

電腦繪圖在波浪表示上之應用

蕭松山* 鄧耀里** 周宗仁***

摘 要

本研究首先利用邊界元素法，對存在任意形狀柱狀體海域的波形及任意形狀港池水面波動等問題，進行數值計算以求解其波浪波形，然後將波浪波形資料利用繪圖程式，繪製波浪波形的自由曲面，再對曲面旋轉、平移並選擇顏色以及光源位置，予以著色與亮暗處理，而繪製出近於實際的波浪圖形於微電腦螢幕上。

由於實際的波浪波形隨著時間連續不斷的變化，為了模擬實際波形的變化，將計算間隔依波週期等分為二十四個時間間隔，分別對每一時間序列繪製波浪波形圖，並由組合語言程式來控制每一畫面的儲存與顯示，使能按照時間序列順序、迅速的於微電腦螢幕上顯示出來，以達到波浪在動態表現的目的。

一、前 言

以往解析波浪波形問題，其研究成果大多需繪製許多圖表，例多：解析海域存在結構物時之波形變化，港池水面波動等問題，其研究結果一般都以等波高圖來表示，如此並不足以表現三維的波浪波形變化特性。近幾年來電腦繪圖技術精進，已有學者將其研究成果，利用電腦及繪圖機繪製三維波浪波形的Wire-Frame圖，即以二維圖形來表現三維的物體。但是Wire-Frame圖形並沒有作隱面處理，在視覺上較不具真實感。

因此，本研究應用一繪圖套裝程式⁽¹⁾，將波浪予以著色，並考慮照射的光源，進行明暗之處理，使得繪製的波浪圖較具有真實感。由於此套裝軟體並無法繪製鄰近結構物邊界的波形曲面，及處理波浪波形動態畫面。所以本研究另行設計程式，處理結構物邊界的波形曲面繪製，與波浪波形的動態畫面，以配合套裝軟體的使用，除了使繪製的畫面能滿足視覺上真實感的要求外，更能使我們清楚的觀察波浪隨時間變化的過程。

* 國立臺灣海洋學院河海工程學系講師

** 國立臺灣海洋學院河海工程學研究所研究生

*** 國立臺灣海洋學院河海工程學系教授

本研究首先應用周(1981)、(1984)解析存在任意形狀柱狀體海域之波浪波形變化⁽²⁾，以及任意形狀港池水面波動⁽³⁾等問題的結果，以無因次週期 $\sigma^2 h/g = 0.5$ 之圓柱、橢圓柱，以及無因次週期 $\sigma^2 h/g = 1.2$ 之矩形、圓形港池為計算例進行數值計算，再利用本學系之 NEC-9801 型微電腦將計算所得海域內各點波浪波形資料繪製成彩色的三維波浪波形圖，並進行動態畫面的處理。

二、MORE 繪圖軟體使用說明及追加

2-1 繪圖軟體使用說明

本研究應用由日本 ASCII 公司開發出的繪圖套裝軟體⁽¹⁾，來繪製彩色的波浪波形圖。此繪圖套裝軟體主要為 MORE 系統，此系統的構成分為四大子系統：Modeling、Motion-Design、Rendering 及 Recording（如圖 2-1 所示）。而各子系統主要用途敘述如下：

1. Modeling 子系統：

將物體幾何形狀資料輸入，並配合對物體的視野（視點的位置、視線的方向及視野角度），表面顏色及光源的選定或再輸入物體運動變化資料，由 Model.c 程式（參考文獻 1）可製作成供繪製物體模型圖的多邊形資料檔（Polygon file），由此多邊形資料檔即可繪出物體的 Wire-Frame 圖。

2. Motion-Design 子系統：

將物體運動有關的資料輸入，經由 Motion.c⁽¹⁾ 程式，可建立描繪物體運動變化的資料檔，配合 Modeling 子系統可將物體運動變化的過程，同時繪製於一個畫面上。而為了模擬連續不斷變化的波浪波形，需對每一時間序列繪製波浪波形圖，再使其按順序顯示於螢幕上，因此，本研究未使用此子系統。

3. Rendering 子系統：

將 Modeling 子系統所得之多邊形資料檔（Polygon file），輸入 Render.c⁽¹⁾ 程式中，由它進行著色、亮暗與隱面的處理，製作成物體影像資料檔（Image file），將此影像資料檔經由系統中之 dither.c 程式⁽¹⁾，使其轉換為彩色畫面繪製於 NEC-9801 型微電腦螢幕上。

4. Recording 子系統：

此子系統具影像處理之功能，並應用閉路或 16mm 攝影機攝取畫面，以製作連續動態畫面。但本研究另外自行開發出 Store-Display 子系統直接在電腦螢幕上製作出連續畫面。

2-2 繪圖軟體的追加

在使用本軟體之 Modeling 子系統之前，必需先求得物體的幾何形狀資料，以做為輸入用之模型資料檔（Model file）的一部份。於本研究即為先求得波浪波形的自由曲面，由於應用於實際海洋波浪波形繪圖時，一般均於較大範圍之海域解析計算波浪波形，若計算網格點分割的相當密集，則會使得繪製的波形曲面截片間的接續相當的平滑連續，但是必需耗費大量的數值計算時間及繪圖時間，因此，計算網格點間距離不可能過於接近，然而為了使波形曲面截片間能平滑連續的接合，本研究採

用雙三次B曲線規，來繪製波浪波形的自由曲面。如圖 2-2 所示，取16個網格點為控制點來繪製曲面，而為了使曲面間更平滑連續，僅取中間九分之一大小的曲面截片。有關雙三次B曲線規的定義式及程式 spline.c 等，請參考文獻 1。

如圖 2-3 所示，是以圓柱附近波形曲面計算為例，實線部份為分割網路，交點為控制點。圖 2-4 中實線部份為由圖 2-3 這些控制點，經 spline.c 所求得的曲面截片群，但是緊鄰結構物邊界之曲片並無法求得，而圖 2-4 中虛線部份曲片則為吾人所需製作的邊界曲片群，製作方式為首先讀取經由 spline.c 所求得的曲面群靠近結構物邊的端點值，再與結構物邊界上相對應的節點值組合為控制點，取每四個控制點來繪製鄰近結構物邊界面片。最後將以上兩部份之曲面資料結合為一波浪波形自由曲面資料檔 (Polygon file)，以作為 Modeling 子系統之輸入。

為了在微電腦螢幕模擬連續不斷變化的波浪波形，本研究另行規劃了顯示動態畫面的 Store-Display 子系統。此子系統包括了兩部份，第一部份為利用以組合語言撰寫之 store.exe 程式 (如附錄一)；將每一時間序列的波浪波形彩色畫面分別儲存於微電腦的記憶體中；第二部份則利用 display.exe 程式 (如附錄二)，控制儲存於記憶體中之彩色波浪波形畫面，按照時間序列順序的顯示於微電腦螢幕上，即可完成動態畫面的顯示。

