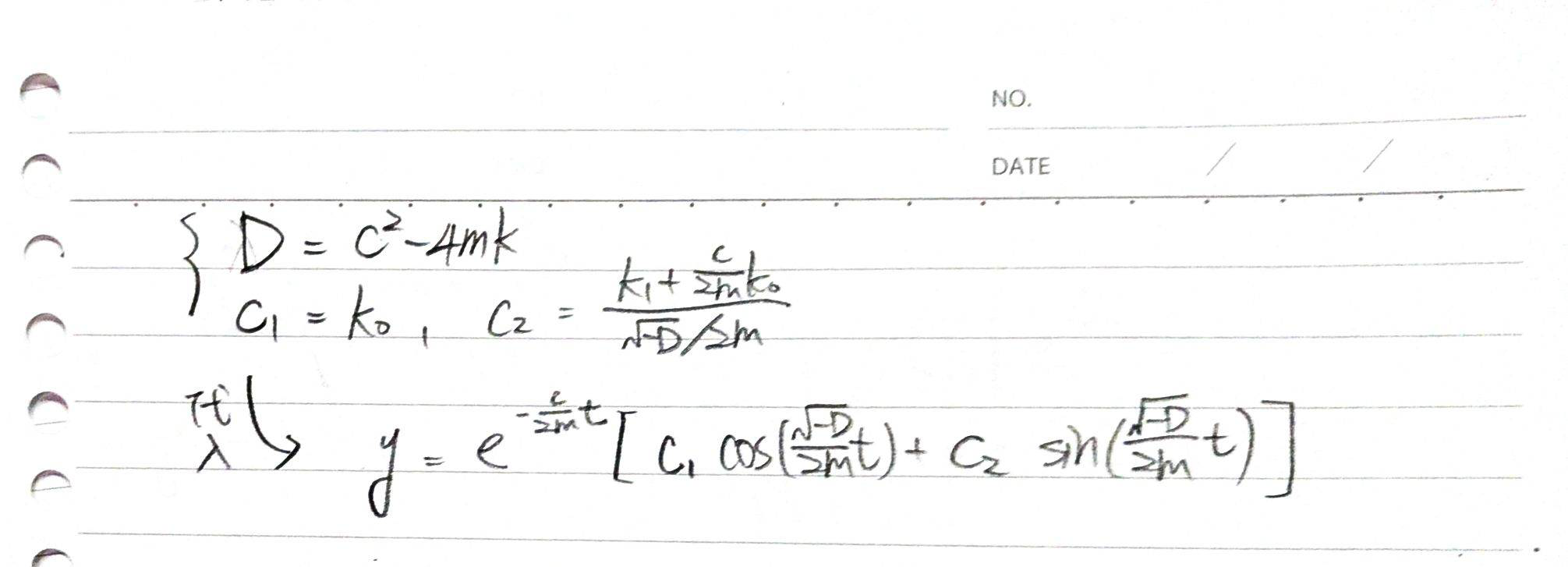
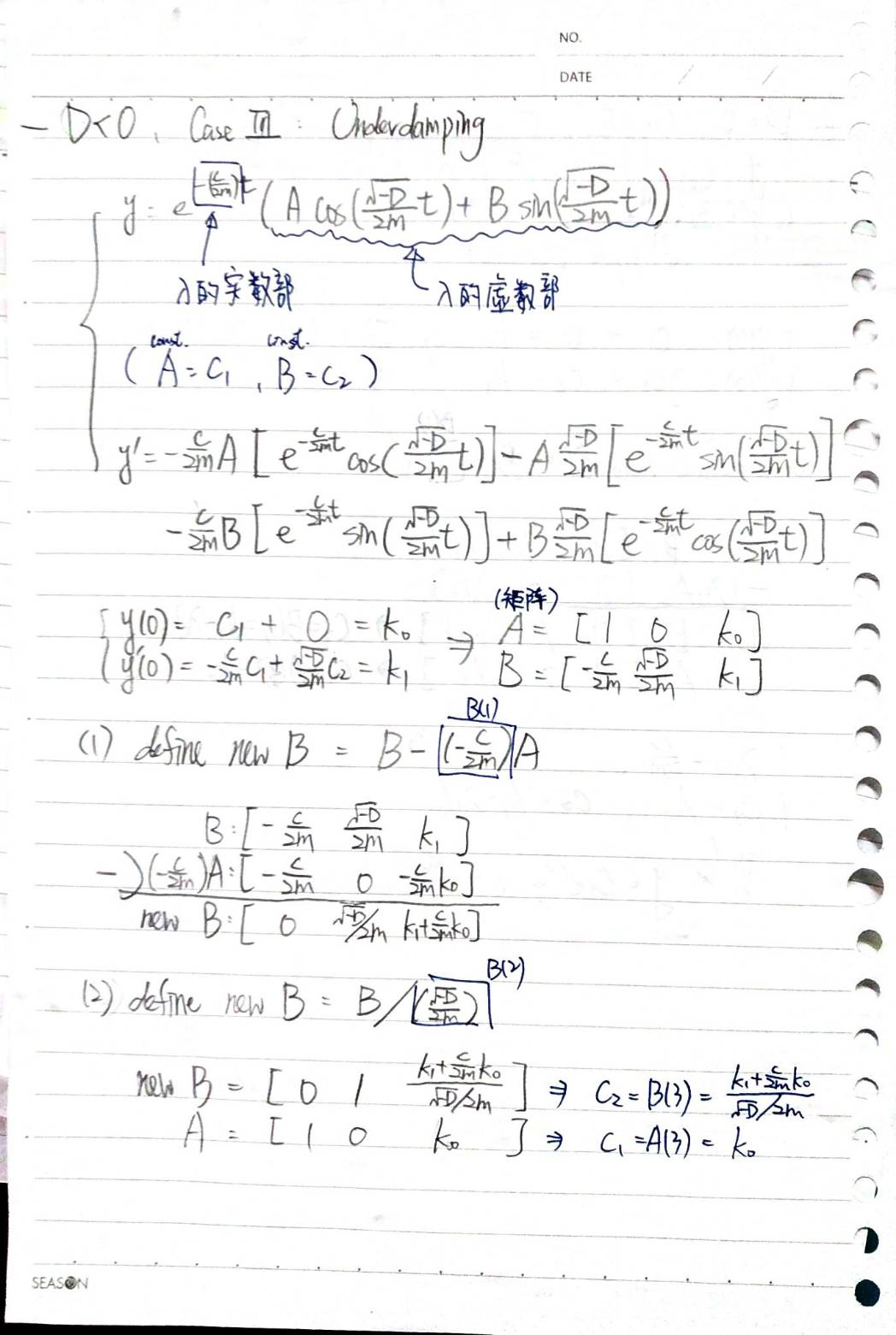
