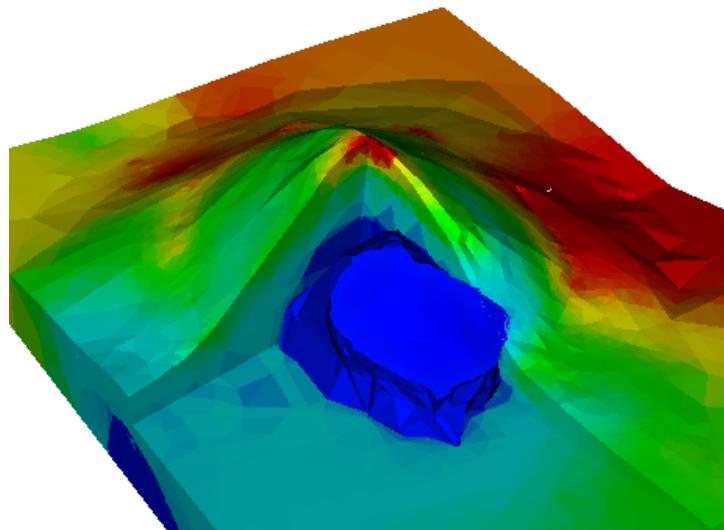


# 無邊界電阻率斷層掃描

## BERT 2 – 使用者教學課程



托馬斯 · 岡瑟\* & 卡斯滕 · 呂克 †

2023 年 4 月 26 日  
版本 2.4.1

在本教程中，我們示範如何使用軟體包 BERT 進行 ERT 反演。討論了一些小但有啟發性的範例，所有真實的現場案例，以展示如何使用個人資料中的不同選項來產生特定於案例的反演結果。

這些範例從有和沒有地形的表面測量的二維反演開始。隨後在 3d 中演示了同樣的情況，其中網格生成變得更加重要。我們展示瞭如何包含結構資訊以及如何處理埋置電極。

此外，還顯示了封閉物體（例如樹木、人類、土柱和模型水箱）的測量結果。最後，我們展示如何處理延時電阻率測量。邀請使用者透過處理範例目錄中的資料來進行操作。

---

\*萊布尼茨應用地球物理研究所，漢諾威  
†柏林工業大學應用地球物理系

# 內容

<b>一、簡介</b>	<b>3</b>
1.1. BERT 1 和 2、DCFEMLib、GIMLi - 歷史和名稱	3
1.2. 選項和命令	4
<b>2. 使用簡單 2D 範例的基本概念</b>	<b>7</b>
2.1. 第一步	7
2.2. 正則化和數據擬合	10
2.3. 網格品質和細化	11
2.4. 規則的網格	12
2.5. 誘導極化	13
<b>3. 其他二維幾何形狀</b>	<b>14</b>
3.1. 帶有地形的 2D ERT	14
3.2. 2D 跨孔資料	15
3.3. 閉合二維幾何樹斷層掃描	17 號
<b>4. 3D 幾何形狀</b>	<b>18</b>
4.1. 平坦的地球表面測量	19
4.2. 3D 地形	20
4.3. 3D 跨孔測量	22
4.3.1. 地形和埋置電極	23
4.4. 封閉的 3D 幾何形狀	23
<b>5. 納入先驗訊息</b>	<b>25</b>
5.1. 結構性限制	26
5.2. 透過水下範例進行區域特定控制	26
<b>6. 延時 ERT</b>	<b>32</b>
6.1. 策略	32
6.2. 處理和選項	32
6.3. 跨孔遊戲中時光倒流測量	33
6.4. 土柱測量	34
<b>一、安裝</b>	<b>38</b>
A.1. 適用於 Windows 使用者的 BERT	38
A.2. 適用於 Linux 使用者的 BERT	39
<b>B、版本歷史</b>	<b>40</b>
<b>C. 文件和程序</b>	<b>43</b>
<b>D. 選項及其預設值的完整列表</b>	<b>44</b>
<b>E. 使用者故事/操作方法</b>	<b>46</b>
E.1。如何利用外部地形資訊進行二維反演	46
E.2。給定 3D 座標的 2D 反演	47
E.3。使用者定義的常規二維網格	50

E.4。如何在 2D 域中反演 1d 電阻率。	51
E.5。對使用者定義的常規 3D 網格進行反演。	53
E.6。3D 地形中 2D 剖面的 3D 反演和視覺化。	55
E.7。使用土柱的 Hydrus3D 網格進行正演計算。	58
E.8。如何使用 Hydrus2D 模擬進行合成資料反演。	60
E.9。模擬傾斜鑽孔中的 CEM 環形電極。	65

## 一、簡介

### 1.1. BERT 1 和 2、DCFEMLib、GIMLi - 歷史和名稱

直流電測量應用廣泛，例如醫學成像、地球物理表面或地下測量或樹木和土壤探針的調查。此反演問題在術語 ERT（電阻率斷層掃描）、ERI (...成像)、EIT (...阻抗斷層掃描) 或直流電阻率反演下稱為已知。我們軟體的目標是提供一種適用於所有幾何形狀的極其靈活的解決方案。

主要優點是可以處理任意幾何形狀。因此我們決定使用非結構化<sup>1</sup>用於正向計算和參數辨識的有限元素網格。透過使用三角形 (2d) 和四面體 (3d)，我們可以追蹤從地下獲得的任何先前的幾何形狀、探測或結構資訊。由於這種普遍性，我們決定稱之為 BERT——無邊界電阻率斷層掃描。

BERT 這個名字可以關聯到

- Günther等人所描述的技術解決方案。(2006)、Rücker (2010) 等作品，
- C++/Python 軟體專案 (上<https://gitlab.com/resistivity-net/bert>)，
- 發行版 (Windows 安裝程式、conda 套件) ，以及
- 主要的命令列工具。

BERT 版本 1 (2009-2013 年發布) 已成為 C++ 庫 DCFEMLib (直流有限元素方法庫) 的一部分，該庫仍包含在 BERT 原始程式碼中。部分基於 DCFEMLib 的網格生成工具仍在使用，但很快就會被 Python 工具取代。從版本 2 開始 (從 2010 年的測試版本開始)，BERT 基於 GIMLi - 廣義反演和建模庫，這是用於各種地球物理方法的多方法 C++/Python 庫。請看一下<https://www.pygimli.org>了解如何檢索、建置和使用程式碼的詳細資訊。pyGIMLi 根據 Apache License 2.0 獲得許可<sup>2</sup>BERT 已獲得 GNU 公共授權 (GPL) 的許可<sup>3</sup>。我們的願景是將我們所學到的知識回饋給學術界，而不讓公司利用它，這就是為什麼我們從不關心圖形使用者介面。

Günther 等人描述了 BERT 的基本理論和技術。(2006) 並基於 Rücker 等人討論的有限元素建模技術。(2006)。有限元素建模最初是針對常見的點電極開發的，後來使用完整的電極模型 (Rücker 和 Günther, 2011) 和長電極模型擴展到任意電極形狀 (Rücker 和 Günther, 2011)。

<sup>1</sup>非結構化或不規則意味著元素沒有順序或形狀規則，規則離散化

四邊形、六面體或稜柱只是特殊的變體。

<sup>2</sup>看<https://www.apache.org/licenses/LICENSE-2.0>Apache 授權條件。

<sup>3</sup>看<https://www.gnu.org/licenses/gpl.html>GPL 授權條件

使用分流電極模型的電極 (Ronczka 等人, 2015)。一般來說，反演是基於 Günther 等人所描述的平滑約束高斯-牛頓反演。 (2006)。後來它被制定為一種靈活的最小化和正則化方案，由 Rücker (2010) 詳細描述。

預設反演方案由三重網格方案表示。大多數反演演算法採用雙網格方案，即正演計算在比反演網格更細的網格上進行。我們添加了另一個，以便使用輔助場方法，從而實現非常快速的前向計算<sup>4</sup> 圖 1 顯示了三個網格：在粗略且與解析度相關的網格上定義了參數。在全域細化和延長的網格上完成前向計算。還有一個非常細的主網格用於計算主勢（對於均勻的地下），但僅在網格創建後直接計算一次。

