

國立交通大學

電信工程學系

碩士論文

基於積分方程電磁場解答器的適應性網格
生成和細分



Adaptive Mesh Generation and Refinement
for Integral-Equation-Based Electromagnetics
Solvers

研究生：林旻靜

指導教授：趙學永

中華民國九十四年六月

基於積分方程電磁場解答器的適應性網格生成和細分
Adaptive Mesh Generation and Refinement for Integral-Equation-Based
Electromagnetics Solvers

研究生：林旻靜

Student : Lin Min-Ching

指導教授：趙學永

Advisor : Chao Hsueh-Yung

國立交通大學
電信工程學系
碩士論文



A Thesis

Submitted to Department of Communication Engineering
College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Communication Engineering

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

基於積分方程電磁場解答器的適應性網格生成和細分

研究生：林旻靜

指導教授：趙學永 老師

國立交通大學電信工程學系（研究所）碩士班

摘 要

利用力差法(method of moments, 簡稱 MOM)解電場積分方程式(electric field integral equation, 簡稱 EFIE)時, 取樣網格的分佈對解的準確性有極大的影響。網格的分佈往往需要先經過人為的調整, 才可能得到較準確的解。有鑑於此, 本論文提出一個適應性網格細分的演算法, 結合將網格元素分解形成更小的單元 (h-refinement) 以及移動節點的位置 (r-refinement) 兩種網格細分的方法, 使網格分佈能適應電場積分方程式的解。利用 divid-and-conquer 方法對欲模擬的結構產生均勻分佈的 Delaunay 三角形網格, 並依據力差法解得的結構表面電流分佈細分網格, 以上適應性網格細分的過程不斷重複進行, 直到電流解收斂或達到使用者或程式預設的停止細分條件 (如最小邊長或最大網格細分次數等)。為了加速網格細分的過程, 此論文亦嘗試忽略力差法矩陣方程式中的遠交互作用來解力差法矩陣方程式。以上自動網格細分的方法亦應用在許多電磁模擬的分析的例子, 如接近帶電流導線的導體平面, 以及金屬導線散射平面波。

Adaptive Mesh Generation and Refinement for Integral-Equation-Based Electromagnetics Solvers

Student: Lin Min-Ching

Advisor: Chao Hsueh-Yung

Department of Communication Engineering
National Chiao Tung University

ABSTRACT

When the method of moments (MOM) is applied to solve the electric field integral equation (EFIE) for electromagnetic radiation and scattering problems, the accuracy of solutions greatly depends on a proper discretization of the simulated domain. Most of time, the grid distribution needs to be manually tuned for getting an accurate solution. In this thesis, we propose a mesh refinement algorithm that adapts meshes to EFIE solutions by splitting elements (h-refinement) and relocating nodes (r-refinement). Using a divide-and-conquer Delaunay triangulation, an initial mesh is generated with equally spaced seeds on the surfaces of the simulated structure. Then the mesh is iteratively refined according the current distribution on the surface. The refinement process automatically terminates when the current distribution converges or when preset criteria, such as the smallest edge length and the maximum pass of refinement, are met. In order to expedite the iterative refinement process, the current is calculated only by the near-interaction terms of the MOM impedance matrix. The adaptive mesh refinement algorithm is further applied to solve radiation and scattering from metallic structures.

誌 謝

能夠在短短的一年之內，完成這篇碩士論文，首先得感謝我的指導教授趙學永老師，趙老師不論在專業知識或是研究的熱忱皆讓人難以頂背，令我在師承他的這段期間內受益匪淺。接著要感謝鍾世忠老師與 912 實驗室的全體同學的支持並鼓勵我朝自己的理想前進。再來要感謝 HFEDA 實驗室的學弟易廷、義志、翔昱，研究路上有了同伴，在怎麼辛苦也有勇氣面對。

同時，感謝信源，不僅是我心靈的支柱，更在我做研究期間以及口試準備時，給我許多建議。

最後僅以此文獻給我摯愛的家人—父母以及兩個弟弟。



目 錄

摘 要.....	i
ABSTRACT.....	ii
誌 謝.....	iii
目 錄.....	iv
表 目 錄.....	vi
圖 目 錄.....	vii
1. 簡介.....	1
2. 適應性網格細分(Adaptive Mesh Refinement).....	3
2.1 網格形狀的選擇.....	4
2.2 Delaunay 三角形分割法.....	4
2.3 力差法解電場積分方程式.....	5
2.4 網格品質需求.....	6
2.5 網格細分的方法.....	7
3. 架構與實作.....	8
3.1 架構概觀.....	8
3.2 資料結構.....	9
3.2.1 基本幾何資料結構.....	9
3.2.2 網格資料結構.....	11
3.2.3 與 Shewchuk 的 Triangle 比較.....	12
3.3 Divide-and-Conquer Delaunay 三角形分割.....	15
3.3.1 演算法.....	15
3.3.2 增進效能的設計.....	19
3.3.3 三維結構表面網格生成.....	23
3.4 初始節點生成.....	24
3.5 邊界完整性.....	25
3.6 網格細分 (Mesh Refinement)	27
3.6.1 決定網格細分的區域.....	29
3.6.2 局部 Delaunay 三角形網格 hr 細分 (Local Delaunay hr-Refinement)	30
3.6.3 網格細分終止條件 (Criteria)	32
4. 應用.....	33
4.1 接近導體平面的導線.....	33
4.1.1 導線在導體平面之上.....	33
4.1.2 導線接近平面邊緣.....	36

4.2 金屬平板散射電磁波.....	39
5. 結論與未來研究方向.....	43
參考文獻.....	44



表 目 錄

表格 3-1：比較 Stella 和 Triangle 的基本幾何資料結構.....	14
表格 3-2：比較 Stella 和 Triangle 的基本幾何資料結構（O：有 X：沒有）.....	15
表格 3-3：比較檢查所有邊和檢查部分邊，Delaunay 三角形分割所需的時間。 .	22
表格 4-1：網格細和計算電流分佈分所需的時間.....	36
表格 4-2：網格細和計算電流分佈分所需的時間.....	39
表格 4-3：忽略遠交互作用與不忽略遠交互作用的執行時間比較.....	42



圖 目 錄

圖 2-1：適應性網格細分的流程	4
圖 2-2：(a)非 Delaunay 三角形分割 (b)Delaunay 三角形分割.....	5
圖 3-1：適應性網格產生器架構概觀	8
圖 3-2：左：一個頂點含有數個指標指向與其鄰接的邊。右：一個頂點含有數個 指標指向與其鄰接的三角形。	10
圖 3-3：三角形含有三個指標指向其三頂點(v_1, v_2, v_3)，三個指標指向其三邊 (e_1, e_2, e_3)，三個指標指向其相鄰三角形(t_1, t_2, t_3)。	10
圖 3-4：一個邊含有兩個指標指向其端點(v_1, v_2)，兩個指標指向其相鄰三角形 (t_1, t_2)。	11
圖 3-5：Mesh 物件是指向一組 Vertex(頂點)、一組 TrianglePatch (三角形)、一組 TriangleEdge (三角形的邊)、一組邊界以及一組 GhostTriangle 的指標集合。 GhostTriangle 圍繞在網格邊界，並以逆時針方向排序。	12
圖 3-6：翻轉邊運算(edge flip operation)：點 d 在三角形 abc 的外接圓裡，點 a 在 三角形 bcd 的外接圓裡，三角形 abc 和 bcd 都不是 Delaunay 三角形。將邊 bc 改成邊 ad (即刪除三角形 abc 和 bcd，形成三角形 abd 和 acd) 可使三角形符 合 Delaunay 條件。	16
圖 3-7：Divide-and-conquer Delaunay 三角形分割	17
圖 3-8：節點分割的過程	18
圖 3-9：網格結合的過程	19
圖 3-10：檢查三角形(24,33,82)內是否含有非其頂點的節點(圖中數字為節點的 ID)	21
圖 3-11：產生一個新邊 e1 時要檢查其是否與其他邊相交，e2 和 e3 可能與 e1 相 交的邊.....	22
圖 3-12：合成網格 (方框編號表示網格邊界) [19].....	23
圖 3-13：正方體挖去一角的合成網格[19].....	23
圖 3-14：Stella 生成的三維合成網格，使用者輸入的網格大小為 1	24
圖 3-15：(a)Delaunay 三角形網格分割後，邊界不完整 (b)翻轉邊的方式達到邊 界完整 (c)增加節點的方式達到邊界完整[6].....	26
圖 3-16：局部 hr 網格細分的過程：(a) 插入新的節點 (o：新插入的節點) (b) 移 除局部網格 (c) 局部 Delaunay 網格生成 (d) 局部網格和全域網格連接 (e) 將移動的局部節點 (f) 移動局部節點後的網格，粗線表示移動節點後被改變 邊.....	31
圖 3-17：比較初始網格和解網格細分後的電流變化劇烈程度。橫座標 C 是電流變 化劇烈參數的值，定義參見 3.6.1 式(5)；縱座標示三角形網格的百分比。虛	

線表示初始網格；實線表示經過細分的網格。	32
圖 4-1：導線在金屬平面上方距離平面 0.01m 處 (a)初始網格 (b)初始電流分佈	34
圖 4-2：導線在金屬平面上方距離平面 0.01m 處，網格細分過程	35
圖 4-3：每一次網格細分與計算電流分佈分所需的時間	36
圖 4-4：導線接近金屬平面的邊緣 (a)初始網格 (b)初始電流分佈	37
圖 4-5：導線接近金屬平面邊緣，網格細分過程	38
圖 4-6：每一次網格細分與計算電流分佈分所需的時間	39
圖 4-7：電場極化的平面波(E-polarized planar wave)以 45 度入射角向金屬平板入射	40
圖 4-8：平面波入射金屬平板，網格細分和電流分佈（計算電流未忽略遠交互作用）	40
圖 4-9：網格細分的過程與忽略遠交互作用計算的電流分佈，只能觀察出金屬平板邊緣的強電流，干涉條紋不明顯	41
圖 4-10：以圖 4-9(d)的網格，不忽略遠交互作用解力差法方程式計算電流分佈。有些電流密度非常小而且變化不大的區域也被細分（虛線包含之區域）	42



1. 簡介

利用力差法 (method of moments) 解電場積分方程式 (electric field integral equation) 以模擬電磁結構前，必須先將模擬結構分成許多小單元，這個前端處理的步驟叫所網格生成。模擬結構的網格取樣對於解的準確性有極大的影響；尤其當欲模擬的金屬導體與輻射源極為接近，或物體之間有強電磁場耦合時，網格取樣的品質更為重要。

傳統上，利用力差法電磁模擬時，網格的分佈必須經過人為調整。在這種方式之下，只有經驗豐富並且對於其模擬結構有深入瞭解的工程師知道網格應該如何分佈才可能得到較準確的解，以減少調整網格或測試的時間。大多時候，工程師必須經過多次嘗試，藉由數種不同密度、分佈的網格來模擬其結構，並交互比對這些模擬結果以確定解是否準確。然而，當模擬的結構大而複雜，且電流變化劇烈時，即使是經驗豐富的工程師也很難推測其中的電流變化，模擬結果不準的情況會更為嚴重。遇到這種情況時，一個可能的解決辦法是在全部欲模擬結構上取緊密分佈的網格（遠小於波長的網格）。如此雖然可能得到較準確的模擬結果，然而必須相對付出相當高的代價，因為電磁模擬的時間和所需的記憶體空間隨著網格的數量成長。另外，即使願意花大量的電腦資源來模擬，仍然很難確定究竟網格密度應該多大才能夠得到準確的解。因此仍必須花費不少時間做人為的網格調整，以及數次電磁模擬的嘗試。除此之外，如果網格分佈太過緊密，由於浮點數運算的有效位數是有限的，數值計算本身的誤差對解的影響會更為嚴重。基於以上理由，當模擬複雜結構時，不論是人為調整網格或者直接選擇較小的網格密度，不僅沒有效率浪費資源而且得到的模擬結果亦不可靠。

由上可知，目前缺乏一個能夠根據解自動修正網格分佈的方法協助力差法模擬電磁結構。過去幾十年間，這種主動網格修正的方式已廣泛地被應用在有限元素法的電磁模擬，相關的研究亦非常成熟[2][8][16]，Ansoft公司的HFSS軟體[23]即是利用有限元素法結合主動網格修正的電磁模擬例子。然而，將主動網格修正結合力差法電磁模擬方面的研究結果少之又少。綜觀市面常見的利用力差法解電場積分方程的電磁模擬商用軟體，幾乎沒有軟體能夠根據解自動調整網格以得到準確的解。例如，廣泛被高頻電路或天線設計工程師使用的軟體IE3D[24]，其網格的設定完全由人為調整，電磁模擬的過程中，網格不會改變，亦不可能隨著電流分佈自動修正。如果一個工程師利用IE3D模擬一個他不甚瞭解的結構，可能因為網格設定的誤失，得到的非常不準的模擬結果。

本文提出了一套自動生成網格並夠根據積分方程電磁場的解自動細分網格方法，並且實作之。這套方法可以不需要人為判斷，完全自動化產生高品質的網格，提供給力差法為基礎的電磁積分方程解答器，以得到較準確的解。在本文第二章，介紹適應性網格細分(adaptive mesh refinement)的流程和方法，其中包含網格形狀的選擇、網格生成的方法、利用力差法快速解電場積分方程式，以及網格細

分的方法。第三章中，提出基於積分方程電磁場解答器的適應性網格生成和細分的軟體架構以及實作的細節，包含幾何圖形的資料結構、網格生成和細分的演算法。第四章中，我們將這套基於積分方程電磁場解答器的適應性網格生成和細分軟體應用於一些實際的例子。最後，本文的第五章，將前幾章所提的做一節論並提出未來可能之研究方向。



2. 適應性網格細分(Adaptive Mesh Refinement)

基於積分方程電磁場解答器的適應性網格細分演算法如下：

ADAPTIVE-MESH-REFINEMENT (G)

Parameter definition:

G: geometry definition of a refined structure

Function definition:

GENERATE-MESH(G): generate mesh according a geometry definition G

SOLVE-CURRENT(M): solve current distribution using MOM according a mesh M and return the current distribution

REFINE-MESH(M, I): refine a mesh M according the current distribution I.

```
1  M ← GENERATE-MESH(G)
2  I ← SOLVE-CURRENT(M)
3  while not reaching criterion
4    do M ← REFINEMENT-MESH(M, I)
5      I ← SOLVE-CURRENT(M)
```

首先，GENERATE-MESH依據輸入的幾何結構定義G在欲模擬的結構（電路、天線等）表面生成初始網格M，以提供電場積分方程解答器SOLVE-CURRENT產生初始電流分佈I。接著，REFINE-MESH利用之前解的電流分佈I，來判斷模擬結構中，哪些地方電流變化劇烈，哪些地方電流分佈沒什麼變化或者幾乎沒有電流。在電流密度變化非常劇烈或大電流的地方，網格密度應該較高，才能較準確地描述電流分佈。並根據解出來的電流分佈來修正之前的網格分佈。直觀地想法是期望相鄰網格間的電流變化是連續的，所以將電流變化劇烈的地方分割成更小的網格，並移動節點，以達到使電流變化連續的目的。細分過的網格會再次經過場電場積分方程解答器SOLVE-CURRENT的數值計算，得到另一組電流分佈I，然後再次細分網格。以上局部網格細分個過程會一直進行，直到電流分佈收斂（達到連續的條件），或使用使用者自訂的其他停止條件，例如：最大的修正次數、最小的網格邊長、最大的節點或網格數量等。以上流程如圖 5-1所示。

這章會介紹網格生成和細分相關的技術，以及設計應用於力差法電磁模擬分析的適應性網格細分器所需要注意的事項和採取的策略。2.1 節討論網格形狀的選擇；2.2 節介紹一個生成三角形網格的方法—Delaunay三角形分割法；為了快速而有效率地修正網格，必須有快而有效的方法來解電場積分方程，這部分在2.3 節介紹；2.4 探討針對力差法解電場積分方程的應用時，網格的需求；最後，2.5 節介紹數種主要的網格細分技術和其優缺點。

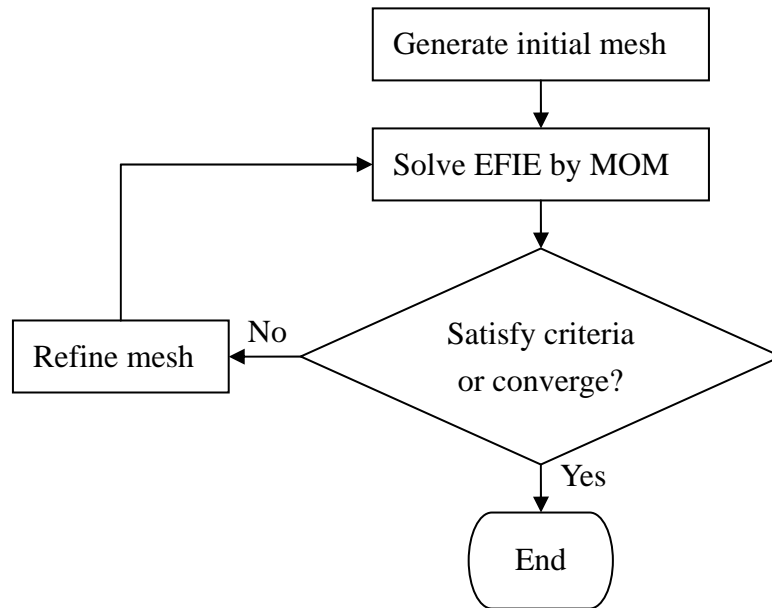


圖 5-1：適應性網格細分的流程

2.1 網格形狀的選擇

關於二維平面上的網格形狀，主要常用的有矩形和三角形兩種形狀，都各有許多演算法可以生成這兩種網格。我們選擇使用三角形，主要是因為在調整網格時，三角形的形變對於網格大小的劇烈改變較不敏感[8]。而且，三角形很有彈性，描述複雜幾何結構和邊界相對比矩型容易[18]。若使用矩形網格描述複雜的結構和邊界，可能在邊界產生太多不必要的小網格。基於以上理由，針對力差法解電場積分方程的應用，三角形是比較理想的選擇。生成三角形網格的方法很多，關於這些方法的介紹可以參考[8]，其中，我們選擇Delaunay三角形分割法來生成三角形網格。在下一節，我們會介紹Delaunay三角形分割法。

