for(R=0;R<mn;R++)
{
x=((double)N)/31.0;
y=((double)M)/31.0;
z=((double)R)/31.0;

m2[a][0]=x;
m2[a][1]=y;
m2[a][2]=z;
a2[a][0]=0.0;
a2[a][1]=0.0;
a2[a][2]=0.0;
v2[a][0]=0.0;
v2[a][1]=0.0;
v2[a][2]=0.0;
r[a][0]=0.0;
r[a][1]=0.0;
r[a][2]=0.0;

a=a+1;
}
}
}
printf("a=%d\n", a);
```

```

fflush(stdout);
N=0;
while(N<dm)
{
    x=((double)rand())/32768.0;
    y=((double)rand())/32768.0;
    z=((double)rand())/32768.0;

    if(sqrt((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)+(z-0.5)*(z-0.5))<=0.2)
    {
        m1[N][0]=x;
        m1[N][1]=y;
        m1[N][2]=z;
        a1[N][0]=0.0;
        a1[N][1]=0.0;
        a1[N][2]=0.0;
        v1[N][0]=((double)rand())/32768.0/mn;
        v1[N][1]=((double)rand())/32768.0/mn;
        v1[N][2]=((double)rand())/32768.0/mn;
        N++;
    }
}

FILE *output;

output=fopen("output0.txt","wb");
for(N=0;N<dm;N++){
    fwrite(&m1[N][0],sizeof(double), 3, output);
    fwrite(&v1[N][0],sizeof(double), 3, output);
}
for(N=0;N<a;N++){
    fwrite(&m2[N][0],sizeof(double), 3, output);
    fwrite(&v2[N][0],sizeof(double), 3, output);
}
fclose(output);

M=0;
N=0;
R=0;
for(R=1;R<turn+1;R++)
{
    for(N=0;N<dm;N++)
    {
        for(M=N+1;M<dm;M++)
        {
            tempd1=m1[M][0]-m1[N][0];
            tempd2=m1[M][1]-m1[N][1];
            tempd3=m1[M][2]-m1[N][2];
            tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
            tempd5=sqrt(tempd4);
            tempd6=1.0*mas / ((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a1[N][0]+= tempd1;
            a1[N][1]+= tempd2;
            a1[N][2]+= tempd3;
        }
    }
}

```

```

        a1[M][0]-= tempd1;
        a1[M][1]-= tempd2;
        a1[M][2]-= tempd3;
    }
}

for(N=0;N<dm;N++)
{
    for(M=0;M<a;M++)
    {
        tempd1=m2[M][0]-m1[N][0];
        tempd2=m2[M][1]-m1[N][1];
        tempd3=m2[M][2]-m1[N][2];
        tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
        tempd5=sqrt(tempd4);
        if(tempd5<=0.25)
        {
            tempd6=黑暗能量密度*mas/((tempd5+r0)*(tempd5+r0))/10.0; //====HERE
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a1[N][0]+= tempd1;
            a1[N][1]+= tempd2;
            a1[N][2]+= tempd3;

            a2[M][0]-= tempd1;
            a2[M][1]-= tempd2;
            a2[M][2]-= tempd3;

        }
    }
    m1[N][0]+=v1[N][0]*deltaT+0.5*a1[N][0]*deltaT*deltaT;
    m1[N][1]+=v1[N][1]*deltaT+0.5*a1[N][1]*deltaT*deltaT;
    m1[N][2]+=v1[N][2]*deltaT+0.5*a1[N][2]*deltaT*deltaT;

    v1[N][0]+=a1[N][0]*deltaT;
    v1[N][1]+=a1[N][1]*deltaT;
    v1[N][2]+=a1[N][2]*deltaT;
}
for(N=0;N<a;N++){
    a2[N][0]-=r[N][0]*(8.0*1.20206)*mas/1.0;
    a2[N][1]-=r[N][1]*(8.0*1.20206)*mas/1.0;
    a2[N][2]-=r[N][2]*(8.0*1.20206)*mas/1.0;

    v2[N][0]+=a2[N][0]*deltaT;
    v2[N][1]+=a2[N][1]*deltaT;
    v2[N][2]+=a2[N][2]*deltaT;

    tempd1=v2[N][0]*deltaT+0.5*a2[N][0]*deltaT*deltaT;
    tempd2=v2[N][1]*deltaT+0.5*a2[N][1]*deltaT*deltaT;
    tempd3=v2[N][2]*deltaT+0.5*a2[N][2]*deltaT*deltaT;

    r[N][0]+=tempd1;
    r[N][1]+=tempd2;
    r[N][2]+=tempd3;

    m2[N][0]+=tempd1;

```

```

        m2[N][1]+=tempd2;
        m2[N][2]+=tempd3;
    }

    if(R%5==0){
        strcpy(filename,"output");
        strcat(filename,gcvt((float)R,4,buf));
        strcat(filename,".txt");
        FILE *output1;
        output1=fopen(filename,"wb");
        for(N=0;N<dm;N++){
            fwrite(&m1[N][0],sizeof(double), 3, output1);
            fwrite(&v1[N][0],sizeof(double), 3, output1);
        }
        for(N=0;N<a;N++){
            fwrite(&m2[N][0],sizeof(double), 3, output1);
            fwrite(&v2[N][0],sizeof(double), 3, output1);
        }
        fclose(output1);
    }

    for(N=0;N<dm;N++)
    {
        a1[N][0]=0.0;
        a1[N][1]=0.0;
        a1[N][2]=0.0;
    }
    for(N=0;N<a;N++)
    {
        a2[N][0]=0.0;
        a2[N][1]=0.0;
        a2[N][2]=0.0;
    }
}
}

```

## 2. 沒有黑暗能量

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define mn 10
#define r0 5.0e-2
#define mas 1.0e-5
#define turn 3000

extern double sqrt();
double tempd1, tempd2, tempd3, tempd4, tempd5, tempd6;

int main(void)
{
    int N,M,R,a;
    double x,y,z,deltaT,m1[mn*mn*mn][3],a1[mn*mn*mn][3],v1[mn*mn*mn][3];
    char buf[32],filename[64];

```

```

deltaT=1.0e-3;
a=1000;
N=0;
while(N<a)
{
    x=((double)rand())/32768.0;
    y=((double)rand())/32768.0;
    z=((double)rand())/32768.0;

    if(sqrt((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)+(z-0.5)*(z-0.5))<=0.2)
    {
        m1[N][0]=x;
        m1[N][1]=y;
        m1[N][2]=z;
        a1[N][0]=0.0;
        a1[N][1]=0.0;
        a1[N][2]=0.0;
        v1[N][0]=((double)rand())/32768.0/mn;
        v1[N][1]=((double)rand())/32768.0/mn;
        v1[N][2]=((double)rand())/32768.0/mn;
        N++;
    }
}

FILE *output;

output=fopen("output0.txt","wb");
for(N=0;N<a;N++){
fwrite(&m1[N][0],sizeof(double), 3, output);
fwrite(&v1[N][0],sizeof(double), 3, output);
}
fclose(output);

M=0;
N=0;
R=0;
for(R=1;R<turn+1;R++)
{
    for(N=0;N<a;N++)
    {
        for(M=N+1;M<a;M++)
        {
            tempd1=m1[M][0]-m1[N][0];
            tempd2=m1[M][1]-m1[N][1];
            tempd3=m1[M][2]-m1[N][2];
            tempd4=tempd1*tempd1+tempd2*tempd2+tempd3*tempd3;
            tempd5=sqrt(tempd4);
            tempd6=1.0*mas / ((tempd5+r0)*(tempd5+r0));
            tempd1 *= tempd6/(tempd5+r0);
            tempd2 *= tempd6/(tempd5+r0);
            tempd3 *= tempd6/(tempd5+r0);

            a1[N][0]+= tempd1;
            a1[N][1]+= tempd2;
            a1[N][2]+= tempd3;

            a1[M][0]-= tempd1;
            a1[M][1]-= tempd2;