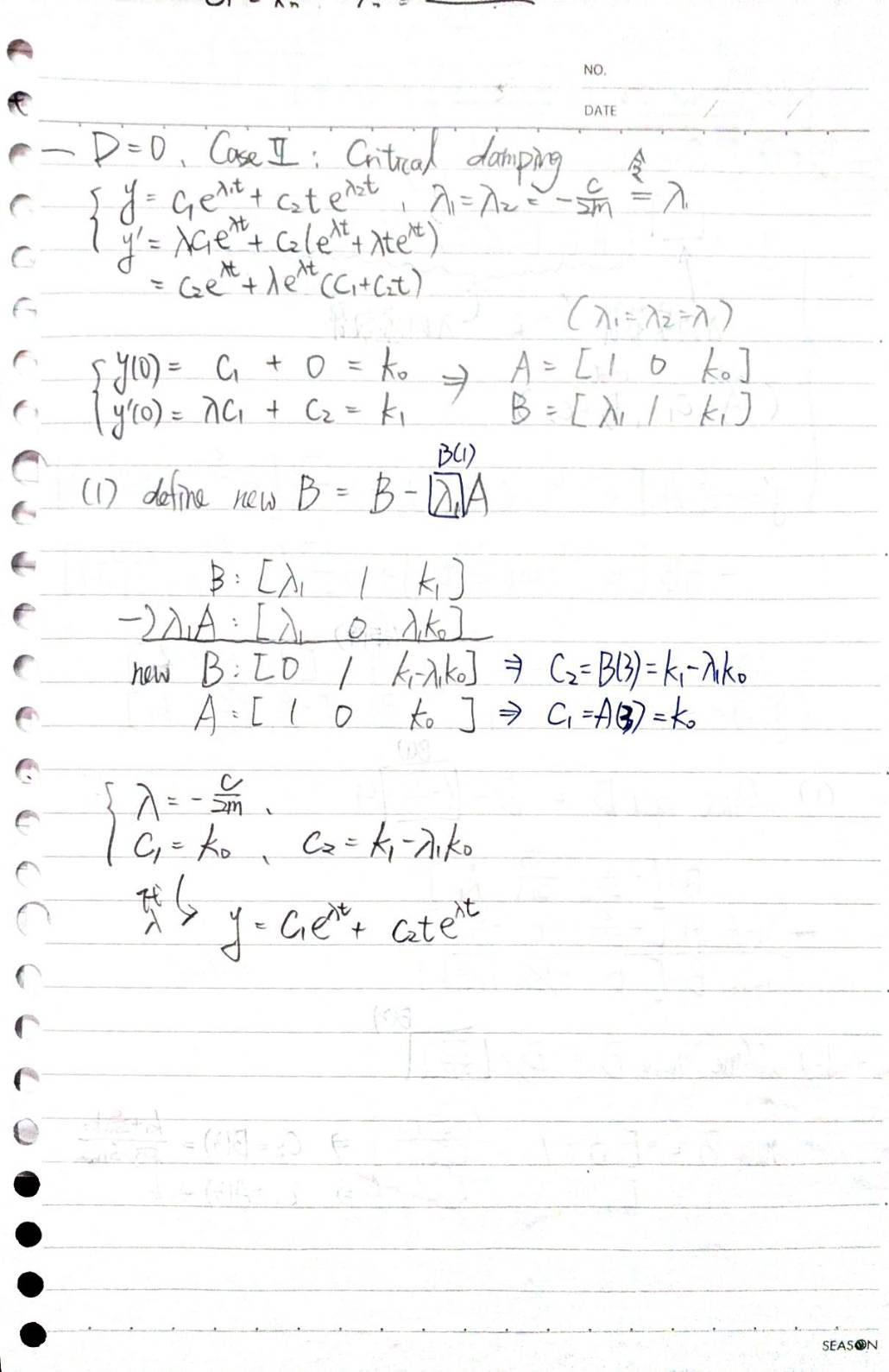
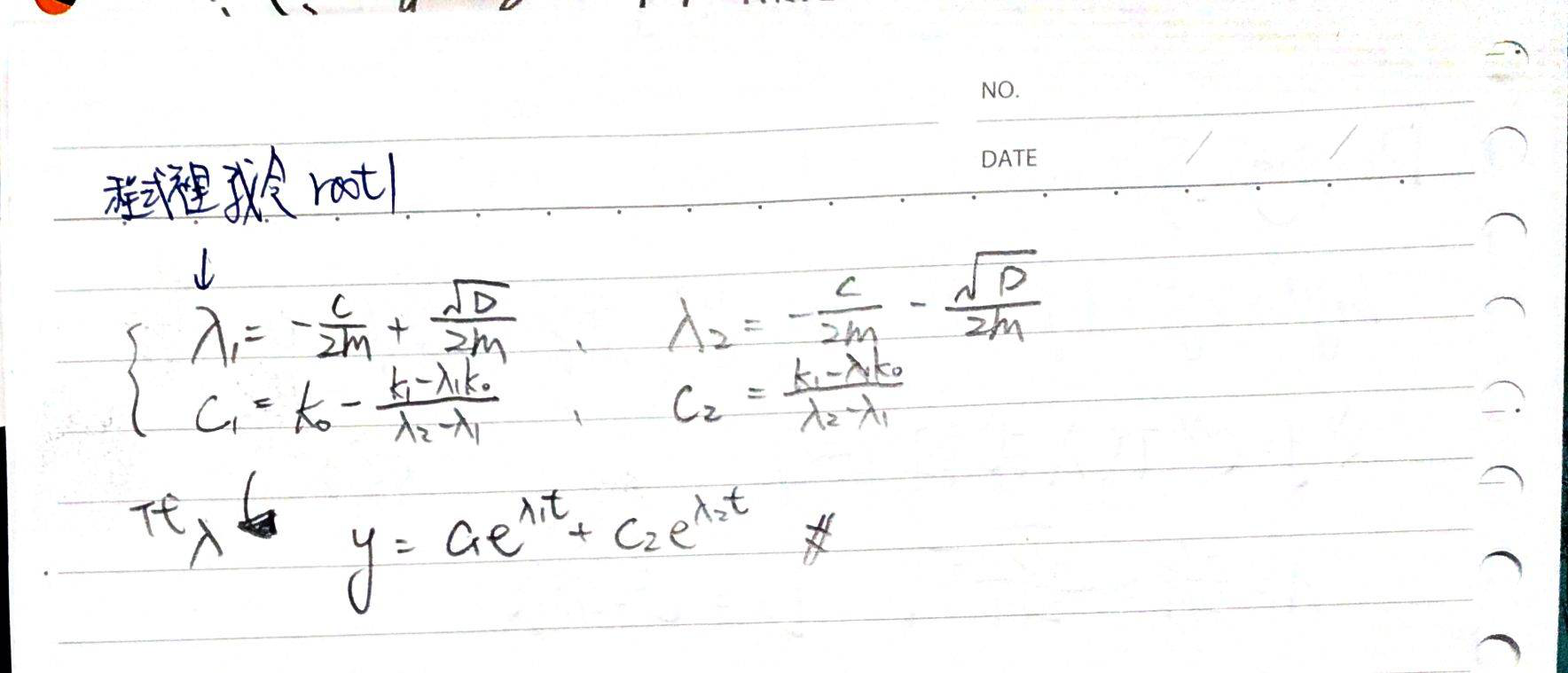
**1. 寫得有點亂:D　　　摻雜廢話的非正式寫法～ （題號亂跳）**