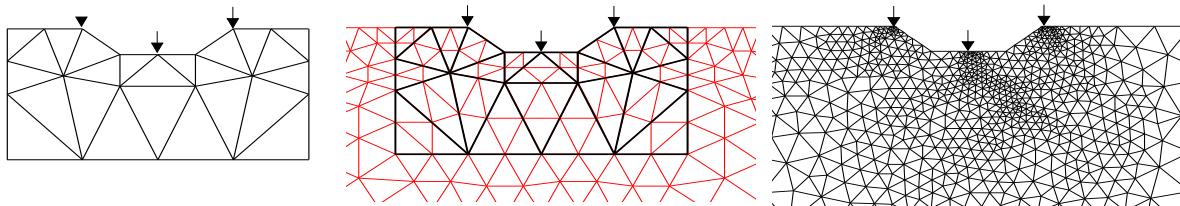


圖 1：二維範例的三個反演網格，來自 Günther 等人。(2006)

奧夫 呀所有方案都是可視化的我n 圖 2. 首先產生三個網格。  
然後 他 計算初級電位並將其插值到次級網格上。從導出此幾何因素，產生視電阻率，並為 h 建立靈敏度矩陣哦 同類案例。最後進行實際反演：使用反演子問題更新電阻率模型，進行正演計算並對照資料進行檢查。這我 直到數據擬合良好或過程停滯。

BERT 可在 Linux 和 Windows 下使用<sup>5</sup>，來自預編譯的二進位檔案或自編譯的程式碼<sup>6</sup>。必須知道二進位檔案和庫的路徑，例如透過設定 \$出口P

在H=\$路徑 : /c/軟體/BERT適用於 Windows 或 Linux 下路  
\$出口P 在H=\$徑 : /路徑/到/BERT/bin  
\$世博會左心室 D\_LIBRARIES\_PATH=\$LD 函式庫路徑:/path/to/BERT/libBERT 伴隨而來  
蟒蛇 米魄上需要透過設定變數找到的pyGIMLi和pyBERT文件  
Python路徑 到其解放軍 ce (最後沒有 pygimli 或 pybert !)。所有變數都可以是 bashrc 文件，  
設定永久 恩特利我 哟。可以是 bert 所在目錄下的所有用戶，也可以是 me 目錄下的變數。  
目前用戶 在浩

## 1.2. 選項 和c 命令

倒轉 n 本身是由程式 bert (一個 bash 腳本) 控制，該程式讀取所謂的 e.我  
配置 (cfg) 菲爾 的 cfg 檔案中所有必要的資訊都以 bash 中的行的形式存儲，# 符  
C霸醜期環 鍵=值 厄蒂 豌豆 號後面的所有內容都將被忽略，並且可以

<sup>4</sup>豪 非常，你 可以使用廣告 雙網格或單網格方法，或完全不同的網格用於前向  
和 鏡向 造型。

<sup>5</sup>看到 彭迪 X A.1 用於 Windows 中的 ng BERT。

<sup>6</sup>看w 資源網 istivity.n 等有關如何獲取二進位/程式碼以及編譯程式碼的信息

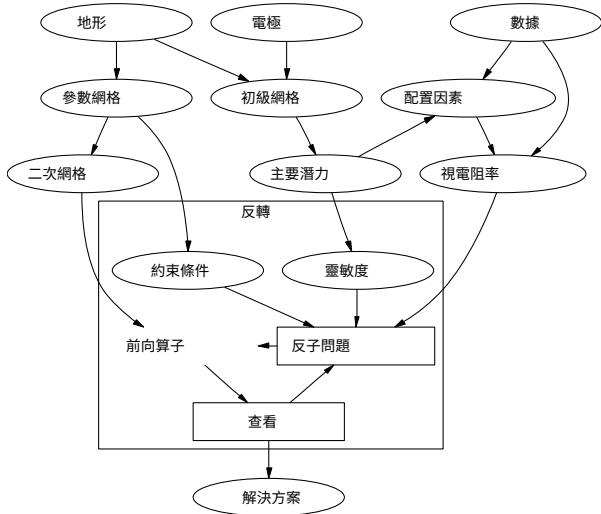


圖 2：BERT 反演方案，來自 Günther 等人。 (2006)：幾何訊息  
用於準備實際的反轉（矩形）。

用於評論。注意按鍵必須是大寫<sup>7</sup>。只有一個強制鍵：資料檔案鍵保存資料檔的名稱。其他重要的鍵是方面（2或3）和地形（0或1，表示假或真）。我們建議為每個問題或不同的解決策略建立一個新目錄。資料檔案必須採用統一的resistivity.net格式（參見www.resistivity.net?unidata）或任何支援的儀器文件（見下文，需要 Python！）。

可以透過呼叫顯示目前版本 \$伯特版本

有關可能選項的清單（帶有預設值），請參閱附錄 D 或致電 \$伯特選擇

但只有少數是常用的<sup>8</sup>。在每一個的開始伯特呼叫時，將搜尋全域和使用者選項並顯示（如果存在）。

為了創建新項目，需要針對各個任務使用特殊命令，即 bertNew2D

• bertNew2DTopo、bertNew3D、bertNew3DTopo，適用於 2d/3d 與或與-輸出拓撲 aphy 情況。例如，

\$伯特紐 2D 資料檔.dat>伯特設定檔

創建一個 ew 設定檔 bert.cfg 包含以下行數據文件=數據文件。和地形=0，而且 數據，DIMEN-錫離子=2 還添加了許多可能的選項，其中大多數是不活動的/由 # 字元註釋。所 r 在這種情況下解釋 有 bert ert 任何帶有結尾的資料文件

使用新指令

嘗試轉換

- TXT (ABEM、Resecs 或 Syscal Pro ASCII 輸出) ，

- 交易 a 點光) ，

<sup>7</sup>有 o 只有一個例外：cMin/cMax/cMap對於色標。我們可能會永久刪除區分大小寫的預設選活力。  
<sup>8</sup>為昌 項，我們建議建立一個檔案 \$首頁/.bertrc 為了我們伯特克在電腦範圍選項（即 ons 或專屬特定選項或一個文件。案選項）安裝 BERT 的目錄中。典型的條目是，例如，SENSMATMAXMEM=8000（或 被覆蓋使用者選擇最喜歡的網格選項 有效記憶體以MB為單位），

- bin (Syscal Pro/Junior 二進位) ,
- stg (AGI SuperSting 數據)
- flw (Geotom 自訂數組資料)
- amp (ABEM LS 系統已棄用的格式) ,
- res2dinv (將重新命名為副檔名 res2dinv 以自動辨識) 。
- udf (DC2dInvRes 定義的統一資料格式)

到 BERT 輸入並在原始檔名中加入 .dat 結尾。

輸入資料轉換也可以透過呼叫手動完成 (無需重新命名) \$bert 轉換資料文件

除了儀器資料之外，還可以使用統一資料格式 (UDF，一旦由 DC2dInvRes 定義)，並在檔案末尾指定拓撲。如果檔案名稱在檔案末端包含地形 (res2dinv 或 udf)，則捲尺測量的電極將沿著地形展開，產生的資料檔案將保留電極部分中的真實位置。請注意，在地形的情況下，我們更喜歡 (並保存) 電阻而不是視電阻率，因為它對於如何計算幾何因子並不明確。

使用者現在 (或稍後) 可以更改選項並運行單一步驟或整個反演 bert cfgfile 指令

其中 cfgfile 是設定文件，命令可以包含以下內容：

版本只繪製 BERT 版本和預設選項

全部使一切成為可能，這可能是大多數 (至少是小) 情況下的第一步

網格只需建立網格來首先檢查它們 (建議用於更大的問題)

諾梅什執行除網格之外的所有操作 (成功產生網格後)

領域僅建立網格輸入 PLC

鍋計算主要勢並將其插值到輔助網格 (在 BERT 1 中，  
這分為兩個工作 primPot 和 interpolate)