```

```

        a1[M][2]-= tempd3;
    }

    m1[N][0]+=v1[N][0]*deltaT+0.5*a1[N][0]*deltaT*deltaT;
    m1[N][1]+=v1[N][1]*deltaT+0.5*a1[N][1]*deltaT*deltaT;
    m1[N][2]+=v1[N][2]*deltaT+0.5*a1[N][2]*deltaT*deltaT;

    v1[N][0]+=a1[N][0]*deltaT;
    v1[N][1]+=a1[N][1]*deltaT;
    v1[N][2]+=a1[N][2]*deltaT;
}

if(R%50==0){
    strcpy(filename,"output");
    strcat(filename,gcvt((float)R,4,buf));
    strcat(filename,".txt");
    FILE *output1;
    output1=fopen(filename,"wb");
    for(N=0;N<a;N++)
    {
        fwrite(&m1[N][0],sizeof(double), 3, output1);
        fwrite(&v1[N][0],sizeof(double), 3, output1);
    }
    fclose(output1);
}

for(N=0;N<a;N++)
{
    a1[N][0]=0.0;
    a1[N][1]=0.0;
    a1[N][2]=0.0;
}
}
}

```

# **Backstage Manipulator for Cosmic Evolution of Galaxy Clusters**

**Yueh-Lin Chung**

March 10, 2007

We use theoretical derivation and numerical simulations to study first the dynamics between dark energy and dark matter and then the effect of dark energy on the evolution of galaxy clusters, in a newly proposed scenario where the dark energy is treated as free particles. A comparison of our numerical predictions and future observations will yield the energy density of cosmic dark energy.



# Backstage Manipulator for Cosmic Evolution of Galaxy Clusters

## Abstract

We study the effect of dark energy on the evolution of cosmic structure in a scenario where the dark energy is treated as free particles and thus can be localized. By theoretical derivation and numerical simulations, we found that:

1. The dark energy particles gain kinetic energy from a moving dark matter particle through gravitational interaction. Due to energy conservation, the dark matter particle will slow down with time

$$E_k(t) = E_{k0} - 9 \times 10^{-5} \left[ 1 + 3w \rho_{DE} \right]^{1.92} t$$

where  $E_k(t)$  is the kinetic energy of the dark matter particle,  $E_{k0}$  is its initial kinetic energy,  $w$  is the coefficient of equation of state for dark energy,  $\rho_{DE}$  is the mean energy density of dark energy, and  $t$  is the time.

2. The formation history and structure of galaxy clusters are different in the presence of localized dark energy. The more the localized dark energy, the earlier the formation of the cluster core. In addition, the kinetic energy  $E_k(R)$  as a function of  $R$  will be different if the  $\rho_{DE}$  is different. Thus we can compare the observed  $E_k(R)$  of clusters with our results to deduce the  $\rho_{DE}$  in our universe.

The results here can be applied to the observations in the near future.

# Introduction

To understand the universe has been a challenging task for centuries. Recent research shows that round 70% of the energy in our universe is Dark Energy (DE) and around 25% is Dark Matter (DM). The study of these mysterious materials is of great importance for the development of physics in the 21<sup>st</sup> century. In Particular, the DE provides repulsive force, confronting the traditional concept of gravity. Such repulsive force is believed to be the origin of the observed accelerated expansion of the universe.

With theoretical derivations and numerical simulations, we studied DE in a scenario where the DE is treated as free particles. Our results may be compared with future observations of galaxy clusters so as to enhance our understanding of our universe.

# Project Goal

1. To understand the dynamics between DE and DM, where the DE is treated as free particles.
2. To investigate the structure formation of galaxy clusters as they are the largest structure in the universe, so as to understand some intrinsic properties of the cosmos, especially the dark energy.

# Equipments

1. Software: C codes developed by ourselves (for all the simulations here), MATLAB
2. Hardware: Sun Blade 2000 Workstation (for simulations) · Pentium PC (for data analysis)

# Theory

In cosmology, dark energy is normally assumed to be uniformly distributed. However, here we assume otherwise. We allow DE to move freely, interacting with each other and with other particles through equations that obey General Relativity. We use this framework to study the influence of DE on the largest structures in the universe, the galaxy clusters.

The main theory we employ is the inverse-square law. In the Newtonian theory, it goes as

$$F = \frac{GMm}{r^2}, \quad (\text{Equation 1})$$

where  $F$  is the force between the two interacting particles,  $G$  is the Gravitational constant,  $M$  and  $m$  are the masses of the two particles, and  $r$  is the distance between them. However, the inverse-square law directly derived from the Einstein equation contains a 'pressure' term. In other words, the  $\rho$  in  $m = \rho V$  needs to be replaced by  $(\rho + 3p)$ . It goes as

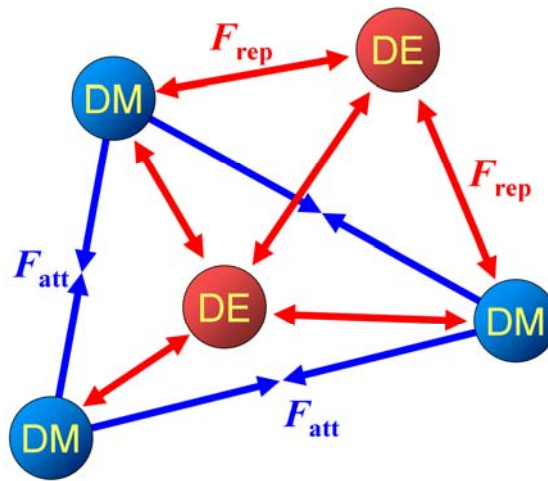
$$F = \frac{GMV(\rho + 3p)}{r^2}. \quad (\text{Equation 2})$$

For dark matter and normal matter  $p = 0$ , so equation (1) is equivalent to equation (2), while for Dark Energy the pressure is negative enough to change the sign on the right-hand side so as to provide a repulsive force. We normally use the Equation of State to define the intrinsic property of the contents in the universe

$$p = w\rho, \quad (\text{Equation 3})$$

where  $w$  is called the coefficient of Equation of State. For DE,  $w < -1/3$ , leading to  $\rho + 3p < 0$ , thus changing the sign on the right-hand side of equation (2). The most common type of DE is the Cosmological Constant, first proposed in the early 20<sup>th</sup> century by Einstein. It has  $w = -1$ , so equation (2) becomes

$$F = \frac{-2GMm}{r^2}. \quad (\text{Equation 4})$$



# Method

Our research steps are the following.

1. Investigate the motion of a dark-matter particle in a sea of dark-energy particles.

- (1) Put a DM particle with an initial velocity in a simulation box, while the DE particles are placed on regular grids in the box.
- (2) Use Equations (1) and (4) to calculate interactions among all particles in the box.
- (3) Compute the net force on each particle and use

$$\mathbf{F} = m\mathbf{a} \quad (\text{Equation 5})$$

to obtain the acceleration  $\mathbf{a}$  of the particle. Next use

$$\mathbf{S} = \mathbf{v}dt + \frac{1}{2}\mathbf{a}dt^2 \quad (\text{Equation 6})$$

to compute the moving distance  $\mathbf{S}$  of the particle after a short time interval  $dt$ .  
Finally use

$$\mathbf{v}(t + dt) = \mathbf{v}(t) + \frac{1}{2}\mathbf{a}dt^2 \quad (\text{Equation 7})$$

to compute the velocity of the particle after  $dt$ .

- (4) For each time step of an interval  $dt$ , output the position  $\mathbf{x}$  and velocity  $\mathbf{v}$  for each particle.
  - (5) Use loops to repeat steps (2) to (4).
  - (6) Use MATLAB to analyze the output data and make animations for the motions of the particles.
2. To save computation time, we derived a new formalism to simplify the computation described above. We studied again the interaction between of a DM particle and all other DE particles.