三、波浪波形的三維圖形繪製

3-1 結構物圖形的繪製

於繪製結構物圖形時，一般將結構物斷面邊界以 n 個節點來分割，連接相鄰之節點而以 n 邊形來表現此結構物斷面，為了使繪製的斷面與原結構物斷面十分近似，分割節點數目需足夠多，而且需分佈均勻。例如圓形斷面之繪製，由於圓形為數學函數圖形，可以參數型或非參數型計算分割節點座標，如 (3-1) 式為圓形參數型之表示式：

$$\left. \begin{aligned} x &= \cos(t) \\ y &= \sin(t) \end{aligned} \right\} \dots\dots\dots (3-1)$$

利用上式計算，以26邊形來繪製之單位圓如圖 (3-1)。另如 (3-2) 式為圓形非參數型之表示式：

$$y = \sqrt{1 - x^2} \quad |x| \leq 1 \dots\dots\dots (3-2)$$

利用上式計算而以26邊形來繪製之單位圓如圖 (3-2)。比較兩圖形可知以參數型表示時，其計算節點分佈較均勻，所繪製之圖形亦較近似原斷面曲線，較具真實感。

由於本研究中所繪製之圓柱、橢圓柱及圓形港池等結構物斷面曲線均屬數學函數型，因此均以參數型式來繪製其頂面與底面。另由於所繪製的結構物為直立體，所以於結構物體身部份之繪製，僅需取頂面二節點與相對應位置之底面上二節點，以此四個控制點來繪製矩形曲面截片，將所有矩形曲面截片連結即可完成結構物體身模型之繪製，並將繪製之結構物圖形資料儲存於結構物模型檔中。

如圓柱之繪製，一般將座標原點置於圓心來繪製單位圓柱，如圖（3-3），然後視解析問題柱體的尺寸及與海域中波形座標的配合等，利用轉換矩陣將其縮小、放大或平移等，以得到各種不同尺寸的圓柱體圖形。

3-2 波浪波形自由曲面之製作

本研究首先利用邊界元素法，對有關的波浪問題進行數值計算（參考第四章），將欲解析海域分割為若干個網格點，輸入網格點座標及無因次週期 $\sigma^2 h/g$ 等有關資料，可計算求得解析海域內任一點 (x, y) 之波高值 z ，亦即可求得含 (x, y, z) 之波浪波形資料。然後取相鄰的16個網格點資料，作為雙三次B曲線規曲面截片計算之控制點，利用自行設計之 menu.c 程式（如附錄三），將整個海域的所有網格點，每16個相鄰點組合在一起，建立計算波浪波形自由曲面之控制點資料檔（命名為 *.ctl），再利用程式 spline.c 將所建立之 *.ctl 檔輸入，即可求得海域內除了鄰近結構物邊界面截片外的波浪波形曲面資料，並將其儲存於一曲面資料檔中（命名為 *.pol）。

對於鄰近結構物邊界面之曲面截片，則利用 2-2 節所述的方法，利用 menu.c 程式讀取與邊界面截片連結線上波形曲面截片的端點值，並與結構物邊面上節點值，每四個點為控制點，建立邊界面控制點資料檔，再由控制點計算出邊界面截片資料檔。最後將所有的曲面截片資料檔（*.pol）結合，組成整個海域之波浪波形曲面資料檔。

3-3 波浪波形的三維圖形繪製

將以上二節所求得之結構物模型資料檔與整個海域之波形曲面資料檔，輸入 model.exe 程式中即可繪製整個海域之 Wire-Frame 圖形。為了求得更具真實感的波浪波形圖，需考慮視野、顏色及光源等因素，而這些因素的控制乃是將對結構物或自由曲面的旋轉角度、平移量、顏色、光源位置的選定書寫於一模型檔 (*.mod) 中，則 model.exe⁽¹⁾ 程式執行時會由此模型檔中讀取上述這些因素及所應含入的模型資料檔。然後執行 render.exe⁽¹⁾ 程式，進行著色、亮暗、及隱藏面處理，並經由 dither.exe⁽¹⁾ 程式將所求之彩色畫面資料繪於微電腦螢幕上。最後依據螢幕上圖形表現的視覺真實度好壞，修改模型檔 (*.mod) 中各控制因素，經過多次修改以求得一最佳視覺真實度之畫面。

3-4 動態畫面之處理

由於實際波浪波形會隨著時間連續變化，對於進行波而言，波形的變化為週期性，本研究將每一計算週期分割為24個時間間隔進行數值計算，求得各個時間序列之波形資料，再將這些波形資料利用以上各節所述方法步驟，分別繪製波浪波形圖。

然而欲於螢幕上模擬實際波浪波形變化情形，需要使每一序列時間的波形圖按順序的顯現在微電腦螢幕上，因此利用（2-2）節所述之 Store-Display 子系統，執行 store.exe 程式，使得繪製完成之每一彩色波浪波形畫面，儲存於記憶體中之不同 address，再執行 display.exe 程式，控制將記憶體中的彩色畫面按順序的於微電腦螢幕上顯示出來，如此即可達到動態畫面的表現。

四、相關的波浪理論

在一定水深之海域，若海域內之波運動為微小振幅波運動而持有速度 $\phi(x, y, z; t) = g \zeta_0 / \sigma \cdot \phi(x, y, z) \cdot \exp(-i\sigma t)$ ，其中 σ 為波之週頻率， ζ_0 表示波之振幅， t 及 g 分別表示時間及重力加速度，其勢函數 $\phi(x, y, z)$ 應滿足 Laplace 方程式：

$$\partial^2 \phi / \partial x^2 + \partial^2 \phi / \partial y^2 + \partial^2 \phi / \partial z^2 = 0 \dots\dots\dots (4-1)$$

4-1 任意形狀柱狀體存在之海域波形解析

如圖(4-1)在一定水深 h 之海域，存在一任意形狀斷面之直立柱體，在水面上取座標原點 O ，水平面上取 x 、 y 軸，垂直向上為 z 軸。由於在水面及不透水海底之邊界條件分別為

$$\left. \begin{aligned} \partial \phi / \partial z = \sigma^2 / g \cdot \phi \quad , \quad z = 0 \\ \partial \phi / \partial z = 0 \quad , \quad z = -h \end{aligned} \right\} \dots\dots\dots (4-2)$$

故滿足(4-1)式之解可以下式表示

$$\phi(x, y, z) = [f_0(x, y) + f^*(x, y)] \cdot \cosh k(z+h) / \cosh kh \quad (4-3)$$

其中 $f_0(x, y)$ ， $f^*(x, y)$ 分別表示入射波及繞射波，而 k 為下列方程式之根。

$$kh \cdot \tanh kh = \sigma^2 h / g \quad \dots\dots\dots (4-4)$$

假設入射波為與 x 軸 ω 角度入射之餘弦波，其水面波形可以下式表示

$$\zeta_i(x, y; t) = \zeta_0 \cdot \cos[k(x \cos \omega + y \sin \omega) + \sigma t] \quad \dots\dots\dots (4-5)$$