篩選使用數據限制過濾輸入數據，添加幾何因子並估計誤差

計算僅過濾和反演 (不更改網格)，例如在更改反演選項後

節省保存所有重要結果 (每次迭代的模型和回應、日誌檔案、cfg 檔案、網格)  
在一個名為結果<日期><時間>

乾淨的清除目錄中的暫存結果

正確先生刪除除輸入和結果目錄之外的所有內容 (完全釋放磁碟記憶體)

繪圖函數 (僅限 2D，使用 Python) :

顯示網格顯示參數網格 (2D)

秀寶顯示 Poly 檔案作為輸入 (2D 和 3D)

展示顯示反演結果或其他模型向量（2D 和 3D）

MKPDF產生所得電阻率或其他模型向量(2D)的 pdf

顯示數據繪製視電阻率偽剖面圖(2D)

顯示錯誤繪圖誤差模型偽截面(2D)

炫耀將資料、模型響應和失配顯示為偽截面(2D)

show 指令可以在專案目錄中使用，也可以在反演仍在運作時使用，也可以在儲存結果的結果目錄中使用。

## 2. 使用簡單 2D 範例的基本概念

### 2.1. 第一步

中的例子範例/反轉/2dflat/畫廊由弗賴堡礦業科技大學(F. Donner)友好提供，並由 Günther (2004) 進行了廣泛討論。它是一個非常小的輪廓，位於已知的採礦坑道上，用於將水排出礦井。它代表了一個完美的二維異常，因為它垂直於輪廓，並且是一個 2x2m 的空腔。

在使用 21 個間距為 2m 的電極的剖面上，應用了偶極子-偶極子測量，數據品質非常好。資料檔案中的輸入圖庫.dat 已經是視電阻率，數據中已經存在誤差模型。

我們建立一個設定檔來重定向初始化腳本的輸出 \$bertNew2D gallery.dat >伯特  
設定檔

資料檔=gallery.dat

維度=2

PARADX=0.25 # 表面細胞的大小 (電極間距)

PARA2D 質量=34.0 # 定義網格成長的速度 (33-快，35-慢)

產生的檔案 bert.cfg 包含資料檔案、維度、一些預設選項以及一些被雜湊停用的可能選項。在運行反演之前，我們想透過呼叫來查看數據<sup>9</sup>

\$bert bert.cfg 顯示數據

圖3顯示了數據，即視電阻率偽剖面。DD表示偶極子-偶極子陣列，數字表示間距。您也可以透過替換來查看錯誤顯示數據和 顯示錯誤。

最簡單的反演運行就是這樣開始的 \$伯特

bert.cfg 全部

包括網格生成和實際反演。輸出也儲存在檔案 bert.log 中，以以下結尾

4 : 型號：最小值=69.271；最大值 = 756.813 4 : 反

應：最小值 = 85.2701；最大值 = 364.752

<sup>9</sup>只有當安裝了 Python (並且在 PATH 環境中找到) 時，繪製任何 show 命令才會起作用  
變數或由 PYTHON 定義)，並且模組 pygimli 和 pybert 透過 PYTHONPATH 找到。

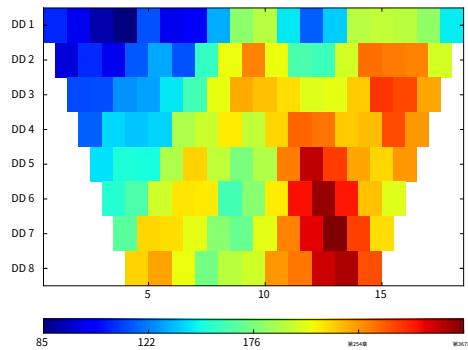


圖 3：使用 showdata 指令的視電阻率偽剖面。

4: rms/rrms(資料、回應) = 3.8314/1.71759% 4:  
 $\chi^2$ (資料、回應、錯誤、日誌) = 1.7069 4: Phi =  
 $198.001 + 19.2525 \times 20 = 583.052$

在每次迭代（開頭的數字）中，為模型及其前向響應指定最小值和最大值。此外，反演指定了資料擬合的幾種度量：以 Ohmm 為單位的絕對均方根 (rms)、相對 rms (rrms)、誤差權重  $\Sigma$

d 卡方擬合  $\chi^2 = \Phi_d/N$  以及由資料失配  $\Phi$  組成的目標函數  $\Phi_d = ((d_{\text{測}} - F_{\text{模}}(\text{米})) / \varepsilon_{\text{測}})^2$  加上正規化參數  $\lambda$ （此處為模型粗糙度的 20 倍）<sup>10</sup>。顯然，數據與約 1.7% 的均方根值吻合，這並不完全在誤差模型內 ( $\chi^2=1$  表示完美配合)。\$伯特 bert.cfg 顯示

使用預設選項顯示圖 4 中顯示的反演結果。它清晰地成像了大約 20m 處的空腔和另一個電阻異常，其起源從 2d 測量結果來看並不完全清楚。有關 3d 資料集的反演，請參閱第 4.1 節。

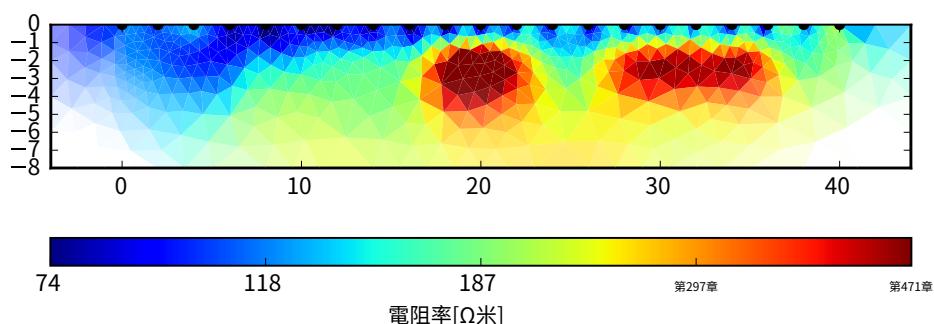


圖 4：使用預設選項的圖庫範例的結果。

它顯示了介質導體上方不斷上升的良好導電的壤土覆蓋層，其中嵌入了兩個不同的電阻體。中央的一座是著名的採礦畫廊，另一座的來歷尚不完全清楚。人們可以點擊模型來獲取有關模型單元的一些資訊。

<sup>10</sup>參見 Günther 等人。（2006）了解詳細資訊和定義。

除非 cfg 檔案中沒有透過關鍵字明確指定，否則將使用 3% 百分位值自動選擇色標（這可以透過設定變數 INTERPERC 進行更改）最小和 c 最大。預設顏色圖取決於您的繪圖庫版本，從v.2.0開始感知綠色彩色地圖取代了舊的噴射規模。作為替代方案，您可以使用類似彩虹的光譜<sup>11</sup>不顯示主要的黃色和青色。您可以使用密鑰地圖選擇其中任何一個繪圖庫色標<sup>12</sup>。關鍵字 使用覆蓋率 (0 或 1 (預設) 定義覆蓋範圍是否用於對模型進行 alpha 著色。鑰匙展示電極定義電極是否繪製為黑點。