- (1) From Equations (4) and (5), we know  $a = \frac{-2GM}{r^2}$ . Thus when a DE particle has a displacement of  $r$  from its original position (see Fig.1), the acceleration it experiences from all other DE particles on its left-hand side is

$$\frac{2GM}{(r+d)^2} + \frac{2GM}{(r+2d)^2} + \dots + \frac{2GM}{(r+nd)^2} + \dots = 2GM \sum_{n=1}^{\infty} \frac{1}{n^2 d^2 \left(\frac{r}{nd} + 1\right)^2}.$$

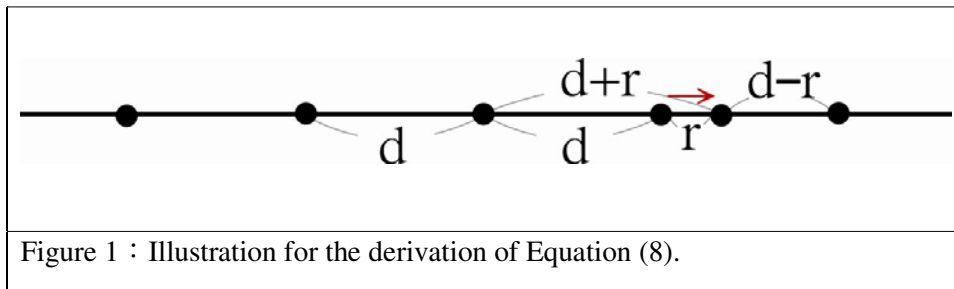
- (2) Similarly the acceleration it experiences from all other DE particles on its right-hand side is

$$\frac{2GM}{(r-d)^2} + \frac{2GM}{(r-2d)^2} + \dots + \frac{2GM}{(r-nd)^2} + \dots = 2GM \sum_{n=1}^{\infty} \frac{1}{n^2 d^2 \left(\frac{r}{nd} - 1\right)^2}.$$

- (3) The difference of the two gives

$$a \cong \frac{8rGM}{d^3} \sum_{n=1}^{\infty} \frac{1}{n^3} = \frac{8rGM}{d^3} * \text{Zeta}(3) \cong \frac{rGM}{d^3} * 8 * 1.20206 \quad (\text{Equation 8})$$

Equation (8) allows us to skip the computation of the interaction between one DM particle and all DE particles and therefore can largely save computation time.



3 · Use the formalism developed in step 2 to speed up the study in step one.

- (1) Study the dependence of the motion of the DM particle on its initial velocity  $v_0$ .
- (2) Study the dependence of the motion of the DM particle on the DE density.
- (3) Study the dependence of the motion of the DM particle on DE of different coefficient of Equation of State  $w$ .

4 · Put a large amount of DM particles in a simulation box to simulate the formation of galaxy clusters. We carried out four runs, with or without dark energy.

(1) With dark energy:

- A. Place DE particles on the regular grids of a 3-dimensional simulation box, and place 1000 DM particles within a sphere of radius of 0.2 times the box size centered at the box center.
- B. Use Equations (1), (4), and (8) to compute the interactions among the particles.
- C. Compute the net force and thus the acceleration of each particle, and then the displacement and the velocity after a time interval  $dt$ .
- D. Save the positions and velocities of all particles for each time step.
- E. Use loops to repeat steps B to D.
- F. Use MATLAB to analyze the results.

(2) Without dark energy:

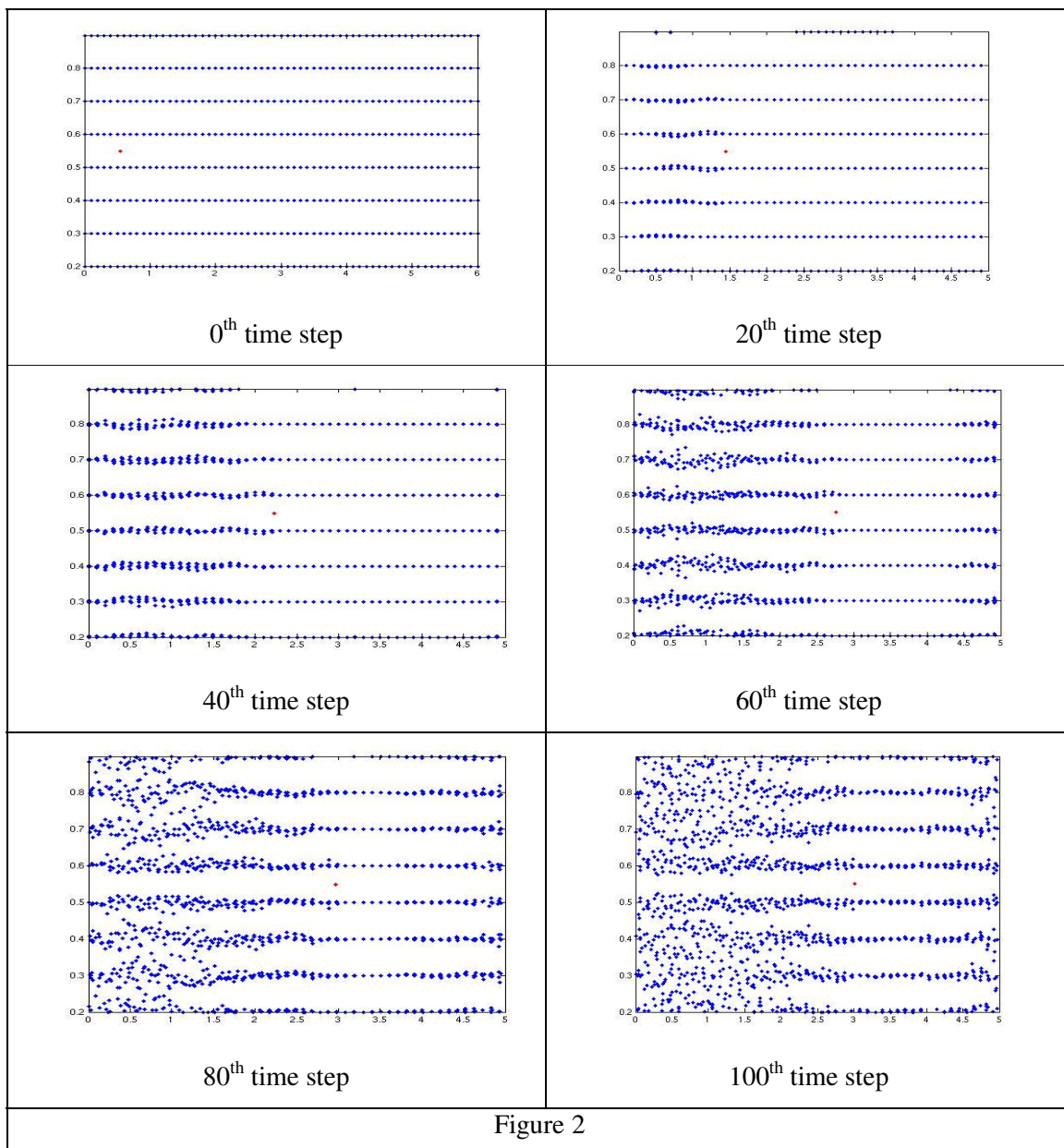
- A. Similar to above but without the DE particles.