則入射波之 $f_0(x, y)$ 應為

$$f_0(x, y) = -i \exp[-ik(x \cos \omega + y \sin \omega)] \quad \dots\dots\dots (4-6)$$

又依(4-1)式 $f^*(x, y)$ 應滿足下列 Helmholtz 方程式

$$\partial^2 f^* / \partial x^2 + \partial^2 f^* / \partial y^2 + k^2 f^* = 0 \quad \dots\dots\dots (4-7)$$

由於柱體為不透水性，故在柱體面上法線方向流速為零，即

$$\partial \phi / \partial n = 0 \quad \dots\dots\dots (4-8)$$

將(4-3)式代入上式得

$$\partial f_0 / \partial n = -\partial f^* / \partial n \quad \dots\dots\dots (4-9)$$

再利用文獻 2、4 中所述之邊界元素法進行數值計算，可求得海域內任意點之 $f^*(x, y)$ ，則任意點之勢函數 $\phi(x, y, z)$ 即可由(4-3)式求得。另由微小振幅波理論，海域內任意點、任意時刻之波形為

$$\zeta(x, y; t) = i \zeta_0 \phi(x, y, 0) \exp(-i\sigma t) \quad \dots\dots\dots (4-10)$$

因此，將所求得之勢函數 $\phi(x, y, 0)$ 值代入上式，即可求得海域內之波形資料。

4-2 任意形狀港池水面波動解析

如圖 4-2 所示為一任意形狀等水深港池，在靜水面上取座標原點 O ，水平面內取 x, y 軸，垂直

向上為 z 軸。現將該水域分成外海領域及數個港內領域（本文以二個為例），並假設 \overline{AB} 、 \overline{EF} 兩段海岸線為完全消波之自然海灘，其餘岸壁則為完全反射之不透水岸壁。

由於水深一定，可將各領域之勢函數以下列形式表示⁽⁴⁾：

$$\left. \begin{aligned} \phi_1(x, y, z) &= [f_0(x, y) + f_1(x, y)] \frac{\cosh k(z+h)}{\cosh kh} \\ \phi_2(x, y, z) &= f_2(x, y) \frac{\cosh k(z+h)}{\cosh kh} \\ \phi_3(x, y, z) &= f_3(x, y) \frac{\cosh k(z+h)}{\cosh kh} \end{aligned} \right\} \dots\dots (4-11)$$

其中 k 為 $kh \cdot \tanh kh = \sigma^2 h/g$ 之根， f_0 及 f_1 分別為外海領域中之入射波及反射波， f_i ($i=2, 3$) 為港內領域 II、III 中之繞射波

若入射波為與 x 軸成 ω 角入射，振幅為 ζ_0 之餘弦波，其水面波形為

$$\zeta_1(x, y; t) = \zeta_0 \cdot \cos [k(x \cos \omega + y \sin \omega) + \sigma t] \dots\dots\dots (4-12)$$

則入射波之 $f_0(x, y)$ 表為

$$f_0(x, y) = -i \exp [-ik(x \cos \omega + y \sin \omega)] \dots\dots\dots (4-13)$$

由 (4-1) 式， f_i ($i=1, 2, 3$) 應分別滿足下列之 Helmholtz 方程式

$$\partial^2 f_i / \partial x^2 + \partial^2 f_i / \partial y^2 + k^2 f_i = 0 \dots\dots\dots (4-14)$$

而有關的邊界條件，除了如 (4-2) 式之自由水面及不透水海底之邊界條件外，尚有

(1) 不透水岸壁之法線方向流速為零（除了假想邊界線 B_1 、 B_2 外之其他邊界上）即

$$\partial \phi / \partial n = 0 \quad i = 1, 2, 3 \dots\dots\dots (4-15)$$

(2) 外海領域 I 與港內領域 II 之假想邊界線 B_1 上，流體之質量流束與能量流束須連續即

$$\left. \begin{aligned} \partial \phi_1 / \partial n &= \partial \phi_2 / \partial n \\ \phi_1 &= \phi_2 \end{aligned} \right\} \dots\dots\dots (4-16)$$

(3) 港內領域 II 與港內領域 III 之假想邊界線 B_2 上，流體之質量流束與能量流束須連續即

$$\left. \begin{aligned} \partial \phi_2 / \partial n &= \partial \phi_3 / \partial n \\ \phi_2 &= \phi_3 \end{aligned} \right\} \dots\dots\dots (4-17)$$

由以上各式及利用文獻 3、4 所述之邊界元素法進行數值計算，可求得各領域內任意點之勢函數 $\phi_i(x, y, z)$ 值。而海域內任意點之波形 $\zeta(x, y; t)$ 如 (4-10) 式所示，因此將所求得之勢函數 ϕ 值代入，即可求得任意形狀港池水面波形資料。

五、應用例與討論

5-1 應用例

本研究分別繪製無因次週期 $\sigma^2 h/g = 0.5$ 的海域中，存在半徑為 $1.6h$ (h 表水深) 之圓柱與長軸 $1.6h$ 、短軸 $0.8h$ 之橢圓柱的波浪波形變化，以及無因次週期 $\sigma^2 h/g = 1.2$ ，港口寬為 $10h$ 、港池長為 $20h$ 之矩形港池與港口寬為 $5h$ 半徑為 $5h$ 之圓形港池的水面波形，並分別將週期分割為 24 個時間間隔，繪製不同時間序列的波浪波形圖，並進行動態畫面的處理。圖 5-1 表示以上繪製波浪波形圖步驟之流程圖。

另利用本學系之 NEC PR-801 型彩色印表機，直接將螢幕上的彩色波浪波形圖印製下來，如圖 5-2 ~ 圖 5-5 為 $\sigma^2 h/g = 0.5$ 、 $\sigma t = 0^\circ$ 、 30° 之圓柱、橢圓柱存在海域之波浪波形圖。而圖 5-6 ~ 圖 5-9 為 $\sigma^2 h/g = 1.2$ 、 $\sigma t = 0^\circ$ 、 30° 之矩形與圓形港池水面波形圖。

5-2 討論

經由以上的研究，已可繪製在視覺上具真實感的波浪波形圖，又由於動態畫面的處理，使我們對波浪週結構物或港池水面波動等波形變化的歷時，可以清楚的觀察。因此，對於相關的波浪問題的研究，可應用本研究繪圖的方法，來繪製相關的波浪波形圖，以表現其研究成果。例如可嘗試應用於短峰波波形變化的歷時等。

六、參考文獻

1. 太田昌孝、竹內あきら、大口孝之 (1986) : "應用グラフィックス" pp309, アスキー出版局 東京。
2. 周宗仁、葉榮椿 (1981) : "有限振幅波作用於海洋結構物之研究", 土木水利季刊第七卷第四期, p53 ~ p61。
3. 周宗仁、林昭圭 (1984) : "應用邊界元素法數值解析任意地形及水深之港池水面波動問題", 國立臺灣海洋學院河海工程學系河海研究第27號, pp21。
4. 周宗仁 (1983) : "邊界元素法在波浪力學之應用" pp227, 國立臺灣海洋學院河海工程學會。

Application of Computer Graphics on Water Waves Profile

ABSTRACT

Firstly, we use Boundary Element Method to find the numerical solution of wave profile in harbors and around cylinders of arbitrary shape. Then, we transform these data of wave profile into graphics on screen of personal computer with computer graphics software. In this transformation, we have considered the factors for visual realism which is rotational angle, translation scale, color, light source etc. about free surface of wave profile.