模型圖可以使用保存 \$伯特 bert.cfg

`mkipdf`

可以透過調用獲得數據擬合的目視檢查 \$bert bert.cfg 表演

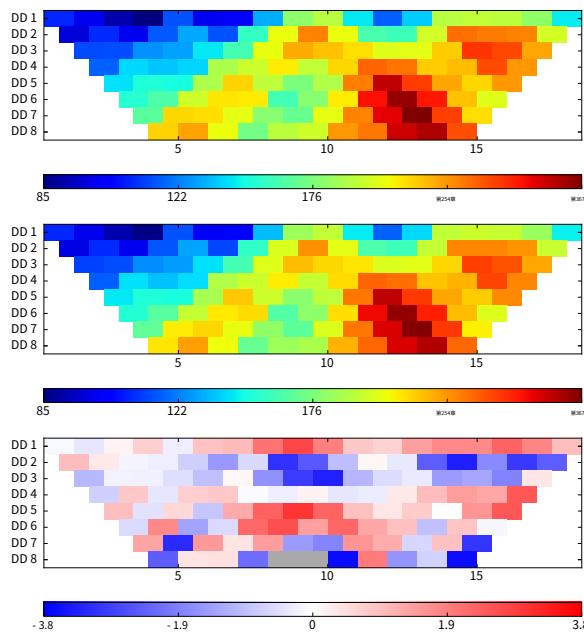


圖 5：測量的視電阻率（上）、模型響應（中）和百分比數據

不合適（底部）。

我們幾乎看不到測量數據和建模數據之間有任何視覺差異。然而，失配圖（也可以透過以下方式作為單一圖獲得）伯特 bert.cfg `showmisfit` 顯示了一些系統的分層，暗示可能還剩下一些資訊。理想情況下，失配圖是不相關的隨機分佈。為了保存結果以供進一步使用，我們使用命令

### \$伯特 bert.cfg 保存

它將所有輸入檔案、網格、日誌檔案和結果（向量、vtk 和 pdf 檔案）保存到結果資料夾中。我們稍後可以轉到該資料夾並進行一些視覺化（甚至開始新的反演）。儲存也會清除工作目錄中的暫存檔案。如果您想手動清潔，請致電

<sup>11</sup>請參閱有關的討論<https://mycarta.wordpress.com/2012/05/29/the-rainbow-is-dead-long-live-the-rainbow-series-out/>  
<sup>12</sup>有關列表，請參閱<https://matplotlib.org/users/colormaps.html>

\$伯特 bert.cfg 乾淨（僅工作目錄）或  
\$bert bert.cfg 非常乾淨（也刪除網格和潛在目錄）。\$伯特 bert.cfg 包甚至壓縮所有結果資料夾。

## 2.2. 正則化和數據擬合

我們現在可能會改變模型的特徵。最重要的關鍵是正則化參數蘭姆達。它控制平滑度約束的強度，從而定義模型的平滑度。因此，我們透過註解掉帶有參數的行來測試不同的值蘭姆達並將其一次設為 200，一次設為 2，而不是預設值 20。bert bert.cfg 計算顯示

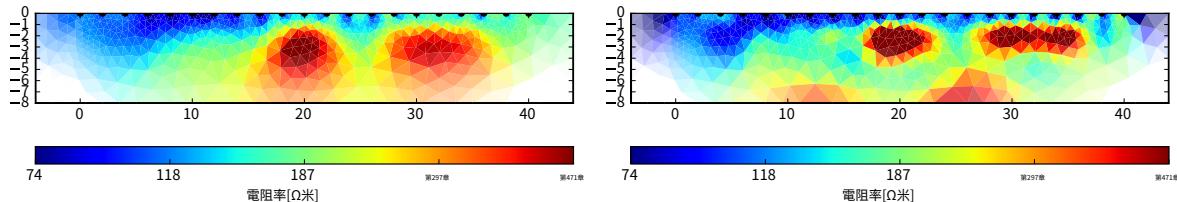


圖 6：正則化參數的結果蘭姆達200（左）和 2（右）。

圖 6 顯示了兩種正則化強度的結果。那個是為了  $\lambda=200$  顯然過度平滑，無法正確擬合資料 ( $\chi^2=7.6$ ，有效值=3.1%)。的價值  $\lambda=2$  可以擬合數據（它停在  $\chi^2=0.8$ ， $rrms=1.2\%$ ）並且可以是一個模型。甚至更低的值也會導致模型振盪。預設情況下，正則化參數保持不變，但在某些情況下，從較高的值開始並在每次迭代中減少它（例如 20%）可能是有益的拉姆達減少=0.8。正則化強度也可以透過兩種方式進行最佳化：一是調整它，使數據在誤差範圍內完美擬合 ( $\chi^2=1$ ) 透過使用CHI1OPT=1。這對於合成數據來說特別有趣。另一個選擇是找到一定範圍內 L 曲線的最大曲率

Günther 等人所描述的 lambda 值。（2006）使用LAMBDAOPT=1。注意 LAMBDA 控制與資料失配項相比的平滑度，其中 rors 用作權重。較高的誤差對應較低的正數據錯譏化，反之亦然。在如何擬合數據的意義上，儀器的疊加誤差並不是適當的措施。估計誤差的一種方法是分析倒數數據，正如 huay 等人所解釋的那樣。（2011）及其參考文獻。如果資料檔案中沒有錯誤，則使用固定百分比進行估計（輸入電平）和電壓誤差

透過 Udp  
錯誤e

（輸入電壓）。如果檔案中沒有電流，則假定值為 100mA。對於不同的電流強度，必須調整電壓誤差。如果檔案中存在錯誤，則可以透過使用新的錯誤估計來丟棄它們覆蓋錯誤=1。請注意，L2 範數（平方）最小化假設（誤差加權）失配的高斯分佈。如果數據中存在顯著異常值，可以使用穩健資料=1，Claerbout 和 Muir (1973) 提出的基於迭代重加權最小平方法 (IRLS) 的 L1 範數方案。然而，人們很容易失去分辨率，這就是為什麼預設不建議使用它，而是應該在觀察資料中的系統異常值後使用。

在許多情況下，只要誤差估計滿足資料質量，預設的正則化強度就會帶來良好的開端。我們建議查看失配的分佈

(圖5c)。在幾乎所有表面測量中，都需要各向異性正則化，即由於主要分層（並且也由方法的較低分辨率支持）而降低垂直對比度的損失。變數權重定義模型中純垂直邊界的相對權重。對於任意邊界（當它們出現在三角形和四面體離散化時），它們按照 Coscia 等人的描述進行計算。  
 (2011)。

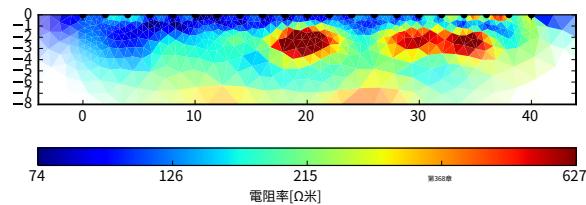


圖 7：使用圖庫範例的結果Z權重=0.4。

圖 7 顯示了垂直權重為 0.4 的結果，這可能稍微改善了解釋。結果，失配函數中的分層消失了。請注意，減少 權重從預設值 1 開始會降低整體平滑度，並可能導致小規模異常導致過度擬合。在這些情況下蘭姆達值應該會增加。為了克服模型中的平滑過渡，可以使用塊模型=1，

即L1範數最小化方案。就像穩健的數據它使用 IRLS 來對各個梯度進行加權，從而產生更強的對比。

到目前為止我們使用的是一階平滑約束，即約束類型約束型1（預設）。也可以使用二階平滑度(2)，即模型的曲率被最小化。零階平滑度(0)意味著僅最小化模型與參考模型差異的範數。這很容易導致模型出現奇怪的振盪，因此僅在已知非常好的起始模型的情況下才建議使用。其他可能性包括使用值 10 和 20 的最小長度約束與一階或二階平滑度的混合。

預設情況下，會產生具有中值視電阻率的均質起始模型。這可以被密鑰覆蓋啟動模型，它可以是電阻率值

或包含整個向量的檔案的名稱。由於長度需要匹配未知型號

單元格，這隨網格選項的不同而變化，這就是為什麼非同質模型更多地透過使用區域來控輕鬆c 制（請參閱第 5.2 章）。預設情況下，起始模型也是ce模型，即模型（包括對數轉換）差參考 異受到約束。發洩這個，使用開關無參考=1。

預

### 2.3. 中號 esh 品質和精細化

當然，反演結果在一定程度上取決於網格。我們回到網格生成，看看網格生成的輸入，即分段線性複數 (PLC)，透過呼叫儲存在 mesh/mesh.poly 中

\$伯特 bert.cfg showpoly並稍微放大一點

我們觀察到兩個物體，一個是紅色的內部（標記 2 的參數域），另一個是更大的標記 1 的外部，這是前向計算所需要的。預設情況下，參數域的深度是根據一維靈敏度研究自動確定的（呼叫命令超深度畫廊.cfg，但可以使用 鍵進行調整

