

電腦數值模擬導論與實習 期中報告

組員： 農化四 林子傑 B02603041

會計三 吳懿峰 B03702030

1. 題目：

太陽、地球、月亮相對運動之數值模擬

2. 題目動機：

線上資料搜尋到模擬雙星運動後，認為非常可行。因此決定設計 VBA 程式，以真實數據模擬一個恆星、一個行星、一個衛星的二維運動。

3. 題目內容：

i. 公式：

行星運動彼此間作用力僅有萬有引力 ($F = \frac{GMm}{R^2}$)。再將此力對兩星作用一時區(設定 $dt=1000$ 秒)得到加速度 a ，以此加速度作用一時區 ($dt=1000$ 秒) 得到速度，以此速度移動一時區($dt=1000$ 秒)。接著不斷執行程式得到每一時區後的位置，再由上萬個位置製作動畫。

ii. 參數：

太陽質量： 1.989×10^{30} kg

地球質量： 5.97219×10^{24} kg

月球質量： 0.07346×10^{24} kg

重力常數： $6.67408 \times 10^{-11} \text{ m}^3 \text{ s}^{-2} \text{ kg}^{-1}$

地球遠日點距離： 1.521×10^{11} m

月球遠日點距離： 4.055×10^8 m

地球遠日點速度：29300 m/s

月球遠日點速度：964 m/s

4. 公式設計

設定一區間時間 dt ，代表恆星受力的時間，在一般情況來說 dt 應該是一非常微小的值，但為了使此模型的觀測結果更加明顯並節省資料的儲存空

間，我們將 dt 設定為 1000。而加速度 a 則可以透過 $F = \frac{GMm}{R^2} = m \cdot a$ 得

出。再利用 $v = v + a \cdot dt$ 得出速度 v ，再利用 $p = p + v \cdot dt$ 得出位置，並將位置數據輸入於 excel 表格中，共產生地球對太陽 x 座標、地球對太陽 y 座

標、月球對地球 x 座標、月球對地球 y 座標以及月球對太陽 x 座標、月球對太陽 y 座標。

5. 程式碼與解釋：

i. 程式碼 timeout()：

```
33 Sub timeout(duration_ms As Double)
34     start_time = Timer
35     Do
36         DoEvents
37         Loop Until Abs(Timer - start_time) >= duration_ms
38
39 End Sub
```

此程式 timeout()可以讓另一個呼叫 timeout(0.01)的程式 A，跳出 0.01 秒後又再回去程式 A。由此可以使程式 A 不斷地輸出資訊，並在 timeout(0.01)與 A 之間切換。由此我們可以得到大量數據，也可以進行繪圖。

ii. 程式碼 planet_model()：_地球繞日軌道

```
1 Sub planet_model()
2     earth_mass = 5.97219 * (10 ^ 24)
3     sun_mass = 1.989 * (10 ^ 30)
4     v_earth_y = 29300
5     v_earth_x = 0
6     v_sun = 0
7     p_earth_x = 152100000000#
8     p_earth_y = 0
9     p_sun_x = 0
10    p_sun_y = 0
11    cons_g = 6.67408 * (10 ^ (-11))
12    dt = 1000
13    T = 0
14    i = 4
15    Do
16        DoEvents
17        T = T + dt
18        F = cons_g * (earth_mass * sun_mass) / ((p_earth_x - p_sun_x) ^ 2 + (p_earth_y - p_sun_y) ^ 2)
19        a_earth = F / earth_mass
20        a_earth_y = a_earth * ((p_earth_y - p_sun_y) / ((p_earth_x - p_sun_x) ^ 2 + (p_earth_y - p_sun_y) ^ 2) ^ (0.5))
21        a_earth_x = a_earth * ((p_earth_x - p_sun_x) / ((p_earth_x - p_sun_x) ^ 2 + (p_earth_y - p_sun_y) ^ 2) ^ (0.5))
22        v_earth_y = v_earth_y + (-a_earth_y) * dt
23        v_earth_x = v_earth_x + (-a_earth_x) * dt
24        p_earth_y = p_earth_y + v_earth_y * dt
25        p_earth_x = p_earth_x + v_earth_x * dt
26        ActiveSheet.Cells(3, 2) = "p_earth_x"
27        ActiveSheet.Cells(i, 2) = p_earth_x
28        ActiveSheet.Cells(3, 3) = "p_earth_y"
29        ActiveSheet.Cells(i, 3) = p_earth_y
30        i = i + 1
31        timeout (1e-10)
32    Loop Until T = 0
33
34 End Sub
```