For the purpose of the time history of wave profile simulation, we segment the wave period into 24 time step and show the pictures of wave profile for each time step on screen. Meanwhile using assembly language program to control storage and display of each picture to show a continuous pictures.

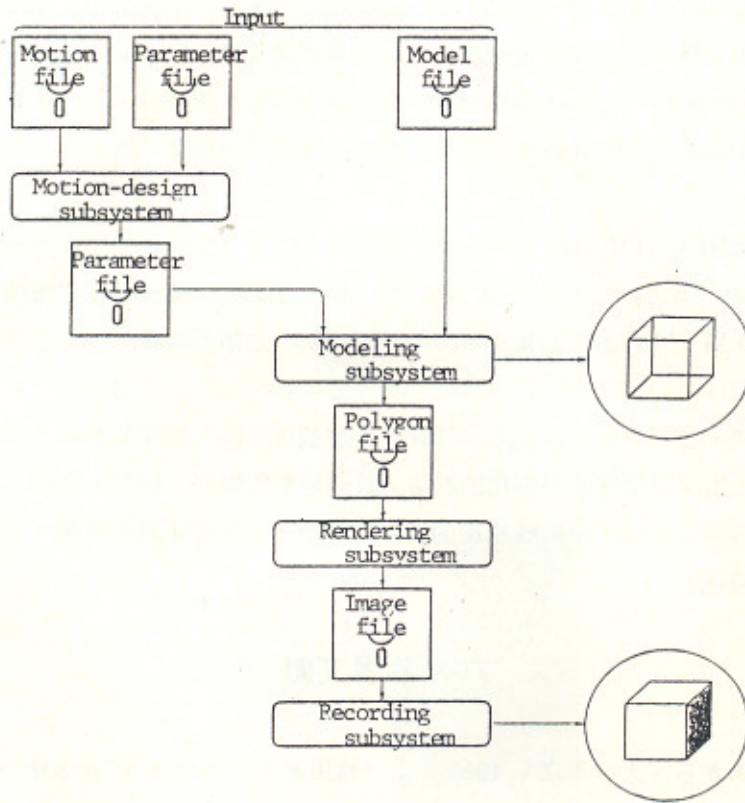


圖 2-1 MORE 系統的構成

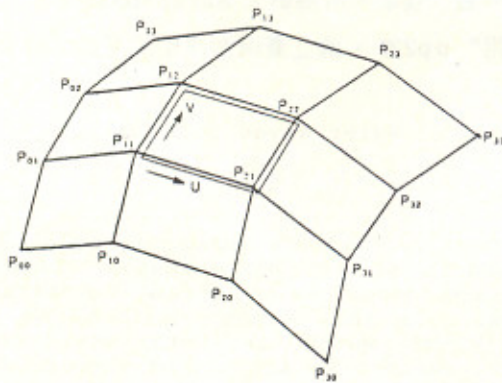


圖 2-2 雙三次 B 曲線規控制點

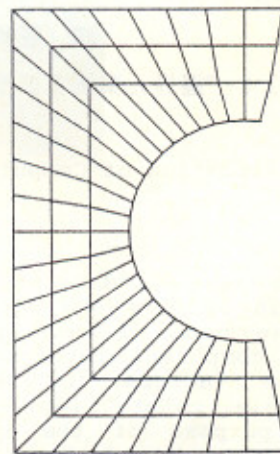


圖 2-3 圓柱附近波形曲面繪製

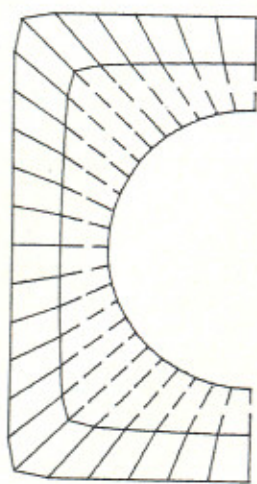


圖 2—4 圓柱邊界波形曲面繪製

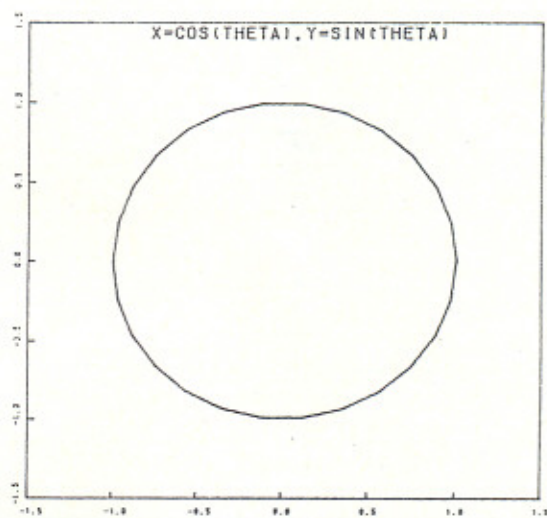


圖 3—1 參數型表示之單位圓

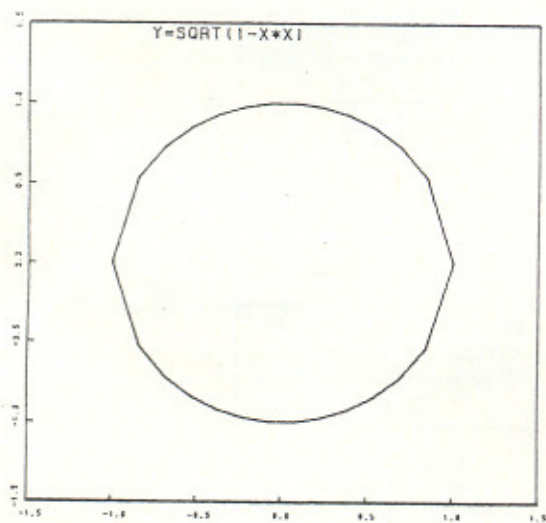


圖 3—2 非參數型表示之單位圓

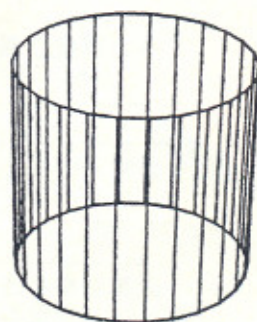


圖 3—3 單位圓柱之繪製

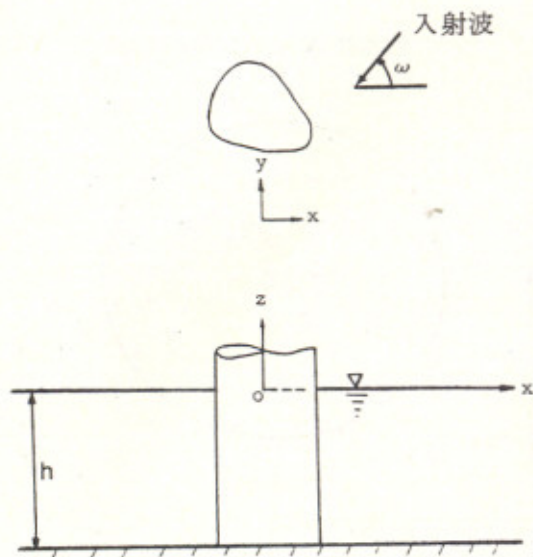


圖 4—1 直立柱體之定義圖

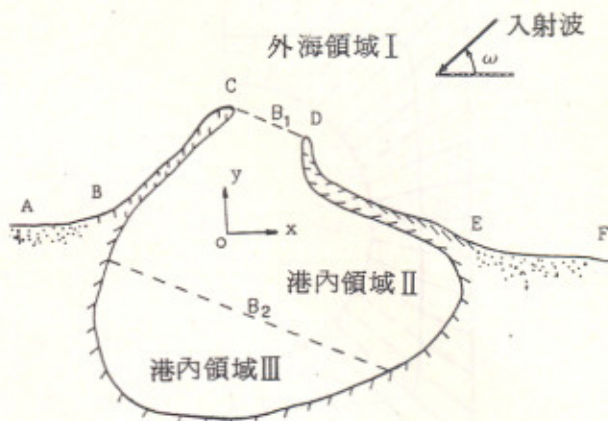


圖 4—2 任意形狀港池定義圖

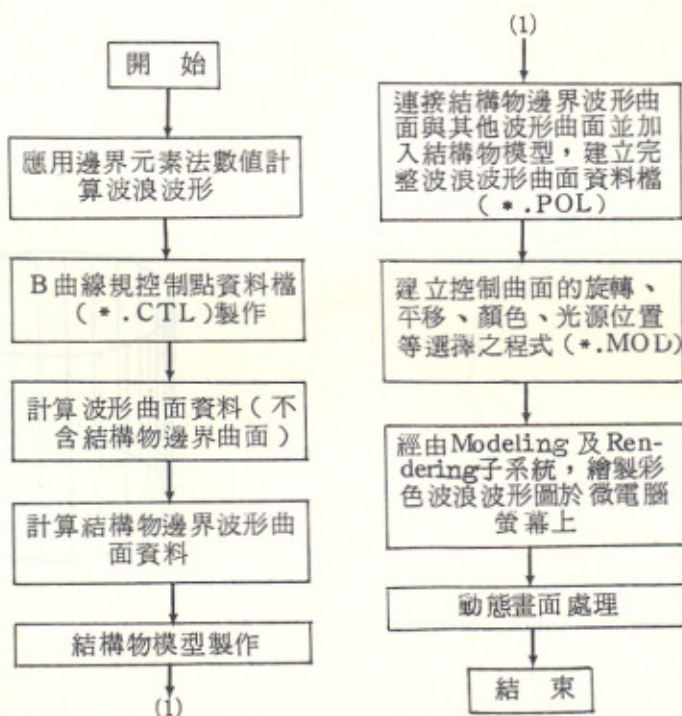


圖 5—1 波浪波形圖繪製流程圖

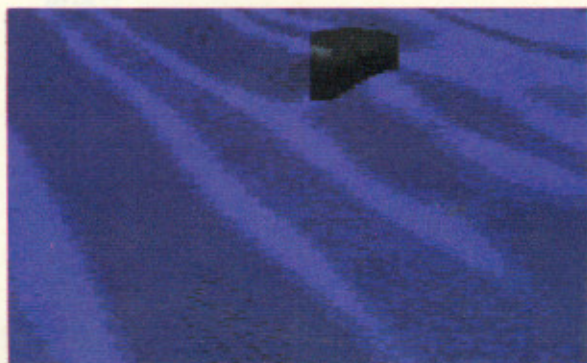


圖5-2 $\sigma^2 h/g=0.5, \sigma t=0^\circ$ 圓柱存在之波形圖

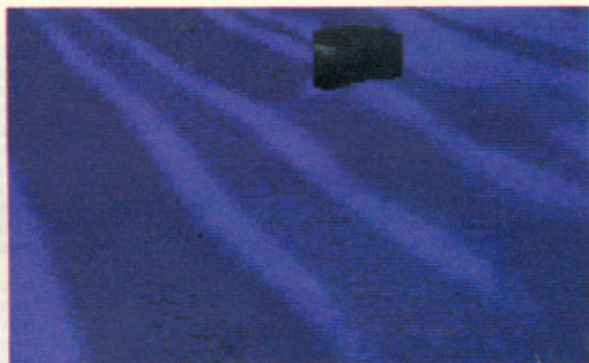


圖5-3 $\sigma^2 h/g=0.5, \sigma t=30^\circ$ 圓柱存在之波形圖



圖5-4 $\sigma^2 h/g=0.5, \sigma t=0^\circ$ 橢圓柱存在之波形圖

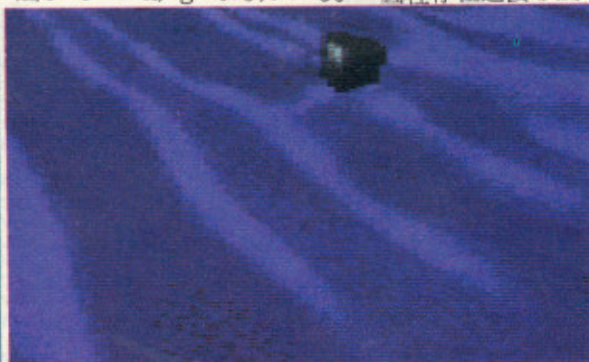


圖5-5 $\sigma^2 h/g=0.5, \sigma t=30^\circ$ 橢圓柱存在之波形圖

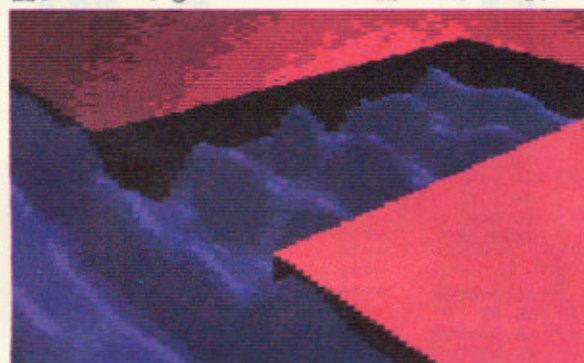


圖5-6 $\sigma^2 h/g=1.2, \sigma t=0^\circ$ 矩形港池之波形圖



圖5-7 $\sigma^2 h/g=1.2, \sigma t=30^\circ$ 矩形港池之波形圖

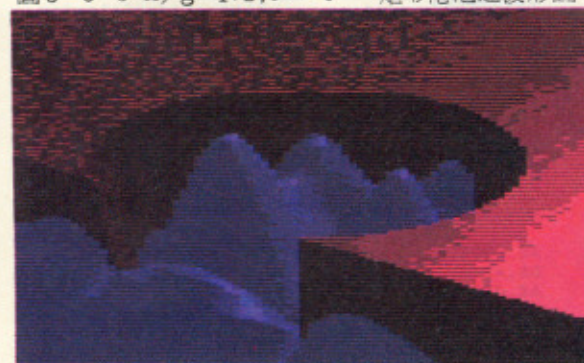


圖5-8 $\sigma^2 h/g=1.2, \sigma t=0^\circ$ 圓形港池之波形圖

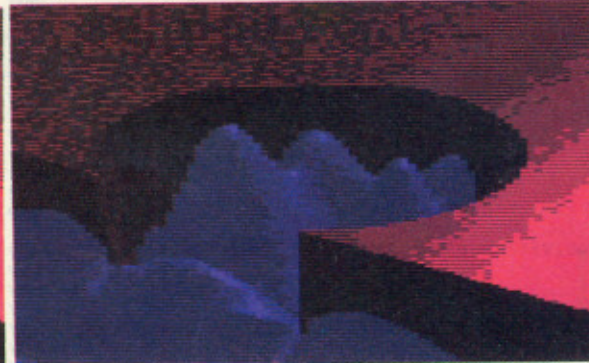


圖5-9 $\sigma^2 h/g=1.2, \sigma t=30^\circ$ 圓形港池之波形圖

附錄一

```

CODE SEGMENT
ASSUME CS:CODE
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH BP
PUSH SI
PUSH DI
PUSH DS
PUSH ES
;
MOV AH,2FH
INT 21H
MOV AL,ES:[BX]
INC BX
MOV DL,0
CMP AL,2
JE GET1
CMP AL,3
JE GET2
JMP QUIT
;
GET2: INC BX