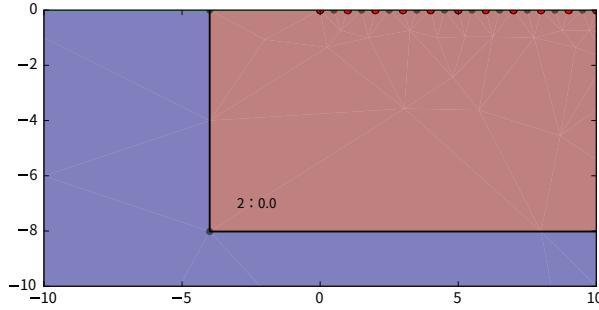


圖 8：圖庫的分段線性複合體（網格輸入）。

閱兵深度。有時建議稍微增加預設值。的價值準邊界定義邊界超出電極（紅點）的距離（以電極延伸的百分比表示，預設=5）。電極之間有輔助點（黑色），用於定義表面的網格細化，由按鍵控制悖論。值為 0.5 表示增加 1 點（如圖 8 所示），0.3 表示增加 2 點。較低的值意味著兩個點設置在每個電極的左側和右側（如果EQUIDIST邊界=0）。如果EQUIDIST邊界=1設定（預設），1/ PARADX 舍入為自然數並設定等距點。標準值為帕拉迪克斯對於平坦地形為 0.25，對於地形為 0.33。

邊界的粗糙度由網格品質控制準二維質量，表示最小角度。品質越高，三角形就越規則，數值精度也會提高，但單元數量和運行時間都會增加。呈三角形<sup>13</sup> (Shewchuk, 1996) 版本 1.6，我們最喜歡的 2d 網格產生器，範圍從 25-30 (品質差) 到 34-35 (品質好)。我們使用預設的起始值 34，但選擇將其增加到 35。Paramax 細胞尺寸 (以米為單位<sup>2</sup>)。或者參數平行邊長用於計算要用於的正三角形的面積Paramax 細胞尺寸。我們建議編輯 cfg 檔案並調用bert bert.cfg 顯示網格直到用戶滿意為止。

在菲古 re 9 顯示不同設定的參數網格結果。顯然，太sh的品質可能會導致下邊界處出現一個大而低我 粗糙的三角形，我們建議  
價值觀 34 到 34.6 之間。限制最大單元格大小可以有所幫助，但也會創建  
模型中某處人為減小三角形尺寸。外模部分有開關邊界，以電極延伸的倍數測量，即值 5 意味著  
在我們的例子中 -x、+x 和 -z 方向有 5x40m 的空間。

#### 2.4. 規則網格

與不規則網格相比，規則網格需要較少的控制，但與三角形網格相比，對於電極處的相同細化，可以導致更多的單元和節點數量。特別是矩形的外邊界變得低效率。因此，常規參數網格被三角形包圍。規則（四邊形）網格可以透過使用輕鬆使用網格=1（需要Python+pygimli）。另外帕拉迪克斯和 閱兵深度，鑰匙帕拉茲可用來定義第一層厚度（否則

<sup>13</sup>看<https://www.cs.cmu.edu/~quake/triangle.research.html>

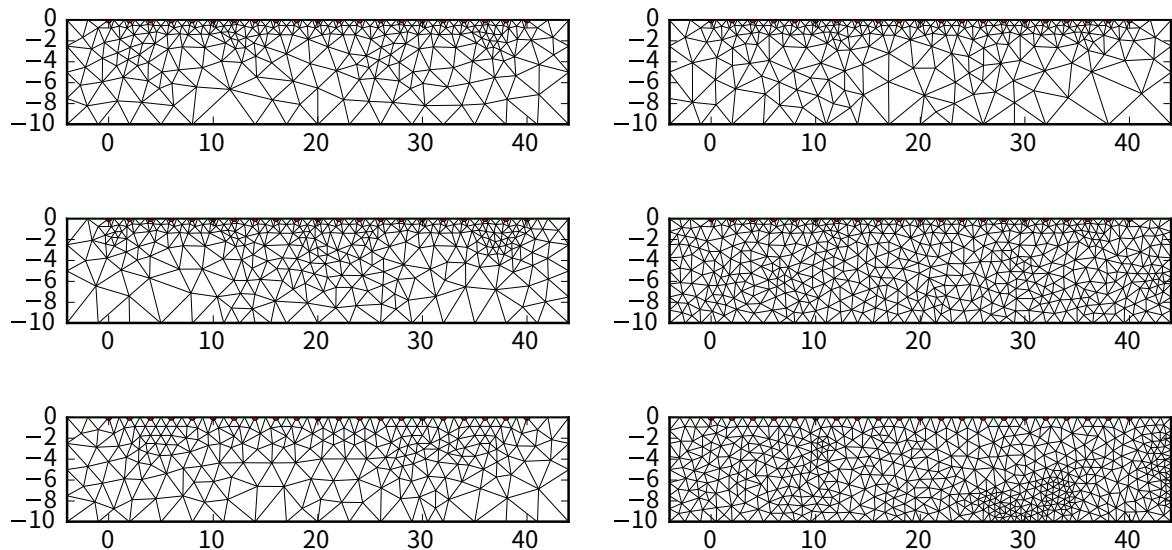


圖 9：具有不同參數的網格： $dx=0.25$  &  $q=34$ （左上，預設）， $dx=0.25$  &  $q=33$ （右上）， $dx=0.25$  &  $q=34.5$ （左中）， $dx=0.25$  &  $q=34$ ， $\text{maxA}=1$ （右中）， $dx=0.5$  &  $q=34$ （左下）， $dx=0.5$  &  $q=34.5$ ,  $\text{maxA}=1$ （右下）

等於悖論）。鑰匙層數定義以層厚度線性增加的方式建立的層數（預設為 11）。圖 10 顯示了預設網格選項的結果，與圖 10 和圖 10 非常相似。6b 或 7。

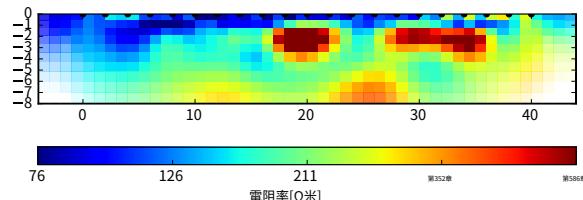


圖 10：常規參數網格的反演結果。

## 2.5. 我誘發極化

那裡 除了電阻率振幅之外，還有幾種類型的激發極化 (IP) 數據，在頻域 (FD) 中測量：

相移使用單一或多個頻率

頻率效應根據兩個頻率下的電阻率測量計算得出

或在時域 (TD) 中：

總充電能力表示衰減曲線的積分（以毫秒為單位）

積分充電能力由閻極長度和電壓歸一化的時間視窗（以 mV/V 為單位）

BERT IP 反演面向單頻 FD 測量，但也可以應用於任何其他類型的數據，只要線性影響很小即可。IP 資料在資料檔中表示為名為 ip 的列，在（直流）電阻率反演後進行反演。如同 Martin 和 Günther (2013) 所描述的，實際的 IP 反演使用實部和虛部之間的 Cauchy-Schwarz 關係來將虛部電阻率的反演表述為線性問題。最後，虛電阻率轉換回相位（以 mrad 為單位）。至於 IP 反轉，可能會使用不同的正規化強度，您可以使用金鑰蘭姆達伊普，否則它正在採取蘭姆達電阻率反演的值。然而，由於小相位的線性度，任何其他類型的資料（以 mV/V 或毫秒為單位的表觀充電能力或以 % 為單位的頻率效應）都可以輸入，因此輸出將具有相同的單位。作為這種非常簡單方法的範例，我們提供了 Martin 和 Günther (2013) 在第 3.3 節中提供的健康橡木的數據。

