HW1

111550143林彥佑

* Implementation

TODO#1 接彈簧

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

AI 產生的內容可能不正確。

照講義的題是先接上Struct Spring，分別以三個不同方向(x, y, z)處理連接避免重複，x, y, z方向的連接程式碼不同之處僅在於iNeighborID，分別用1, particleNumPerEdge, particleumPerFace找連接的particle。

再來Bending Spring的接法與Struct Spring類似，差別只是Bending會跨兩個接，所以在處理時會以2, 2\*particleNumPerEdge, 2\*particleumPerFace找連接的particle。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。



在做Shear Spring時，我選擇把20個不同方向都列出來，並且在處理時增加判斷條件，只有在iParticleID<iNeighborID時才做連接，這樣雖然說會增加判斷，但是最直覺確保連接埠重複的方法。

TODO#2 Spring Force & Damper Force & Internal Force

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。

算彈力套用particle講義p10所提供的公式: 

算Damper Force套用講義p11所提供的公式: 

然後在計算internal force時將這兩個力納入: 一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

AI 產生的內容可能不正確。

TODO#3 處理平面碰撞與contact force

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

AI 產生的內容可能不正確。

在處理斜坡時，我先將法向量寫出，因為是XZ平面並轉-20度，故法向量為(sinf(PI/9), cosf(PI/9), 0)

，平面上一點我選定(0, 0, 0)，並將速度分成垂直平面以及平行平面，再來判斷particle是否與斜坡發生collision: (particle p15)



更新碰撞後的速度，因為斜坡沒速度，所以套用particle p17之公式

接著處理contact force與friction force，參考particle p18提供的contact conditions:

一張含有 文字, 字型, 螢幕擷取畫面, 圖形 的圖片

AI 產生的內容可能不正確。

並在N·f<0時需計算contact force與friction force，套用particle p19提供的公式



一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。

Elevator的話要考慮其本身的速度，故用HW1提供的碰撞公式，而非particle講義中的

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

TODO#4 Integrator

1. ExplicitEuler

一張含有 文字, 電子產品, 螢幕擷取畫面, 陳列 的圖片

AI 產生的內容可能不正確。

利用Explicit Euler處理jelly中每個particle的速度與位置，公式為

用當前位置、速度、加速度，更新delta t後的位置與速度。

sceneID=1時表示為elevator，將elevator當作一個particle，用相同方法更新速度與位置。

1. ImplicitEuler

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 軟體, 作業系統 的圖片

AI 產生的內容可能不正確。

Implicit是要用現在位置與下個時間點的速度來計算下個時間點之位置，故我先將此時間點之位置與速度存入oldPos與oldVel，並用現在位置與速度去計算Force，得出下個時間點的速度與加速度，得出在此時的下個時間點之位置與速度，並clear force。



同理，將elevator視為一個particle同樣進行implicit Euler，但要記的把elevatorCounter更新回去，因為這裡只是為了模擬，並不是要推進Counter。

1. MidpointEuler

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

AI 產生的內容可能不正確。

與implicit類似，但這次要用現在的位置與預測的下一個位置的中點之速度去預估下一個點，公式



所以存完現在位置與速度後，設定位置是中點位置與速度，不再是現在位置與速度，然後計算force得新的速度與加速度與舊位置與速度得出下一個預估之位置與速度，最後clear force。

同樣將elevator計算midpoint Euler，如implicit的方法。

1. RungeKutta

一張含有 文字, 螢幕擷取畫面, 陳列 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

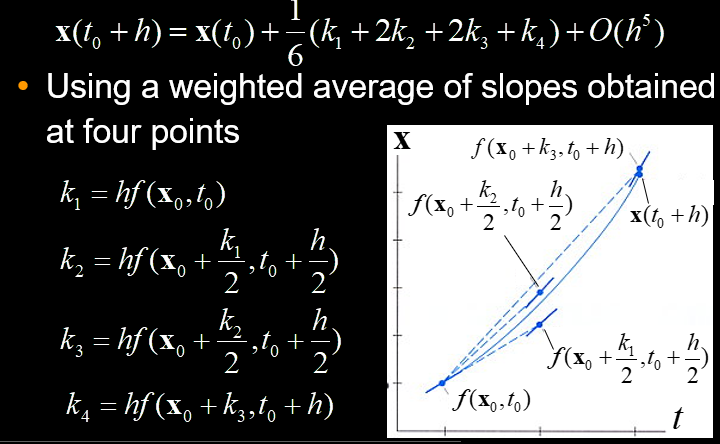
AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面 的圖片