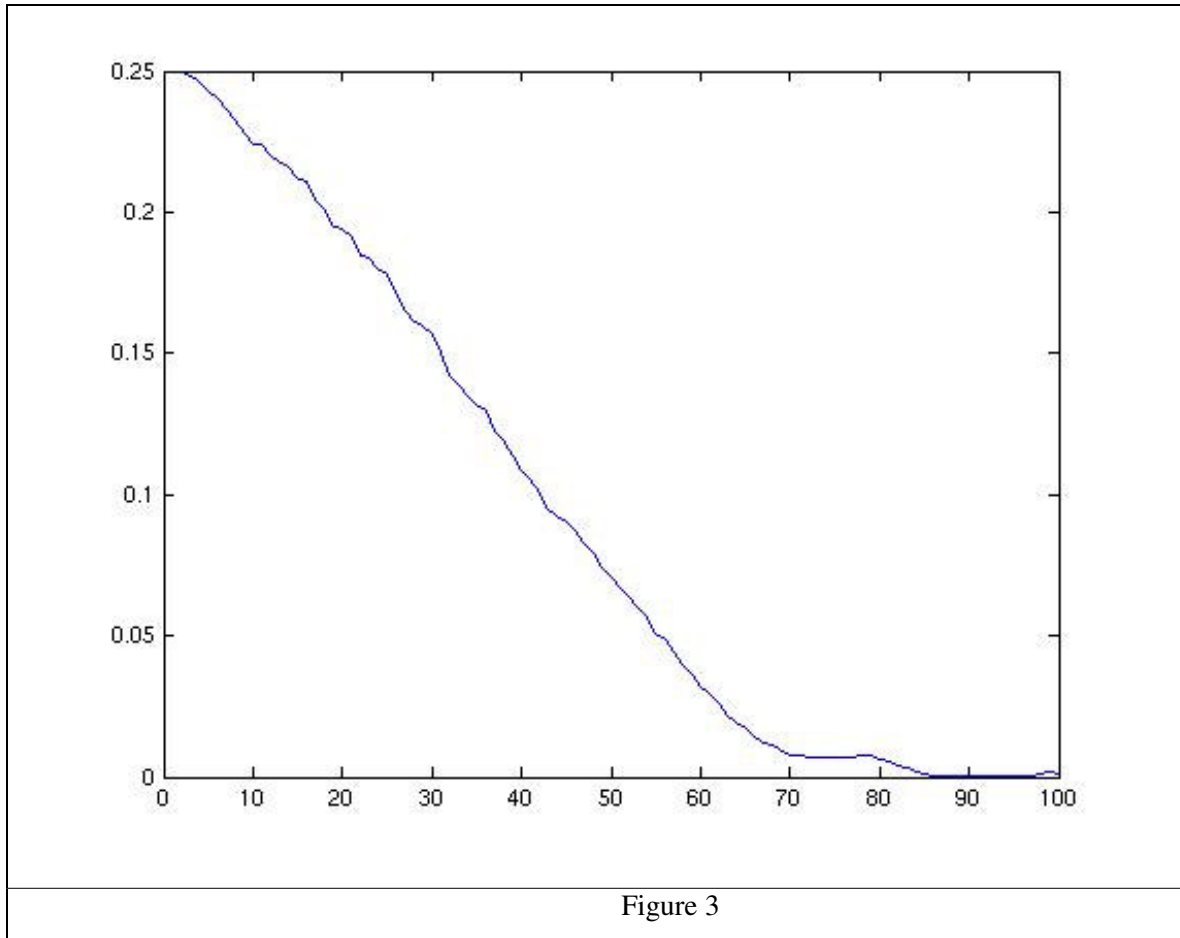
(3) Analyze and compare the results from (1) and (2).

# Results

## I. The motion of a dark-matter particle in a sea of dark-energy particles

Here are the snapshots of the simulation at different time steps. The simulation was carried out in a 3-dimensional box and here are the results projected on a 2-dimensional plane.





The figure 3 shows how the kinetic energy decays with time. Apparently, the kinetic energy decays linearly with time, and obeys this empirical formula (When  $\rho_{DE} = 1$ ):

$$\frac{dE_k}{dt} = -9 \times 10^{-5}$$



## II. Use of a new formalism to save computation time

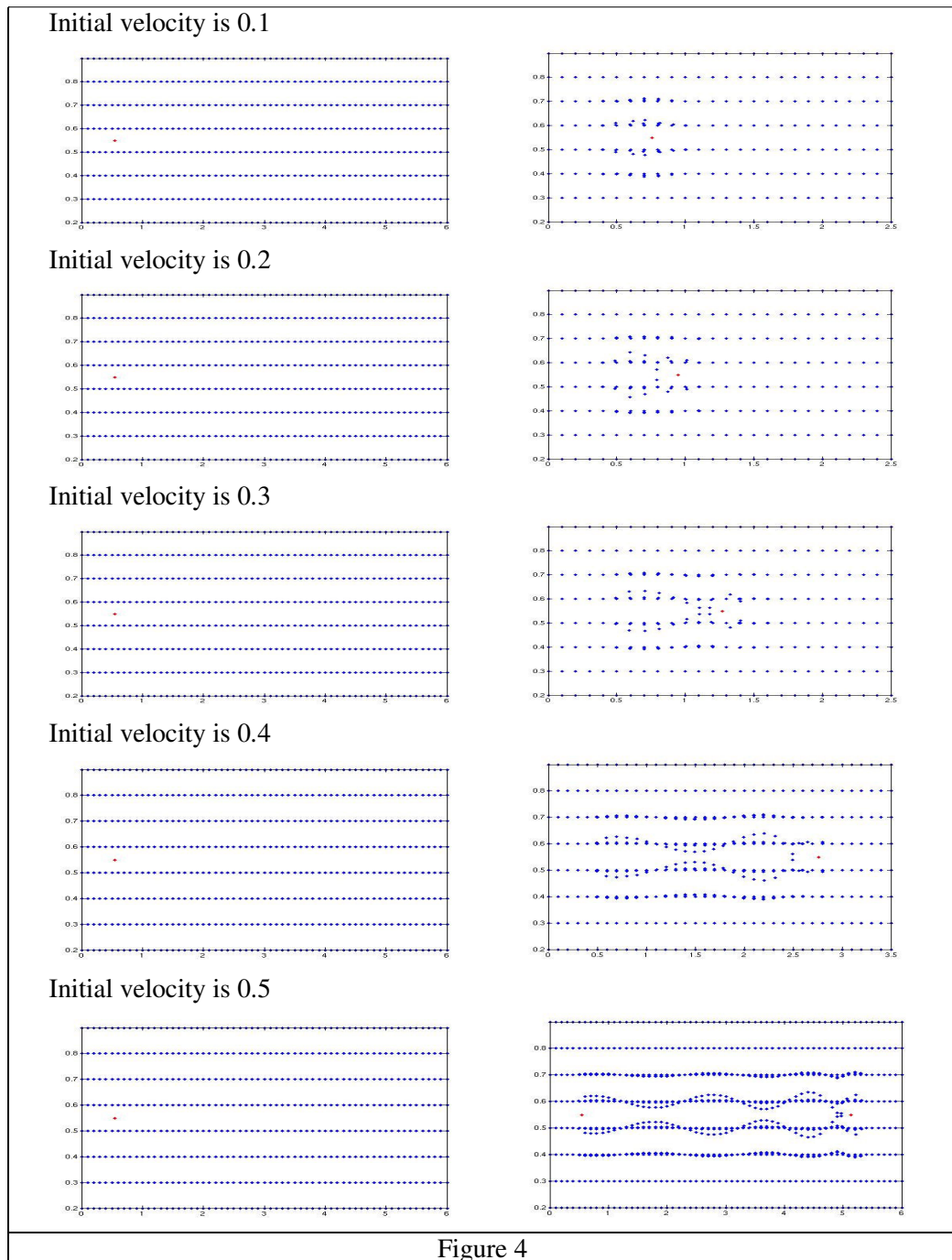


Figure 4 shows the snapshots for simulations with a DM particle of a different initial velocity, at time zero (left) and the time step of a full stop (right). It is clear that the larger the initial velocity of the DM particle, the farther it can travel before a full stop. Along the moving path, it leaks its kinetic energy to the DE particle causing the DE particles to oscillate locally.