在許多情況下，光譜 IP 數據是透過測量多個頻率或分析衰減曲線的不同部分來獲得的。由於這超出了資料格式和分析的範圍，因此頻譜分析不是命令列 bert 的一部分。相反，我們提供了一個用於 SIP 資料處理、反演和後處理的 Python 類別 (Günter et al., 2016)。它直接讀取現場 SIP 數據，並且可以對所有頻率進行頻譜約束反演，然後進行 Cole-Cole 分析，如 Günther 和 Martin (2016) 所述。

### 3. 其他二維幾何形狀

#### 3.1. 有地形的 2D ERT

在二維反演中，透過設定電極的高度可以輕鬆積分形貌。其餘的一切都應該自動完成，如有必要，必須插入額外的電極。然而，很少會對所有電極進行形貌測量。通常有幾個點就夠了。請注意，在當前階段，BERT 需要第一部分中的地形訊息，而不是文件末尾的地形資訊。對於這種情況，我們建議使用 DC2dInvRes (Günther, 2007)，它將沿著表面滾動位置。對於此任務，請使用資料：保存歐姆文件並指定 x 值是沿捲尺還是真實 x。

#### 渣堆概況

該現場案例是 Günther 等人描述的第一個案例。(2006)，基礎 BERT 文章，作為 3D 案例，由漢諾威聯邦地質和自然資源研究所 (BGR) 的 M. Furche 友好提供。數據在範例/反演/2dtopo/slagdump 是穿過渣堆的剖面之一 (圖10)。溫納陣列 A=採用 2m 間距，產生 222 個資料點。透過平整在 8 個點測量地形並附加到原始文件，然後使用 DC2dInvRes 進行轉換。我們使用以下命令使用標準選項初始化反轉：

\$bertNew2DTopo slagdump.ohm>伯特設定檔

在非平凡幾何的情況下，我們還有另一個數值任務，就是計算主勢（否則透過分析計算），它與以 PRIM 開頭的幾個關鍵字相關聯。與參數網格類似，普里米德克斯值指定電極處的（絕對的，而不是相對的！）細化PRIM2D質量定義網格增長。此外，PRIMP2網格（預設值 1）決定由二次形函數計算的主勢（P2 細化）。正如 Rücker 等人所述。

(2006) P2 網格的必要細化約為  $A/10$  和  $A/P1$  網格為 100。通常表面上的點是線性內插法的，樣條邊界=1強制進行樣條插值，這對於「圓形」幾何形狀或平滑地形很有用。

計算主要潛力（包括插值到次要前向）後

網格（由伯特 bert.cfg 鍋），有兩個文件 PrimaryPot/pot.ohm 和 PrimaryPot/pot.collect 分別描述所選陣列和所有電極組合的單位電導率的潛力。這些用於計算幾何因子和視電阻率（bert bert.cfg 過濾器）。透過致電伯特 bert.cfg 托波夫人們可以看到地形效應（Rücker et al., 2006），即有地形和無地形的幾何因素的比率  $t = G_{\text{平坦}} / G_{\text{地形}}$ 。bert bert.cfg 顯示 topocorr 另外也顯示基於平坦和地形幾何因素的視電阻率（請參閱圖 11）。有些異常現像只能用地形起伏來解釋。電阻率分佈顯示了 Günther 等人討論的導電內部和電阻硬碟。（2006）。

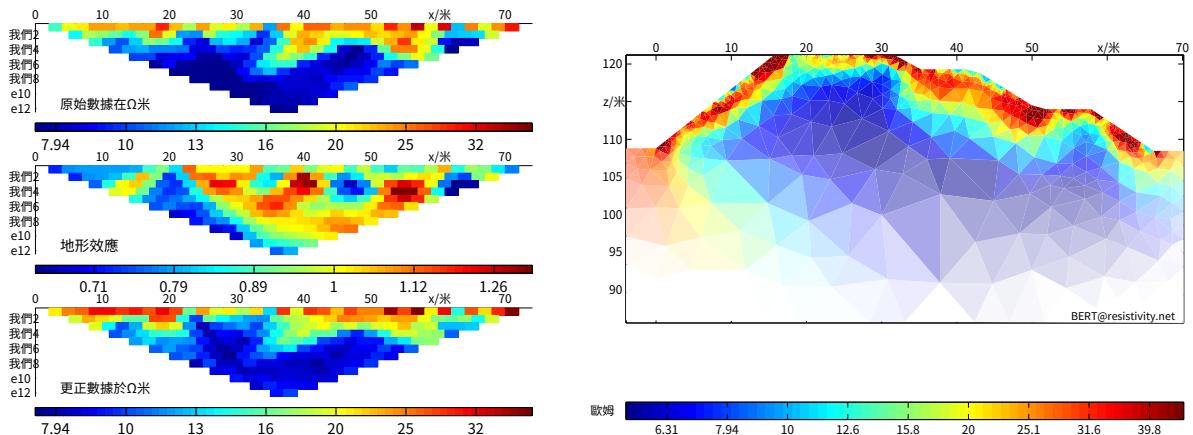


圖11：渣場地形影響及反演結果。

### 3.2.二維跨孔數據

當然，跨孔測量也可以使用 BERT 進行反演。每個電的高度電極必須設定為高程減去深度。然而，由於我們無法區分是否它是地形或埋置電極，我們必須手動建立幾何形狀。中的例子範例/反轉/2dxh 由英國地質調查局 (BGS) 的 O. Kuras 在 ALERT 項目 (Kuras et al. (2009)) 中製作，用於時移反演（參見第 6.3 節）。它代表了 5 個非常之間的跨孔測量所獲得的約 1300 個數據淺 (0-1.6m) 鑽孔。為了建立反轉網格，我們將在一個大框內建立一個帶有標記 2 (反轉) 的小框，用於透過 8 個點和 8 個邊進行正向計算 (標記 1)。

這是由 PLC 腳本更嚴格地實現的多平面世界這會自動根據參數計算模型的尺寸和電極周圍的邊界、準邊界、悖論和閱兵深度。它是由

PARAGEOMETRY="polyFlatWorld 2dxh.ohm" <sup>14</sup>。和以前一樣，我們可以使用帕拉迪克斯

<sup>14</sup>先前的版本包含單字「來源」（在目前 shell 中運行），這可能會導致錯誤。和我們一樣檢測到它是一個命令，我們自動獲取來源以便能夠存取 cfg 檔案中給出的參數。

透過每個 0.1m 分離電極之間的節點來細化電極處的模型，使用PARADX=0.05。請注意，與二維表面測量不同，它絕對是透過聚扁平世界。由於分層，我們使用Z權重=0.1，一個更大的 拉姆達=100和穩健資料=1。圖 12 顯示了示蹤劑實驗開始時所獲得的電阻率分佈。

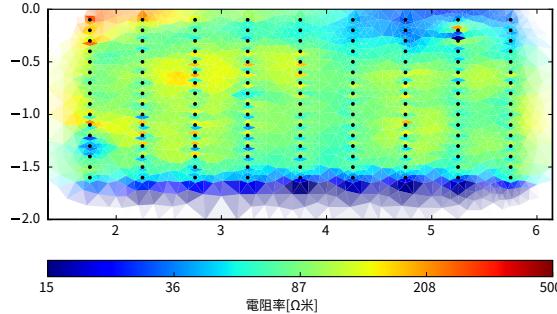


圖12：井間資料集反演結果。

在 BERT 2 中，電極不再必須是點。然而，對於表面測量，使用它們的位置作為節點來建構網格幾乎總是有意義的。這在跨孔幾何形狀中是不同的。與腳本類似多平面世界有一個腳本聚自由世界它取代了 cfg 檔案中的前者。

由於三角形的約束太少，因此生成的網格在整個區域上非常粗糙。我們可以放置一些點來確保局部較小的三角形，並使用良好的網格質量來緩慢增加邊長。或者，我們可以（全局）提供最大三角形面積，例如透過使用選項副邊長度，其中面積是根據正三角形計算的。透過將實際值設為 0.05，我們獲得了足夠細的網格，與電極位置無關。

