

專題一 資料內插

摘要

實作 Laplace interpolation 方法為專題一主要核心，其目的為修補圖像中全部遺失的數值。

Laplace Interpolation 介紹

Laplace interpolation 是一種用於還原圖像中缺少數據之插值方法，雖然簡單，但在大部分情況下效果良好。

其修補方程式如下：

線性方程式	說明
y_u	為 y 點上方數值
y_d	為 y 點下方數值
y_l	為 y 點左方數值
y_r	為 y 點右方數值
$y_0 = y_0$	在點已知的情況中
$y_0 - \frac{1}{4}y_u - \frac{1}{4}y_d - \frac{1}{4}y_l - \frac{1}{4}y_r = 0$	通用求未知點方程式，為應用均值定理。
$y_0 - \frac{1}{2}y_u - \frac{1}{2}y_d = 0$	求未知點在左右邊界時方程式
$y_0 - \frac{1}{2}y_l - \frac{1}{2}y_r = 0$	求未知點在上下邊界時方程式
$y_0 - \frac{1}{2}y_r - \frac{1}{2}y_d = 0$	求未知點在左上邊界時方程式
$y_0 - \frac{1}{2}y_l - \frac{1}{2}y_d = 0$	求未知點在右上邊界時方程式
$y_0 - \frac{1}{2}y_r - \frac{1}{2}y_u = 0$	求未知點在左下邊界時方程式
$y_0 - \frac{1}{2}y_l - \frac{1}{2}y_u = 0$	求未知點在右下邊界時方程式

數值線性方程式介紹

Gaussian Elimination method:

數值線性方程式中其一算法，可用來為線性方程組求解，求一矩陣時，高斯消去法會產生出一個行梯陣式。

時間複雜度為 $O(n^5)$ 。

Gauss-Seidel method:

Gauss-Seidel method 是數值線性方程式疊代法其中之一，可用來求出線性方程組解的近似值。

時間複雜度為 $O(n^3)$ 。

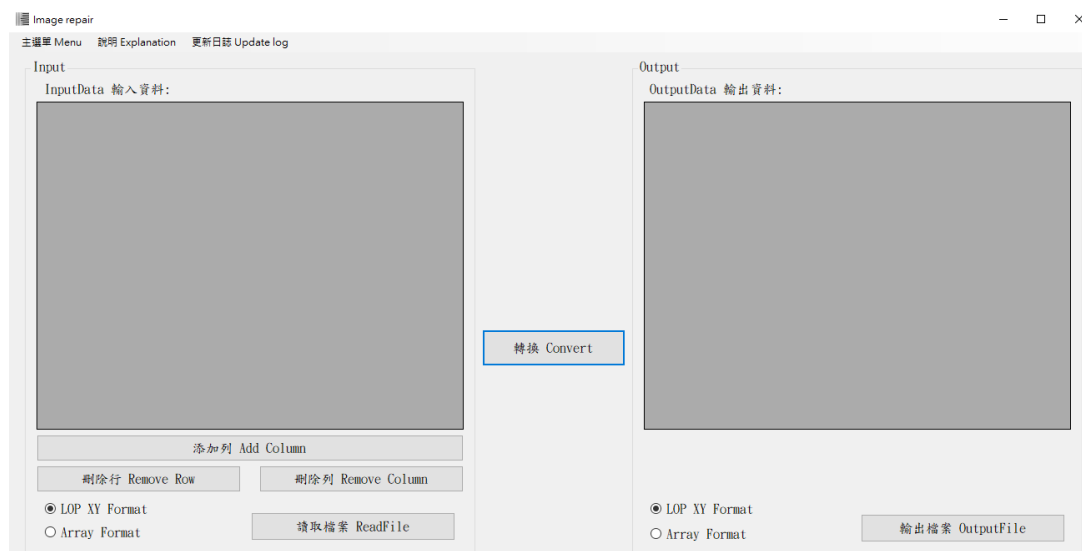
Conjugate gradient method:

Conjugate gradient method 可求解係數矩陣在對稱正定矩陣中線性方程組數值解的方法。Conjugate gradient method 是一個疊代方法，適用於係數矩陣為稀疏矩陣的線性方程組。