MOV AL,ES:[BX]
SUB AL,30H
JC QUIT
CMP AL,10
JNC QUIT
MOV CL,10
MUL CL
MOV DL,AL
INC BX
MOV AL,ES:[BX]
SUB AL,30H
JC QUIT
CMP AL,10
JNC QUIT
ADD AL,DL
CMP AL,0
JE QUIT
OUT 0ECH,AL
;
CLD
MOV AX,0A800H
MOV DS,AX
MOV AX,08000H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW
;
MOV AX,0B000H
MOV DS,AX
MOV AX,08800H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW
;
MOV AX,0B800H
MOV DS,AX
MOV AX,09000H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW
;
QUIT: POP ES
POP DS
POP DI
POP SI
POP BP
POP DX
POP CX
POP BX
POP AX
INT 20H
CODE ENDS
END

```

附錄二

```

CODE SEGMENT
ASSUME CS:CODE
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH BP
PUSH SI
PUSH DI
PUSH DS
PUSH ES
;
MOV AH,2FH
INT 21H
MOV AL,ES:[BX]
INC BX
MOV DL,0
CMP AL,2
JE GET1
CMP AL,3
JE GET2
JMP QUIT
;
GET2: INC BX
MOV AL,ES:[BX]
SUB AL,30H
JC Q
CMP AL,10
JNC Q
MOV CL,10
MUL CL
MOV DL,AL
INC BX
MOV AL,ES:[BX]
SUB AL,30H
JC Q
CMP AL,10
JNC Q
ADD AL,DL
CMP AL,0
JE Q
MOV AH,00H
MOV BP,AX
;
START: MOV BX,0001H
COUNT: MOV AX,BX
OUT 0ECH,AL
AND AL,01H
OUT 0A6H,AL
XOR AL,01H
OUT 0A4H,AL
;
CLD
MOV AX,08000H
MOV DS,AX
MOV AX,0A800H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW
;
MOV AX,08800H
MOV DS,AX
MOV AX,0B000H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW
;
MOV AX,08800H
MOV DS,AX
MOV AX,0B800H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW
;
MOV AX,09000H
MOV DS,AX
MOV AX,0B800H
MOV ES,AX
MOV SI,00000H
MOV DI,00000H
MOV CX,04000H
REP MOVSW

```

```

REP MOVSW
;
MOV AX,BX
AND AL,01H
OUT 0A4H,AL
XOR AL,01H
OUT 0A6H,AL
;
PUSH BX
MOV CX,01000H
MOV AH,05H
INT 18H
TEST BH,01H
WAIT JNE WAIT
L LOOP L
POP BX
JMP NEXT
;
WAIT: MOV AH,00H
INT 18H
JMP QUIT
;
NEXT: INC BX
CMP BX,BP
JLE COUNT
JMP START
;
QUIT: MOV AL,00H
OUT 0A6H,AL
OUT 0A4H,AL
POP ES
POP DS
POP DI
POP SI
POP BP
POP DX
POP CX
POP BX