2.2 Delaunay 三角形分割法

Delaunay三角形分割法可以應用在任意形狀二維結構的網格分割。Delaunay三角形分割中，所有的三角形網格都要符合Delaunay條件，即任意三角形網格的外接圓內不存在其他不屬於此三角形的節點，如圖 5-2所示。這種方法能夠針對一組給定的節點和邊界產生最佳化的三角形網格。

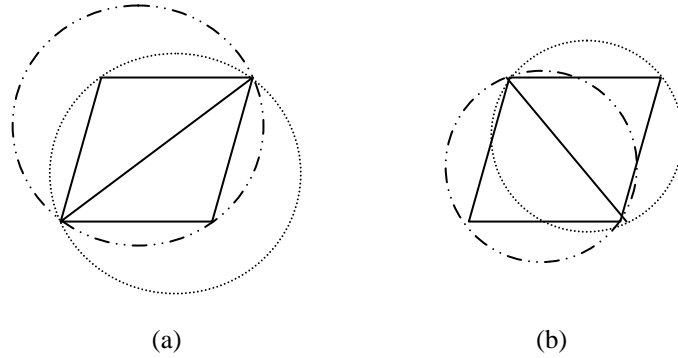


圖 5-2：(a)非 Delaunay 三角形分割 (b)Delaunay 三角形分割

有許多演算法可以產生 Delaunay 三角形分割，包含 sweepline, incremental construction, 和 divide and conquer。在這些演算法中，divide and conquer 是最快的方法，而 incremental construction 是最慢的。以上幾種生成 Delaunay 三角形的方法，在 [20][21][22] 中，有更詳細比較和討論。另外，執行時間的比較可以參考 [3]，Shewchuk 用同樣的資料結構，實作這些不同的演算法，並實際執行以比較其效能。

divide and conquer 是最佳的 Delaunay 三角形分割演算法，若能搭配有效率的資料結構和資料存取的方式，理論上，時間複雜度可達 $O(n \log n)$ ，其中 n 為節點的數量。我們以 divide and conquer Delaunay 三角形分割法為主，加上一點修改，以符合力差法電磁模擬應用的需求，實作自動化網格生成和局部網格修正的程式，關於實作的細節在第三章有詳細的討論和說明。

2.3 力差法解電場積分方程式

力差法可將電場積分方程式離散化成為矩陣方程式，接者，解此矩陣方程式即可得到電流分佈。矩陣方程式可以用高斯消去法 (Gauss elimination) 或迭代法 (iterative method) 求解，前者適用於未知數不多的小問題。未知數很多的大問題通常用迭代法求解，因為迭代法結合一些快速演算法，如快速多極法 (fast multipole method, 簡稱 FMM)，可降低時間複雜度 [11]。

FMM 與傳統的力差法不同點在於，傳統的力差法必須計算所有電流基底函數之間的交互作用，而 FMM 將電流基底函數分群，以一群基底函數為單位，考慮這些群體之間的交互作用，減少電流基底之間交互作用的數量與計算時間。簡而言之，對一群基底而言，由其他群基底散射的場先由這群基底的群體中心接收，再分配都屬於這群基底的基底元素。

在這裡，解電場積分方程式的目的在於提供電流分佈給後續網格細分的步驟，以判斷出結構中需要被細分的區域，不期望解非常準確，但要能快速描述模擬結構中電流的相對變化和大小。我們使用以下方法快速地得到電流的近似解。

利用力差法解電場積分方程式會得到一個矩陣方程式如下：

$$ZI = V \quad (1)$$

其中， Z 矩陣描述電流基底之間的交互作用， I 為電流基底係數（待解的未知數）， V 為外界激發的電源。 Z 矩陣可分成三部分，如下式（1）所示， Z_0 與 Z_1 為近交互作用項（near-interaction terms），其中 Z_0 為近交互作用中的方塊對角項（block-diagonal terms）， Z_1 為近交互作用中的非方塊對角項（non-block-diagonal terms）， Z_2 為遠交互作用項（far-interaction terms）。

$$(Z_0 + Z_1 + Z_2)I = V \quad (2)$$

由於 Z_0 為方塊對角矩陣（block diagonal matrix），可直接求其反矩陣，左右兩邊乘上 Z_0^{-1} 得

$$I + Z_0^{-1}(Z_1 + Z_2)I = V \quad (3)$$

遠交互作用（far-interaction） Z_2 矩陣中的元素值遠小於於 Z_0 與 Z_1 中的元素值，忽略遠交互作用（far-interaction）得

$$(1 + Z_0^{-1}Z_1)I = V \quad (4)$$

忽略遠交互作用（far-interaction）之後，節省許多計算時間，解得的電流分佈雖為近似解，但已能描述電流的相對大小和變化量，並符合適應性網格細分的反饋電流需求。不過在模擬某些結構時，忽略遠交互作用可能造成網格不必要的細分。

2.4 網格品質需求

網格的品質需求直接與應用的問題相關，網格的品質也根據不同的應用而有不同的定義。網格細分的過程中，主要藉由修改網格的密度和分佈來得到更準確的解。但在修改網格的過程中，網格的形狀和大小可能導致嚴重的數值計算誤差，必須避免產生這類會導致嚴重誤差的網格。

一般幾何品質的網格細分主要是修正網格的形狀，例如，對三角形網格而言，修正其最小角度下限及最大角度上限，盡可能生成接近正三角形的網格。實際上，在有限元素法的應用，網格的角度也是一項重要的條件，需避免形成小角度的三角形，因為那會造成嚴重的計算誤差。

然而，在利用力差法解電場積分方程的應用上，網格的需求除上述條件之外，還必須避免太短的邊長，或太小的網格，因為太小的網格會導致低頻崩潰（low-frequency breakdown）[10]。除此之外，網格的分佈須均勻，相鄰的網格間密度變化不能過度劇烈，解才會準確。

為了確保生成的網格符合以上條件，不論在生成初始網格或網格細分的過程中，都必須檢查新產生的網格是否合乎以上條件。

2.5 網格細分的方法

網格細分的方法主要包含分解網格單元形成更小的單元 (h-refinement)，增加基底函數的階層 (p-refinement)，以及移動節點位置 (r-refinement)。其中，分解網格單元形成更小的單元 (h-refinement) 是最容易實作的方法，能夠快速將元素分解成遠小於原始網格大小的元素，如果網格細分的過程中常常需要將特定小區域 (例如：奇異點發生的區域) 分解成遠小於初始大小的網格，這是很好的局部網格細分方法；但若需要細分成極小網格元素的網格區域很大，使用 h-refinement，可能要經過很多次的細分，解才能達到需要的準確度，另外，h-refinement 細分過的網格往往不均勻，所形成網格元素的幾何品質亦較難維持。相較於 h-refinement，r-refinement 最大的優點是可以不用增加網格節點的數量，亦即不必增加計算力差法方程式的未知數數量，而達到使數值計算的解更精確的目標。然而，大部分電磁模擬的應用的結構中，常常有電流突然劇烈的變化的地方，在這些區域，利用 r-refinement 細分網格能增進解的準確性是很有限的。因為，r-refinement 增進解的準確性的極限，被限制在初始節點的數量，如果初始網格形成的時候，節點的數量本來就太少，或網格的大小原本就非常粗糙，花再多時間去調整節點的位置，改變網格的形狀，頂多只是平衡全部解的誤差，整體誤差還是很大。

如果要避免以上網格細分的方法的缺點，一個可能的作法是結合不同的方法。在實作上，我們結合分解網格與節點的移動這兩種方法，對局部電流變化劇烈區域的網格細分。在3.6 節中會詳述實作網格細分的步驟和流程。

3. 架構與實作

3.1 架構概觀

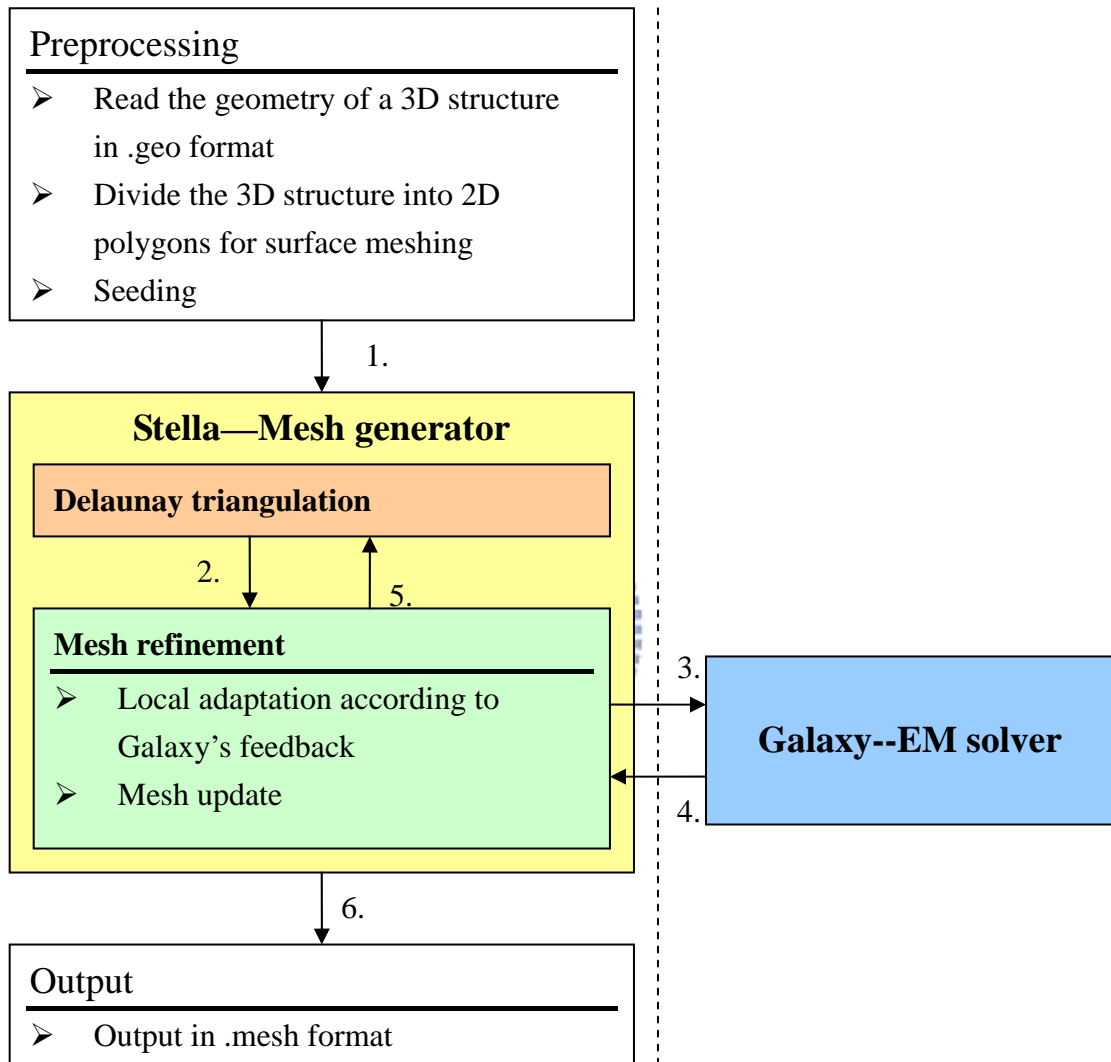


圖 5-3：適應性網格產生器架構概觀

圖 5-3是適應性網格產生器軟體架構概觀。其中，Stella是適應性網格生成和細分的程式，幾何圖形相關的處理都在這部分完成；Galaxy是利用力差法解電場積分方程的程式。Stella中主要含有兩部分模組：Delaunay網格生成模組（Delaunay triangulation module，簡稱DTM）和網格細分模組（mesh refinement module，簡稱MRM）。DTM可針對一組給定的節點和邊界產生受制Delaunay三角形分割（constrained Delaunay triangulation，簡稱CDT）；MRM是針對現有的三角形分割，以及每一個三角形元素上的電流大小，決定需要細分的區域並調整該區域

的網格分佈和密度。在網格調整的過程中，MRM會呼叫DTM執行局部CDT，如圖 5-3 步驟 5。

前置處理 (preprocessing) 的目的是為了確保輸入資料的正確和完整，以及產生初始節點給DTM，若輸入的幾何結構是三維結構，則需將三維多面體的每個平面，旋轉到二維平面。Stella的DTM和MRM只處理二維平面的網格生成和細分，三維平面需旋轉至二維平面，資料輸出前才轉回原來的座標。前置處理完成後，將合法的幾何定義資料傳給Stella (圖 5-3步驟 1)，DTM生成初始網格，傳給MRM (圖 5-3步驟 2)，MRM先呼叫並傳初始網格給Galaxy解電場積分方程回傳電流分佈 (圖 5-3步驟 3、4)，MRM找出需要細分的網格區域，細分該區域並更新網格。圖 5-3步驟 2 到 5 會重複執行直到解收斂或達到使用者設定的其他停止條件 (網格細分的最大次數、網格最小寬度等)，最後輸出網格資料 (圖 5-3步驟 6)。

3.2 資料結構

實作上，有三種基本幾何資料結構：Vertex (頂點)、TrianglePatch (三角形)、TriangleEdge (三角形的邊)。這些資料結構之間，互相以指標 (pointer) 連接，並組成 Mesh (網格)。Vertex 儲存 x 和 y 座標的位置，以及與其相鄰的邊和三角形；TrianglePatch 和 TriangleEdge 分別是兩組關係：TriangleEdge 記錄一個邊與其兩個頂點以及兩相鄰三角形的關係；TrianglePatch 是一個三角形與其頂點、邊、相鄰三角形的關係。

在本節中，3.2.1 和 3.2.2 分別詳述基本的幾何圖形資料結構和網格資料結構。由於以上資料結構的設計參考Shewchuk的Triangle[25]，但有一點修改。本文會在3.2.3 中，比較與Triangle資料結構的異同。

3.2.1 基本幾何資料結構

Vertex為三角形頂點，記錄 x 和 y 座標的位置，以及與其鄰接的邊和三角形。頂點指向鄰邊和相鄰三角形的指標數量為無限多個，分別用兩個串列 (list) 實作之。圖 5-4為一個頂點含有五個鄰接的邊以及五個鄰接的三角形。

TrianglePatch表示一個三角形，含有三個指標分別指向其頂點，三個指標分別指向其頂點的對邊，三個指標分別指向對應頂點的相鄰三角形，這些指標都沿著逆時針方向的順序排列，如圖 5-5所示。

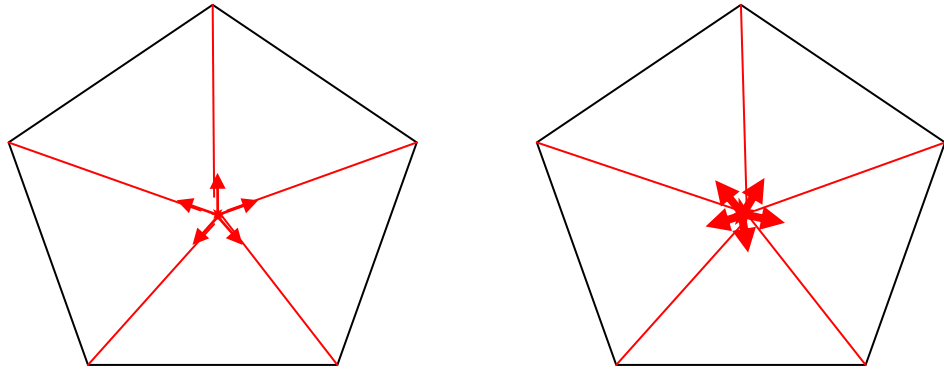


圖 5-4：左：一個頂點含有數個指標指向與其鄰接的邊。右：一個頂點含有數個指標指向與其鄰接的三角形。

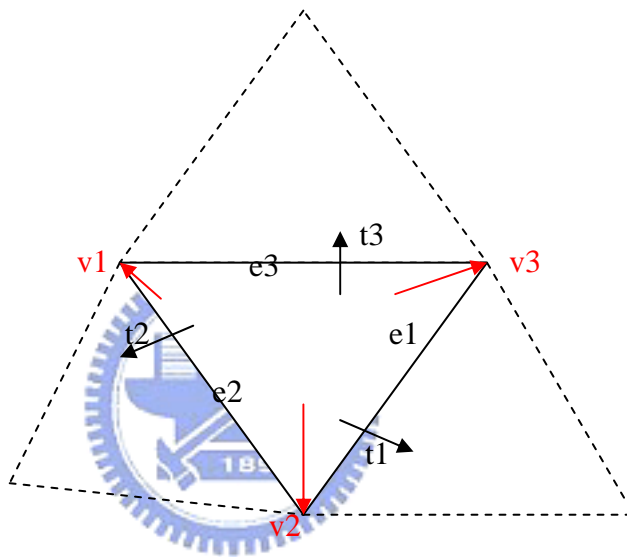


圖 5-5：三角形含有三個指標指向其三頂點(v_1, v_2, v_3)，三個指標指向其三邊(e_1, e_2, e_3)，三個指標指向其相鄰三角形(t_1, t_2, t_3)。