只針對解的形式再推算其λ、C1、C2的值，用程式加減消去法的表示（？）

圖片中的define new A等只是為了方便看懂而已，實際上應該是計算後的結果整個取代原陣列A。



Case I:

Overdamping

用解析解和初始條件得出常數的方程式組，再利用加減消去法（以陣列形式?）得出常數，最後將常數與λ帶回y中

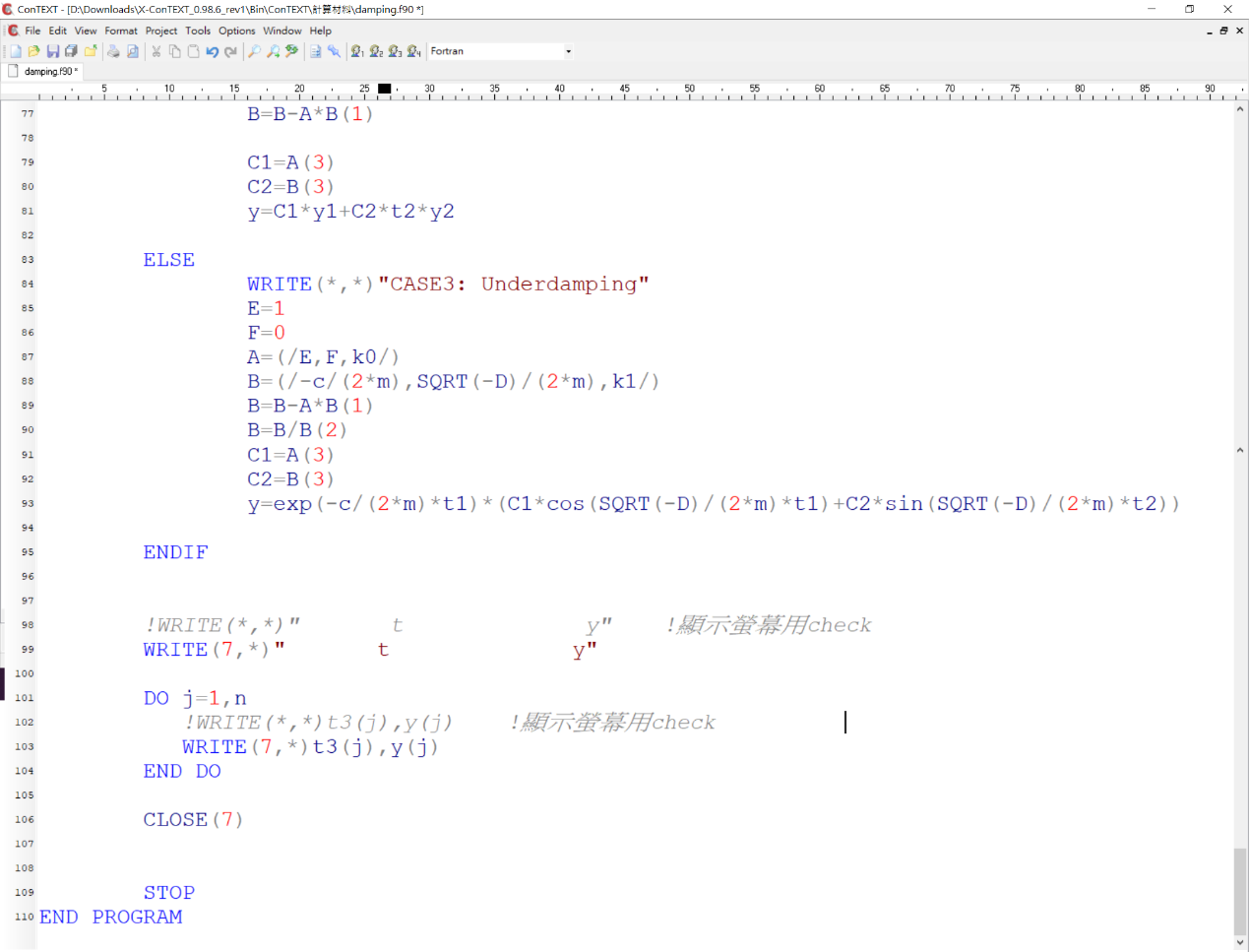
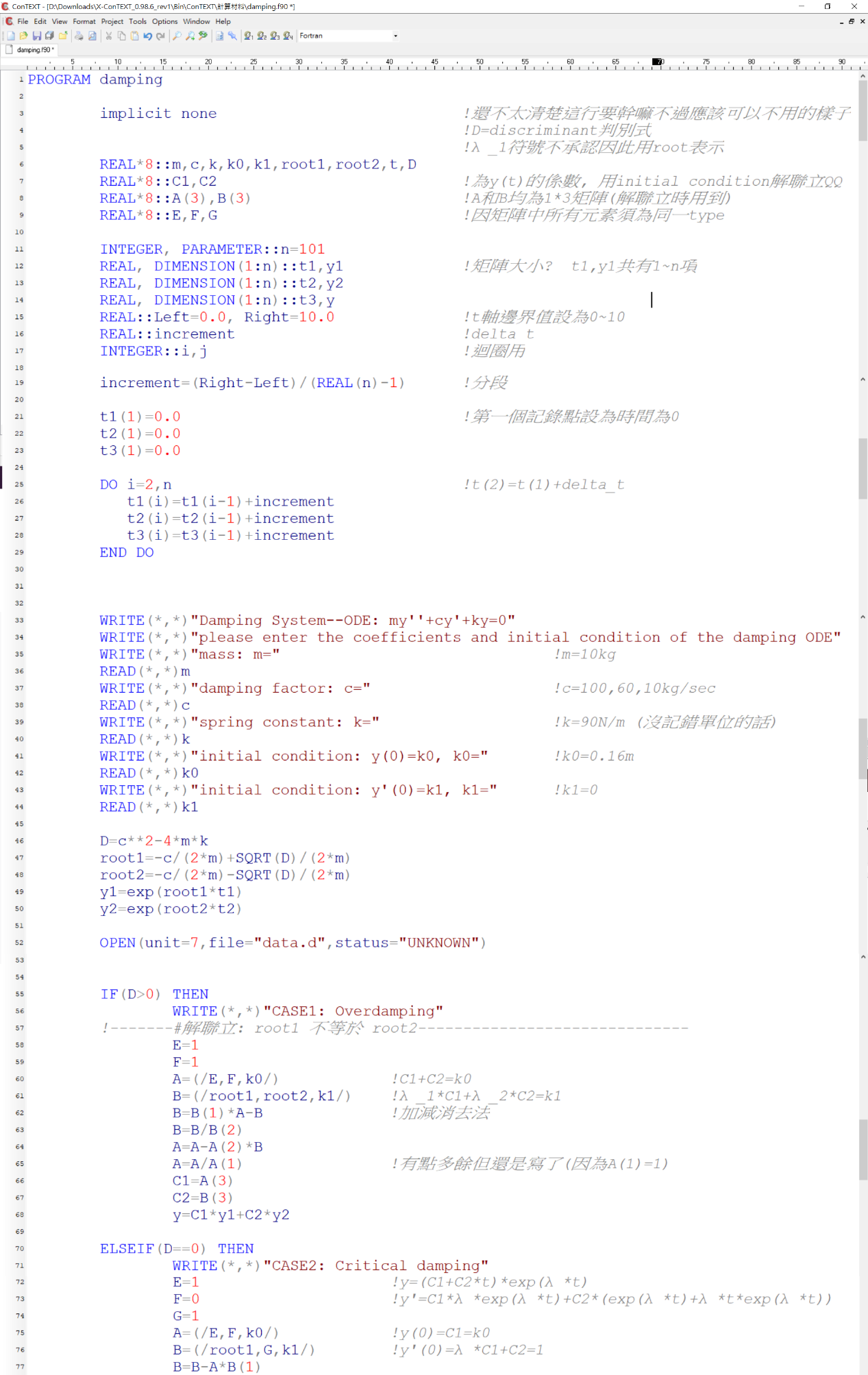
Case II:

