

第三章 基本概念

在光學系統設計軟體裡,基本上所需要的概念是雷同的,首先要界定透鏡規格。爲了程式處理,各家程式公司對輸入透鏡規格有其固定的格式規定。但由於所用的視窗界面雷同,所以大體而言格式是差不多。但在摹擬的眼光裡,有一件事是與以前不同:這裡是把一個真實的透鏡組視作一個電腦文件來處理。程式讀取文件檔案進行分析,產生各式輸出,讓讀者評估透鏡組效能乃至作最佳化分析與規劃。雖然這裡用的是 Sinclair Optics 公司的 OSLO LT 軟體,對其他公司其格式雷同。

3-1. 「透鏡就是一份文件」

在透鏡設計中,製作一份文件是用文字、圖等等去描述主題。而設計一個透鏡是用數字、文字、草樣等等去把透鏡的訊息給表示出來。撰寫文件與透鏡設計都有個重點,那就是用「語言」去把訊息給表示出來,一本書如果沒有把它的訊息給表達出來,那不叫一本書,同樣的,透鏡設計也是。文件軟體協助作者去製作文件,而 OSLO 協助光學設計者製作其透鏡。換言之,在 OSLO「透鏡」就是一份文件。

在早期的透鏡設計裡,設計稿包含:曲率 (curvatures) 厚度 (thickness) 和玻璃 (glasses) 種類,再加上孔徑 (aperture) 視場 (field of view) 及波長 (wavelengths)。這些正好足夠用來作光束覓跡 (或光束描光) (traces rays)。現在,一個完整的設計需包含更多的訊息—諸如容忍度 (或是光學界常使用之公差) (tolerances) 鍍膜規格 (coating specification), 製作細節—像鉋光等級 (polishing grades), 斜面程度 (bevels) 和表面要求 (cosmetic requirements)。雖然 OSLO 尚未能處理設計的所有層面,但是它的資料結構遠比一個簡單透鏡設計稿包含更多。

要想將 OSLO 發揮到淋漓盡致,就必須瞭解 OSLO 程式所處理的數據、會產生的數據及其基本特性。OSLO 數據分成四大類:

1. 面數據 (surface data)
2. 透鏡設定 (operating conditions)
3. 程式設定 (preferences)
4. 試算表數據暫存庫 (Spreadsheet Buffer)

另外有一類則是雜項,包含諸如指令變數 (command arguments), 固定內建設定 (fixed defaults) 及各式內部使用暫時數據。

3-2 面數據 (surface data) :

面數據包含平常透鏡草樣的項目(lens prescription item)如 :

1. 曲率半徑 (radii of curvature)
2. 厚度 (thickness)
3. 玻璃種類 (glasses)
4. 非球面常數 (aspheric constants)

等等。所有的面都以數字標記, 其第零面為物面 (object surface)。在一般情況連續的透鏡組, 其面數 (surface numbers) 依光束穿過順序依序增加。

在 OSLO PRO 版本裡, 多組態系統可以被處理。多組態系統意指各面組 (groups of surface) 有不同的空間分布, 但各分佈中面數仍依序增加。而在 OSLO SIX 版本中, 則更可以處理透鏡陣列 (lens arrays) 及非循序面組 (non-sequential groups)。所謂非循序面指的是光束並不是依照循序面數穿過各個鏡面。

在 OSLO 裡有四種方式輸入或修改面數據 :

1. 面數據試算表 (surface data spreadsheet) (如圖 3-1 所示)
利用更新主選單 (update menu) 或利用 F5 工具列鍵 (toolbar icon)
2. 全編輯 (global editing)
各項數據、面數及值, 直接由指令列 (command line) 輸入。
3. 內建透鏡編輯器 (internal lens editor)
程式可以利用此一編輯器, 可以切入輸入某一鏡面的數據。
4. 離線輸入 (off-line)
可以利用任何編輯器以文件檔案 (text file) 型或編輯面數據輸入。