一個不完整的 TrianglePatch (有任意邊或任意頂點尚未指定的 TrianglePatch)，叫做 GhostTriangle。若一個三角形的任意邊沒有相鄰的三角形，其相鄰三角形的指標並非 NULL，而是指向一個 GhostTriangle。因此，在一個有邊界的幾何結構中生成的網格，所有實體的 TrianglePatch (非 GhostTriangle) 都有三個相鄰的 TrianglePatch，而且，只在結構的邊界上可能出現 GhostTriangle。

TriangleEdge表示三角形的邊，含有兩個指標分別指向其兩端點，兩個指標分別指向其相鄰三角形，如圖 5-6所示。有些邊可能沒有相鄰三角形或只有一個相鄰三角形，這些邊相鄰三角形的指標並不是 NULL，而是指向一個 GhostTriangle。因此，在任何網格中，所有的 TriangleEdge 都有兩個相鄰的 TrianglePatch，如果其中一個是 GhostTriangle，則此邊就是網格邊界的一部份。

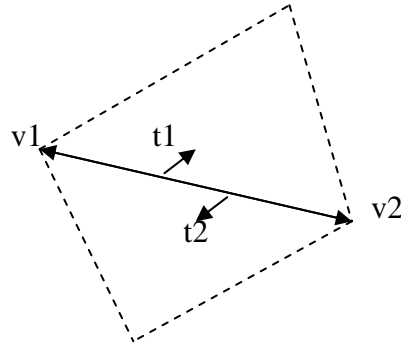


圖 5-6：一個邊含有兩個指標指向其端點(v1,v2)，兩個指標指向其相鄰三角形(t1,t2)。

另外，TriangleEdge 還存有其邊長，但只有在第一次存取時才會計算。如果不曾存取 TriangleEdge 的邊長，邊長的值為 NULL，不會去計算。

3.2.2 網格資料結構

平面直線圖 (planer straight line graph, 簡稱 PSLG) 是一個二維圖形，其中每個線段必須含有兩個端點，而且線段之間只能在其兩端點相交。Mesh (網格) 物件存放的是對一個 PSLG 做三角型分割，所得到的網格資料。

Mesh (網格) 物件針對三角形網格設計，主要由以下五部分組成：一組 Vertex 的集合、結構邊界、一組 TriangleEdge 的集合、一組 TrianglePatch 的集合，以及一組 GhostTriangle 的集合。幾何物件(Vertex, TrianglePatch, TriangleEdge) 之間的關係已經記錄在這些物件本身。邊界和 GhostTriangle 集合依逆時針方向排列；Vertex 集合在 divide-and-conquer Delaunay 三角形分割時，要依照座標位置排列。

圖 5-7 是一個簡單的 Mesh 物件例子，由四個頂點，四個邊界(b1, b2, b3, b4)，兩個三角形網格(t1, t2)，以及四個 GhostTriangle(Gt1, Gt2, Gt3, Gt4) 組成。實作上，Mesh 含有許多不同資料結構的指標串列，分別指向包含在其中的所有幾何物件。

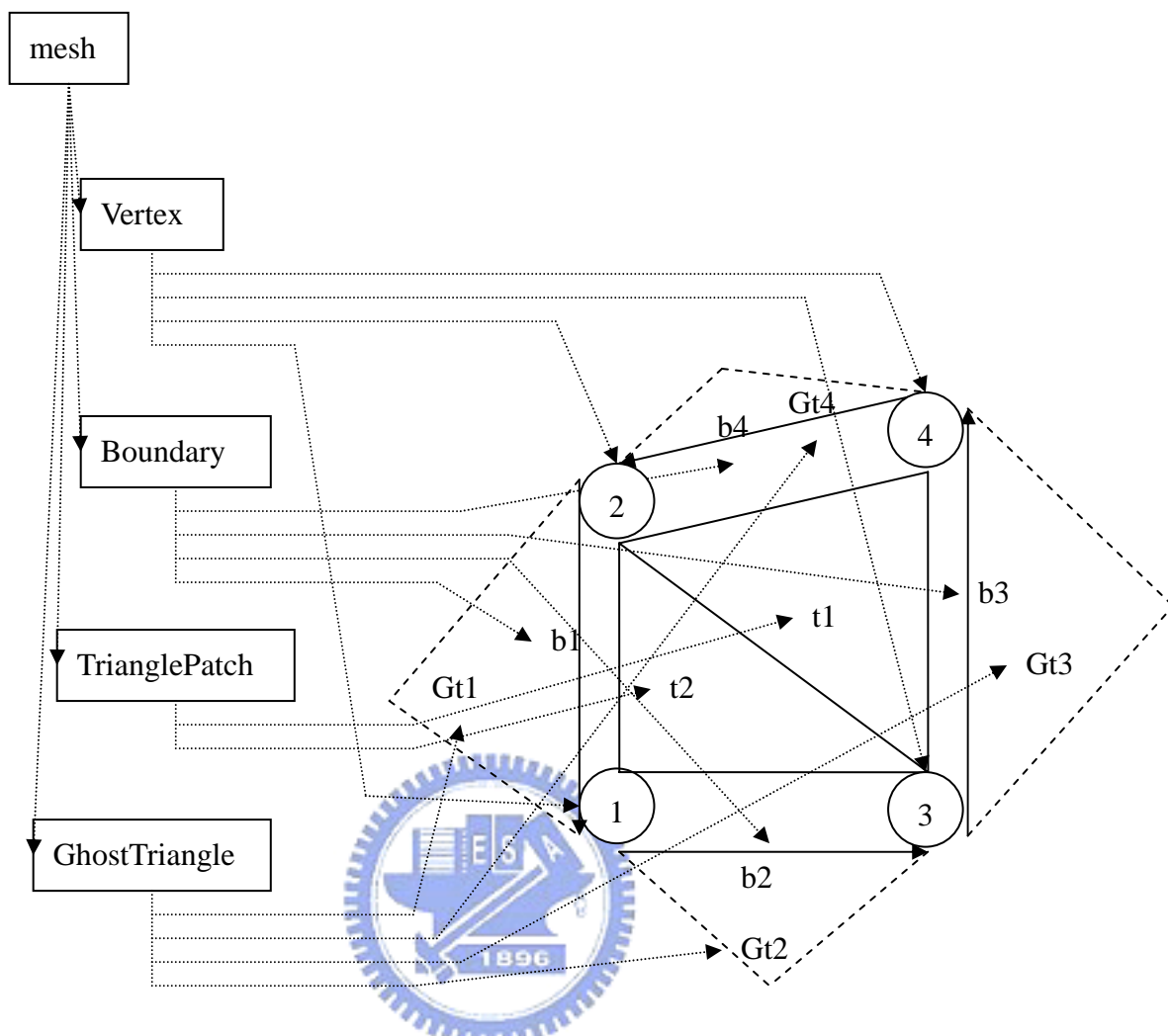


圖 5-7：Mesh 物件是指向一組 Vertex(頂點)、一組 TrianglePatch (三角形)、一組 TriangleEdge (三角形的邊)、一組邊界以及一組 GhostTriangle 的指標集合。GhostTriangle 圍繞在網格邊界，並以逆時針方向排序。

3.2.3 與 Shewchuk 的 Triangle 比較

Triangle 是一個 C 語言程式的二維 Delaunay 三角形網格產生程式，可在 Shewchuk 的 Triangle 網頁免費下載 [25]。Triangle 不僅快速，省記憶體空間，亦功能強大，關於 Triangle 程式的介紹可以參考 [3]。Stella 以 C++ 程式語言設計，物件為基礎的程式，用於二維結構或三維多面體的 Delaunay 三角形網格生成與適應性網格細分。Stella 的設計中，許多部分參考 Triangle 資料結構的設計。在這一節，比較 Stella 與 Triangle 的基本幾何資料結構設計上異同，並討論這些不同設計的優缺點。

Stella 與 Triangle 最大的不同點是：在 Triangle 中，任何幾何資料結構，不管是頂點、邊、三角形，都可以接上任意數量的使用者自訂屬性參數，屬性參數的數量在程式執行時才決定。在 Stella 的實作中，沒有這一部份，所有需要的

屬性參數都已經定義在資料結構中。Triangle 的設計較為彈性，因為它不是為特定應用設計的網格產生器，所以必須含有使用者自訂的參數。實作上，Triangle 僅用一個指標指向一個存放所有的屬性參數的陣列，此陣列在程式執行時才會定義。

其餘部分的資料結構設計，Stella和Triangle是大同小異的。三角形資料結構都含有三個節點、三個邊、三個相鄰的三角形。邊的資料結構都含有兩個節點、兩個相鄰三角形，以及一個記錄此邊是否為邊界的參數。其中有兩點不同，Triangle的邊含有兩個指標分別指向兩組相鄰的邊，而Stella的邊沒有；Triangle不記錄邊的長度，而Stella有，因為Stella在網格細分時，常利用頂點周圍的邊長檢查網格是否均勻分佈。頂點資料結構的部分，Stella和Triangle都含有x和y座標位置，不同的是，Stella的頂點含所有相鄰三角形的資料，而Triangle的頂點只含有一個『可能』包含此頂點的三角形資料；換言之，在Stella程式中，可以迅速找到含有一個頂點的所有三角形，而在Triangle程式中，頂點只記錄一個可能包含此頂點的三角形，如果這筆記錄剛好是要找的三角形，那可以大大地節省搜尋的時間，最壞的情況（這筆三角形的記錄是錯的），還是要搜尋全部的三角形。Stella的頂點含有所有相鄰邊的資料，雖然Triangle的頂點不記錄其相鄰邊，但Triangle邊資料結構中記錄該邊的相鄰邊的功用，其實與Stella的頂點記錄其相鄰邊的功用相同。表格 5-1和表格 5-2列出Stella和Triangle基本幾何資料結構的比較。

此外，Stella亦保留了Triangle中的GhostTriangle（不完整的三角形，只含有一個邊和兩個點），因為GhostTriangle在divide-and-conquer Delaunay 三角形分割演算法的實作中，非常有用。

表格 5-1：比較 Stella 和 Triangle 的基本幾何資料結構

	Shewchuk' s Triangle	Stella
TrianglePatch	相似點	
	皆記錄三個節點、三個邊、三個相鄰的三角形	
	若三角形之任一邊沒有相鄰三角形時，皆有 GhostTriangle	
	相異點	
	任意數量的使用者自訂浮點數屬性或指標(三角形資料結構的大小在程式執行後才決定)	不含任意數量的使用者自訂屬性(三角形資料結構的大小明確指定)
TriangleEdge	相似點	
	皆記錄兩個頂點、兩相鄰三角形，以及此邊是否為邊界	
	相異點	
	兩個指標指向分別指向兩組相鄰的邊	不記兩組相鄰的邊
	不記錄邊長	記錄邊長
Vertex	相似點	
	含有 x 和 y 座標值	
	相異點	
	任意數量的使用者自訂浮點數屬性或指標	不含任意數量的使用者自訂浮點數屬性或指標
	含有一個三角形指標指向『可能』含有此頂點的三角形	含有一串列記錄所有含有此頂點的三角形
	不記錄含有此頂點的邊	含有一串列記錄所有含有此頂點的邊

表格 5-2：比較 Stella 和 Triangle 的基本幾何資料結構 (0：有 X：沒有)

	比較項目	Shewchuk' s Triangle	Stella
TrianglePatch	三個頂點	0	0
	三個邊	0	0
	三個相鄰三角形	0	0
	任意數量使用者 自訂屬性	0	X
TriangleEdge	兩個端點	0	0
	兩個相鄰三角形	0	0
	邊界標示	0	0
	相鄰的邊	0	X
	邊長	X	0
Vertex	x 和 y 座標	0	0
	任意數量使用者 自訂屬性	0	X
	邊界標示	0	X
	相鄰三角形	X	0

3.3 Divide-and-Conquer Delaunay 三角形分割

3.3.1 演算法

以下divide-and-conquer Delaunay三角形分割的演算法與Guibas和Stolfi在[5]所提出的相近，唯資料結構參考Shewchuk[3]，以三角形為主的資料結構。divide-and-conquer Delaunay三角形分割的流程如圖 5-9所示。輸入幾何結構的定義後，將所有的節點分成兩部分，接著，用遞回的方式分別處理這兩部分，產生網格，最後，將處理好的部分網格結合起來。

輸入欲分割的幾何結構後，需要先經過一些前置處理的步驟，才能執行divide-and-conquer Delaunay 三角形分割。首先，需對所有節點依照 x 座標排序，x 座標相同的節點之間再依照 y 座標排序，接著，移除相同座標的節點和含有相同兩端點的邊，這些相同的節點和邊會使之後的 Delaunay 三角形分割程式無法正常執行。除此之外，還需調整圖形，確保符合 PSLG 的條件，若任何邊上除了兩端點之外，含有其他節點，則將此邊分成數段僅含有兩端點的邊。

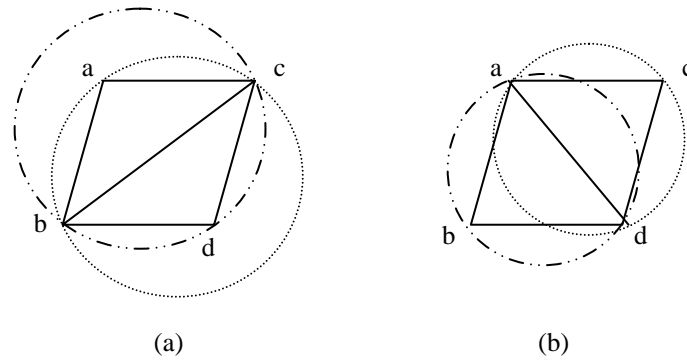


圖 5-8：翻轉邊運算(edge flip operation)：點 d 在三角形 abc 的外接圓裡，點 a 在三角形 bcd 的外接圓裡，三角形 abc 和 bcd 都不是 Delaunay 三角形。將邊 bc 改成邊 ad（即刪除三角形 abc 和 bcd，形成三角形 abd 和 acd）可使三角形符合 Delaunay 條件。

用遞迴的方式將排序過的節點資料均分成兩部分 L 和 R，當其中一部分的節點數量在三個以下，或所有該區域的節點在同一直線上，才將這個部分的節點連接起來，形成一個三角形或直線。緊接著，把兩個已經生成網格的小部分結合起來，直到所有的部分網格都合成一個網格。

圖 5-10和圖 5-11是一個簡單的divide-and-conquer Delaunay三角形分割生成網格例子。這個例子連接 10 個節點，形成Delaunay三角形網格。在圖 5-10中，把排序過的節點分成兩部分，此時每個部分有五個節點（圖 5-10上），再分別將這五個節點再分成兩部分，則每個小部分有兩個或三個節點（圖 5-10中）。由於每個小部分已經少於四個節點，只需直接將每個小部分各自的節點連接成三角形或直線（圖 5-10下），就生成這些小區域的網格。其中，灰色的三角形是 GhostTriangle，圍繞著給各區域網格的邊界。GhostTriangle對於結合兩組相鄰區域網格時很有幫助，僅需將兩區域接口上的GhostTriangle一組一組拼在一起，就像拼合兩個齒輪的齒鋸一樣，如圖 5-11。接合的過程中，有些邊也許需要翻轉（edge flip，如圖 5-8），以確保網格中的所有三角形符合Delaunay條件。