POP AX
INT 20H
CODE ENDS
END

```

附錄三

```

/*
** FILENAME :MENU.C
** VERSION :6.0
** WRITEN BY :Y.L.Dern
*/
#include "stdio.h"
#include "math.h"
#include "ctype.h"
#include "malloc.h"
#define CTL1 3828
#define CTL2 3832
#define CTL3 3836
#define CTL4 3944
#define CTL5 4052
#define CTL6 4160
#define CTL7 4268
#define CTL8 4376
#define CTL9 4484
#define CTL10 4596
#define CTL11 4600
#define CTL12 4604
#define ROW 73
#define COLUMN 33
#define XYZ 3
typedef float xyz[XYZ];
typedef float data[25][6];
typedef float point[4][6];
typedef xyz data1[ROW][COLUMN];
typedef xyz point1[4][4];
typedef float mix[25][6];
FILE *in,*out;
char fname[10],foname[10];
int row,column,r,c,mu,md,n,count;
int inside,infirst,inlast,aa,bb,ctl,fflag;
main()
{
char *memloc,option;
unsigned k=1024;
data d;
point p;
data1 d1;
point1 p1;
mix mm;
mix nn;
unsigned nbytes=32*k;
printf("\t:;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;\n");
printf("\t: \n");
printf("\t: this software used to creat three dimation datafile \n");
printf("\t: \n");
printf("\t: \n");
printf("\t: \n");
printf("\n\tA:whole area consist of sixteen point");
printf("\n\tB:whole area consist of four point");
printf("\n\tC:outside area consist of sixteen point");
printf("\n\tD:outside area consist of four point");
printf("\n\tE:interface out option ");
printf("\n\tF:interface in option ");
printf("\n\tG:inside 25 option ");
printf("\n\tH:four point spline ");
printf("\n\tI:clipping source data ");
printf("\n\tJ:middle polygon option ");
printf("\n\n");
printf("\tplease input your option --->");
scanf("%c",&option);
switch(option){
case 'a':
case 'A':
memloc = malloc(nbytes);
r=4;c=4;
control(d1);
if(memloc != NULL)
free(memloc);
break;
case 'b':
case 'B':
memloc = malloc(nbytes);
r=2;c=2;
control(d1);
if(memloc != NULL)
free(memloc);
break;
}
}