Critical damping

Case III:

Underdamping

**2. 程式說明**



Real, dimension(1:n)::t,y1,y2,y

改成這樣更為簡略且方便

⑦

⑥

⑤

④

③

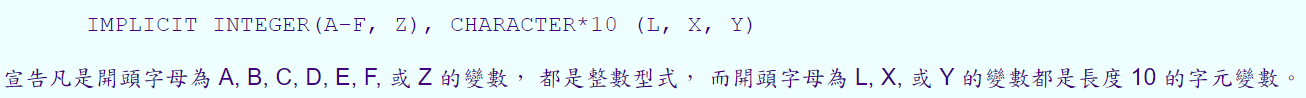
②

①

⑧

① 第3行

Implicit算是一種宣告複數變數型態的方式，網路上的例子：



而implicit none代表不採用implicit預設的型態(I, J, K, L, M,或N開頭的變數，一律視為整數，其他則為實數)，需另外宣告變數型態（視清況需要，不過在這應該可有可無）。

② 第6~9行 宣告變數

**6, 7**:

令所需要的係數，包括質量m、阻尼係數c、彈性係數k、初始條件k0和k1、解λ1和λ2（以root表示）、判別式D。　**（t多令了忘了拿掉）**

**8:**

令1\*3的陣列Ａ和B（如果我理解沒錯的話），給之後解常數方程組用（詳見⑦）。

**9:**

令實數E、F、G。由於後面（第60, 75, 76, 87行）在陣列中的元素須為同一型態，像是同為整數（1, 2, 3），或同為實數。而k0、k1在為實數，因此無法直接定義A=（/1, 1, k0/）（非同一型態），因此令E、F、G再給予數值。**（但後來想想好像可以直接用第19行的方法，強行將整數轉為實數，雖然沒試過就是了）（real(1)行得通吧? 這樣就省了不少行了）**

③ 第11~29行　宣告變數、陣列，但與作圖相關（參見⑧？）

**突然圈太大範圍了:D**

這裡是從這個影片上看的

<https://www.youtube.com/watch?v=kDXv1hxCdoc&list=PLNJwxvvxYU0gmCuK1AuHHEzvL_GcLxGey&index=101&t=0s>

**11:**

宣告參數n=101，數字隨便，但就想挑101（和後續與第19行increment定義有關）（和共有多少紀錄點有關）。

**12~14:**

宣告有從1~n項（共n項）實數元素的陣列t1, y1, t2, y2, t3, y

後來發現其實不用設那麼多個t，單一t就好了。 (我笨蛋)

所以應改為**Real, dimension(1:n):: t, y1, y2, y**

這樣可省去第13, 14, 22, 23, 27, 28行的敘述。　（試過了可行）

**15:**

宣告t的邊界值（作圖範圍），從0~10單位（在此指秒）。

**16:**

Increment中文是增量，當初照影片的方式命名，應指Δt。

所以為宣告Δt。

**17:**

宣告整數ｉ, j，用於迴圈（第25~29、101~104行）。

**19:**

定義increment（Δt）。

假設把從1~9的範圍五等分，則每一等分為(9-1)/5＝1.6，而1.6則為Δt。

而在此程式，範圍為0~10（第15行），分成100等分（每等分0.1秒）。

**Q: 至於為什麼要定義n=101，而分100等分要寫成real(n)-1**

n=101代表共有101項的陣列（第12行），而這101項為從第０秒（t=0）到第１０秒（t=10），總共紀錄101個點（分成一百等分的話，紀錄點有101個）（想像成種樹問題然後頭尾都要算XD）。

real(n)是（強行）將n的型態轉為實數型態。

**21~23:**

定義時間第一個紀錄點為t=0.0的時候，用0.0是因為將型態定為實數（畢竟Δt並不是整數）。

**25~29:**

記錄用的迴圈

i從第2項開始，到第n項為止，執行t(i)=t(i-1)+increment的動作，也就是說t(2)=t(1)+Δt ; t(3)=t(2)+Δt ……。每個記錄點t(i)各差Δt。