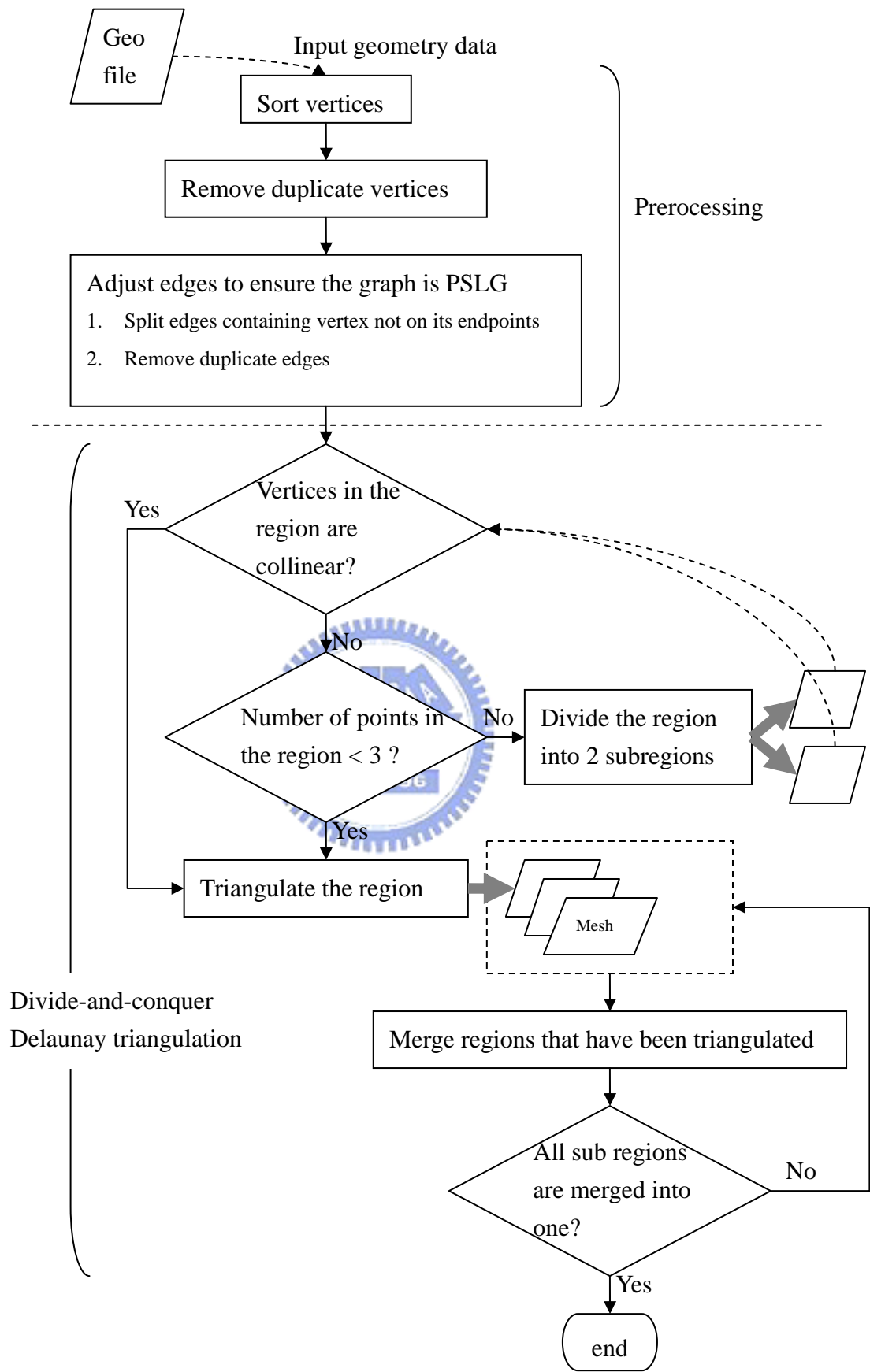
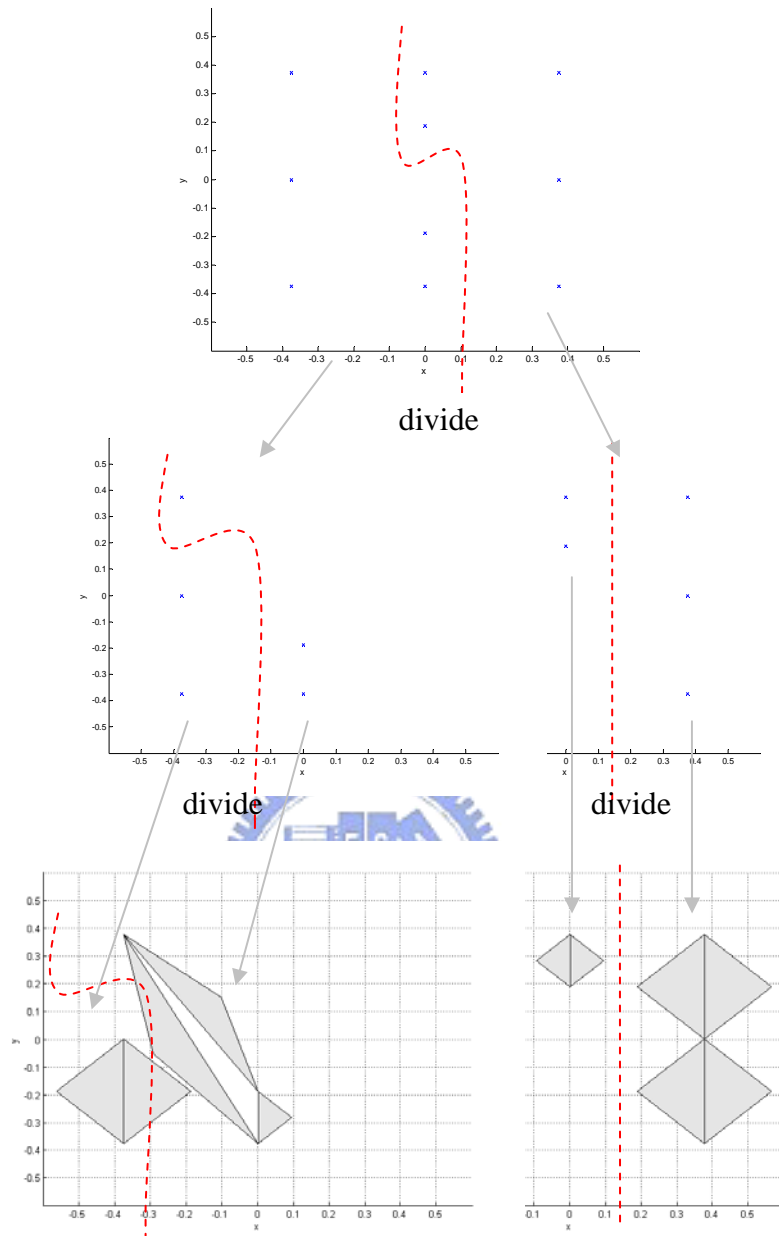


圖 5-9：Divide-and-conquer Delaunay 三角形分割



If the number of vertices in the subregion is smaller than 4 or vertices in the subregion are collinear, connect the vertices in this region to form a triangle or edges.

圖 5-10：節點分割的過程

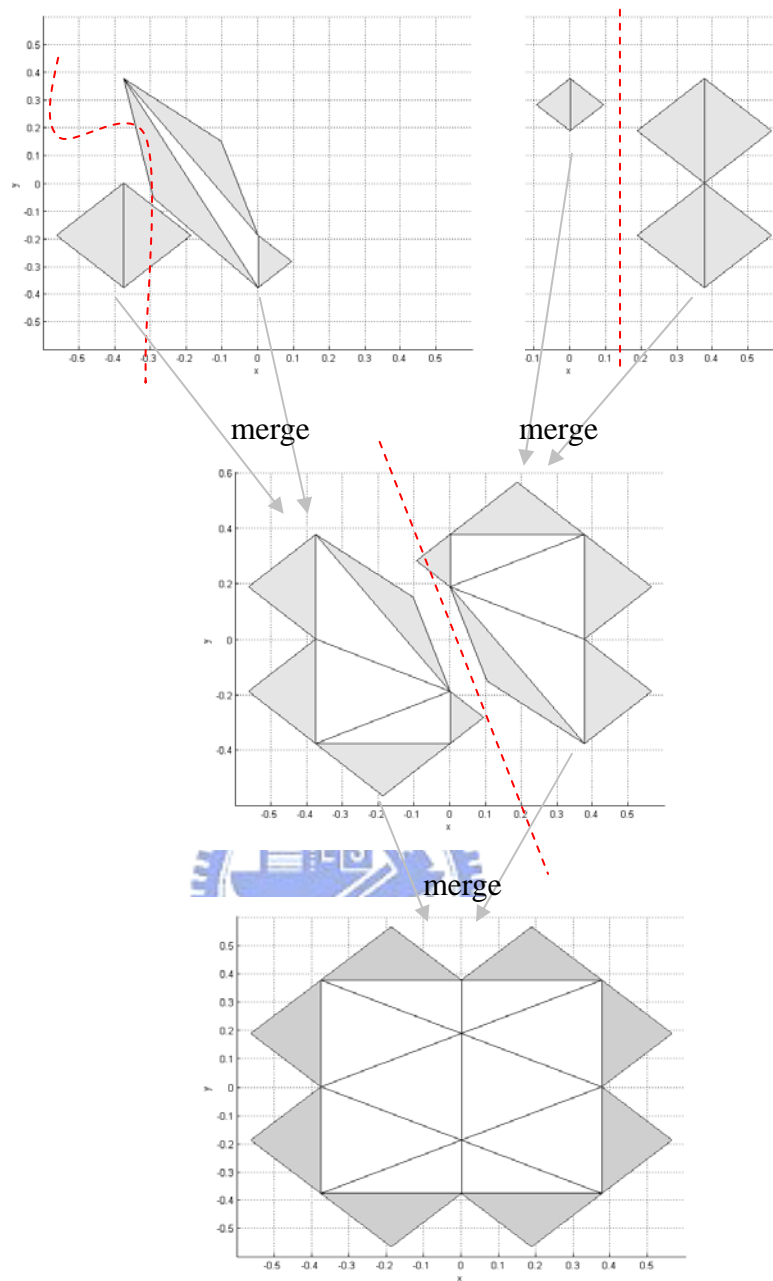


圖 5-11：網格結合的過程

3.3.2 增進效能的設計

對資料節點排序，不僅是 divide-and-conquer Delaunay 三角形分割的必要步驟，更能提供許多增進效能的設計，以減少程式執行的時間。在本節中，我們提出一些 Stella 網格生成演算法中增進效能的設計，這些設計雖然無法改變 divide-and-conquer Delaunay 三角形分割演算法的時間複雜度，但能縮短數倍程式執行的時間，在實際應用上很有幫助。

在Guibas和Stolfi提出的divided-and-conquer Delaunay三角形分割演算法

中 [5]，遞迴分割的步驟直到分割後的小區域中，節點的數量少於四個，才會停止。然而，在Stella的實作中，若一個區域中的節點全部在同一條直線上，或節點的數量少於四個，分割的動作都會停止，並在該區域生成網格。在一些情況中，例如：同一分割區域的節點都在同一直線上，這樣的改進可以較早結束分割的動作，並開始生成區域網格，增進程式的效率。

另外，Delaunay三角形分割的過程中，常常需要檢查形成的三角形網格是否覆蓋住其他的節點。如果每一次形成一個新的三角形網格，都要去檢查所有節點是否在此三角形網格之內，這是浮點數乘法運算，相當費時。排序的節點，可以提供一個快速的方法做這項檢查，使檢查節點是否在此三角形網格之內做浮點數運算時間的次數和時間從 $O(n)$ （其中， n 為網格節點的數量）減至常數時間。假想一個矩型，長含寬分別沿著 x 方向和 y 方向，外接於欲檢查的三角形，只有矩型內的節點才可能在三角形之內，由於節點已排序，要找出在此矩型內的節點非常快。如圖 5-12所示，假設欲檢查一個連接節點(24, 33, 82)的三角形是否覆蓋到其他節點，所有的節點已經沿著 x 座標排序過，因此不需考慮順序在節點 24 之前及節點 33 之後的所有節點。如圖 5-12上圖所示，除了 x 座標在點 24 到點 33 之間的節點以外，其他節點全部蓋住不看。剩下來的節點，只考慮 y 座標介於點 33 和點 82 的 y 座標間的節點，在圖 5-12下圖中，只需要計算ID為 29, 81, 85 這三個點是否在三角形(24, 33, 82)之中。不管全部網格的數量有多少，需要真正經過浮點數計算才能決定是否在一個三角形網格內部的節點其實非常少。

類似的做法，也可應用於檢查一個邊是否與其他邊相交。假想一個矩型，長含寬分別沿著 x 方向和 y 方向，外接於欲檢查的邊（即此邊為假想矩型的對角線之一）。只有通過假想矩型的邊才可能與此邊相交，如圖 5-13所示。由於節點已經排序，要找出可能相交的候選邊（candidates）只需要檢查此邊兩端點的順序。以圖 5-13為例，只有通過假想矩型的候選邊 e_2 和 e_3 需要經過浮點數運算來決定是否與邊 e_1 相交。

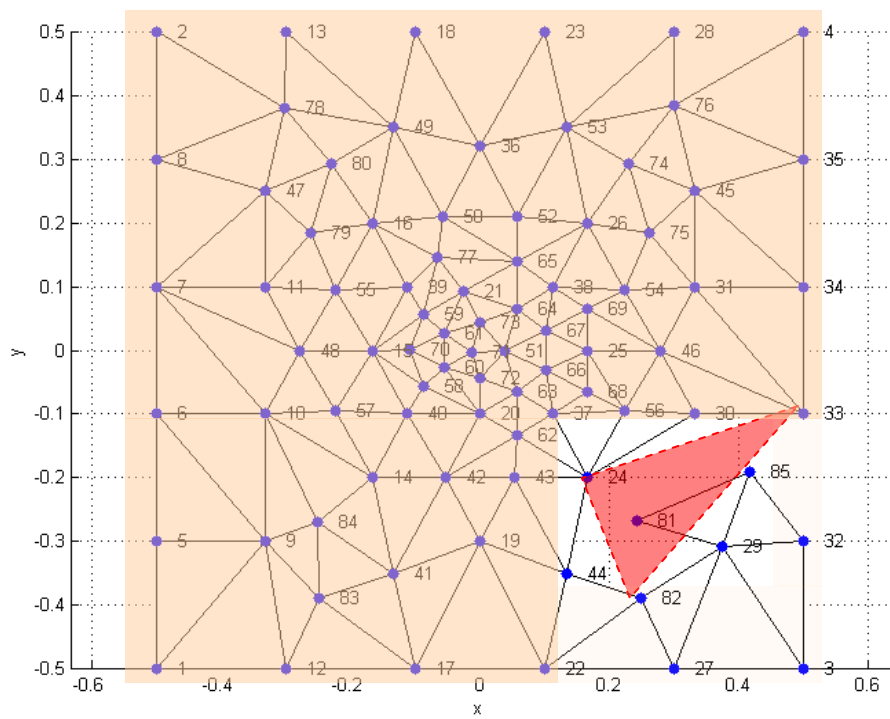
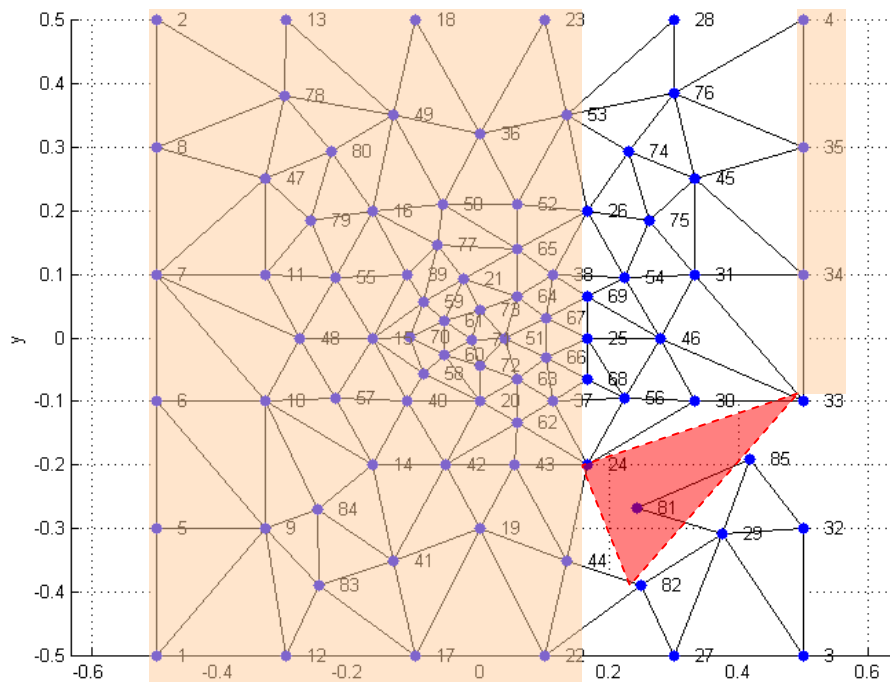


圖 5-12：檢查三角形(24, 33, 82)內是否含有非其頂點的節點（圖中數字為節點的 ID）

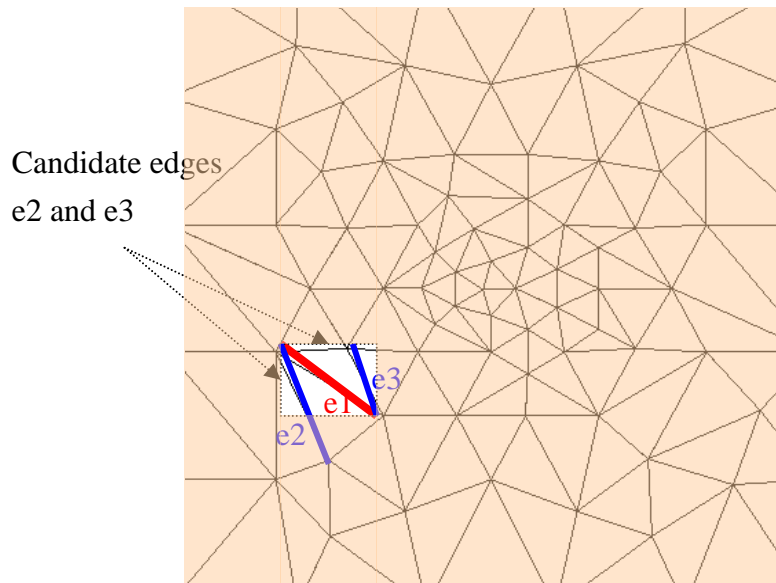


圖 5-13：產生一個新邊 e1 時要檢查其是否與其他邊相交，e2 和 e3 可能與 e1 相交的邊

表格 5-3：比較檢查所有邊和檢查部分邊，Delaunay 三角形分割所需的時間。

Number of meshed vertices	A. Execution time for checking all edges (s)	B. Execution time for only checking part of the edges (s)	Ratio of $\frac{A}{B}$
100	0.156	0.047	3.4
1000	3.657	0.390	9.5
10000	28.390	4.878	5.7
100000	413.254	56.279	7.3