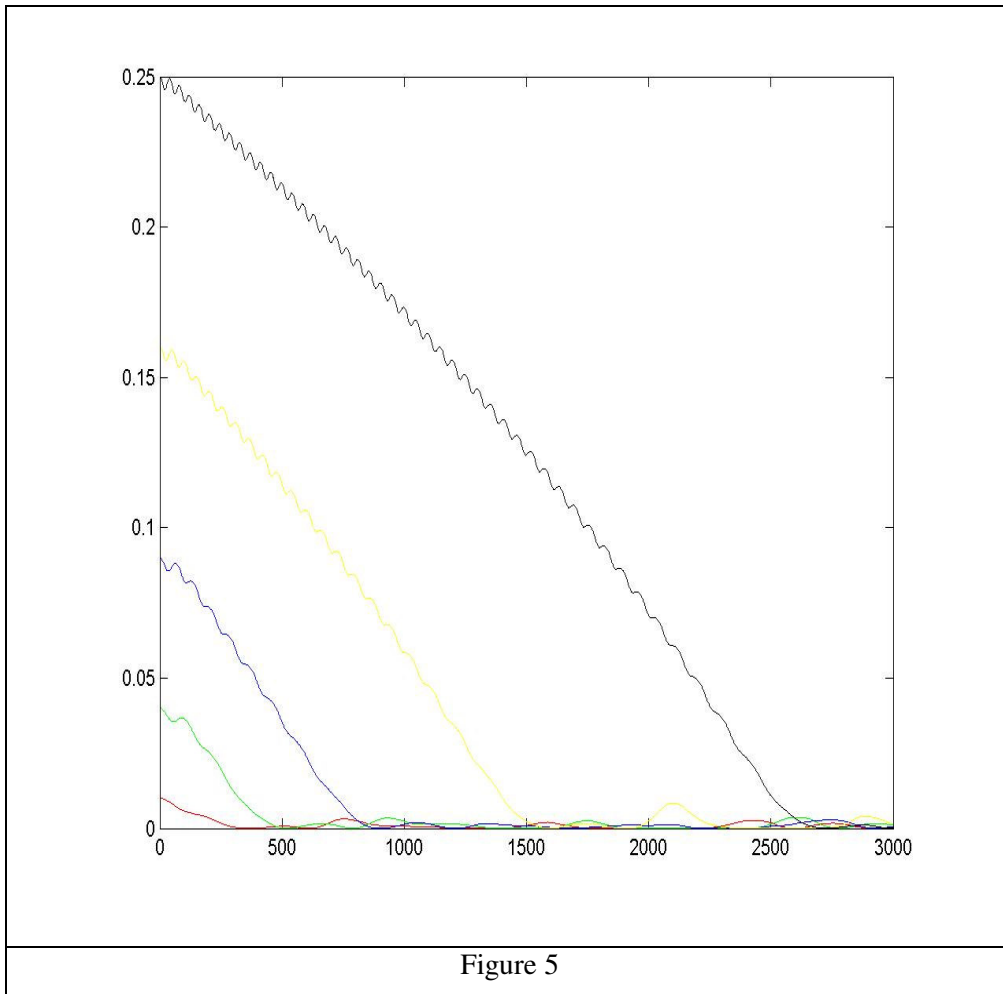


Figure 5 shows the kinetic energy as a function of time. It agrees perfectly with the result in step I, which is much more time consuming by a factor of at least 20. It is also clear that the decay rate of kinetic energy is independent of the initial velocity. It obeys the empirical formula (when  $\rho_{DE} = 1$ )

$$\frac{dE_k}{dt} = -9 \times 10^{-5}. \quad (\text{Equation 9})$$

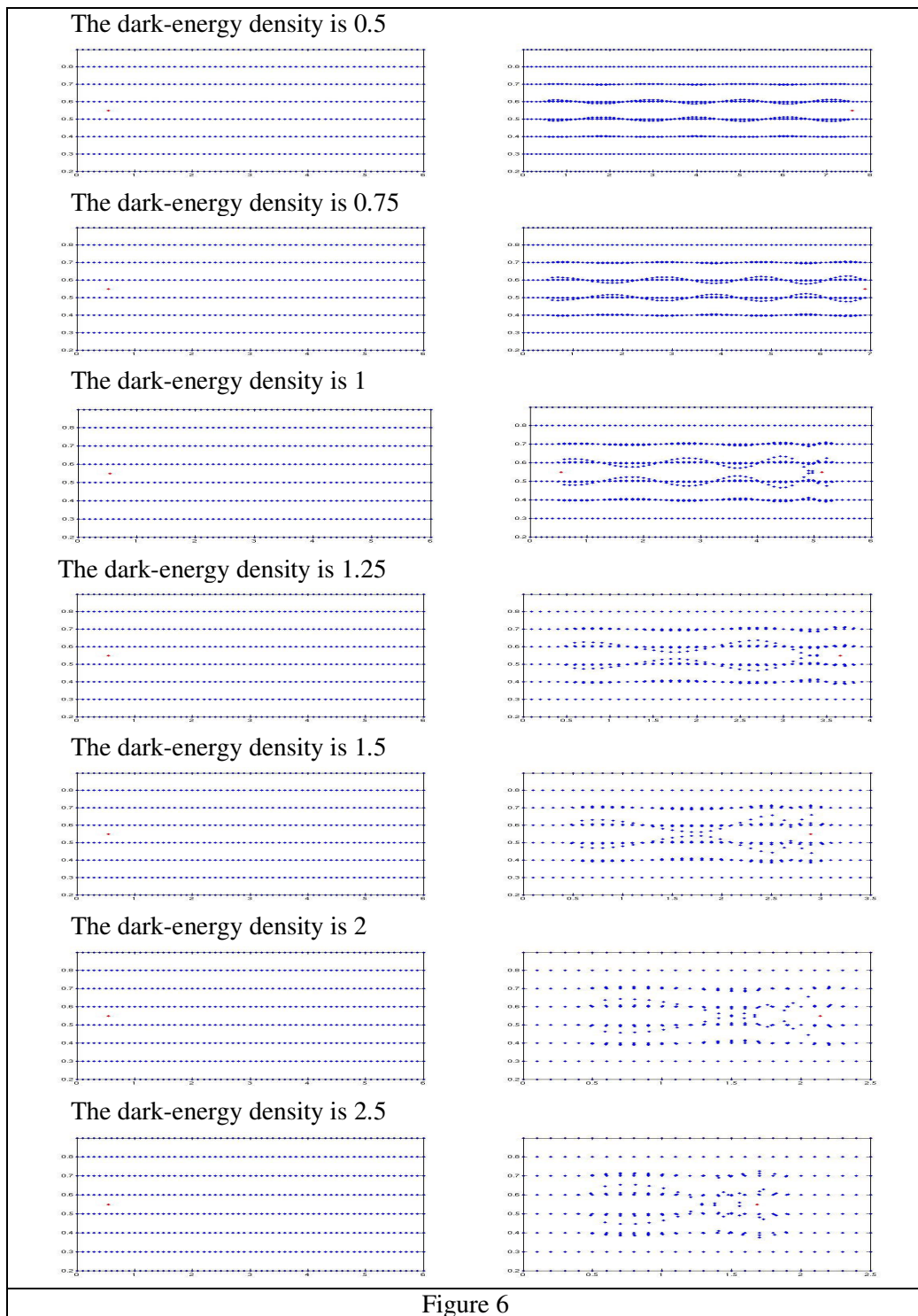


Figure 6

Figure 6 shows that the larger the dark-energy density, the shorter the distance a dark-matter particle can travel within a fixed amount of time. In such case the amplitude in the local oscillation of dark-energy particle is larger.