AI 產生的內容可能不正確。

Runge-Kutta 4 使用多階段預測。首先備份原始位置與速度，並依序計算四組微小變化量 k1 ~ k4，其中每組皆透過暫時更新particle狀態並重新計算速度與加速度來取得。最後將這四組權重平均後，計算出下一時間點的位置與速度。

同理，Elevator只是另一個particle，用同樣方法計算出速度與位置，並reset counter。

參考公式:



* Result & Discussion

在變數皆不改變，四種方法經實測後僅有些微差異，implicit Euler在斜坡滾動較少，較早開始滑行，應該為他採用速度為下一個預測之時間點之速度導致，Runge-Kutta因為計算量較大，無論斜坡還是升降梯顯示效果相較其他方法像是慢動作執行，但也最為自然。

然後進行不同變數修改的測試(springCoef, damperCoef, coefResist, coefFriction)

1. springCoef

當springCoef越大，Jelly的彈性力越強，變形時產生的回復力越大，會有更強的反彈力，如implicit在把springCoef從1400調到1500後，他在斜坡上因彈跳多一次翻滾，表示彈力增加，更極端的例子，將springCoef調低至100，則發現jelly下半部被壓扁看不出回復的跡象，第一次碰觸地面也沒有反彈之跡象，調高至2800發現，在彈力上的視覺效果好很多，剛度更好。

1. damperCoef

damperCoef 越大，彈簧的振動就會被抑制越快，Jelly看起來會越黏、重，反彈與彈性效果減弱。

我將damperCoef調低至10，jelly彈跳後的滯空時間明顯比50的多，彈跳效果較好，調高至80後，在elevator場景時第一次接觸平面時幾乎沒有彈跳，故與理論相符。

1. coefResist

在handleCollision中修改coefResist，並用斜坡場景測試，當coefResist為1時為完全彈性碰撞，垂直斜面的速度經碰撞後不改變，彈跳比0.5好，當等於0時，幾乎沒有彈跳只會翻滾，表示只剩下平行斜面的速度。

1. coefFriction

將他調到10.0，在斜坡場景時幾乎不發生彈跳只翻滾，且無滑行，調至0後，明顯增加滑行距離與時間。

但3、4在elevator上均無明顯差別。

最後在穩定測試，當springCoef為50000，然後delta t為0.0012，explicit, implicit都出現錯誤無法執行，midpoint與Runge-Kutta活了下來，更進一步把springCoef調高至100000，delta t為0.0014時，midpoint出現不合理的抖動與緊縮，Runge-Kutta仍穩定。

* Conclusion

在本次作業中，我實作了Explicit Euler、Implicit Euler、Midpoint Euler 與 Runge-Kutta 4，並於斜坡與elevator場景中測試其行為差異，進一步調整模擬參數，如springCoef、damperCoef、coefResist、coefFriction，以觀察系統對這些參數的反應。

模擬結果顯示， Explicit Euler、Implicit Euler雖計算簡單、效率高，但在高剛性參數下極不穩定，容易導致系統崩潰。Midpoint Euler在穩定性與物理真實性之間取得良好平衡，惟在極端參數條件下仍會出現形變與抖動。Runge-Kutta 4 雖然計算量較大、動畫呈現稍慢，但整體效果最為平滑且符合真實物理行為，即使在極端設定下仍能穩定運作。

在物理性質方面，提升springCoef能有效增加彈性與回復力，而增加damperCoef則可快速抑制彈跳與振動。碰撞參數方面，coefResist越大會帶來更強的反彈效果，coefFriction越小則會顯著增加滑動距離。不過，在電梯場景中這兩個參數的影響較不明顯，推測可能因為重力與垂直運動為主而掩蓋其作用。

總結而言，模擬行為對於積分器選擇、delta t以及物理參數的組合非常敏感。本作業使我更深入理解數值方法與物理建模在電腦動畫與特效的關係與重要性。