上述不論是檢查三角形內是否含有節點，或檢查一個邊是否與其他邊相交的改善，時間複雜度還是一樣為 $O(n)$ (n 為節點的數量)，主要節省的是浮點數運算的時間，排除不必做浮點數計算就能淘汰的資料。不管結構多大，節點數多少，必須經過浮點數運算來檢查邊是否相交或節點是否在三角形中的時間大約是常數。表格 5-3 中，比較上述改進前後，Delaunay 三角形分割所需的計算時間。由實驗結果看，整個 Delaunay 三角形分割所需的時間可以因此改進減至原來的 1/3 到 1/10，而且，節點的數量越多，節省的時間也越多。

3.3.3 三維結構表面網格生成

三維結構若能以數個空間中的平面來描述，就不難透過分別處理這些空間中平面，來產生三維結構表面網格。作法為分別將這些空間中的平面轉到二維空間（ xy 平面上），以二維網格生成的方式在這些平面上產生網格後，再將整個平面轉回原來的座標位置，即生成三維結構的表面網格。要實作以上三維網格生成，必須記錄每一個三維平面的面積向量使平面能在空間中轉動，以及不同平面之間相接的邊。圖 5-15 是一個正方體挖去一角的立體結構，可以分別以許多空間中的平面描述之，這些平面間以共同的邊界相接。

兩個不同平面的網格集合起來，叫做合成網格（composite mesh）。圖 5-14 是一個簡單的二維合成網格例子，網格 A 和 B 各含有兩個三角形，邊界 7 是 A 網格和 B 網格相接的邊。圖 5-16 是 Stella 生成三維合成網格的例子。

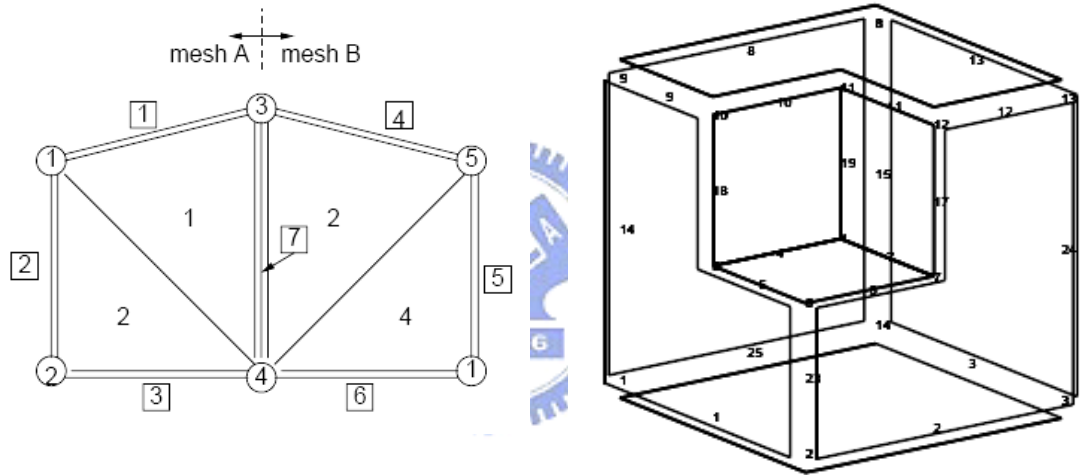


圖 5-14：合成網格（方框編號表示網格邊界） 圖 5-15：正方體挖去一角的合成網格[19]
[19]

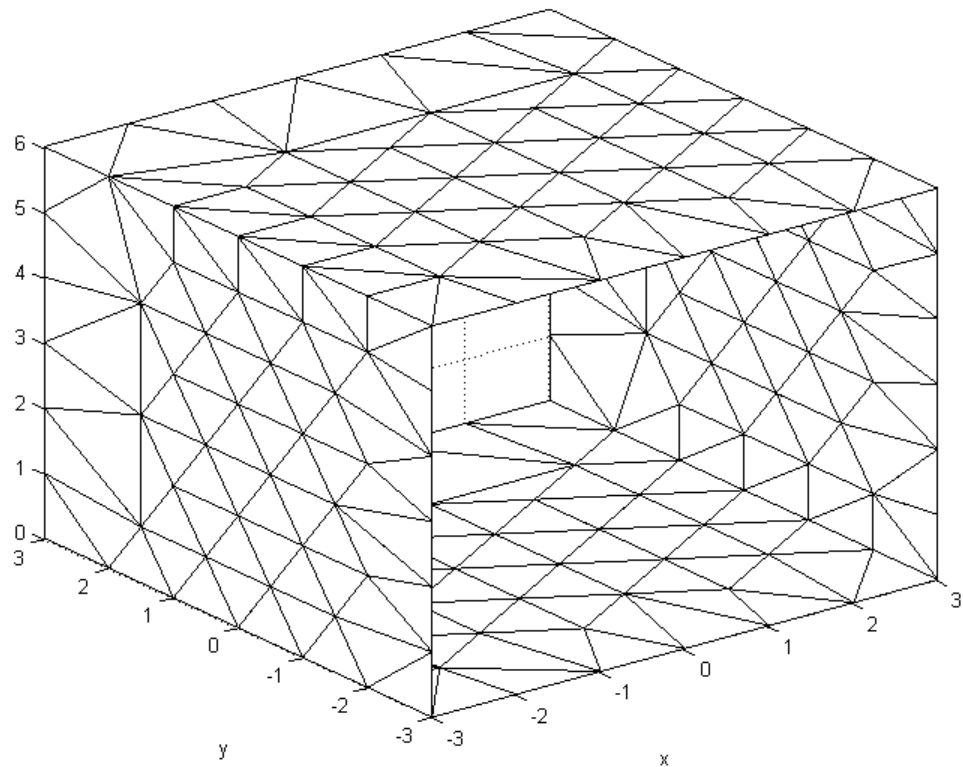


圖 5-16：Stella 生成的三維合成網格，使用者輸入的網格大小為 1



3.4 初始節點生成

前述 Delaunay 三角形網格分割演算法，是針對一組已知的節點生成網格，至於節點應該如何分佈，並沒有規定。一個完整的網格生成器，也應包含前端自動產生初始節點的部分。在利用力差法解電場積分方程的電磁模擬應用上，網格分佈品質的好壞，直接與結構中電流分佈有關。因此，不可能在不做任何電流分佈的計算或估計的前提下，設計一個演算法能夠決定最佳的網格分佈。鑑於上述理由，初始節點生成程式的設計目標並非期望一開始就產生最佳節點分佈，而是有效率地在結構中灑上均勻分佈的網格。初始網格分佈雖然粗糙，但後續網格細分 (mesh refinement) 的過程可漸漸改善網格品質。

初始節點生成的方法設計上很簡單，演算法如下：

SEEDING(G, d)

Parameter definition:

G : geometry definition

d : user-specified desired initial grid edge length

Function definition:

SEED-EDGE(e, d): seed on the edge e with the distance $\cong d$

SEED-INSIDE(G, d): seed inside the geometry defined region G with the distance $\cong d$ and finally return the seeded nodes

```
1  for each boundary edge  $e \in G$ 
2    do SEED-EDGE( $e, d$ )
3   $V \leftarrow$  SEED-INSIDE( $G, d$ )
4  for each node  $v \in V$ 
5    do for each boundary edge  $e \in G$ 
6      do if  $v$  near  $e$ 
7      then remove  $v$ 
```

使用者輸入的網格邊長為 d ，先在邊界灑上間距大約是 d 的節點，接著，在結構內部亦灑上間距大約是 d 的節點，同時移除結構中太靠近邊界的點。這個方法可以有效率地產生均勻分佈的節點，根據這些節點生成的Delaunay三角形網格的幾何品質也不錯。節點的分佈若是均勻的，就不會形成含有小角度的病態三角形網格。圖 5-16是利用以上方法生成的節點，Delaunay三角形分割產生網格的例子。

3.5 邊界完整性

Delaunay三角形分割演算法本身並不保證能符合結構的邊界。換言之，若結構中有邊界，需要額外處理以確保邊界的完整性。一般來說，處理這類的問題有兩種做法，一種是邊界優先法 (p priori approach)，另一種是邊界其次法

(posteriori approach)。在第一個邊界優先法中，邊界會自然生成，使用這種方式處理有邊界結構的Delaunay三角形網格生成時，初始節點的位置分佈就已決定邊界必然存在。另一個邊界其次法中，有些邊界元素可能在Delaunay三角形分割後消失，需要經過其他操作強制讓這些邊界存在。讓邊界強制存在的方法主要有兩種。其中一種是將邊界分成許多小段，直接插入已形成的網格中，如圖 5-17(c)所示；另一種是對所有與邊界相交的格線做翻轉，直到沒有任何邊與邊界相交，再將邊界插入，如圖 5-17(b)所示。

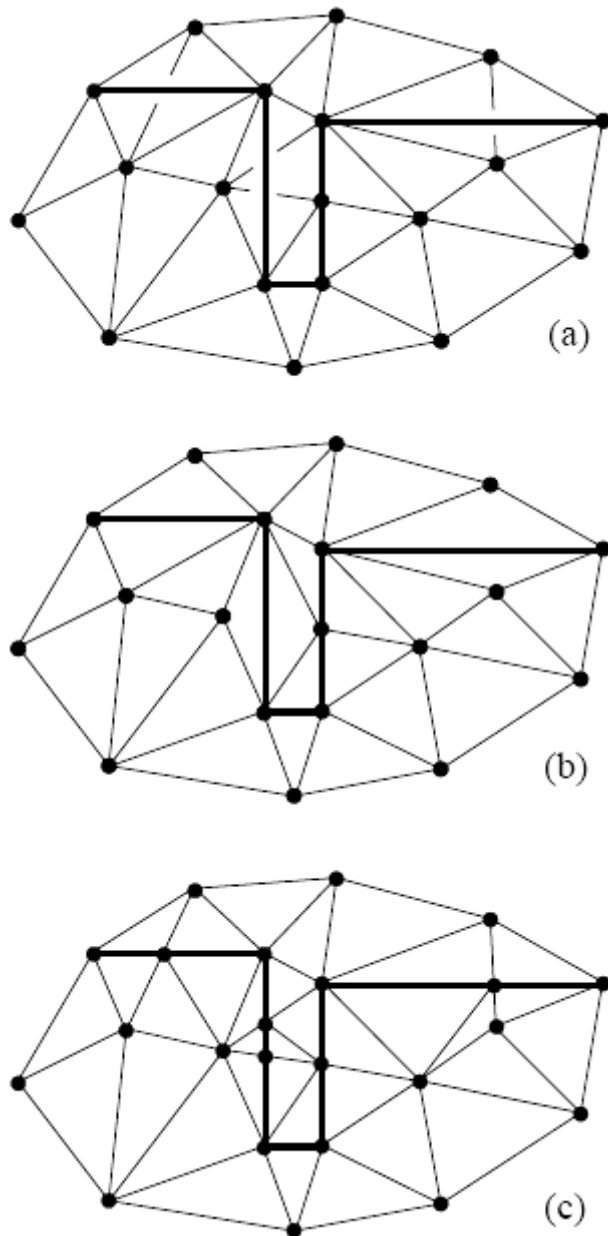


圖 5-17: (a)Delaunay三角形網格分割後，邊界不完整 (b)翻轉邊的方式達到邊界完整 (c)增加節點的方式達到邊界完整[6]

實作上，邊界優先法 (p priori approach) 需要花許多時間處理初始節點的生成。網格細分時，因為常需要調整網格，使用邊界優先法來維持邊界的完整性更加費時。相較之下，邊界其次法比較容易實作，加上一點改進也能達到不錯的效率。

改進的邊界其次法中，所有的邊界元素在執行三角形網格分割之前就已經存在，但是隱形看不見的。在每次生成新的邊或執行翻轉邊運算(圖 5-8edge flip operation)時，都先檢查邊是否已經存在，若發現即將生成的邊或翻轉後的邊是隱形的邊界元素，則將此邊界元素顯現出來，顯現出來的邊界元素從此不再改變，即使它需要翻轉邊以符合Delaunay條件的情況也一樣。以上的改善使得當

Delaunay三角形網格分割執行完後，大部分的邊界元素已經存在，少數仍然隱形的邊界元素再經過翻轉邊運算(edge flip operator)，如圖 5-17(b)所示，邊界的完整性就確定了。

3.6 網格細分 (Mesh Refinement)

網格細分的演算法可根據力差法解電場積分方程式的解，調整局部需要細分的網格，以達到更準確的電流計算答案。首先，須對欲模擬的結構生成初始網格，提供給 Galaxy 解電流分佈，電流解不必很準，但要能描述網格間相對的電流分佈和大小，以提供後續網格細分的步驟。決定網格細分的區域後，插入新的節點並移除原先該區域的網格，重新在該區域執行局部 Delaunay 三角形網格生成 (h-refinement)。最後，移動局部節點 (r-refinement) 使網格分佈較均勻。每一次區域網格改變時，都要對變動的網格檢查 Delaunay 條件，以翻轉邊(edge flip)的運算使網格符合 Delaunay 條件。網格細分的過程重複執行，直到達到停止條件。

網格細分的演算法如下：



REFINE-MESH(M)

Parameter definition:

M: original mesh

Function definition:

ENCLOSE-REFINED-REGION(M,I): enclose the refined region according a mesh M and its current distribution I, finally return the enclosed region

GENERATE-MESH(G): generate a mesh of the enclosed region R and return the mesh

GALAXY(M): solve current distribution using MOM according the mesh M and return the current distribution

INSERT-VERTICES(R): insert vertices into the region R

MERGE-MESH(M,MR): merge two mesh M and MR and return edges connecting to these two mesh

LEGALIZED(e): flip the edge e until it satisfy the Delaunay criterion

SMOOTH-MESH(M): smooth mesh M and return the changed edges

```
1  I ← GALAXY(M)
2  R ← ENCLOSE-REFINED-REGION(M,I)
3  INSERT-VERTICES(R)
4  for each triangle patch  $t \in R$ 
5    do remove t
6  MR ← GENERATE-MESH(R)
7  E ← MERGE-MESH(M,MR)
8  for each edge  $e \in E$ 
9    do LEGALIZED(e)
10 E ← SMOOTH-MESH(M)
11 for each edge  $e \in E$ 
12  do LEGALIZED(e)
```

3.6.1 決定網格細分的區域

在此，我們針對電流連續性做網格細分。相鄰的網格間，若電流密度相差很大，表示該區域的電流變化劇烈，需要更密的網格才能清楚描述電流分佈。網格細分前，由前一次 Galaxy 計算的電流分佈決定相對電流變化劇烈的區域。使用以下公式來描述一個三角形網格元素上電流變化劇烈的程度參數 C ：

$$C = w(I)\Delta I \quad (5)$$

其中，

$$\Delta I = \frac{\sum_{j=1}^n |I - I_{neighbor_j}|}{n} \quad (6)$$

I 表示一個三角形網格上的平均電流密度的大小， n 表示此三角形網格相鄰三角形的數量（對於邊界三角形網格 $n=2$ ，其他三角形網格 $n=3$ ）， $I_{neighbor_j}$ 表示此三角形網格的第 j 個相鄰三角形， $w(I)$ 是一個與此網格上電流密度大小相關的函數，其函數值隨著電流密度增大而增大。 ΔI 描述的是電流密度不連續的程度，乘上 $w(I)$ 函數的原因是，電流變化劇烈的區域以及強電流區域都可能有較大的誤差，該處的網格需要細分。 C 參數越大的網格，表示誤差可能越大，越需要被細分。在網格細分時，就是挑出 C 參數大的網格區域，進行局部網格細分的動作。

電流密度變化的部分其實也可以用下式表示：

$$\Delta I = \frac{\sqrt{\sum_{j=1}^n (I - I_{neighbor_j})^2}}{n} \quad (7)$$

為了計算速度的考量，不選擇用式(7)，而以式(6)來表示電流變化的程度。

以上誤差判斷的方式只是一個局部誤差指示器 (local error indicator)：一個網格元素上電流的誤差是與該網格中所有網格元素相對比較而得。這種局部誤差指示器計算快速而且節省記憶體空間。另外有一種較為嚴謹的全域誤差指示器 (global error indicator)：誤差的大小必須與之前每一次網格修正產生的所有網格以及計算的電流大小比較來決定，不僅實作上較為複雜，且執行起來亦相當費時費空間。鑑於以上原因，我們仍選擇快而省空間的局部誤差指示器來決定網格細分的區域。

3.6.2 局部 Delaunay 三角形網格 hr 細分

(Local Delaunay hr-Refinement)

Delaunay 三角形網格分割的技術亦適用在局部網格的細分。只需定義局部的節點集合，就可以用 Delaunay 三角形網格分割的方法在該處生成局部網格，再將這些局部網格與原來的網格結合即可。關於局部 Delaunay 三角形網格分割的演算法可以參考 [18]。

然而，使用以上演算法做網格細分會產生一些問題：網格只在局部區域符合 Delaunay 條件，而且，相臨網格之間的網格大小不均勻，以上兩個問題常常會出現在局部網格和全域網格的交接處。我們對 [18] 提出的區域 Delaunay 網格細分演算法做兩點改進以改善細分後的網格品質：第一是調整節點的位置 (r-refinement)；第二是在區域與全域網格的交界處做翻轉邊運算，使這些交界的邊符合 Delaunay 條件。改進後的局部 Delaunay 三角形網格 hr 細分演算法能使細分後的網格較為均勻。