```

```

case 'c':
case 'C':
    memloc = malloc(nbytes);
    r=4;c=4;
    bcontrol(d1);
    if(memloc != NULL)
        free(memloc);
    break;
case 'd':
case 'D':
    memloc = malloc(nbytes);
    r=2;c=2;
    bcontrol(d1);
    if(memloc != NULL)
        free(memloc);
    break;
case 'e':
case 'E':
    memloc = malloc(nbytes);
    fflag=6;
    point25(p,d);
    if(memloc != NULL)
        free(memloc);
    break;
case 'f':
case 'F':
    memloc = malloc(nbytes);
    fflag=6;infirst=1;inlast=24;aa=0;bb=1;
    inside25(p,d);
    if(memloc != NULL)
        free(memloc);
    break;
case 'g':
case 'G':
    memloc = malloc(nbytes);
    fflag=3;infirst=1 ;inlast=24;aa=3;bb=2;
    inside25(p,d);
    if(memloc != NULL)
        free(memloc);
    break;
case 'h':
case 'H':
    memloc = malloc(nbytes);
    spline3(p1);
    if(memloc != NULL)
        free(memloc);
    break;
case 'i':
case 'I':
    memloc = malloc(nbytes);
    clipping(d1);
    if(memloc != NULL)
        free(memloc);
    break;
case 'j':
case 'J':
    memloc = malloc(nbytes);
    mixture(mm,nn);
    if(memloc != NULL)
        free(memloc);
    break;
default :
    inerror();
}
printf("\n\tif you want to try again please type menu\n");
}
mixture(mm,nn)
mix mm;
mix nn;
{
int i,k;
openr();
for(i=0;i<25;i++){
    for(k=0;k<6;k++){
        fscanf(in,"%f",&mm[i][k]);
    }
}
fclose(in);
openr();

```

```

for(i=0;i<25;i++){
    for(k=0;k<6;k++){
        fscanf(in,"%f",&mm[i][k]);
    }
}
fclose(in);
openr();
for(i=0;i<25;i++){
    for(k=0;k<6;k++){
        fscanf(in,"%f",&nn[i][k]);
    }
}
fclose(in);
openw();
for(i=0;i<24;i++){
    fprintf(out,"p\t(%f,%f,%f,%f,%f,%f),\n",mm[i][0],mm[i][1],mm[i][2],
        mm[i][3],mm[i][4],mm[i][5]);
    fprintf(out," \t(%f,%f,%f,%f,%f,%f),\n",mm[i+1][0],mm[i+1][1],mm[i+1][2],
        mm[i+1][3],mm[i+1][4],mm[i+1][5]);
    fprintf(out," \t(%f,%f,%f,%f,%f,%f),\n",nn[i+1][0],nn[i+1][1],nn[i+1][2],
        nn[i+1][3],nn[i+1][4],nn[i+1][5]);
    fprintf(out," \t(%f,%f,%f,%f,%f,%f),\n",nn[i][0],nn[i][1],nn[i][2],
        nn[i][3],nn[i][4],nn[i][5]);
}
fclose(out);
}
inside25(p,d)
point p;
data d;
{
    int ch,i,k;
    count=0;
    inside=0;
    openr();
    do{
        switch(ch=skip()){
            case 'p':
                count++;
                pinside(p,d);
                break;
            case EOF:
                break;
            default:
                inerror();
        }
    }while( ch != EOF );
    openw();
    output(d);
    fclose(in);
    fclose(out);
}

pinside(p,d)
point p;
data d;
{
    int i,k;
    for(i=0;i<4;i++){
        if(i<3){
            if((fscanf(in,"\t(%f,%f,%f,%f,%f,%f),\n",&p[i][0],
                &p[i][1],&p[i][2],&p[i][3],&p[i][4],&p[i][5])) != 6)
                inerror();
        }
        if(i==3){
            if((fscanf(in,"\t(%f,%f,%f,%f,%f,%f),\n",&p[i][0],
                &p[i][1],&p[i][2],&p[i][3],&p[i][4],&p[i][5])) != 6)
                inerror();
        }
    }

    if(count >= infirst && count <= inlast ){
        if(inside != 23){
            for(k=0;k<6;k++){
                d[inside][k]=p[aa][k];
            }
        }
        if(inside == 23){
            for(k=0;k<6;k++){
                d[inside][k]=p[aa][k];
                d[inside+1][k]=p[bb][k];
            }
        }
        inside++;
    }
}

```

```

}
}
clipping(d1)
datal d1;
{
  int i,j,a,b,count=0;
  openr();
  printf("\n\tplease enter row_number ---->");
  scanf("%d",&row);
  printf("\n\tplease enter column_number ---->");
  scanf("%d",&column);
  printf("\n\tplease input clipping_row ---->");
  scanf("%d",&a);
  printf("\n\tplease input clipping_col ---->");
  scanf("%d",&b);
  getdatas(d1);
  fclose(in);
  openw();
  for(i=0;i<row;i++){
    for(j=0;j<column;j++){
      if(i>=a && i<row-a && j>=b){
        count+=1;
        fprintf(out,"%f %f %f ",d1[i][j][0],d1[i][j][1],d1[i][j][2]);
        if((count % 4)==0) fprintf(out,"\n");
        if((count % 16)==0) fprintf(out,"\n");
      }
    }
  }
  fclose(out);
}

getdatas(d1)
datal d1;
{
  int i,j,k;
  for(i=0;i<row;i++){
    for(j=0;j<column;j++){
      for(k=0;k<3;k++){
        fscanf(in,"%f",&d1[i][j][k]);
      }
    }
  }
}

control(d1)
datal d1;
{
  int i,j,k,flagr,flagc,mn;
  mn=(r==4) ? 16 : 4 ;
  printf("\n\t$$this option used to creat %d control point datafile$$\n",mn);
  openr();
  printf("\n\tplease input row-number ---->");
  scanf("%d",&row);
  printf("\n\tplease input column-number ---->");
  scanf("%d",&column);
  getdatas(d1);
  printf("\n\tread datas completely !!\n");
  fclose(in);
  openw();
  flagr=flagc=0;
  for(i=flagr;i<flagr+r;i++){
    for(j=flagc;j<flagc+c;j++){
      for(k=0;k<3;k++){
        fprintf(out,"%f ",d1[i][j][k]);
      }
      fprintf(out," ");
    }
    if(i==flagr+r-1 && column==flagc+c){
      i=flagr;
      flagc=0;
      flagr+=1;
      fprintf(out,"\n");
      if(row==flagr+r-1){
        fclose(out);
        break;
      }
    }
    if(i==flagr+r-1){
      i=flagr-1;
      flagc+=1;
      fprintf(out,"\n");
    }
  }
  fprintf(out,"\n");
}