第 2~14 行：設定參數，如地球質量、位置、距離、重力常數、時間間距(1000sec)。

第 15~32 行：while 迴圈設定 Loop until T=0 (但 T 會一直增加，得手

動停)，每 1000 秒(dt=1000)移動輸出一次地球位置，並假定太陽幾乎不動。F 為兩星作用力，以萬有引力公式算出。再計算加速度、速度、移動距離。其中 timeout(1e-10)，每 1e-10 秒跑一次 planet_model()。

iii. 程式碼 planet_model_part2()：_月球繞地軌道

```
1 Sub planet_model_part2()  
2 moon_mass = 0.07346 * (10 ^ 24)  
3 earth_mass = 5.97219 * (10 ^ 24)  
4 v_moon_y = 964  
5 v_moon_x = 0  
6 v_earth = 0  
7 p_moon_x = 0.4055 * (10 ^ 9)  
8 p_moon_y = 0  
9 p_earth_x = 0  
10 p_earth_y = 0  
11 cons_g = 6.67408 * (10 ^ (-11))  
12 dt = 1000  
13 T = 0  
14 i = 4  
15 Do  
16 DoEvents  
17 T = T + dt  
18 F = cons_g * (moon_mass * earth_mass) / ((p_moon_x - p_earth_x) ^ 2 + (p_moon_y - p_earth_y) ^ 2)  
19 a_moon = F / moon_mass  
20 a_moon_y = a_moon * ((p_moon_y - p_earth_y) / ((p_moon_x - p_earth_x) ^ 2 + (p_moon_y - p_earth_y) ^ 2) ^ (0.5))  
21 a_moon_x = a_moon * ((p_moon_x - p_earth_x) / ((p_moon_x - p_earth_x) ^ 2 + (p_moon_y - p_earth_y) ^ 2) ^ (0.5))  
22 v_moon_y = v_moon_y + (-a_moon_y) * dt  
23 v_moon_x = v_moon_x + (-a_moon_x) * dt  
24 p_moon_y = p_moon_y + v_moon_y * dt  
25 p_moon_x = p_moon_x + v_moon_x * dt  
26 ActiveSheet.Cells(3, 4) = "p_moon_x"  
27 ActiveSheet.Cells(i, 4) = p_moon_x  
28 ActiveSheet.Cells(3, 5) = "p_moon_y"  
29 ActiveSheet.Cells(i, 5) = p_moon_y  
30 i = i + 1  
31 timeout (1e-30)  
32 Loop Until T = 0  
33  
34 End Sub
```

與 planet_model()一樣，只是將參數修改。將地球設定不動，只讓月球移動。

iv. 程式碼 all_draw()：將得到的地球、月球軌道以動畫方式呈現

```
1 Sub all_draw()  
2 T = 0  
3 x = 4  
4  
5 Do  
6 DoEvents  
7 ActiveSheet.Shapes("圖表 7").IncrementLeft 0  
8 ActiveSheet.Shapes("圖表 7").IncrementTop 0  
9 ActiveChart.FullSeriesCollection(3).XValues = "=工作表1!$D$" + Trim(Str(x)) + ":$D$" + Trim(Str(x + 200))  
10 ActiveChart.FullSeriesCollection(3).Values = "=工作表1!$E$" + Trim(Str(x)) + ":$E$" + Trim(Str(x + 200))  
11  
12 ActiveSheet.Shapes("圖表 3").IncrementLeft 0  
13 ActiveSheet.Shapes("圖表 3").IncrementTop 0  
14 ActiveChart.SeriesCollection(1).XValues = "=工作表1!$B$" + Trim(Str(x)) + ":$B$" + Trim(Str(x + 200))  
15 ActiveChart.SeriesCollection(1).Values = "=工作表1!$C$" + Trim(Str(x)) + ":$C$" + Trim(Str(x + 200))  
16 ActiveChart.SeriesCollection(2).XValues = "=工作表1!$F$" + Trim(Str(x)) + ":$F$" + Trim(Str(x + 200))  
17 ActiveChart.SeriesCollection(2).Values = "=工作表1!$G$" + Trim(Str(x)) + ":$G$" + Trim(Str(x + 200))  
18  
19 ActiveSheet.Shapes("圖表 4").IncrementLeft 0  
20 ActiveSheet.Shapes("圖表 4").IncrementTop 0  
21 ActiveChart.SeriesCollection(1).XValues = "=工作表1!$B$" + Trim(Str(x)) + ":$B$" + Trim(Str(x + 400))  
22 ActiveChart.SeriesCollection(1).Values = "=工作表1!$C$" + Trim(Str(x)) + ":$C$" + Trim(Str(x + 400))  
23 ActiveChart.SeriesCollection(2).XValues = "=工作表1!$F$" + Trim(Str(x)) + ":$F$" + Trim(Str(x + 400))  
24 ActiveChart.SeriesCollection(2).Values = "=工作表1!$G$" + Trim(Str(x)) + ":$G$" + Trim(Str(x + 400))  
25  
26 timeout (1e-20)  
27 x = x + 100  
28  
29 Loop Until T = 1  
30 End Sub
```