（算是時鐘的概念）（可以延伸做計時器？？）

**但覺得多此一舉了，可以直接定義t(i)=t(i-1)+1，這樣就不用定義increment，也不會有類似t(9)=8.000001之類的東西出現（吧）。**

**④ 第33~50行**

**33~44:**

從螢幕上輸入和輸出一些訊息（隨便帶過）。

**46~50:**

一些參數的公式。（請詳閱第一題的圖片）

**⑤ 第52行 開檔案**

打開data.d的檔案並將檔案代號設為7。 （狀態隨便打的）

**其實這行放到第97行也是可，不知為什麼當初把它擺在這麼前面。**

**第104行關檔案（？）**

**⑥ 第55, 70, 83, 95行 IF statement**

在第55~95行範圍間：

若判別式D>0，則執行56~68行

若判別式D=0，則執行71~81行

若判別式D<0，則執行84~93行

**⑦ 第56~68行 在滿足D>0情況下的執行結果**

**58~61:**

**詳見第一題overdamping的圖片**，可見有令陣列A ,B。其中

A= [ 1 1 k0 ]

B= [ λ1 λ2 k1 ] (λ1= root1, λ2= root2)

因此令E=1, F=1。

而A=(/E,F,k0/) 代表A(1)=E, A(2)=F, A(3)=k0。

**62~67:**

**詳見第一題陣列加減消去法**，但程式碼多了一些不必要的步驟，像是第64, 65行的A(1), A(2)數值均為1，所以有乘沒乘、有除沒除都一樣。而第62行定義new B，雖然圖片和程式碼順序調換，但在第63行除以B(2)之後的new B是一樣的。