雖然試算表方式 (Spread sheet) 是最方便的, 但其它方式亦有其方便之處 (raison d'être)

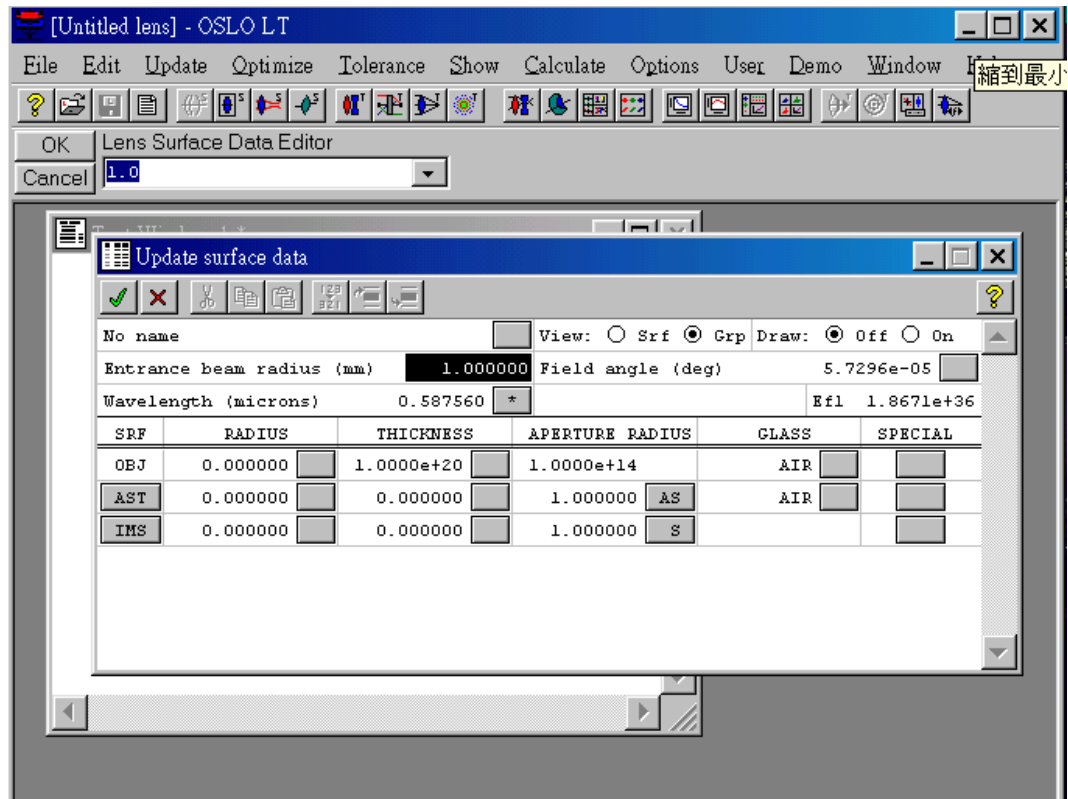


圖 3-1 面數據試算表

3-3. 透鏡設定 (Operating conditions):

透鏡設定或條件包含的是跟整個透鏡組有關的項目或者指定鏡組使用條件。傳統上透鏡設定是指孔徑 (aperture) 及視角 (field of view)。OSLO 則包含許多其它操作條件。有些則是已被光學設計軟體納入的如波長、誤差函數 (error function); 有些則是新的, 例如:

- 1 元件草樣數據 (element drawing data):
可以更完整描述整個設計。
- 1 斑點圖操作條件 (spot diagram operating conditions)
可以進一步進行一致性分析 (consistent analysis)
- 1 透鏡草樣製作條件 (lens drawing operating conditions)
可以避免重新鍵入 (re-entry) 大量數據。