局部 Delaunay 三角形網格 hr 細分的步驟如下 (同時參考圖 5-18)：

1. 插入新的節點：在需要被細分的三角形網格上插入新的節點。若三角形最長的邊為邊界，則直接在最長邊的中點插入節點；內部的三角形則採用以下規則：對一個銳角三角形，新的節點置於其外心位置；鈍角三角形，新的節點置於其最長邊的中點 (圖 5-18a)。
2. 決定重新生成網格的局部區域：定義一個區域，使所有被插入新節點的三角形網格都在此區域中，並移除該區域原有的網格 (圖 5-18b)。
3. 對上述局部區域做 Delaunay 三角形網格生成 (圖 5-18c, d)。
4. 確保局部區域的邊界符合 Delaunay 條件：若局部網格細分區域與全域網格的交界邊不符合 Delaunay 條件，做翻轉邊運算使其符合 Delaunay 條件 (圖 5-18e)。
5. 移動局部節點的位置：為了得到較均勻的網格分佈，移動部分節點的位置 (圖 5-18f)。新的節點位置為其周圍所有節點的中點。

$$r' = \frac{\sum_{i=1}^n r_{\text{adjacent}_i}}{n} \quad (8)$$

r' 為新的節點位置， r_{adjacent_i} 表示只以一個邊與此節點相連的周圍節點，其中， $i = 1, \dots, n$ ， n 為周圍節點的數量。

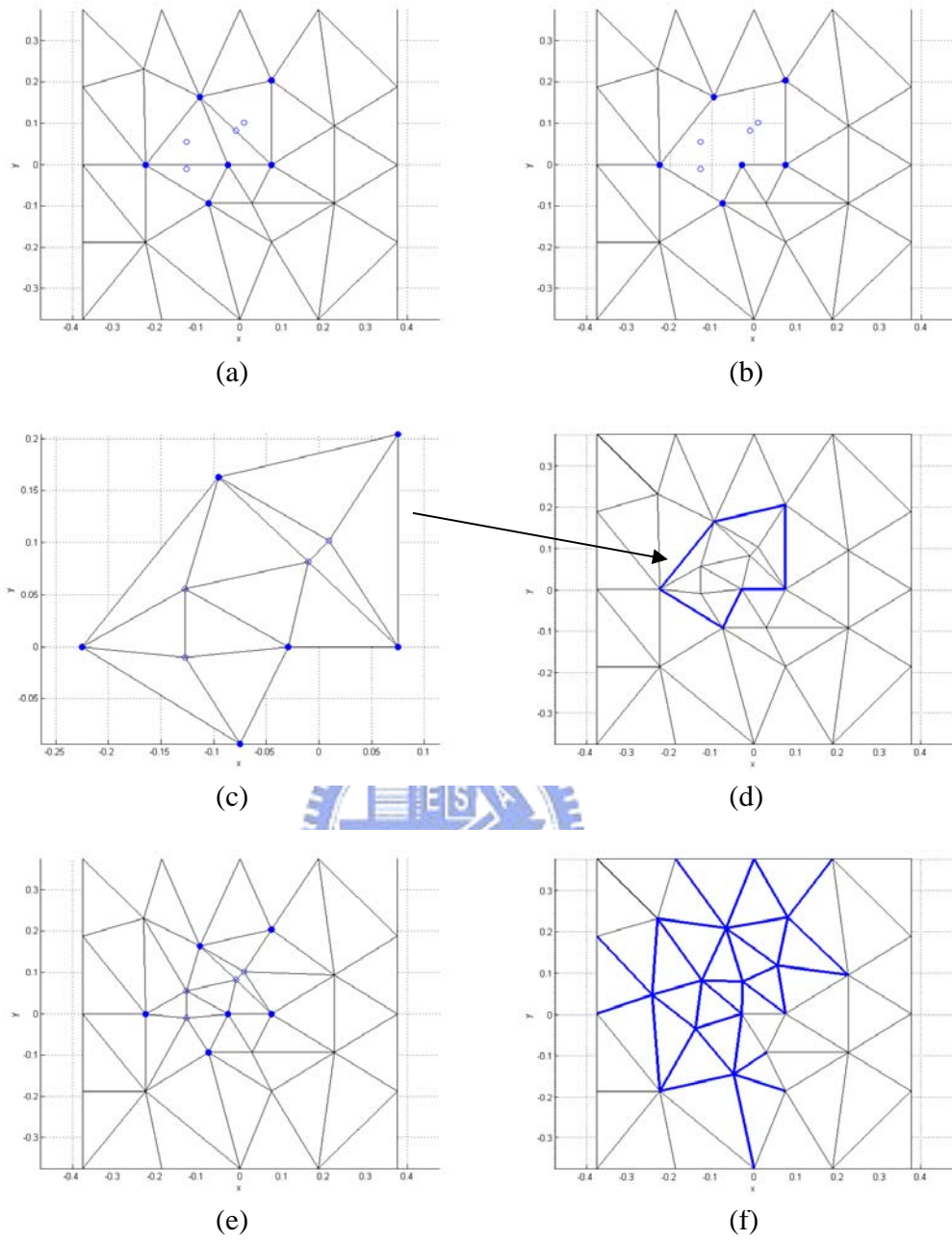


圖 5-18:局部 hr 網格細分的過程:(a) 插入新的節點(o:新插入的節點) (b) 移除局部網格 (c) 局部 Delaunay 網格生成 (d) 局部網格和全域網格連接 (e) 將移動的局部節點 (f) 移動局部節點後的網格，粗線表示移動節點後被改變邊

[18]中提出的區域Delaunay網格細分演算法，只做了以上步驟 1 到 3 (圖 5-18a-d)。我們加上步驟 4 和 5，可使網格分佈更均勻，觀察圖 5-18f的網格分佈，比圖 5-18d的網格分佈更為均勻。

3.6.3 網格細分終止條件 (Criteria)

由於網格細分是依據電流的連續性，故當電流分佈達到『連續』的標準的時候，網格細分的動作就終止。如圖 5-19所示，網格細分的動作會挑出3.6.1 式(5)中定義的C參數較大的網格進行細分的動作，如此一來，含有C值大的網格比例會減少。當細分網格無法再減少C值大的元素比例，即達到收斂條件，並停止網格細分。除上述收斂條件之外，當C值小於某值的網格元素達到一定的比例，或C值大於某值的網格元素少於一定的比例，網格細分亦會停止。

有些情況下，例如電流奇異點，電流理論上是無限大，即使網格不斷細分，電流的大小永遠不會收斂。此時，網格大小若已達到使用者輸入或程式預設的最小網格限制，將不會再細分。另外，當網格數達到使用者輸入的網格最大數量或網格細分的次數達到使用者輸入的網格細分最大次數，也會終止網格細分的動作。

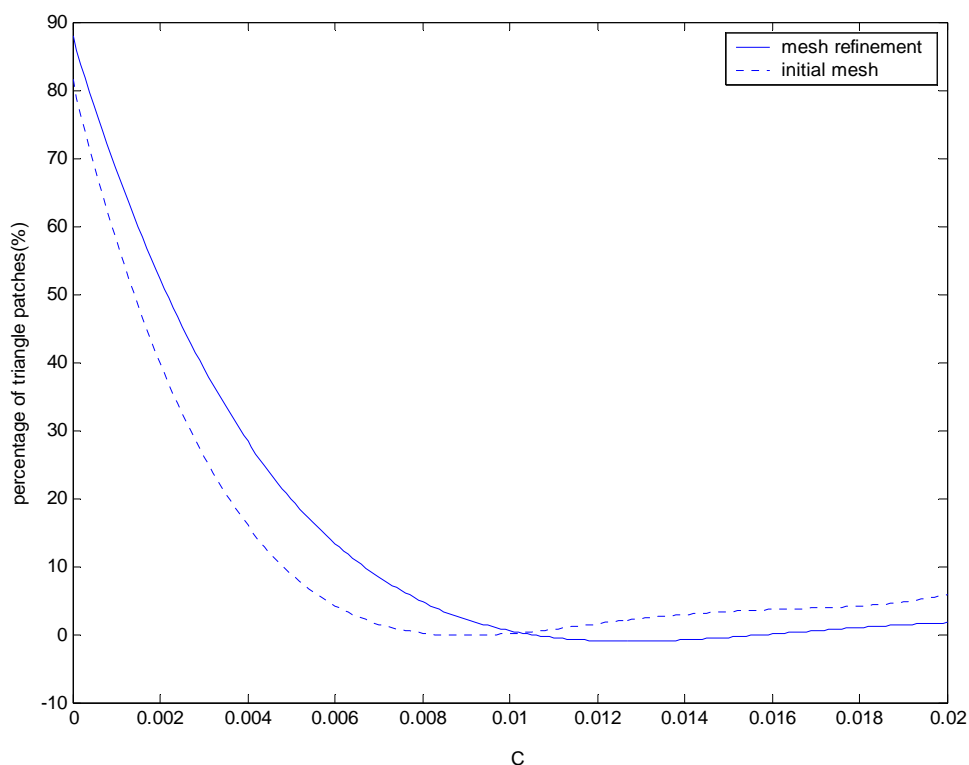


圖 5-19：比較初始網格和解網格細分後的電流變化劇烈程度。橫座標C是電流變化劇烈參數的值，定義參見3.6.1 式(5)；縱座標示三角形網格的百分比。虛線表示初始網格；實線表示經過細分的網格。

4. 應用

本章提出一些適應性網格生成的應用實例。在4.1中，模擬一條接近導體平面的帶電流導線，觀察導體平面上網格細分的過程；在4.2中，模擬金屬平面散射平面波的干涉現象。

4.1 接近導體平面的導線

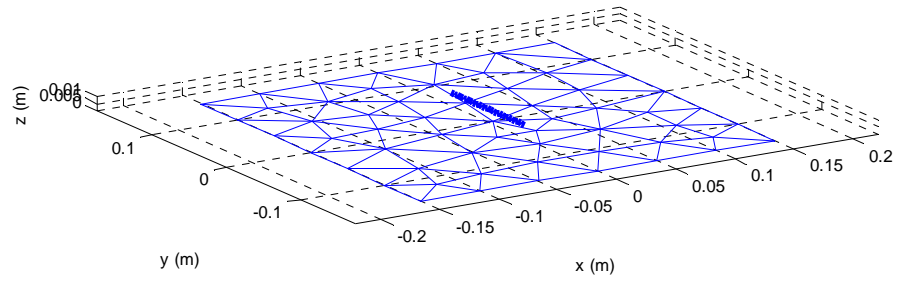
帶電流的導線，會在導體平面上感應出電流，越靠近導線的部分，感應電流越強烈。以下兩個例子，分別將導線接近導體平面的中心位置和將導線接近導體平面邊緣，在適應性網格細分的過程中，不需人為的判斷介入，網格的分佈就能精準地描述電流分佈。

4.1.1 導線在導體平面之上

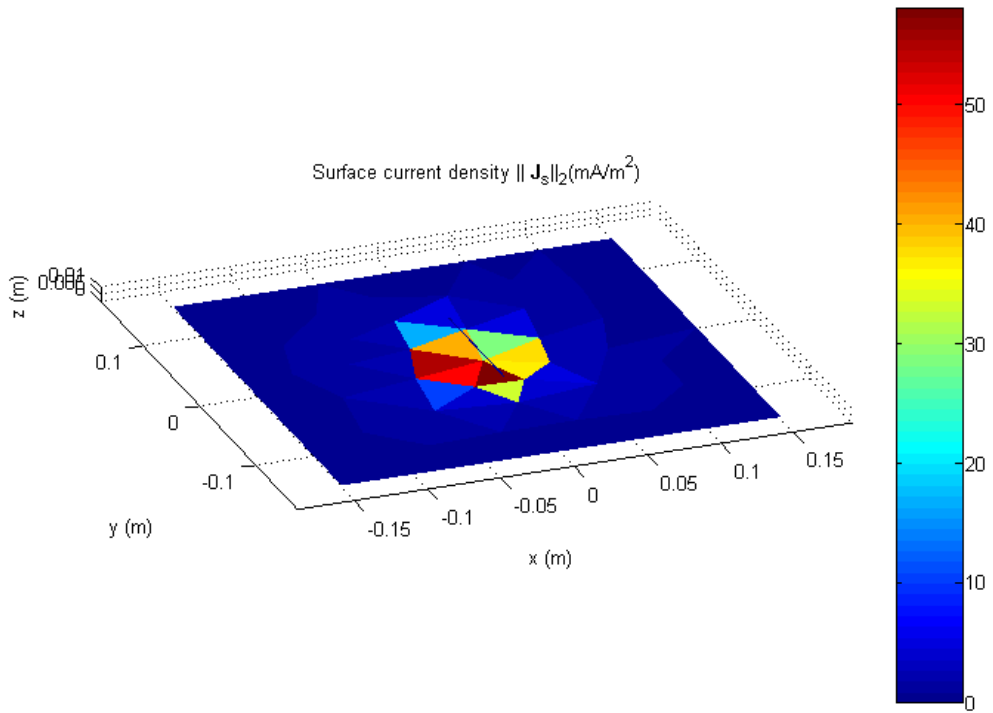
如圖 5-20所示，一根沿著y方向的導線平行置於導體平面上方 0.01 公尺處，電壓源在此導線中央，座標為(0, 0, 0.1)的位置，導體平面的長寬皆為 0.3 公尺，中心在原點，電磁模擬頻率為 1GHz。初始網格的大小設為 $\lambda/5$ (λ 為波長，1GHz 時，真空中的電磁波波長約為 0.06m)，如圖 5-20(a)所示；圖 5-20(b)是Galaxy計算的初始電流分佈。

三次適應性網格細分的過程中，網格的分佈與Galaxy計算的電流分佈如圖 5-21所示。每一次網格細分的過程，都是挑出電流變化較劇烈的網格區域，灑點並在該局部區域重新分割網格。在這個例子中，可以清楚看出導體平面接近導線的部分，有大而且劇烈的電流變化，因此，網格在這個區域的密度也相對較大，才能夠更準確地描述電流的分佈。

在表格 5-4中列出每一次網格細分和計算電流分佈所需的時間。由於我們做局部網格細分，網格細分所需的時間主要和需要被細分的網格數量有關。由圖 5-22可以看出，每一次網格細分所需的時間都遠小於計算電流分佈所需時間。

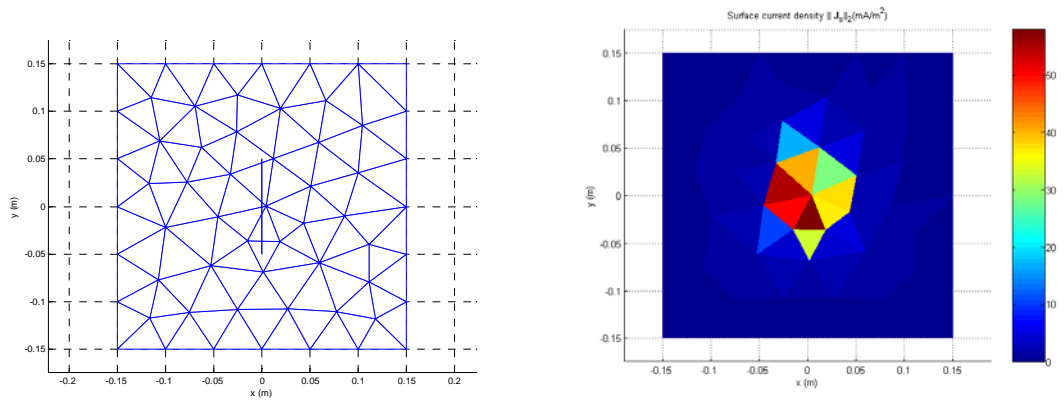


(a)

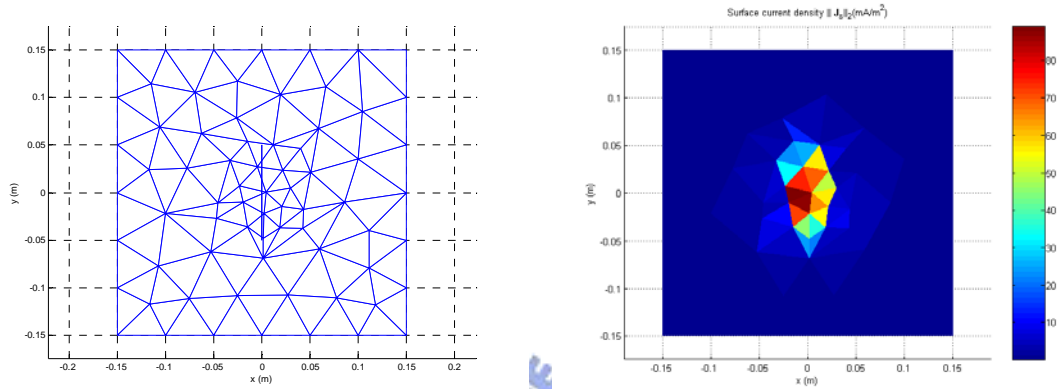


(b)

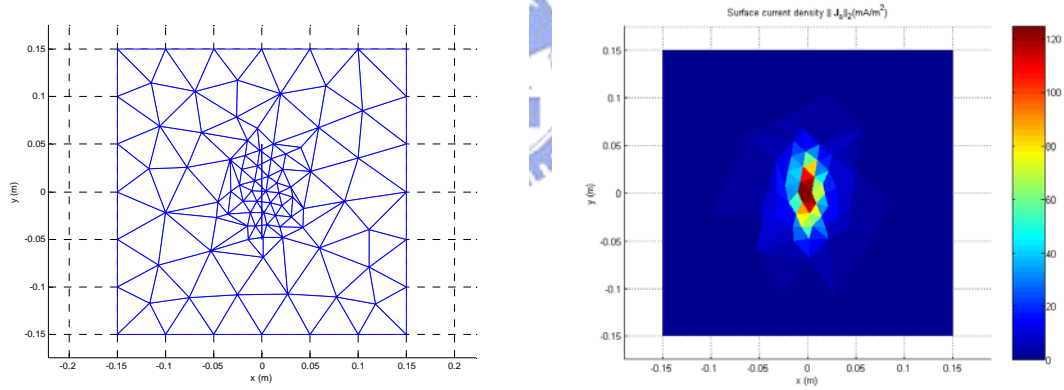
圖 5-20：導線在金屬平面上方距離平面 0.01m 處 (a) 初始網格 (b) 初始電流分佈