**關於new A和new B：為了方便理解（是嗎？），將等號左側的Ａ或Ｂ定義為newＡ或new B，但實際上是等號右側的運算結果取代等號左側的值。**

**68:**

在D>0的情況下，y=C1\*y1+C2\*y2 **（解析解）**（定義ｙ值）

**備註：第71~81, 84~93行執行原理如⑦，請以此類推。**

**⑧ 第98~104行　資料輸出**

**98, 99:**

將t和y分別輸出於螢幕和7號檔案(data.d)（第52行）中。（中間空格自己喬）

備註：目前貼上的程式碼將輸出於螢幕的那行變成註解，畢竟只是檢查用的。

**101~104:**

輸出用的迴圈

ｊ從第1項開始，到第n項為止，輸出t(j)和y(j)於螢幕和7號檔案中（第52行）。

t(j)的定義詳見第25~29行。

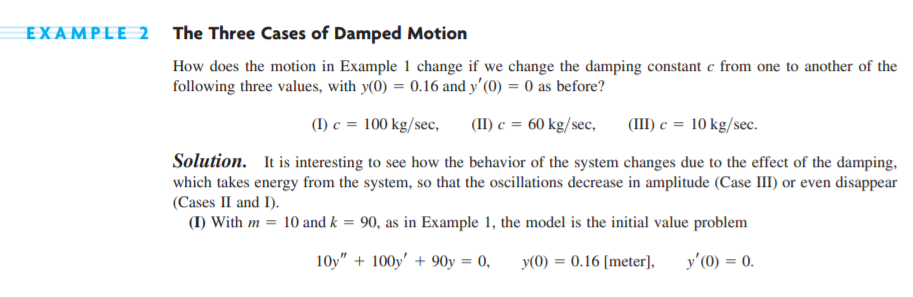