第三種方法是使用下列任一方法建立規則網格網格=1或使用者定義的Python腳本（例/inversion/2dhw/reg/mymesh.py）。圖 13 顯示了兩個替代結果。一般來說，電阻率分佈非常相似，即深度的良導體和

電阻器 在左上角。但小規模含水層異常有差異。

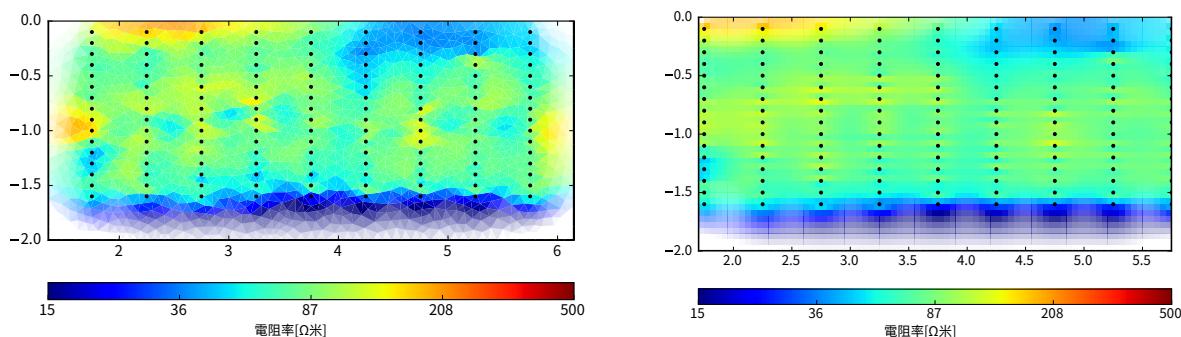


圖 13：替代網格類型的反演結果：自由電極（左）和矩形網格（右）。

### 3.3. 封閉二維幾何 - 樹斷層掃描

#### 空心椴樹上的 ERT

對於二維物體，電極通常位於整個邊界上，並且 PLC 可以很容易地由閉合多邊形形成。對於樹幾何形狀，創建了名為 TreeBERT（在 DC2dTree 之前）的專用 GUI，可以輕鬆地以視覺方式處理資料。樹木 EIT 已成功建立，用於研究樹木腐爛情況（Martin 和 Günther，2013）。中的例子範例/圓/樹由 Niels Hoffmann（原 HAWK Göttingen）測量和提供。它代表一棵椴樹，透過插入樹皮的 24 個鋼電極進行測量。偶極子-偶極子測量已使用 Geotom 儀器進行應用。

設定檔內容如下

```
資料檔案=hollow_limetree.ohm  
尺寸=2  
地形=1          # 啟動主網格  
氣缸=1          # 定義一個封閉的幾何體  
表面光滑=1      # 打造更好的表面  
EQUIDIST邊界=1 # 等距細化  
PARADX=0.2       # 電極之間有 5 段  
PARA2D 質量=34.8 # 品質非常好，幾乎是上限  
樣條邊界=1      # 圓形幾何形狀  
PRIMDX_R=0.001   # 徑向細化初級網格  
拉姆達=10        # 正規化強度  
塊模型=1          # 透過穩健的 (L1) 方法增強對比度
```

對於這種情況，等距細化，使用樣條線和具有圓形邊界的高品質網格。初步細化是在肯定不錯徑向進行的。我們使用穩健的建模以獲得高分辨率的更清晰的對比度  
另外活力。

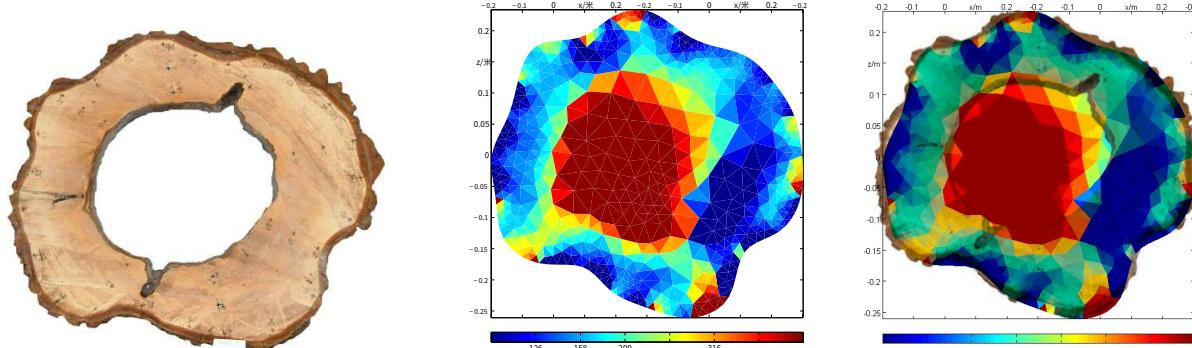


圖 14：樹切割（左）、反演結果（中）和疊加。

測量完成後，這棵樹被砍伐，發現內部有一個由十水合物造成的空洞，顯示了照片、y。圖14  
反演結果以及兩者的疊加。顯然，高電阻率形成的電洞與照片幾乎完全一致。y 被標記

## 健康橡木上的 ERT 和 IP

中提供的數據範例/反轉/圓/橡木/Martin 和 Günther (2013) 對此進行了描述，他們還提供了有關測量和反演的詳細資訊。在夏季和冬季對一棵健康的橡樹（論文中的 HI 部分）進行了測量。我們在夏天進行測量 (hp5-s.dat)並使用給定的 cfg 檔案反轉它hp5-s.cfg。查看也儲存在中的輸出伯特日誌我們看到以下內容：

線性搜尋 tau = 0.12

3 : 型號：最小值=211.068；最大值 = 435.779 3 : 反應：  
最小值 = 252.394；最大值 = 359.383 3: rms/rrms(data,  
回應) = 82.4875/25.4984% 3: chi^2(資料, 回應, 誤差, 日  
誌) = 68.29 3: Phi = 19052.9+4.1727336.

達到資料擬合標準 (delta phi < 2%) 。停止。處理ip資料

min=9.84261 max=37.4019

◦◦◦ # 這裡還有一些東西取決於詳細程度

反轉 DC 數據後，dcinv 會尋找 IP 數據，如果沒有則停止 (min/max=0) 。然後，IP 資料被反轉，預設輸出是靜默的，我們看不到單獨的反轉（因為它會妨礙查看 DC 資料擬合），而只能看到最後一次迭代。

15 : 型號：最小值=0.00531699；最大值 = 33.0494 15 : 反  
應：最小值 = 1.49463；最大值 = 17.6232 15: rms/rrms(資  
料、回應) = 0.893923/12.1318% 15: chi^2(資料、回應、錯  
誤、日誌) = 7.36795 15: Phi = 2055.66+290.928

結果相位：最小值=0.0178055 最大值=111.453 mrad。

在這種情況下，模型（虛電阻率）位於幾毫歐姆和幾歐姆之間，對應於超過 100mrad 的相位值。資料擬合度為 12%，但是當 IP 資料接近零時，此測量值可能沒有意義。卡方誤差是基於假設的 1mrad 相位誤差。所得的相位影像可以透過以下方式顯示

\$bert hp5-s.cfg 顯示phase.vector或使用導出為 pdf \$bert hp5-s.cfg mkpdf 相位.向量。

預設正規化強度（蘭姆達在這種情況下，對於 DC 和 IP 反演都有好處。一般來說，不同的值可能是合適的。如果參數蘭姆達IP給出後，該值將僅用於 IP 反演。請注意，需要重新執行計算 (calc)。透過 Python 類電阻率（用於單一 DC/IP 反演）和 SIPdata（用於頻譜 IP 反演）可以提供更靈活的 IP 反演可能性。