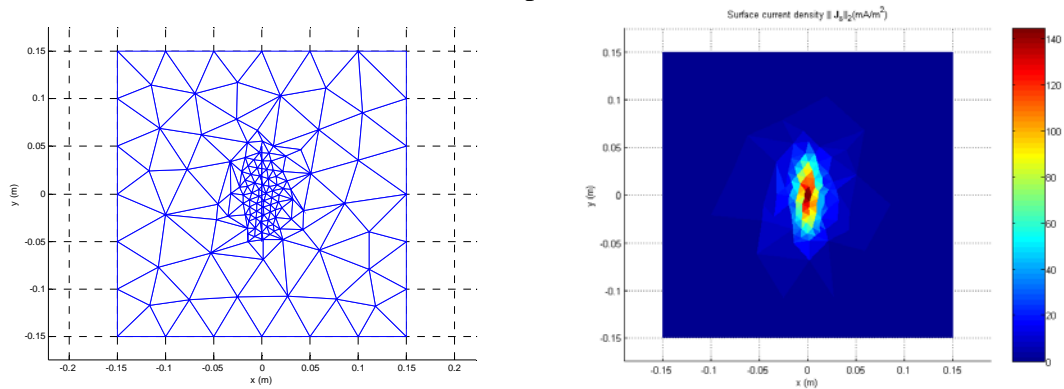
(a) Initial mesh



(b) First pass



(c) Second pass



(d) Third pass

圖 5-21：導線在金屬平面上方距離平面 0.01m 處，網格細分過程

表格 5-4：網格細和計算電流分佈分所需的時間

	Initial	Pass 1	Pass 2	Pass 3
Time for solving current bypass far-interaction (s)	1	3	6	9
Time for mesh refinement (s)		0.032	0.062	0.094
Number of nodes	59	74	94	117
Number of refined triangles		25	23	20

Time for solving current and mesh refinement

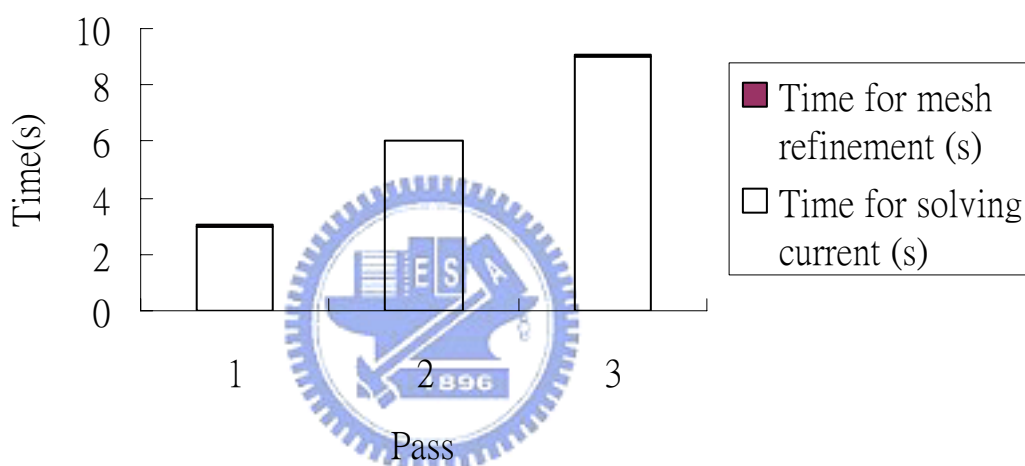
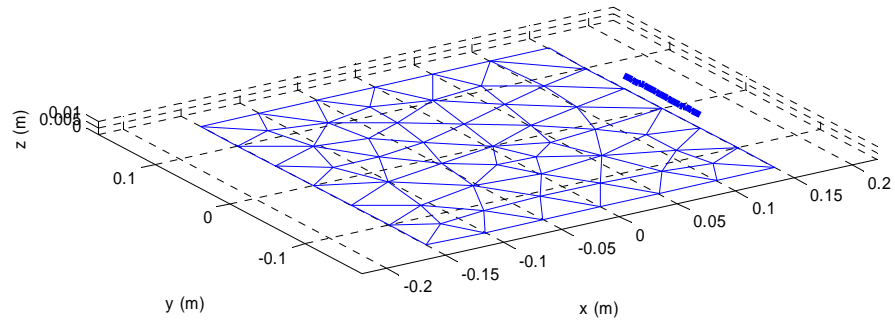


圖 5-22：每一次網格細分與計算電流分佈分所需的時間

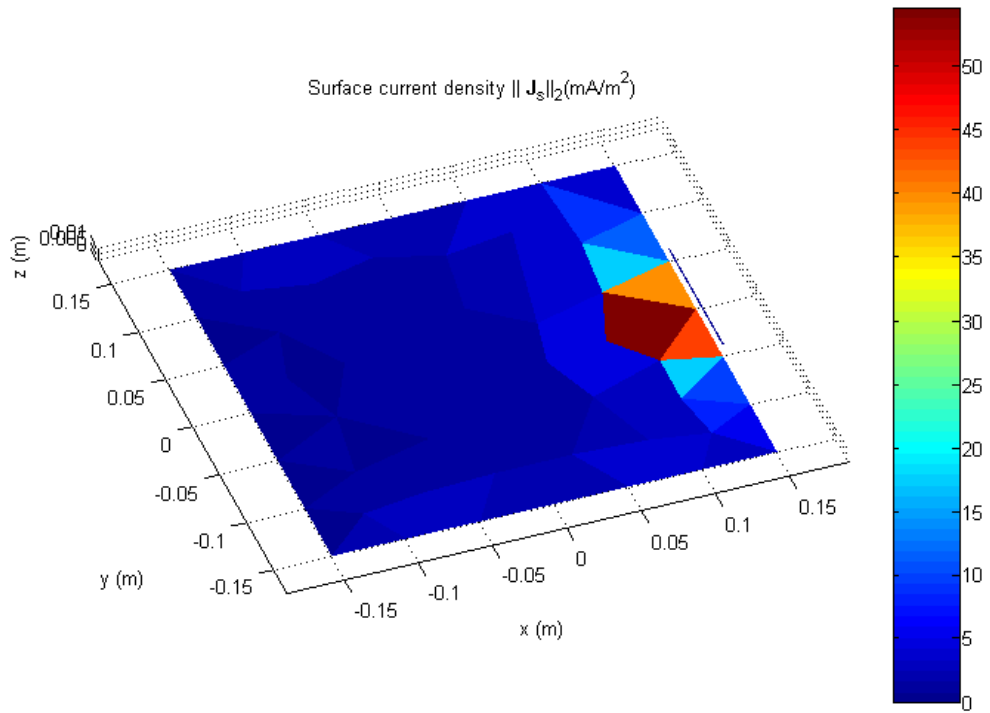
4.1.2 導線接近平面邊緣

若將4.1.1例子中的導線移至金屬平面的邊緣，適應性網格的分佈應該如何？以下我們使用與4.1.1例子相同的均勻初始網格開始計算，如圖 5-23a，並觀察每一次網格細分的情形，如圖 5-24所示。適應性網格細分的過程中，導體平面接近導線的部分不斷地被細分，以期能夠準確地描述該處的電流分佈。

表格 5-5和列出這個例子中，每一次網格細分和計算電流分佈所需的時間，與前述4.1.1 導線在導體平面之上的例子一樣，每一次網格細分所需的時間都遠小於計算電流分佈所需時間。

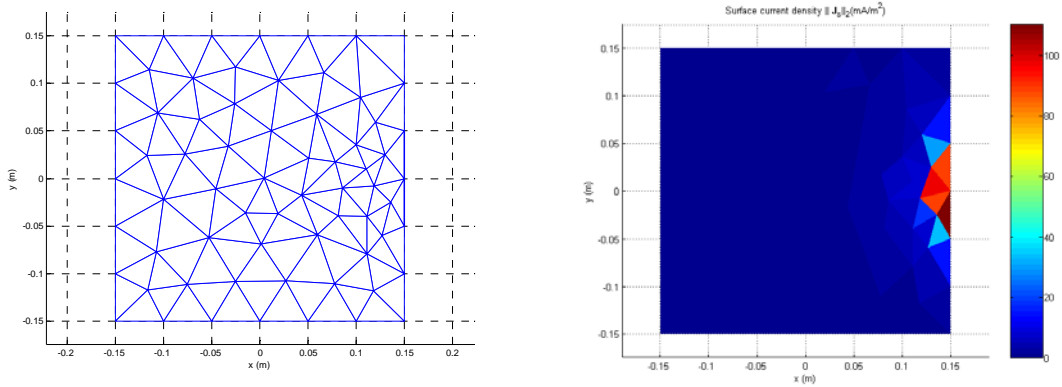


(a)

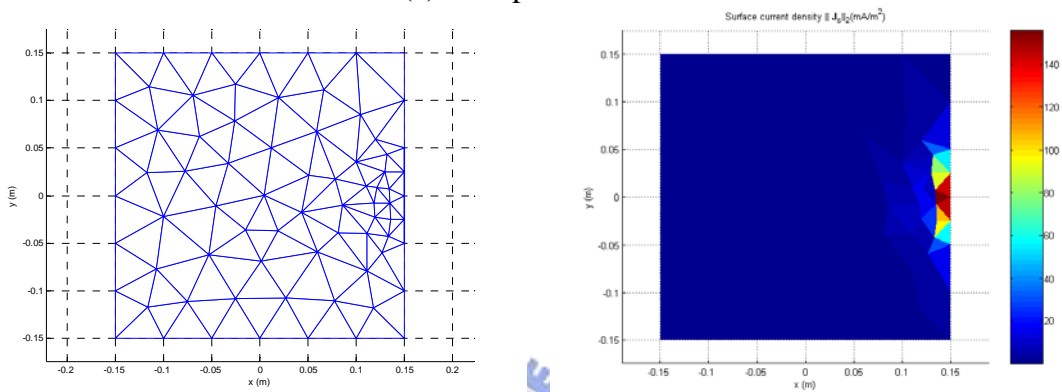


(b)

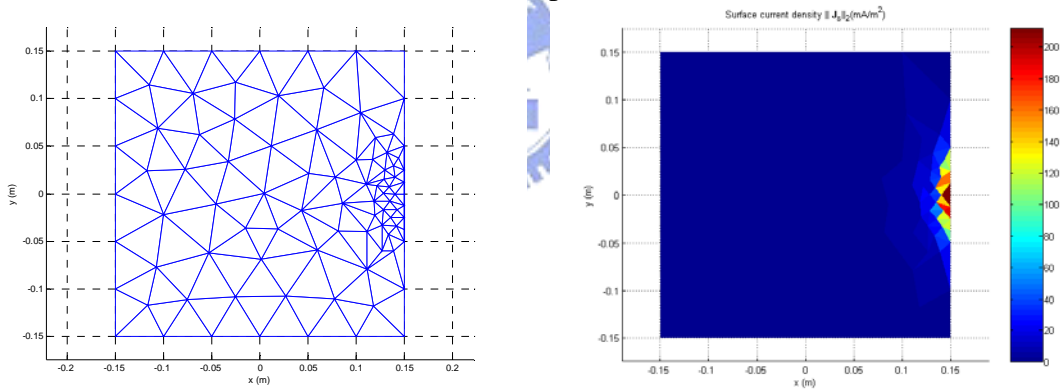
圖 5-23：導線接近金屬平面的邊緣 (a)初始網格 (b)初始電流分佈



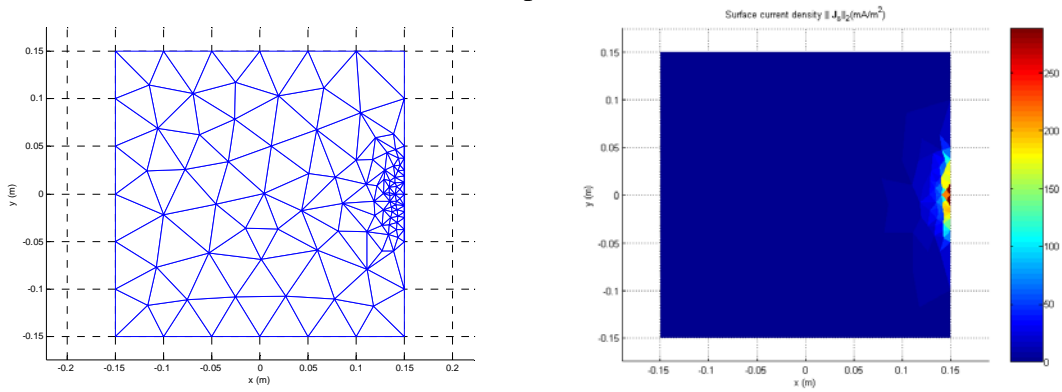
(a) First pass



(b) Second pass



(c) Third pass



(d) Fourth pass

圖 5-24：導線接近金屬平面邊緣，網格細分過程

表格 5-5：網格細和計算電流分佈分所需的時間

	Initial	Pass 1	Pass 2	Pass 3	Pass 4
Time for solving current bypass far-interaction (s)	1	1	3	4	7
Time for mesh refinement (s)		0.031	0.062	0.062	0.203
Number of total nodes	59	67	76	92	111
Number of refined triangles		8	9	16	19

Time for solving current and mesh refinement

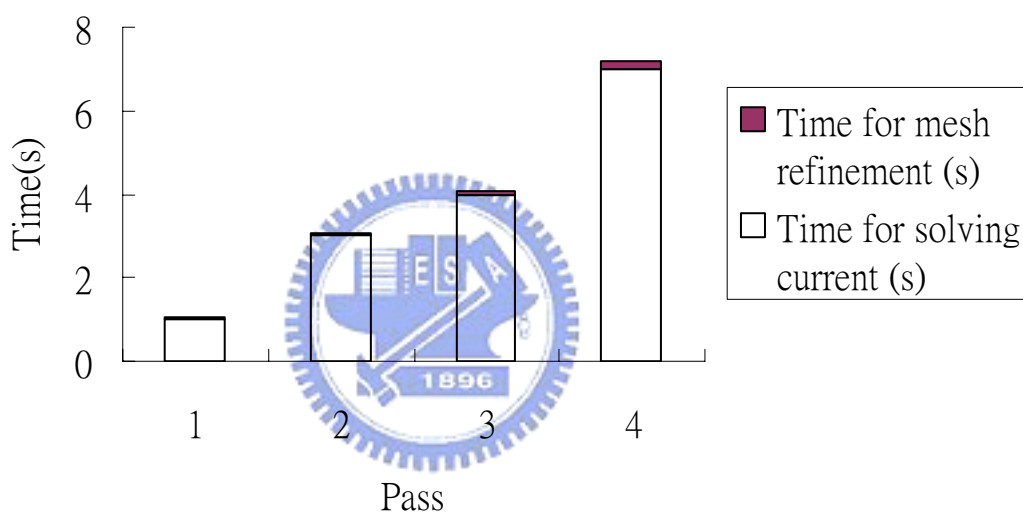


圖 5-25：每一次網格細分與計算電流分佈分所需的時間

4.2 金屬平板散射電磁波

這個例子是電場極化的平面波(E-polarized planar wave)以 45 度入射角向金屬平板入射，觀察入射波和反射波在金屬平板上產生的干涉現象。平面電磁波的頻率為 3GHz，金屬平板的寬度為一倍波長，長度為九倍波長，如圖 5-26 所示。

入射波和反射波會產生干涉，金屬平板上的電流分佈形成干涉條紋，而且，靠近邊緣的電流密度會遠大於中央，如圖 5-27 所示，網格細分的區域主要為電流變化最劇烈的區域，即在建設性干涉的導線邊緣區域，網格不斷被細分。

