**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 |
|  | 為y點上方數值 |
|  | 為y點下方數值 |
|  | 為y點左方數值 |
|  | 為y點右方數值 |
|  | 在點已知的情況中 |
|  | 通用求未知點方程式，為應用均值定理。 |
|  | 求未知點在左右邊界時方程式 |
|  | 求未知點在上下邊界時方程式 |
|  | 求未知點在左上邊界時方程式 |
|  | 求未知點在右上邊界時方程式 |
|  | 求未知點在左下邊界時方程式 |
|  | 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。

**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。

**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。

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。可是我們的矩陣只要大於就沒辦法塞入阿！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。