y(j)的定義依IF statement執行結果決定：

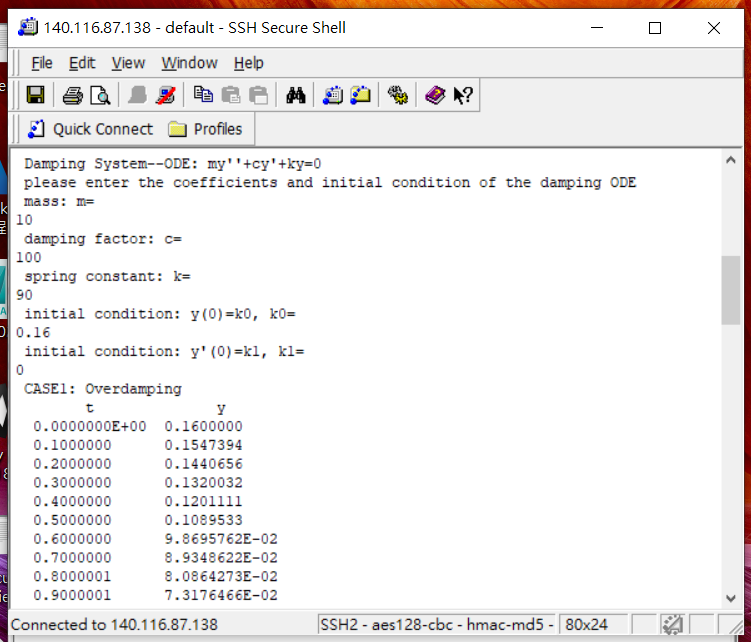
備註：目前貼上的程式碼將輸出於螢幕的那行變成註解，畢竟只是檢查用的。

**備註：上網查到好像可以利用gnuplot這個東西來做圖，可是目前還是不會，所以頂多把資料輸出而已。**

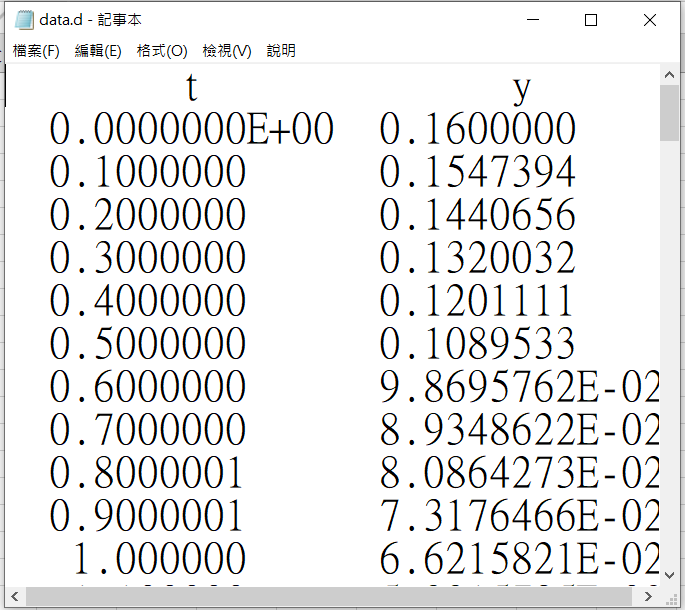
**3. 題目實測與比較**

用工數課本p.67頁example 2做測試

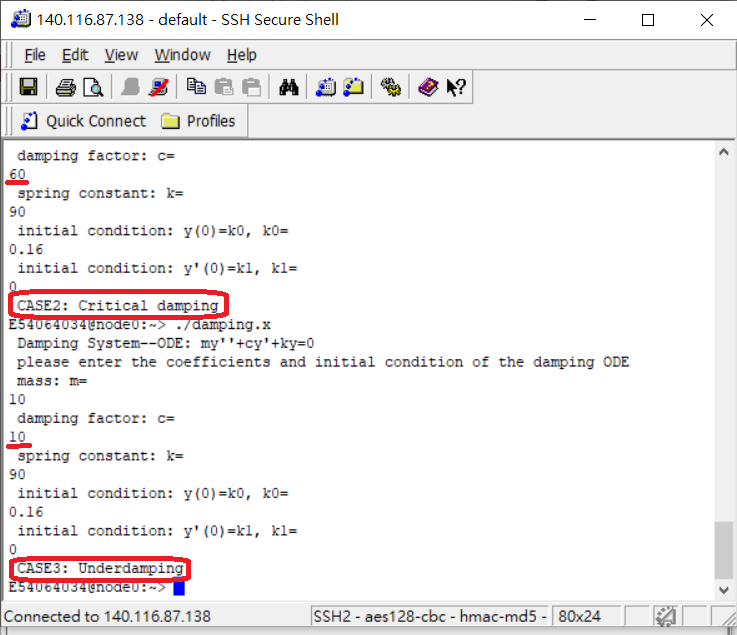




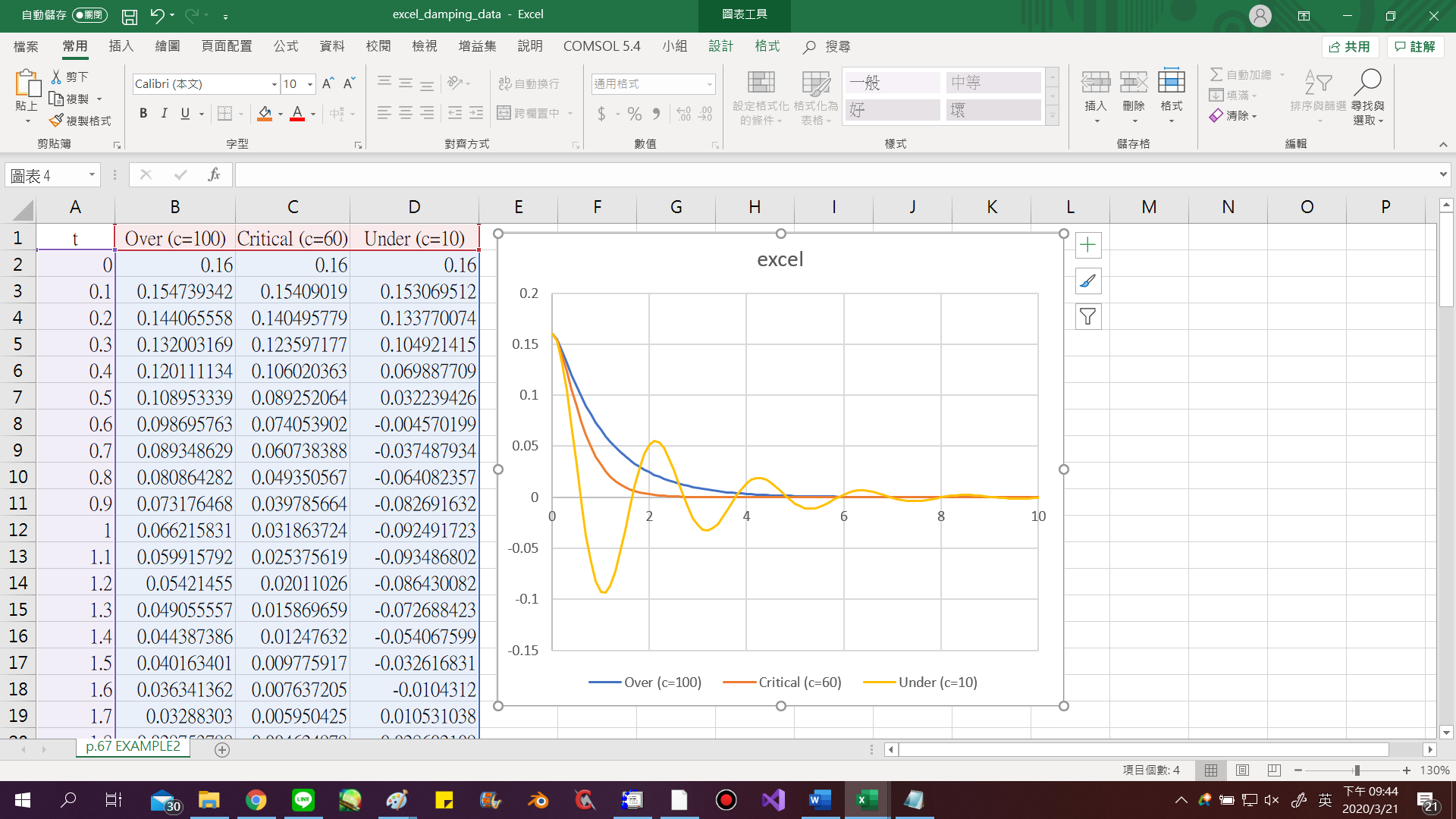
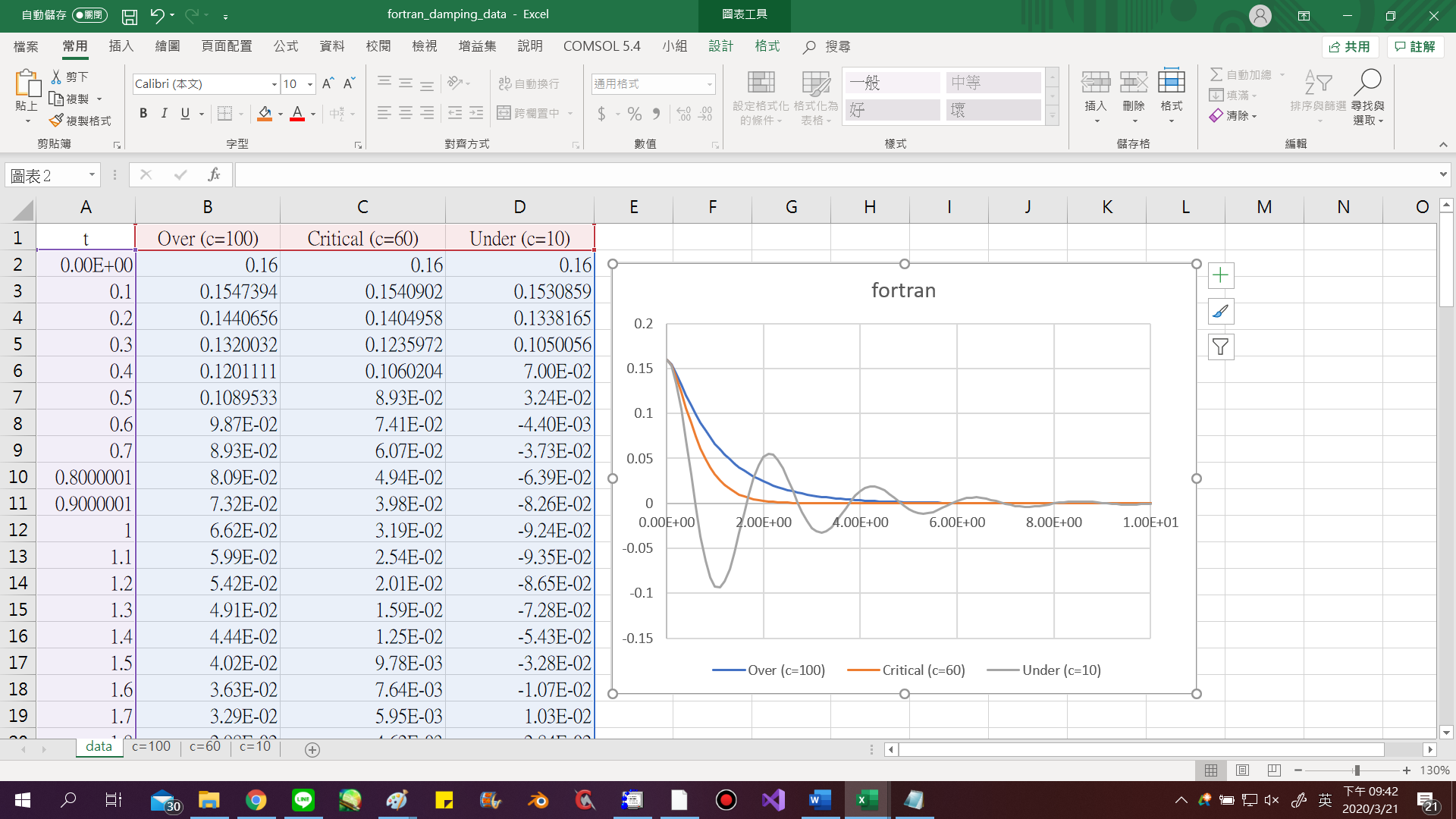
此為CASE I: Overdamping的輸出數據截圖，此時程式碼第98, 102行尚未變成註解。