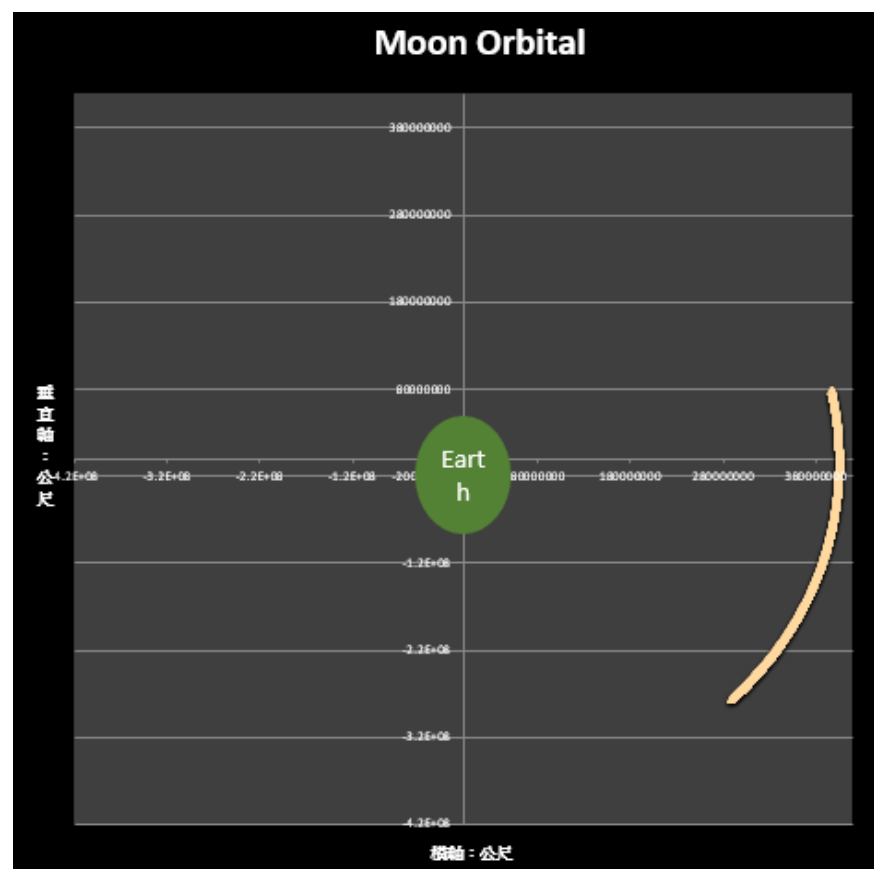
程式一開始就進入 while 迴圈，設定 Loop Until T = 1 (但是 T=0，得手動停止)。

第 7~10 行：為做圖表 7，將工作表 1 的 D 行的 4~204 列 (起初 $x=4$) 為 X 項，工作表 1 的 E 行的 4~204 列 (起初 $x=4$) 為 Y 項。接著不斷地增加 x ($x = x + 100$)，產生動畫效果。

第 12~24 行：一樣是做圖，只是畫其他圖表。

第 26~27 行：timeout(1e-20)，才可讓動畫呈現，否則 x ($x = x + 100$) 增加過快，沒有動畫效果。

6. 結果：



以月球對地球的繞日軌道截圖。

7. 參考資料：

<https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html>

<http://stackoverflow.com/questions/26636127/excel-adding-animation>