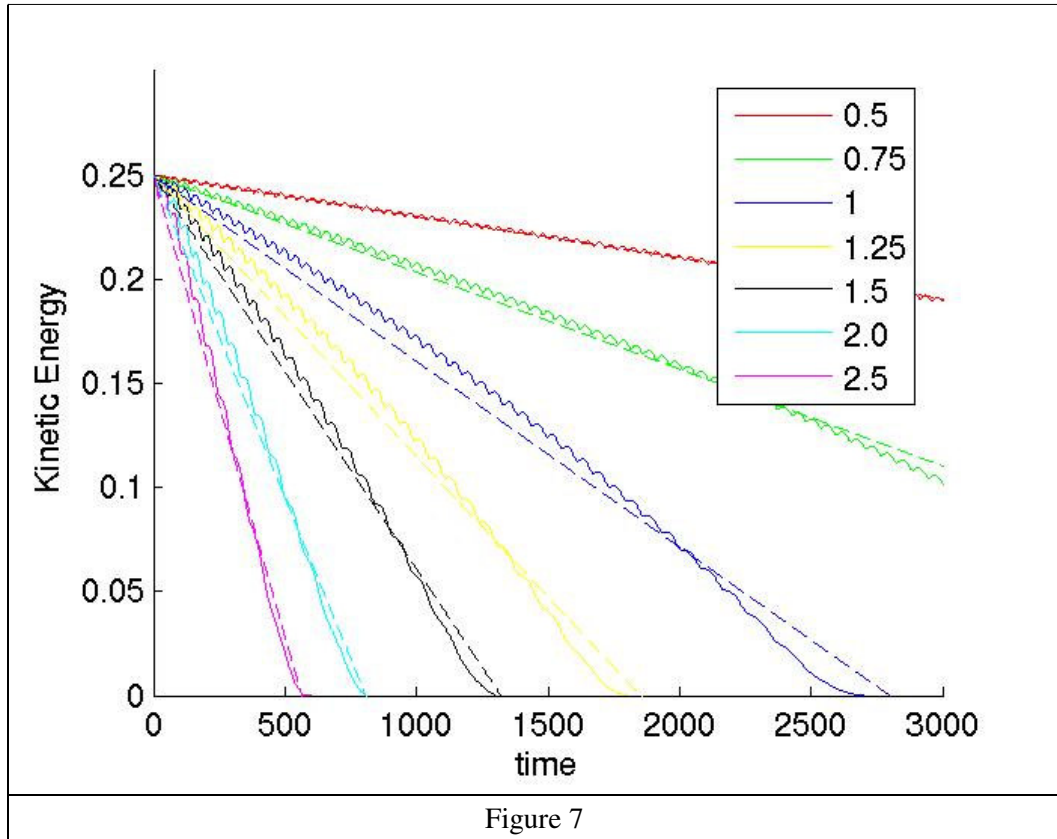
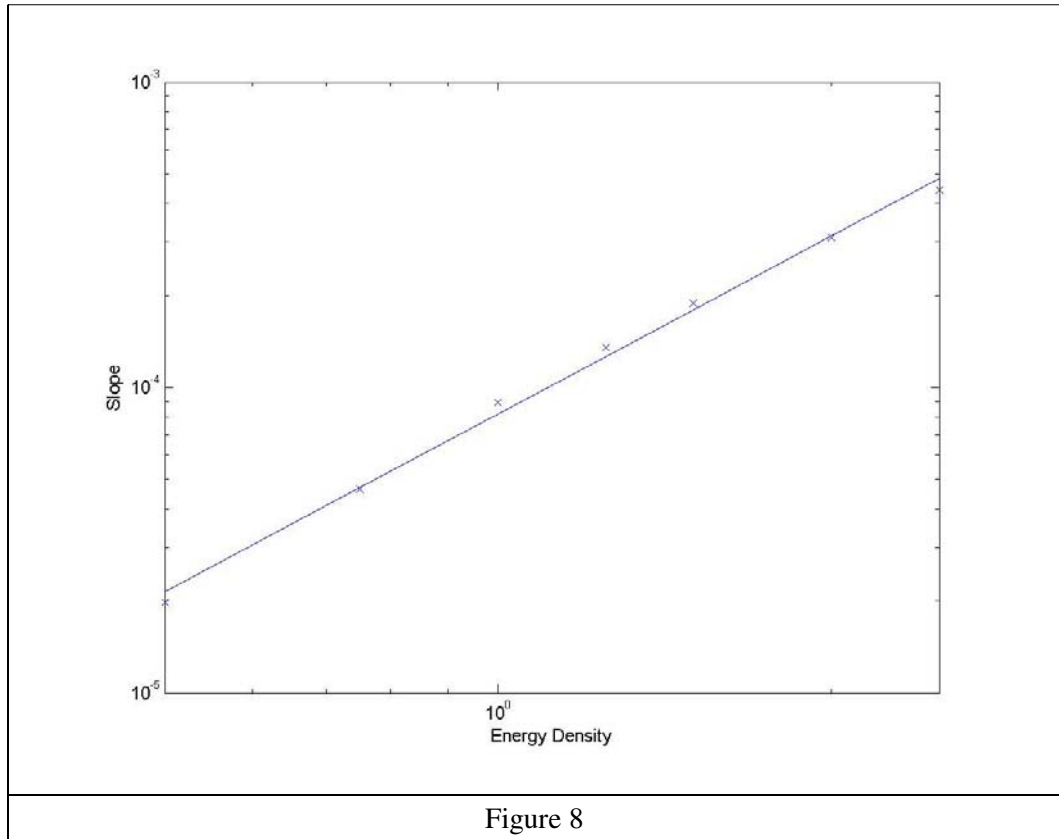


Figure 7 shows not only that the kinetic energy decays linearly with time, but also that the large the dark-energy density the larger the decay rate. Figure 8 shows that the decay rate of the DM kinetic energy and the DE density obeys a power law. These results obey the empirical formula

$$\frac{dE_k}{dt} \propto \rho_{DE}^{1.92} \quad (\text{Equation 10})$$



We also investigated the dependence of the dark-matter kinetic energy on the coefficient of equation of state  $w$ . We revealed the empirical formula

$$\frac{dE_k}{dt} \propto (1 + 3w)^{1.92} \quad (\text{Equation 11})$$

Combining formulae (9) (10) and (11) gives formula (12):

$$E_k(t) = E_{k0} - 9 \times 10^{-5} [1 + 3w|\rho_{DE}|]^{1.92} t \quad (\text{Equation 12})$$

Therefore as long as we know the dark-energy density,  $w$ , and the initial velocity of the dark-matter particle, we can use formula (12) to predict the time dependence of the dark-matter kinetic energy.

### III. Simulation of galaxy clusters

Now we put a large amount of DM particles to simulate the formation of galaxy clusters in the universe. We carried out four runs, with or without dark energy. Figure 9 shows a comparison of the four runs. In the figure, the red, green, and blue dots indicate the positions of the dark-matter particles at the initial, the 1000<sup>th</sup>, and the 2000<sup>th</sup> time steps respectively.

