5 坐標算法二〔教學說明〕

教學目標

用「樸素」的平面幾何與坐標觀念，發展平面直角坐標上的圖形平移與三角形面積算法。

知

知道坐標平面上點坐標的加減相當於點的移動，而平移是整個圖形一起移動。

行

能做三角形的平移，能計算給定三頂點坐標的三角形面積。

識

知道坐標平面上的點坐標，相當於已經知道許多資訊，應當可以利用那些資訊來解決問題。例如：如果知道三頂點坐標，就能算出任意三角形的面積；這是很偉大的一項成就。

主要設計理念

1. 有感於高中坐標幾何課程與國中平面幾何的銜接太少，使得高中的坐標幾何基礎教學顯得不足，所以特別設計了這兩課。
2. 學生在國中階段學習了不少平面幾何的知識與技能，到了高中多半無用武之地。事實上，笛卡兒的坐標幾何是延續平面幾何而發展的，本文希望在學生心中具體地建立坐標幾何承襲平面幾何的感覺，用基本的平面幾何知識（例如三角形面積的不變性）推論坐標幾何的算法。
3. 幾何的「剛性」操作無非就是鏡射、旋轉、平移；前一課已經針對常用的特例，介紹前兩項，本課介紹第三項。這些操作都是伴隨著坐標幾何一起被「發明」的算法，也將成為坐標幾何的最基本概念與最基本操作。
4. 雖然「拼貼法」經過詳細分類之後，也能獲得一般性的三角形面積公式，但是作者刻意不這樣做。原因是「平行線法」提供連結直線方程式的機會，而且「平行線法」呈現較高層次的數學思維範例。但是本課篇幅無法寫完「平行線法」所發展的公式，留到下一課再詳細完成。
5. 這一課示範了一種拆解問題的數學思維方式：只要能計算頂點在原點的三角形面積，就能用平移計算任意位置三角形的面積。雖然這並不是數學專有的思維方式，例如程式設計就很需要這樣的思維，但數學確實提供很多拆解問題的學習機會。

教學備忘

1. 這一課用到「過點 P 平行於直線 L 的直線方程式」，假設學生已經在「正課」學過這項先備知識。
2. 教師同仁一定看得出來，這一課講的「就像」平面向量。確實如此，但是如課文所述，這些知識在 17 世紀中期（在牛頓和萊布尼茲出場之前）就已經發展出來了，如課文標題所示，它們就是「坐標算法」而已，並不是「向量」。讀過這一課，相信老師們也同意：這些算法並不需要向量。事實上，「平面向量」在數學史中並不存在，向量是從「空間向量」開始的。這一課的坐標算法，反而可以作為向量的「前置經驗」。請教師不要在這一課引進向量，讓學生有機會從「樸素」的平面坐標，連結平面幾何，逐步體會坐標的妙用。

3. 請不要將 $|\Delta OPQ|$ 讀作「絕對值…」，應讀作「三角形 OPQ 的面積」。
4. 本課的隨堂練習一再要求同學畫圖，請務必請同學照做，養成畫圖的習慣，肯定是一個受用無窮的好習慣。
5. 教師可使學生閱讀／朗讀「平行線法」小節標題之前的兩段話，它們說明數學「公式」的價值，以及公式皆以「參數」撰寫的特徵。
6. 請注意教學語言：移動一個點，不叫做「平移」；它就是「移動」。當其它點跟著一起保持相對關係地移動時，才叫做「平移」。譬如我們要「平移」一個三角形，使得其中一個頂點被移動到原點。「平移」雖然不是日常用語，但是應該容易望文生義，教師應該可以不必特意解釋它，也不宜正式定義它，只要在說明的時候配合正確的「動作」，讓學生從「例中學」，就可望能夠理解這個動詞的意義了。

教學素養

幾何的「剛性」操作無非就是鏡射、旋轉、平移。第 4、第 5 課在坐標平面上做了小規模的示範，讓學生將直觀上的剛性幾何操作，對照到坐標的變換關係。作者有感於國中時期所學的平面幾何，與高中發展的坐標幾何，連結太弱，並相信這將減損高中數學的整體教學成效，《別冊》嘗試做出一份適當的銜接教材。

我們在坐標幾何的課程裡，直接講方程式圖形的平移：直線、圓，然後講函數圖形的平移：二次函數、三次函數、三角函數，卻沒有正式說明它們的共同基礎：點的平移。但是，如果要講點的平移，一句話也就講完了。所以本課介紹一項基本應用：三角形的平移。而這項應用，將要延續用到下一課。在學生熟悉點的平移之後，我們才有機會讓她／他們明白方程式／函數的平移公式。

課文裡特別為「公式」寫了兩段話。作者希望表達的主要概念有三：

(1) 數學真正的威力就在於抽象化的公式（以及定理），可以用在一切符合定義與前提條件的情況下。所以數學教師不要「害羞」教公式，只要別過度了就好。
(2) 我們常說數學要理解而不要死背，其實說穿了，我們推崇理解的原因還是要幫助記憶。理解之後，不必死背也能長期記憶，或者可以在需要的時候推論出來。
(3) 所謂公式當然要有一般性，所以都是用符號寫出來；換句話說，公式都是「代數式」。這些符號可以根據情境而代入適當的數，但它們並不是未知數的意思（不需要求解），也不是變數的意思（沒有自變的概念，當然也不隨機）。公式裡的代數符號，稱為參數。

總結一下：代表數的符號有三種角色：**參數**、**未知數**、**變數**。

電腦程式語言所使用的符號，程式設計師也習慣稱它們為「變數」，但事實上它們都是「參數」。程式語言必須定義（宣告）每一個參數的「型態」——有些程式語言表面上不需要宣告，但其實都在背後做了內定的自動宣告。所謂「型態」就是我們說的正整數、整數，或者有理數，甚至複數。我們可以在觀念上自由轉換這些型態，但是電腦（硬體）不行，所以必須事先指定型態，或者在運算過程中明確地指定它轉換型態。