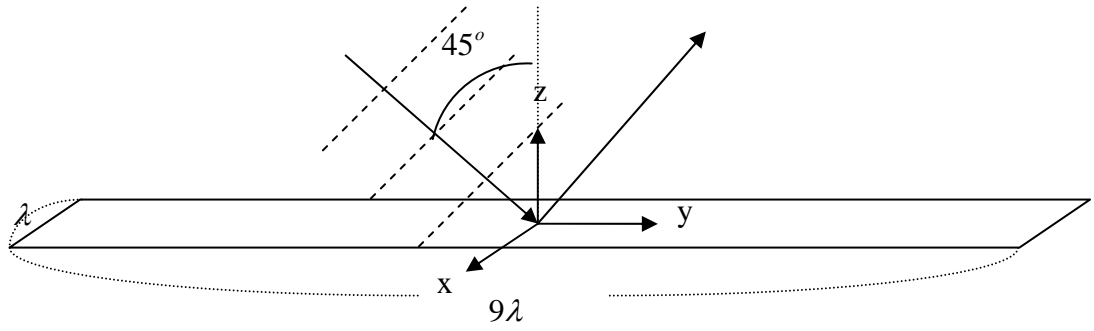


圖 5-26：電場極化的平面波(E-polarized planar wave)以 45 度入射角向金屬平板入射

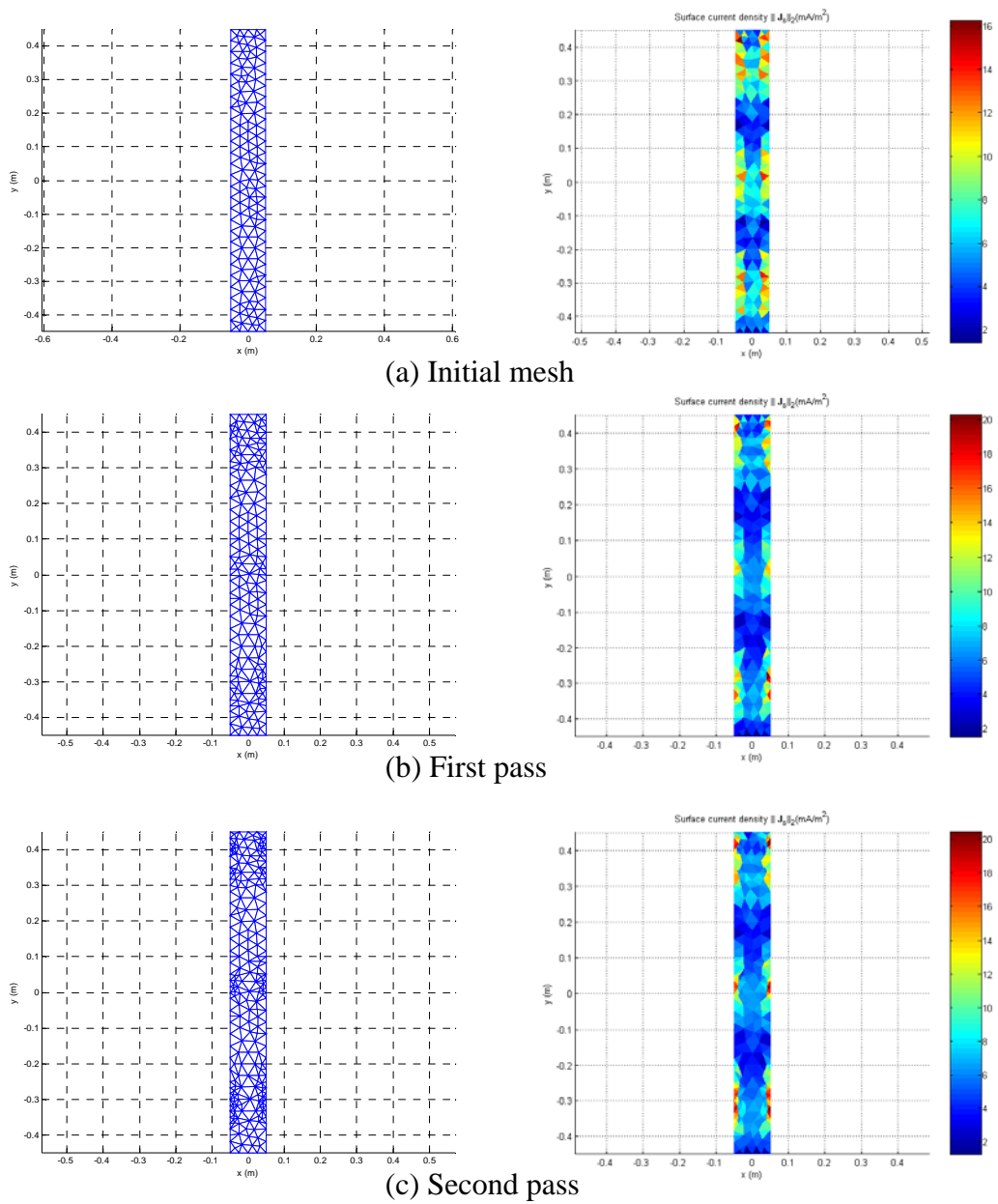


圖 5-27：平面波入射金屬平板，網格細分和電流分佈（計算電流未忽略遠交互作用）

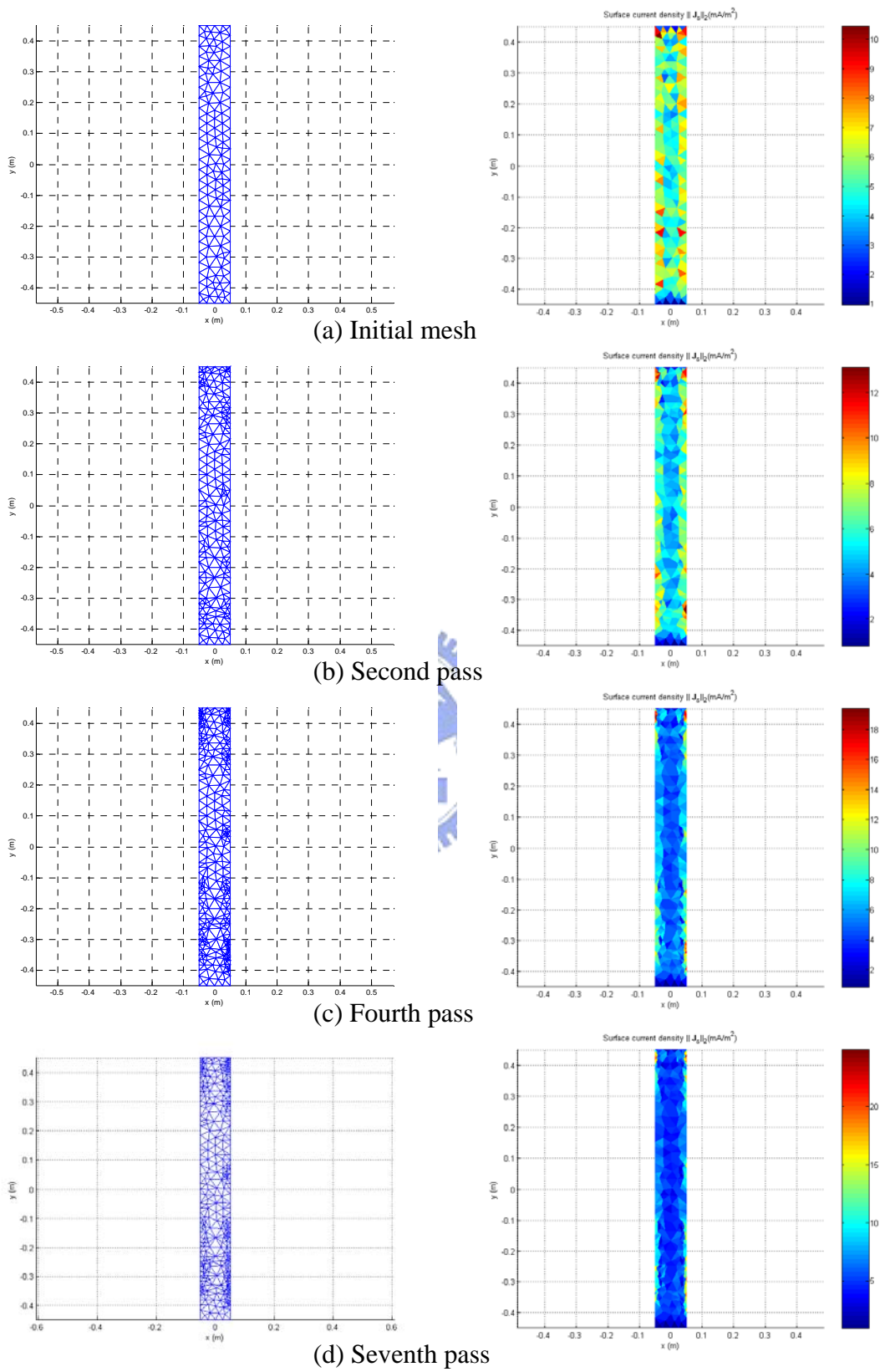


圖 5-28：網格細分的過程與忽略遠交互作用計算的電流分佈，只能觀察出金屬平板邊緣的強電流，干涉條紋不明顯

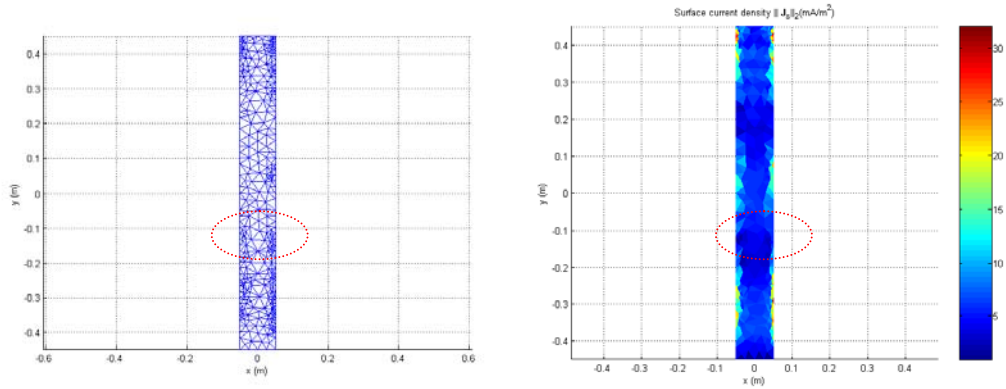


圖 5-29：以圖 5-28(d)的網格，不忽略遠交互作用解力差法方程式計算電流分佈。有些電流密度非常小而且變化不大的區域也被細分（虛線包含之區域）

表格 5-6：忽略遠交互作用與不忽略遠交互作用的執行時間比較

		Initial	Pass 1	Pass 2
Bypass far-interaction	Time for solving current (s)	1	2	3
	Time for mesh refinement (s)		0.25	0.265
	Number of nodes	150	192	232
	Number of refined triangles		42	41
Not bypass far-interaction	Time for solving current (s)	9	14	16
	Time for mesh refinement (s)		0.25	0.328
	Number of nodes	150	192	247
	Number of refined triangles		42	55

在這個例子中，若忽略遠交互作用來解力差法方程式，雖然節省不少計算電流分佈的時間（表格 5-6），但可能觀察不到明顯的干涉現象，只能觀察到導線邊緣的強電流密度，如圖 5-28所示，即使細分網格七次，也幾乎觀測不到電流干涉。因為忽略遠交互作用解得的電流分佈，會在強電流區域附近產生一些假電流（實際上不存在的電流），擾亂了原本明顯的干涉現象，因此，無法從解得的電流分佈觀測到明顯的亮暗紋間格，但是導線邊緣的強電流仍非常明顯。由此可知，忽略遠交互作用的力差法方程式解，可能會誤判部分需要被細分的區域，額外花時間去細分這些不必被細分的區域（即假電流分佈的區域），實際上並不影響解的準確度，然而不必要的網格細分增加了計算的時間，如圖 5-29所示。

5. 結論與未來研究方向

本文利用 divide-and-conquer Delaunay 三角形分割法生成網格，並結合以力差法解電場積分方程式的電磁模擬結果自動細分網格。局部誤差指示器標示出相對電流變化劇烈的區域，只針對該區域的網格進行局部網格細分，因此，網格細分所需的時間幾乎只與被細分區域的節點數量有關，每次網格細分的時間複雜度為 $O(N_{local} \log N_{local})$ ，其中 N_{local} 為細分區域節點的數量。在網格細分的過程中，不只是將網格單元直接分解成更小的單元，還調整節點的位置以使網格分佈較為均勻，細分後的網格品質符合以力差法解點場積分方程式電磁模擬器的網格需求。

這個針對電場積分方程電磁模擬器的適應性網格細分演算法已應用在許多實例中，都得到不錯的結果。靠近帶電流導線的金屬平面上，越接近導線的地方會感應越大的電流，接近導線的區域網格不斷被細分，細分後的網格更能準確地描述金屬平板上電流的變化與分佈；金屬平板散射平面波時，入射波和反射波會在金屬平板上干涉，使金屬平板上的電流分佈呈現干涉條紋，除此之外，大部分的電流會集中在平板的邊緣，因此，金屬平板邊緣且建設性干涉處，網格會自動細分，以高密度的網格來描述該區域劇烈的電流變化。

此適應性網格細分演算法的目標是期望不需任何人為介入的情況下，網格的分佈能夠自動調整，有效地利用有限的資源（控制未知數的數量），使利用力差法解電場積分方程式的電磁模擬器能解出更準確的解。與現今市面上以力差法為基礎的商用電磁模擬軟體相較之下：一般商用軟體大多使模擬結構上的網格均勻分佈，但是會取較小的網格來模擬，如 IE3D 預設網格寬度為波長的 1/20；或是建立一些經驗式，如：導線的邊緣與電源附近需要較密的網格，其他沒有經驗式但需要細分才能得到準確解的區域就必須由工程師來判斷。本文所提出的適應性網格細分演算法以力差法的電流解為基礎來細分網格，能自動將初始較為粗糙的均勻網格（初始網格寬度大約為波長的五分之一），經過幾次局部網格細分修改網格的分佈和密度，以得到準確的解。

然而，還有許多可以改進的地方值得研究。首先，誤差指示器的部分目前用局部誤差指示器，誤差大小只由該次網格與電流分佈來決定，局部誤差指示器的優點是計算速度快，節省記憶體空間，也較容易實作，缺點主要是誤差估計不夠嚴謹；如何針對力差法解電場積分方程的應用，設計快速且更為嚴謹的全域誤差指示器，是往後重要的研究課題。另外，網格本身的大小，以最大最小網格寬度的比值，主要侷限在浮點數運算的準確位數，此侷限亦為電流解準確度的極限，除了硬體的限制以外，如何利用軟體設計的方式增加浮點數運算的準確位數也有許多相關研究，可以參考 Shewchuk 的論文[4]。最後，若欲應用在三維結構表面的網格細分，目前需要的，是研究如何灑點在面與面之間的交界邊，以得到品質好的網格。

參 考 文 獻

- [1] H. Y. Chao and M. C. Lin, "Adaptive mesh refinement for integral-equation-based electromagnetics solvers," accepted by 2005 USNC/URSI National Radio Science Meeting.
- [2] P. L. George, *Automatic Mesh Generation: application to finite element methods*, Wiley, 1991.
- [3] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator," *ACM First Workshop on Applied Computational Geometry*, pp. 124-133, May 1996.
- [4] J. R. Shewchuk, "Robust adaptive floating-point geometric predicates," in *Proc. of the 12th Annu ACM Symp. Comput. Geom.*, May 1996, pp. 141-150.
- [5] L. Guibas and J. Stolfi., "Primitive for the manipulation of general subdivisions and the computation of Voronoi diagrams", *ACM Trans. on Graphics*, vol. 4, no. 2, pp. 74-123, April 1985.
- [6] P. Fleishmann, "Mesh generation for technology CAD in three dimensions", PhD dissertation, [Technische Univ. Wien, Fak. für Elektrotechnik](#), 1999.
- [7] H. Y. Chao, "A multilevel fast multipole algorithm for analyzing radiation and scattering from wire antennas in a complex environment", PhD dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.
- [8] S. H. Lo, "Finite element mesh generation and adaptive meshing," in *Prog. Struc. Engng Master.*, 4:381-399, 2000.
- [9] H. Edelsbrunner and E. P. Mucke, "Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithm", *ACM Trans. on Graphics*, 9(1):66-104, 1990.
- [10] C. C. Lu, W. C. Chew, and J. M. Song, "A study of disparate grid sizes for an irregular-shape scatterer on EFIE, MFIE, and CFIE," in *IEEE AP-S Int. Symp. Dig.*, vol. 3, Baltimore MD, July 1996, pp. 1746-1749.
- [11] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microw. Opt. Tech. Lett.*, vol. 10, no. 1, pp., 15-19, Sept. 1995.
- [12] M. Krizek, "On the maximum angle condition for linear tetrahedral elements," *SIAM J. Numer. Anal.*, 29:513-520, 1992.
- [13] R. Lee and A. Cangellaris, "A study of discretization error in the finite element approximation of wave equations," *IEEE Trans. Antennas Propagat.*, vol. AP-40, pp. 542-548, May. 1992.
- [14] A. F. Peterson and R. J. Baca, "Error in the finite element discretization of the scalar Helmholtz equation over electrically large regions," *IEEE Microwave Guided Wave Lett.*, vol. 1, pp. 219-222, Aug. 1991.

- [15] M. Bachtold, M. Emmenegger, J. G. Korvink, and H. Batles, "An error indicator and automatic adaptive meshing for electrostatic boundary element simulations," *IEEE Trans. on Computer-Aided Design of Integrated Circuit and Systems*, vol. 16, no. 12, Dec. 1997.
- [16] R. Hertel and H. Kronmuller, "Adaptive finite element mesh refinement techniques in three-dimensional micromagnetic modeling," *IEEE Trans. on Magnetics*, vol. 34, no. 6, Nov. 1998
- [17] J. F. Thompson, et al., *Handbook of Grid Generation*, CRC Press, 1999.
- [18] G. R. Liu, *Mesh Free Methods*, CRC Press, Boca Raton, 2003.
- [19] R. B. Simpson, "C++ classes for 2-D unstructured mesh programming", May 2003. Available: <http://www.scicom.uwaterloo.ca/~rbsimpso/>
- [20] P. Su and L. S. Drysdale, "A comparison of sequential Delaunay triangulation algorithms," in *Proc. of the 12th Annu ACM. Symp. Comput. Geom.*, June 1995, pp. 61-70.
- [21] S. Fortune, "Voronoi diagrams and Delaunay Triangulations," in F. K. Hwang and D.-Z. Du, editors, *Computing in Euclidean Geometry*, pp. 193-233, World Scientific, Singapore, 1992.
- [22] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," in F. K. Hwang and D.-Z. Du, editors, *Computing in Euclidean Geometry*, World Scientific, 1992.
- [23] Ansoft HFSS: <http://www.ansoft.com/products/hf/hfss/>
- [24] Zeland IE3D: <http://www.zeland.com/ie3d.html>
- [25] Triangle: <http://www-2.cs.cmu.edu/~quake/triangle.html>