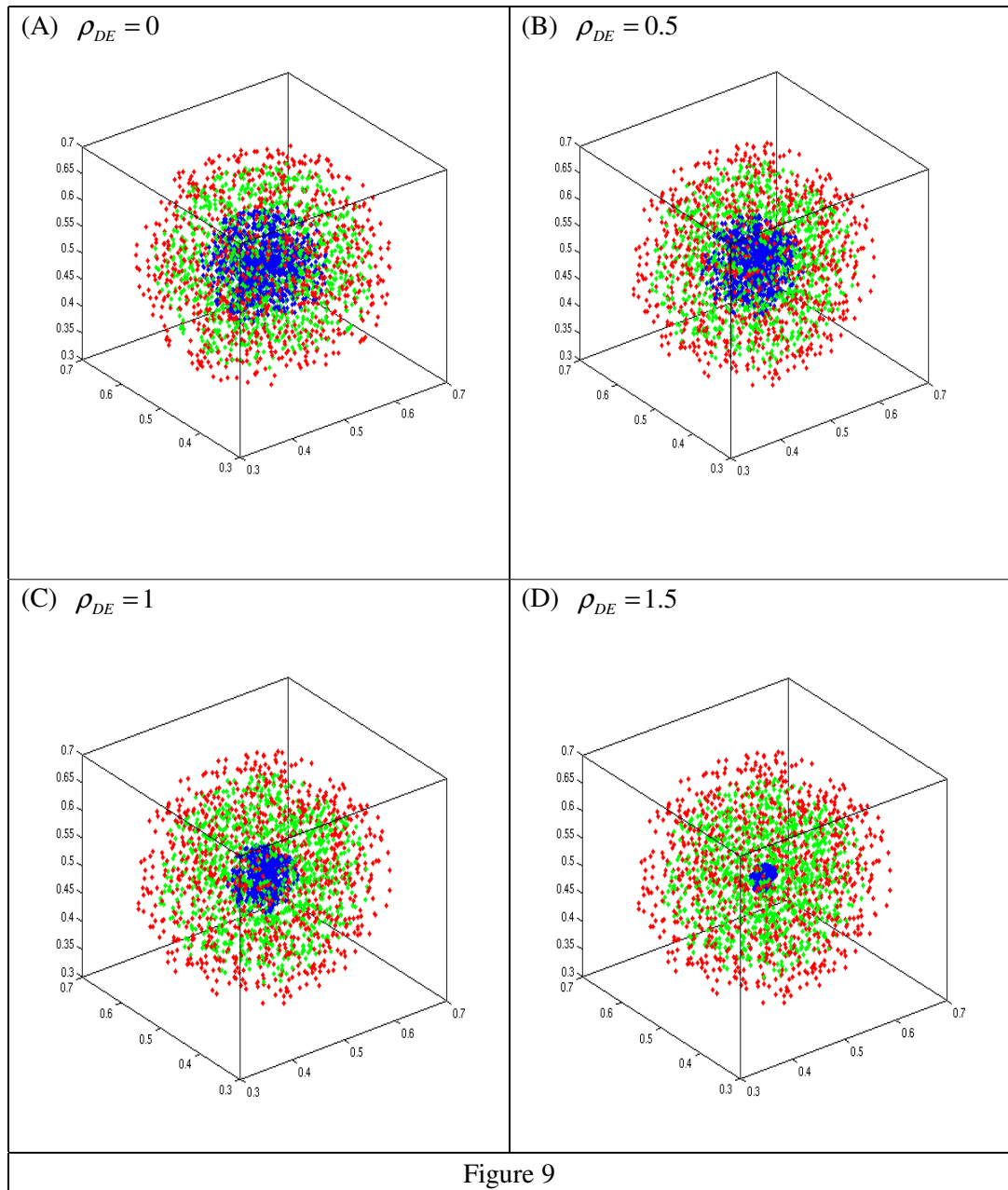
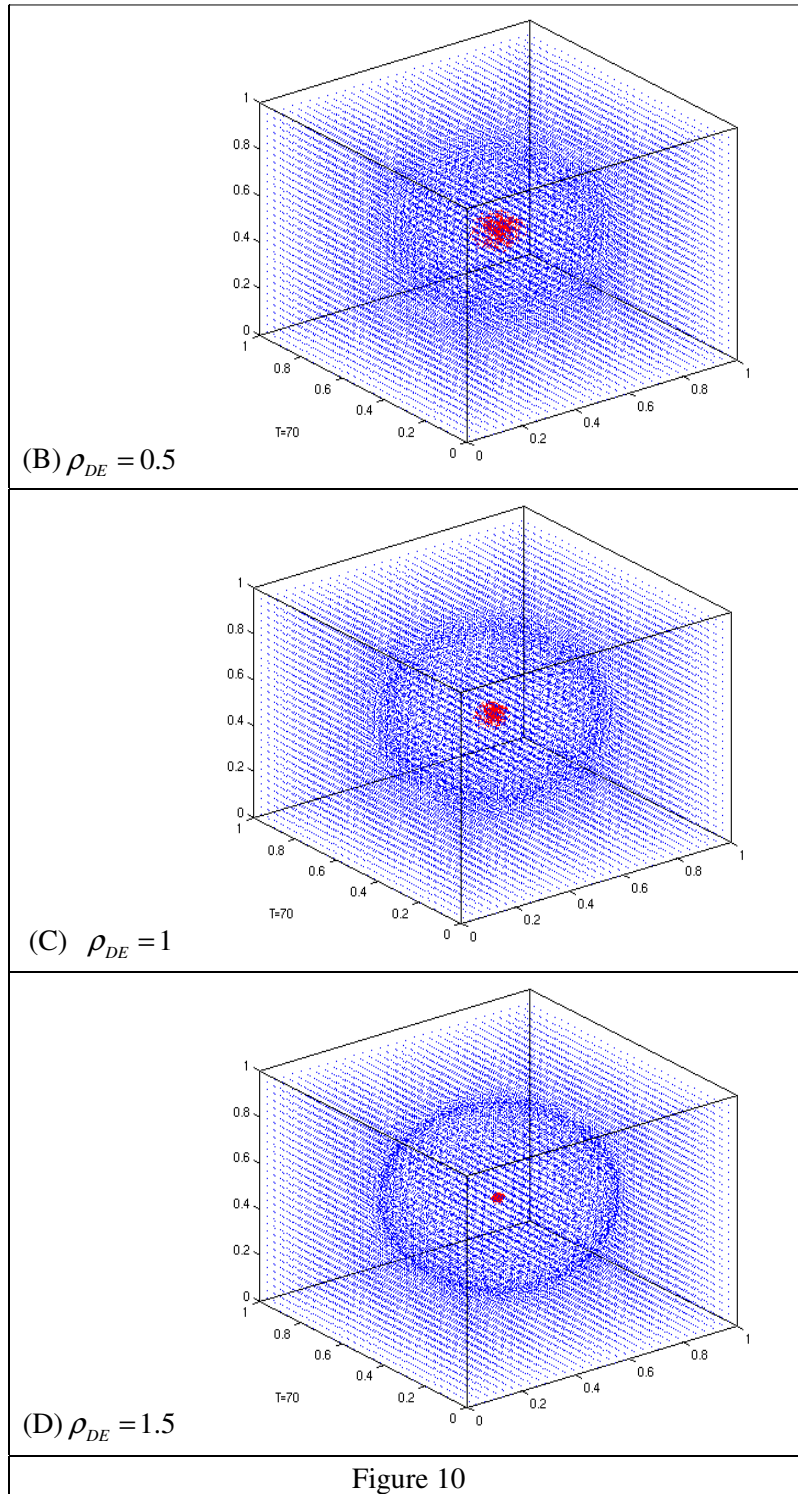


Figure 10 shows the distribution of DM and DE particles at the same time step for simulations with different amount of dark energy. From figures 9 and 10, it is obvious that the larger the dark-energy density, the sooner the cluster forms. In such case the dark-energy particles are further pushed away from the center while squeezing the dark-matter particles more strongly to the center.



To quantitatively study this property, we carried out the following analysis. Figure 11 shows the number of dark-matter particles within a sphere of a radius equal to 0.1 times the box size, as a function of time. The red, green, black, and purple curves represent respectively the cases when the dark-energy density is equal to 0, 0.5, 1, 1.5.

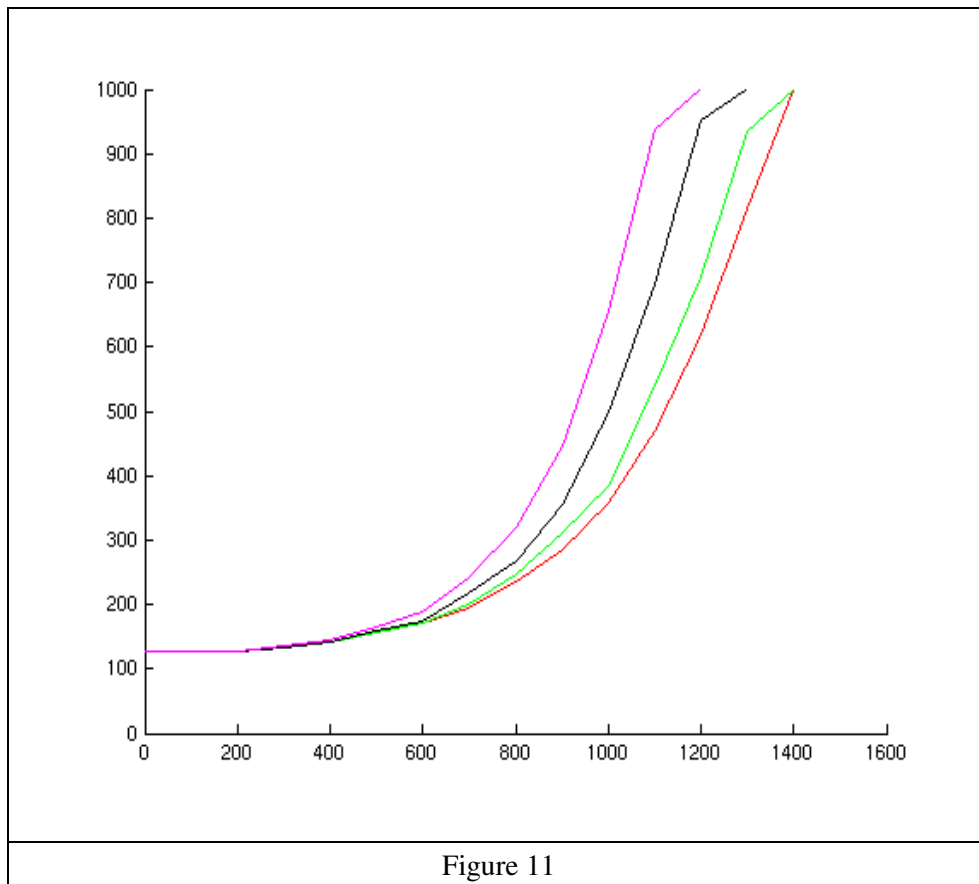
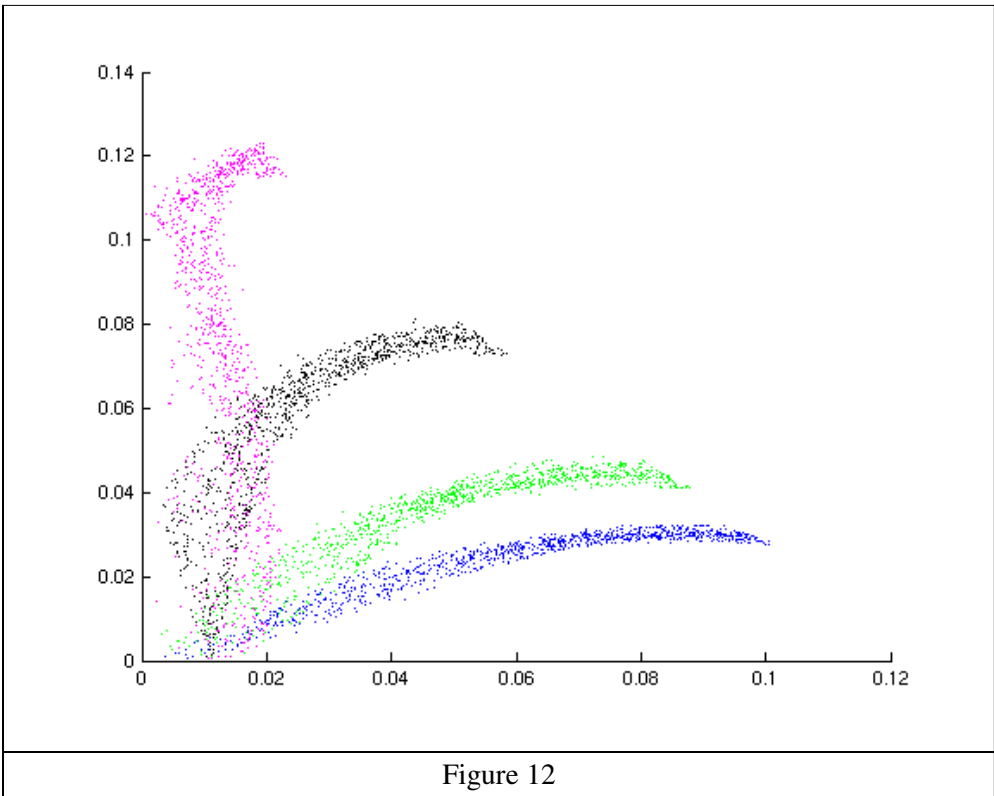