重點是透鏡設定是附在透鏡數據, 而且是跟面數據 (surface data) 儲存在透鏡檔案 (lens files)。這簡化了設計手續, 一旦完成設定則之後若有選定, 就自動引入。例如一組特定光束用於一透鏡草樣 (lens drawing), 則不論何時, 一旦啓用這一個含有特定光束的草樣, 此一特定光束透鏡設定即可使用。

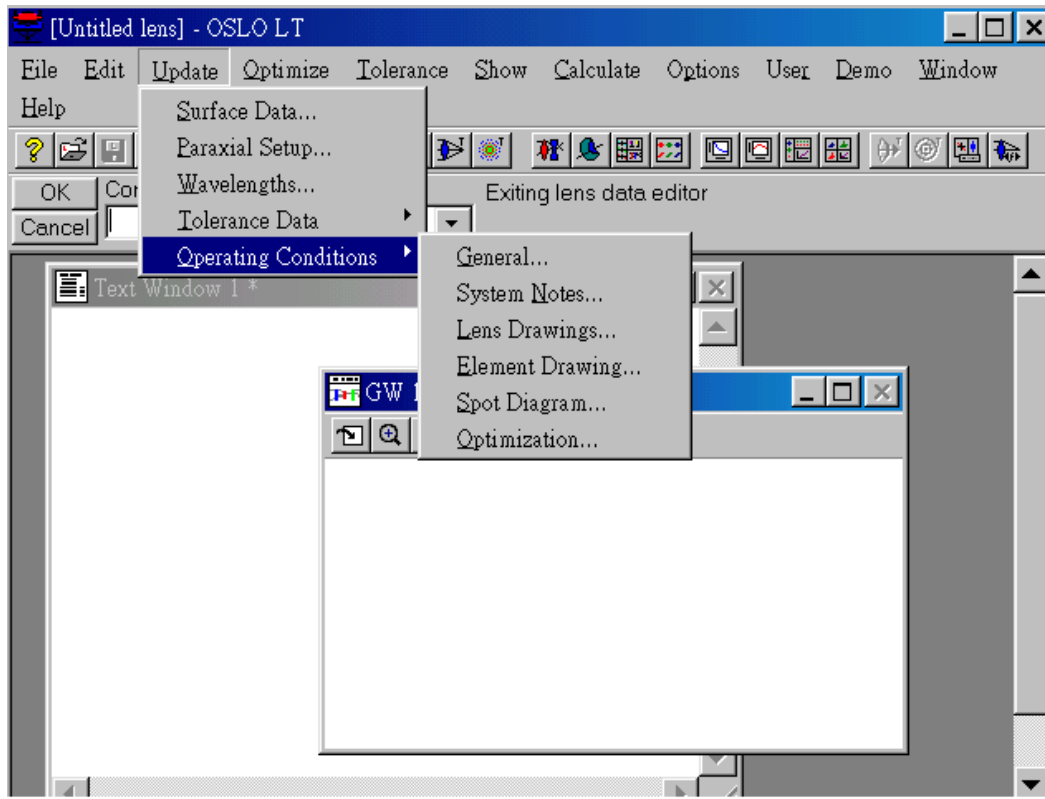


圖 3-2 透鏡設定控制選項

3-4. 程式設定 (preferences)

程式設定與透鏡設定有些不同。程式設定是跟著 OSLO 整個程式, 其影響是對所有透鏡都相同。例如圖形視窗的顯示與數字格式等等。大多數程式設定是一旦被設定, 立即生效, 但有些則必須重新啟動 OSLO。程式設定的例子有諸如面數的最大上限, 斑點圖的光束及程式所可以用的波長及字型 (fonts) 的選擇。一旦一個程式設定被決定, 其內容會被存入結構檔 (configuration file) (*.ini), 且將在啟動時被讀取。所以所修改的這些及原有之內容將一直被保留。

3-5. 試算表數據暫存庫 (the Spreadsheet Buffer)