```



```

}
printf("\n\input completely !!");
}
openr()
{
int flag;
do{
flag=0;
printf("\n\tnow enter input file name --->");
scanf("%s",filename);
if((in=fopen(filename,"r"))==NULL){
puts("can't open input file !!");
flag=1;
}
else
printf("\n\topen %s file for reading.....\n",filename);
}while(flag==1);
}
openw()
{
int flag;
do{
flag=0;
printf("\n\tnow enter output file name --->");
scanf("%s",foname);
if((out=fopen(foname,"w"))==NULL){
puts("can't open output file !!");
flag=1;
}
else
printf("\n\topen already and waiting for output....\n");
}while(flag==1);
}
point25(p,d)
point p;
data d;
{
int ch;
count=0;
openr();
do{
switch(ch=skip()){
case 'p':
count+=1;
inpoint(p,d);
break;
case EOF:
break;
default:
inerror();
}
}while( ch != EOF && count < 4610 );
openw();
output(d);
fclose(in);
fclose(out);
}
output(d)
data d;
{
int i,k;
for(i=0;i<25;i++){
for(k=0;k<fflag;k++){
fprintf(out,"%f ",d[i][k]);
}
fprintf(out," ");
if((i+1)%2==0) fprintf(out,"\n");
}
}
inerror()
{
fprintf(stderr,"\n\t?? input error in %s ??\n",filename);
}
skip()
{
int ch;
while(isspace(ch=getc(in)));
return ch;
}

```

```

inpoint(p,d)
point p;
data d;
{
  int i,k;
  for(i=0;i<4;i++){
    if(i<3){
      if(!fscanf(in,"\t(%f,%f,%f,%f,%f,%f);\n",&p[i][0]
        ,&p[i][1],&p[i][2],&p[i][3],&p[i][4],&p[i][5])) != 6)
        inerror();
    }
    if(i==3){
      if(!fscanf(in,"\t(%f,%f,%f,%f,%f,%f);\n",&p[i][0]
        ,&p[i][1],&p[i][2],&p[i][3],&p[i][4],&p[i][5])) != 6)
        inerror();
    }
  }

  if(count==CTL1+2){
    for(k=0;k<6;k++){
      d[5][k]=p[2][k];
    }
  }
  if(count==CTL1+4){
    for(k=0;k<6;k++){
      d[4][k]=p[2][k];
    }
  }
  if(count==CTL2+2){
    for(k=0;k<6;k++){
      d[3][k]=p[2][k];
    }
  }
  if(count==CTL2+4){
    for(k=0;k<6;k++){
      d[2][k]=p[2][k];
    }
  }
  if(count==CTL3+2){
    for(k=0;k<6;k++){
      d[1][k]=p[2][k];
    }
  }
  if(count==CTL3+4){
    for(k=0;k<6;k++){
      d[0][k]=p[2][k];
    }
  }
  if(count==CTL4+3){
    for(k=0;k<6;k++){
      d[6][k]=p[1][k];
    }
  }
  if(count==CTL4+4){
    for(k=0;k<6;k++){
      d[7][k]=p[1][k];
    }
  }
  if(count==CTL5+3){
    for(k=0;k<6;k++){
      d[8][k]=p[1][k];
    }
  }
  if(count==CTL5+4){
    for(k=0;k<6;k++){
      d[9][k]=p[1][k];
    }
  }
  if(count==CTL6+3){
    for(k=0;k<6;k++){
      d[10][k]=p[1][k];
    }
  }
  if(count==CTL6+4){
    for(k=0;k<6;k++){
      d[11][k]=p[1][k];
    }
  }
  if(count==CTL7+3){
    for(k=0;k<6;k++){
      d[12][k]=p[1][k];
    }
  }
}

```

```

if(count==CTL7+4){
    for(k=0;k<6;k++){
        d[13][k]=p[1][k];
    }
}
if(count==CTL8+3){
    for(k=0;k<6;k++){
        d[14][k]=p[1][k];
    }
}
if(count==CTL9+4){
    for(k=0;k<6;k++){
        d[15][k]=p[1][k];
    }
}
if(count==CTL9+3){
    for(k=0;k<6;k++){
        d[16][k]=p[1][k];
    }
}
if(count==CTL9+4){
    for(k=0;k<6;k++){
        d[17][k]=p[1][k];
        d[18][k]=p[2][k];
    }
}
if(count==CTL10+1){
    for(k=0;k<6;k++){
        d[19][k]=p[1][k];
    }
}
if(count==CTL10+3){
    for(k=0;k<6;k++){
        d[20][k]=p[1][k];
    }
}
if(count==CTL11+1){
    for(k=0;k<6;k++){
        d[21][k]=p[1][k];
    }
}
if(count==CTL11+3){
    for(k=0;k<6;k++){
        d[22][k]=p[1][k];
    }
}
if(count==CTL12+1){
    for(k=0;k<6;k++){
        d[23][k]=p[1][k];
    }
}
if(count==CTL12+3){
    for(k=0;k<6;k++){
        d[24][k]=p[1][k];
    }
}
}
spline4(p1)
point1 p1;
{
    openr();
    openw();
    while(getpoints(p1))
        polygon(p1);
    fclose(in);
    fclose(out);
}
getpoints(p1)
point1 p1;
{
    int i,j;
    for(i=0;i<2;i++){
        for(j=0;j<2;j++){
            if(fscanf(in,"%f %f %f",&p1[i][j][0],&p1[i][j][1],&p1[i][j][2])!=3)
                return 0;
        }
    }
}
return 1;
}

```

```

polygon(p1)
point1 p1;
{
    int k,r,c,i;
    float p2[4][3],p3[4][3],nx,ny,nz,nl;

    for(k=0;k<3;k++){
        p2[0][k]=p1[1][0][k]-p1[0][0][k];
        p3[0][k]=p1[0][1][k]-p1[0][0][k];

        p2[2][k]=p1[0][1][k]-p1[1][1][k];
        p3[2][k]=p1[1][0][k]-p1[1][1][k];

        p2[1][k]=-p3[0][k];
        p3[1][k]=-p2[2][k];

        p2[3][k]=-p3[2][k];
        p3[3][k]=-p2[0][k];
    }
    fprintf(out,"p");
    for(i=0;i<4;i++){
        if(i==0){r=0;c=0;}
        if(i==1){r=0;c=1;}
        if(i==2){r=1;c=1;}
        if(i==3){r=1;c=0;}
        nx=p2[i][1]*p3[i][2]-p2[i][2]*p3[i][1];
        ny=p2[i][2]*p3[i][0]-p2[i][0]*p3[i][2];
        nz=p2[i][0]*p3[i][1]-p2[i][1]*p3[i][0];
        nl=sqrt(nx*nx+ny*ny+nz*nz);
        nx /=nl;
        ny /=nl;
        nz /=nl;
        fprintf(out,"\t(%f,%f,%f,%f,%f,%f)%c\n",p1[r][c][0],p1[r][c][1],
            p1[r][c][2],nx,ny,nz,((i==3)?':':' '));
    }
}

```