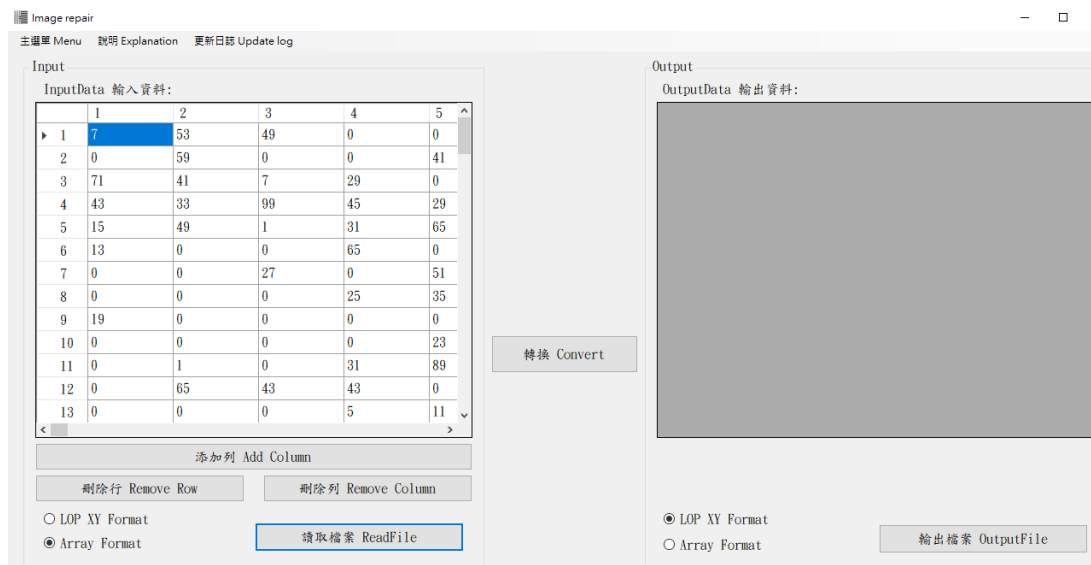
時間複雜度為 $O(n^3)$ 。

目前我們在程式中寫出 Gaussian Elimination method、Gauss-Seidel method，並在比較中發現 Gauss-Seidel method 速度較快。即將在暑假中實現 Conjugate gradient method 並與 Gauss-Seidel method 比較後，選擇出效率最快速演算法，應用在 Laplace Interpolation。

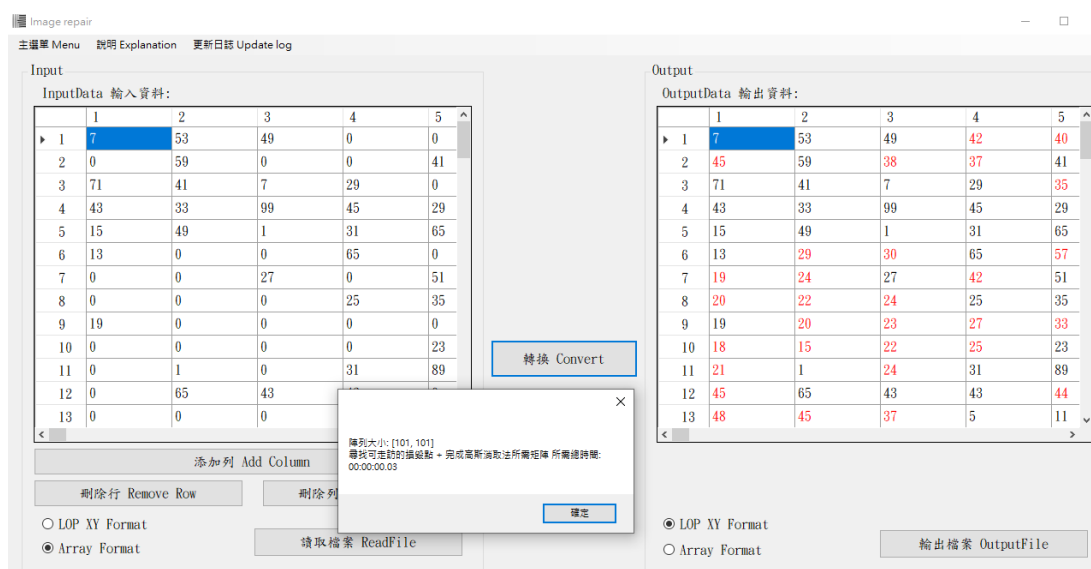
應用程式介紹



▲應用程式介面



▲輸入介面，若資料元素大於 2250000 將不顯示資料



▲輸出介面，若資料元素大於 65535 將不顯示資料，在右方 Output 畫面中紅字即為修補後的數值

此應用程式其餘功能：

1. 可輸入 LOP XY 格式:序列格式
2. Array XY 格式: 二微陣列格式
3. 載入資料，可使用拖曳。
4. 可輸入資料類型為 csv, xlsx, txt
5. 可輸出資料類型為 csv, xlsx, txt
6. 添加列需先按按鈕，添加行則是直接輸入數值

時間複雜度比較

根據我們反覆測試之後資料如下，且遇一問題將再結論做說明。

Gauss–Seidel method 運行效率表如下：

矩陣大小	運行時間	作業系統	記憶體使用量
2000	37s	32 位元	≈2GB
3000	79s	64 位元	≈6GB
5000	322s	64 位元	≥16GB
8000	880s	64 位元	≥16GB

註: Gaussian Elimination method 在 5000 時運行時間已無法再 10min 內執行完畢故不在此加入 Gaussian Elimination method 效率表

結論

我們在實現 Gauss–Seidel method 發現兩個問題：

1. 當我們在開一個陣列時候，陣列[0,0] 與陣列[1000,1000] 速度不一致，再呼叫陣列時[0,0] 比 [1000,1000]快了約 10 倍的速度找出，於是我們認為，陣列的大小會影響到我們從陣列提出數值的速度。後來我們在 Gauss–Seidel method 疊代陣列中進行壓縮後，效率比未壓縮效率快約 42 倍。
2. C# 在一個陣列裡最多可以有多少元素阿！根據 MSDN 預設，陣列的最大大小只能夠有 2GB，假設此陣列為 int 時，最大元素最多只能有 $2^{32}/4$ 。可是我們的矩陣只要大於 $3000 * 3000$ 就沒辦法塞入阿！MSDN 有給予些許的解決方法，當電腦為 64 位元且再 app.config 加入 `gcAllowVeryLargeObjects`，假如你的 RAM 夠大，就能使陣列擁有 40 億元素，若你的 RAM 無法負載 40 億元素那只能到 RAM 能負載最大值。

未來我們想要在增進兩點，將 Gauss–Seidel method 的原先矩陣嘗試是否能夠用疊代的方式尋找陣列中值而使效率增加？與 Conjugate gradient method 實作後是否能夠比 Gauss–Seidel method 效率更好來的更快速。