試算表數據暫存是 OSLO 數據結構中一個重要部份, 就 OSLO 而言, 試算表數據暫存是個輸出數據結構, 但其實它是介於程式與使用者巨指令集 (見下節星指令 (star commands) 討論) 的主要連通。試算表數據暫存庫是附在每一個文字視窗 (text window), 一旦 OSLO 將浮點數字輸出寫入視窗, 同時此份備份輸出含將放在試算表數據暫存庫, 且其全部精準度均被保留。而各個暫存元素均可在數據輸入區 (data entry), 以 SCP 及 CCL 巨指令語言加以定址或者以按住在文字視窗的數字來顯示暫存元素的內容。

3-6. 星指令 (Star Commands)

雖然 OSLO 對一般使用者來看, 可能只是個平常的視窗應用, 但它事實上是個視窗應用與巨指令集語言的綜合。部份指令是由程式所提供且屬於核心部分, 而其它則是以巨指令集方式撰寫。對使用者而言, 那個是那個並不是馬上可以看出來。在 OSLO 中有兩個巨指令集語言。OSLO Light (即 OSLO LT) 包含 SCP—中斷語言。OSLO PRO 及 OSLO SIX 則另外包含組合語言 CCL, 這兩個語言均用 C 語法撰寫。

在 SCP 中的巨指令之執行是在其前冠以星號, 是以命名叫做「星指令」(star command)。使用者可以撰寫其個人的星指令並與 OSLO 的指令集整合在一起。OSLO 的主選單 (menu) 是可以由使用者改變。因為這些指令通常只是用來提供特殊需求, 所以它們不必像內建指令那般需界定內容, 所以把這些加入程式是非常容易。

3-7. 按／下指令界面 (Click/Command Interface)

如前所述, 在 OSLO 中加入星指令是可以的。這些指令可以加入主選單 (menu) 或直接在指令行輸入指令名並執行。雖然許多視窗程式誇稱不用學任何指令就可以上手應用, 但事實上所有程式都是建構於指令集上。問題在於是這些指令是否由使用者直接使用。在 OSLO 裡, 傳統指令界面被保留以給予使用者多一個選擇。這提供一高效率的方法去運用程式。「按／下指令」讓使用者可以在任何時候任一點, 選擇輸入指令或用滑鼠 (mouse) 按下一個圖樣 (icon) 或主選單 (menu)。

當試算表 (Spreadsheet) 打開時就可以輸入指令。OSLO 把指令、數據分開並且會接續完成適當動作。所以在 OSLO 試算表數據的夾格 (cell) 被叫做精靈夾格 (Smart cell)。

3-8. 互動操作 (interactive operation)

OSLO 程式基本上是個互動式。在輸入指令之後, OSLO 就執行所需動作並更新顯示界面。舉例而言, 面數據試算表 (surface data spreadsheet) 是一直更新。但也有少數例外—特別是跟最佳化數據輸入有關的。雖然如此, 整個 OSLO 設計是基於互動模式。從這理念來的是波長、組態、物點 (objective point), 及斑點圖的概念的使用。一旦這些量被決定, 就一直不變, 除非使用者改變或其它的指令作用使之改變。這使得整個程式的速度與效率提高。以斑點圖為例, 一旦一個斑點圖被算出, 則很大部份的分析特性可順帶完成而不必再重畫光束。

程式互動結構的顯著例子是 OSLO 中所用的「互動式設計視窗」。在這互動式設計視窗中, 透鏡參數是附在圖形式滑表 (graphic slider)。一旦使用者改變滑

(2001 年第一版; 僅供教學與研究參考, 請勿作商業用途。)

編者: 許阿娟、何承舫、張國輝、陳志隆

表, 則透鏡參數改變, OSLO 會更新特定的評估功能(如光束覓跡, 斑點圖等等), 使用者將可感受到整個程式以即時方式在進行。

參考文獻

1. OSLO user's guide (version 5) (Sinclair Optics, 1996) 。 OSLO user's guide 可以自 on-line help 得到。