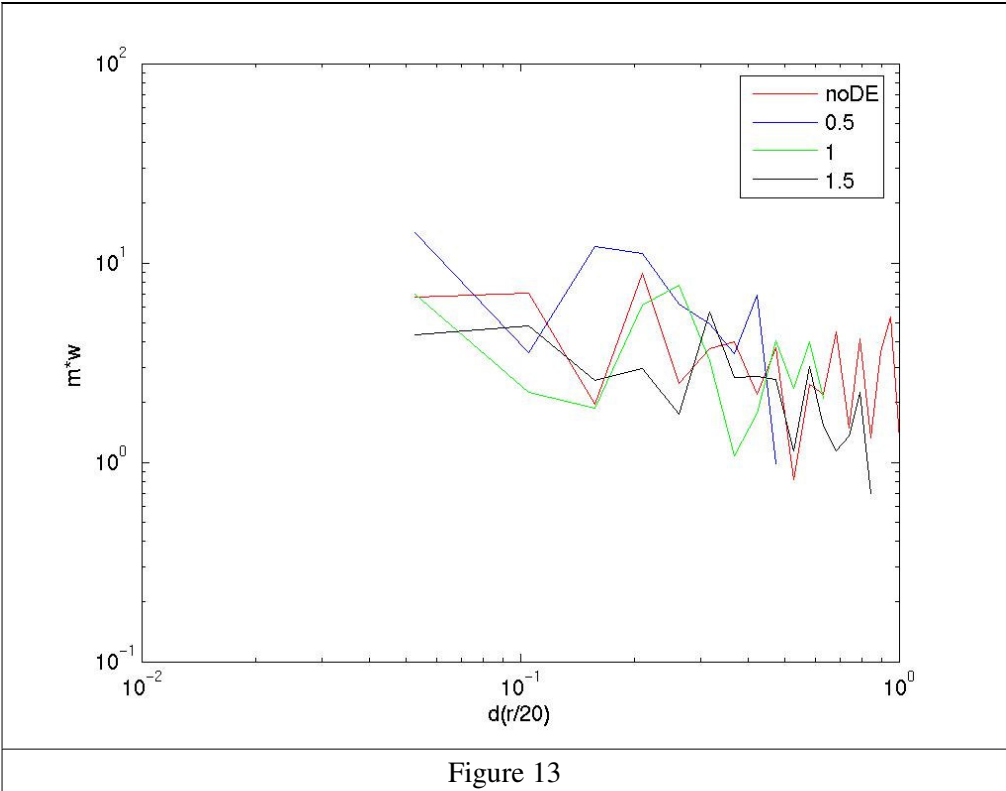
Figure 11



We also studied the distribution of kinetic energy of the dark matter as a function of the distance to the cluster center. Again the four colours represents results of different dark-energy density. The difference is dramatical. Thus we can apply this property to observations to estimate the density of dark energy in our universe.



Finally we investigate the influence of dark energy on the rotation of clusters. Figure 13 shows the mean angular velocity as a function of the distance to the cluster center. We found that although the existence of dark energy does largely affect the formation of clusters, it has little effect on the rotation of clusters. This is somehow surprising and further investigation along this track is required.



# Discussion

1. In Figures 5 and 7, the curves show fluctuations of small amplitude. These fluctuations result from the fact that the distribution of DE in the simulation is discrete rather than continuous, so the DM particle experience cycles of increasing and decreasing repulsive forces when it approaches and leaves a nearby DE particle. We verify that the oscillation period of the curves scales with the size of the spacing of the DE particles. With try and error, we found that to keep both a sufficient resolution and a good efficiency for the simulation a good choice for the size of the simulation is to have  $31^3 = 29791$  DE particles and 1000 DM particles.
2. In Figures 5 and 7, we saw that the kinetic energy of the DM particle decays with time, and this can be explained by the energy conservation. In Figures 4 and 6, we saw that when the DM particle passed through the DE particles the DE particles started oscillating along the directions perpendicular to the moving path of the DM particle. This means the kinetic energy of the DM particle is leaking to the oscillation energy of the DE particles.
3. It requires a further study about why the decay rate of the kinetic energy of a moving DM particle in a sea of DE particle is linear in time.
4. It requires a further study about why the power index in Equation (12) is 1.92.
5. If we can release a DM-like particle in space and observe its motion, then we can use Equation (12) to estimate  $\rho_{DE}$  or equivalently  $w$ .
6. In Figure 10, the fact that the more the DE the more strongly the DM particles are squeezed to the center can be explained by Newton's 3<sup>rd</sup> Law of Motion. More DE means a stronger repulsive force not only pushing the DM to the center but also pushing DE itself outwards.
7. In the simulations, we induced a softening length to avoid the divergence of the force when the distance between two particles is too short.

# Conclusion

From the research presented above, we obtain the following conclusions.

1. A dark-matter particle in motion will leak its kinetic energy to the surrounding dark energy while slowing itself down. It obeys this equation.
2. The structure and formation process of a galaxy cluster is largely influenced by the existence of dark energy. The more the dark energy, the earlier the cluster forms.
3. In a cluster, the kinetic energy of dark matter as a function of the distance to the cluster center is dramatically different for cases of different amount of dark energy. In observation, we may use this property to estimate the density of dark energy in our universe.
4. The rotational property of galaxy cluster is largely independent on the density of dark energy.

# Future Work

In the near future when high-accuracy observations are available for galaxy clusters, we may apply our research results directly to the observations so as to estimate the amount of dark energy better and to understand them better.

# References

1. Physics, Vol. 1, Senior-High-School Textbook.
2. Scientific American, NO.25
3. “A first course in general relativity”, B. F. Schutz, Cambridge University Press.

## 評語

本作品以目前天文物理關注的熱門課題--黑暗能量--為主，探討占宇宙能量百分之七十三的黑暗能量，與占百分之二十三的黑暗物質之間的交互作用，發現黑暗能量能夠促使黑暗物質加速形成星系團等大型結構。研究主題明確，分析過程清楚，值得鼓勵。