而data.d中也有剛在螢幕中產生的數據



將第98, 102行變成註解後，進行CASE II和CASE III 測試



CASE I: 藍

CASE II: 橘

CASEIII: 灰

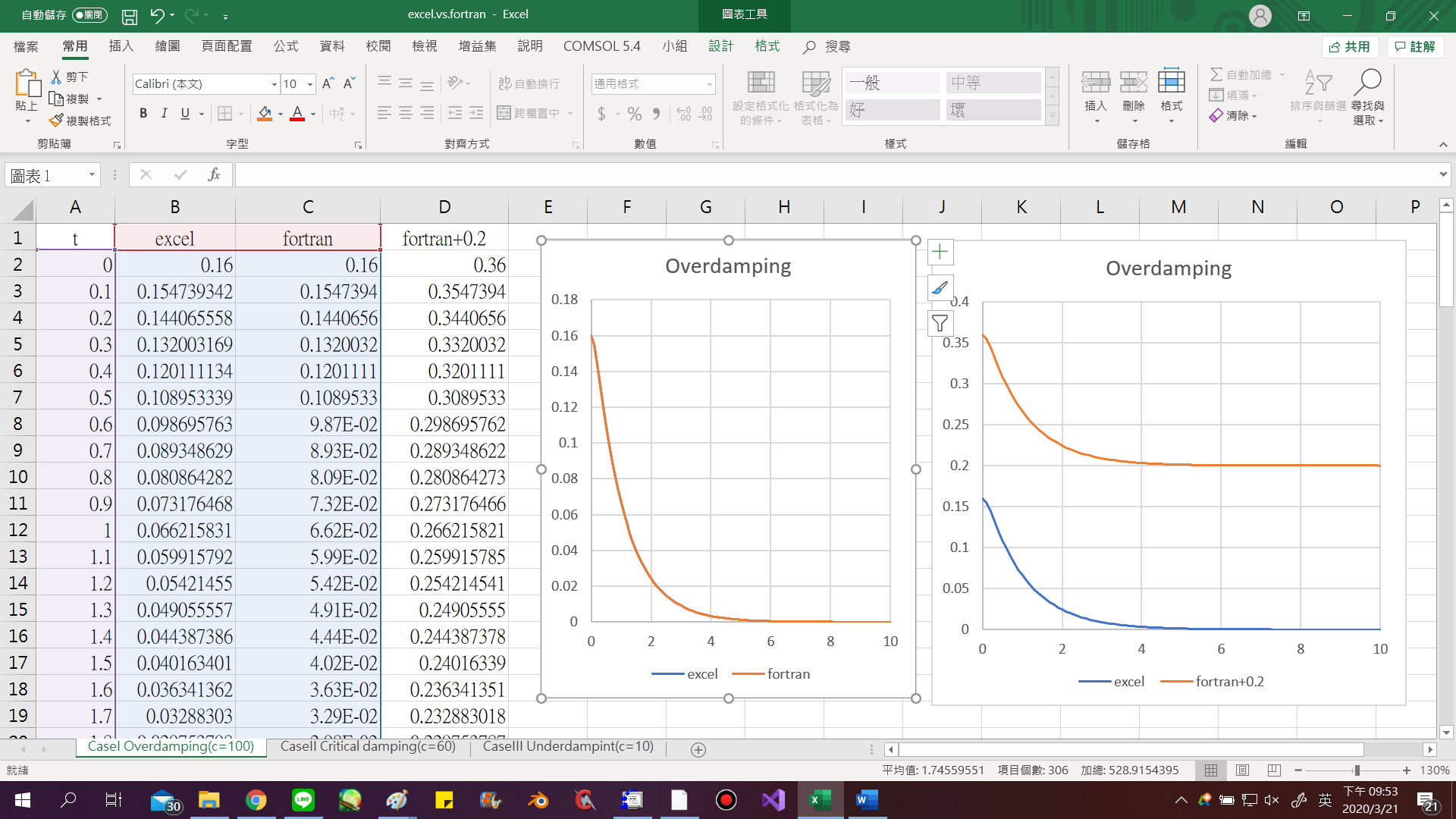
將fortran產生的data貼在excel上並作圖比較三個CASE

CASE I: 藍

CASE II: 橘

CASEIII: 黃

此為excel直接帶入課本上所得出之公式作圖比較三個CASE



抱歉圖超小

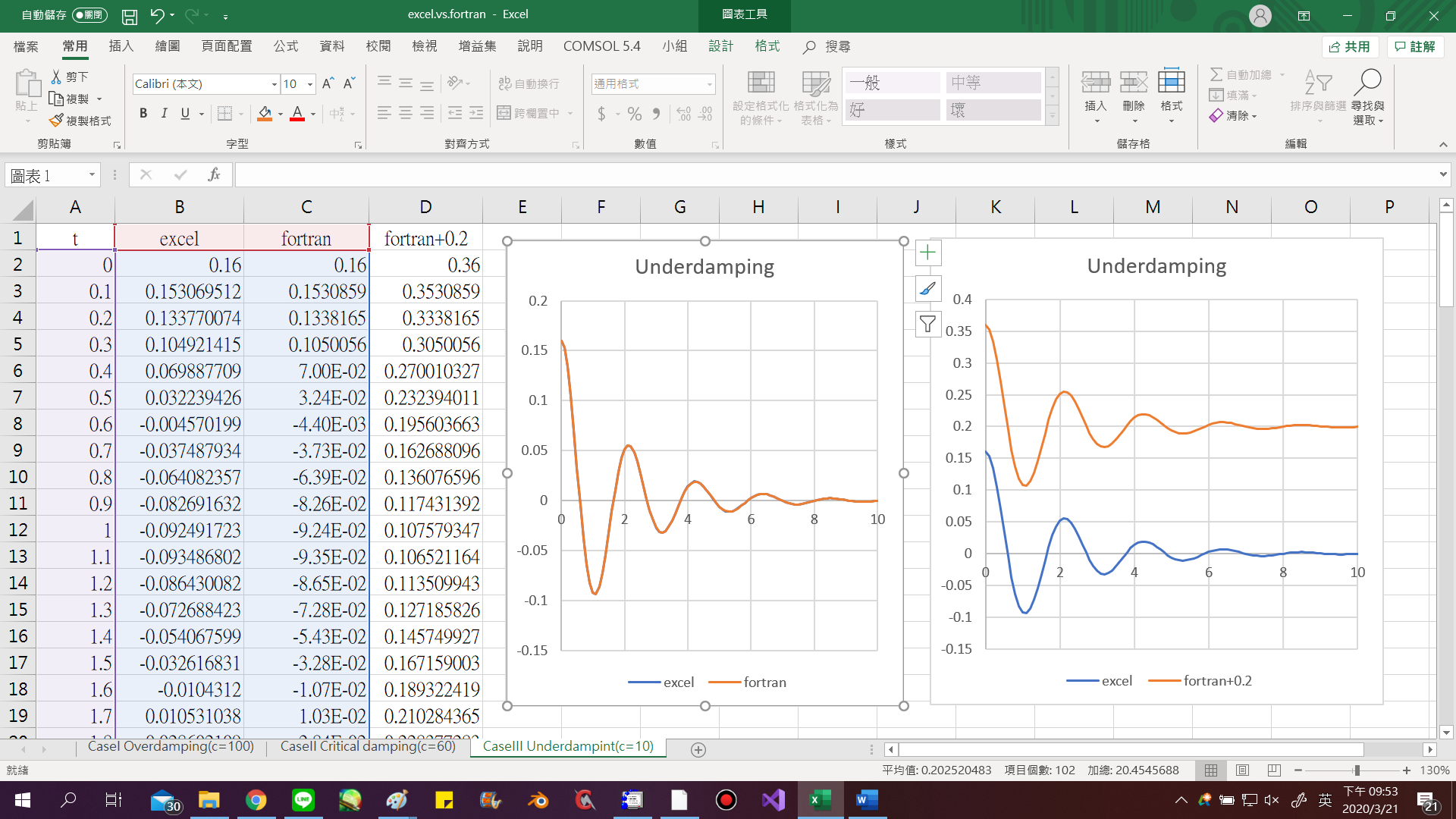
左圖都疊在一起XD

將CASE I, fortran和excel的結果進行比較

右圖橘線為fortran計算值+0.2

將CASE II, fortran和excel的結果進行比較

右圖橘線為fortran計算值+0.2



將CASE III, fortran和excel的結果進行比較

右圖橘線為fortran計算值+0.2

在t=4(sec)後誤差趨近於0

（就肉眼觀察上）

雖說一起做圖的話會疊在一起，但將excel(代入公式值)減去fortran數據後，仍可觀察出兩者有些微的差距（也太小）

橫軸為t

在t=2(sec)後誤差趨近於0

（就肉眼觀察上）

莫名爆衝XDD

和CASE I, CASE II不相同的異類

誤差看起來有週期性，但隨著時間的增加，其鋒值有衰減的情形。好像連誤差都是underdamping XD

而且誤差尺度突然暴增

把三個CASE的誤差疊起來

CASE III真的太誇張了

雖說和Excel畫圖比用肉眼真的看不出來

有可能是程式碼打錯（但個人覺得既然圖都那麼像了所以可能性應該比較低）

有沒有可能是兩者（excel & fortran）定義運算子的方式不同（像是exp()和sin(), cos()的定義有些微的差距所導致的??）

這大概是我寫作業最認真的一次吧？（先不管好不好）

可能之後會每況愈下XD

畢竟花的時間有點太多了

所以會越來越隨便也說不定

和老師先說聲抱歉~~