## 4. 3D 幾何形狀

### 3D 表面測量

3D 表面測量可以透過多種方式進行：

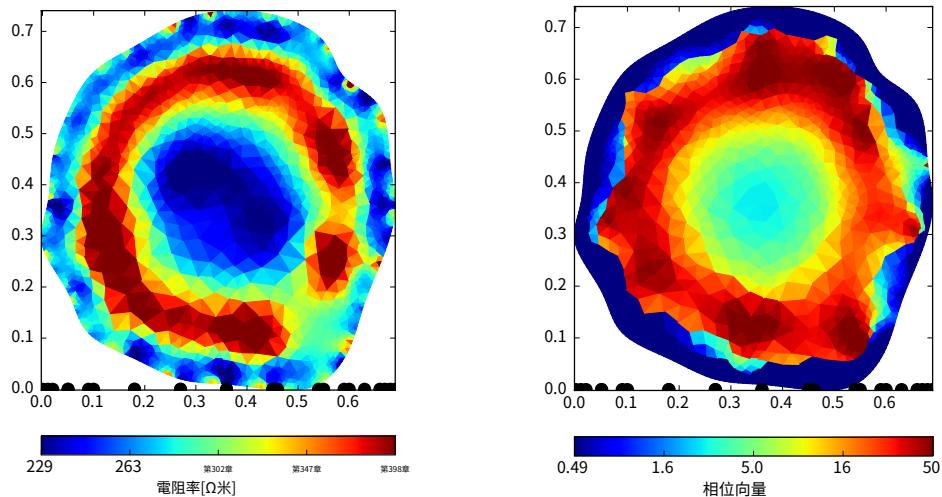


圖 15：健康橡樹（夏季）的電阻率（左）和相位（右）影像。

1. 電極網格的佈局。然而，由於電極數量有限，網格僅限於小區域。
2. 沿座標軸的平行（和垂直）輪廓。
3. 由於可達性限制，可在任意方向上進行輪廓。
4. 非剖面佈局，例如大規模偶極-偶極實驗。

無論如何，電極位置和測量必須根據統一的資料格式定義。前兩種類型的資料可以輕鬆地手動組織。對於第 3 個（和第 2 個），我們建議準備 2d 檔案並編寫一個設定文件，其中包含具有 2d 檔案名稱的線以及該線所在的點對。該檔案可讀入 DC3dInvRes Günther (2008) 並用於寫入 3d 檔案。對於地形，最好在使用 DC2dInvRes 和 Export Ohm 之前對 2d 檔案進行磁帶校正。

3d 中最靈活的元素是四面體。四面體化是由網格產生器完成的，我們的主要選擇是 Tetgen (Si, 2008)，這是一個免費且多功能的高品質網格產生器 (Si, 2015)，可以從 <https://tetgen.org>，但是，您也可以使用網格產生器 GMesh (Geuzaine 和 Remacle, 2009 年)，如附錄所述。質量度量與 2d 不同，描述的是半徑與邊緣的比率，請注意，較小的值表示較高的質量。（主場）正向計算的適當值為 1.12 至 1.2，對於 1.2-1.5 的反向（因此次要）網格值是適當的，關鍵是 PRIM3D 質量和超三維品質。

#### 4.1. 平坦的地球表面測量

在範例/反轉/3dflat/畫廊是測量 2dflat 範例（第 2.1 節）的現場資料集。更多資訊請參閱 Günther (2004)。它由 9x14 電極網格組成。偶極子-偶極子測量已在所有 x 和 y 剖面上進行了測量。資料檔案中存在常數 0.0 的錯誤，該錯誤將會自動覆寫。具有預設參數的反演專案檔案是透過以下方式建立的：

\$bertNew3D gallery.dat >伯特設定檔

然後完全運行反轉（使用命令全部）收斂到約 2 的卡方失配 ( $rrms=4-5\%$ )。為了更好地擬合數據，使用減少正則化參數 拉姆達=5，這導致約 3% 的相對均方根誤差 ( $\chi^2=1$ )。另一種方法是使用各向異性正規化（權重以下 1）也導致  $\chi^2 \approx 1$ 。

反演結果已經儲存為vtk檔名dcinv.結果.vtk可透過ParaView等3D視覺化軟體查看<sup>15</sup>。

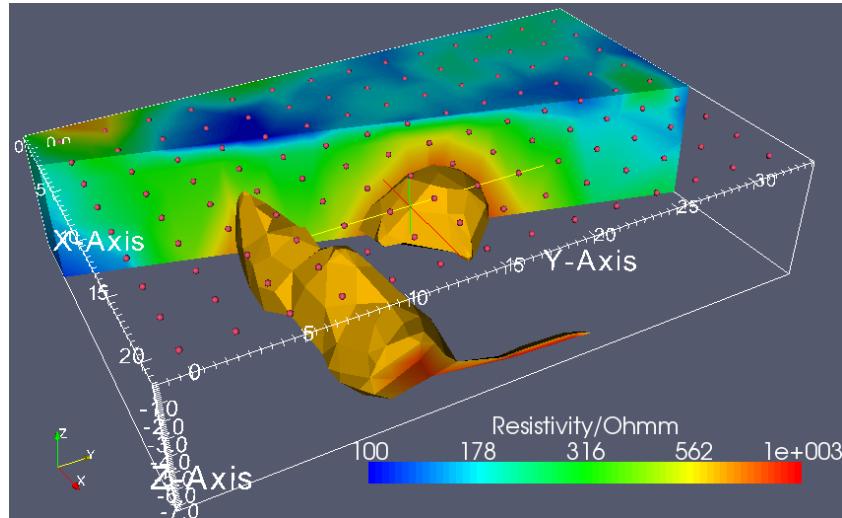


圖 16：使用  $650\Omega\text{m}$  平滑等值面的 3d 圖庫資料集反演結果  
和一個平面夾，紅色球體是使用的電極。

圖 16 顯示了透過以下步驟建立的 Paraview 視覺化：i) 單元資料到點數據，ii) 透過標量  $650 (\Omega\text{m})$  進行剪輯，iii) 提取表面，iv) 平滑表面，v) 基於 Cell2Point 的另一個剪輯與 Plane, vi) 將輸入表示為輪廓軸和立方體軸。彩條為對數，手動範圍為  $100-1000\Omega\text{m}$ 。電極已包含為點 vtk 文件，並由 Glyph 顯示為半徑 0.05 的球體。經過一些練習後，讀者將能夠創建漂亮的圖像、繪圖併計算結果，例如地質體的延伸或體積。

#### 4.2. 3D 地形

3D 地形的定義比 2D 地形複雜得多，其中每個形狀都可以用簡單的多邊形來描述。輸入 PLC 由面而不是邊組成，產生的多邊形檔案具有相似但不同的格式<sup>16</sup>。一般來說，過程如下：i) 建立一個平面網格，ii) 根據地形資訊插入高度，iii) 在其周圍製作一個小（反轉網格）和一個大（正向網格）框，iv) 如有必要，進行細化和 v) 使用 tetgen 建立網格。整個過程由 Günther 等人介紹。（2006）並由 Udpuhay 等人更詳細地解釋。（2011）以陡峭的懸崖為例。

為了指定地形訊息，有兩種不同的方式：

<sup>15</sup>看<https://www.paraview.org>

<sup>16</sup>看<https://tetgen.org>。

- 數據檔案中的電極有一個標高，所有其他點都經過內插
- 有一個數字高程模型 (DEM) 或至少一個測量的地形點列表 (在包含 x、y 和 z 的 3 列文件中)

雖然第一種情況足以實現平滑的地形和/或密集的電極覆蓋，但後者更為普遍。地形點可以是（大多數是規則的）數位高程模型 (DEM) 或不規則測量點的清單。在任何情況下，它們都經過 Delaunay 三角測量，為網格表面的每個點（即電極以及整個前向網格）提供標高，因此實際電極分佈之外的點特別有用。兩種方法都可以結合使用，但是我們建議將資料檔案中的電極高度設為零，以便從 DEM 中進行內插。確保與 DEM 一致的測量高程的電極可以附加到地形檔案中。包含 xyz 列地形點的 ASCII 檔案由下列方式指定 TOPOPOINTS=檔名。

在例/反演/例/acucar有一個項目由漢諾威聯邦地質和自然資源研究所 (BGR) (M. Furche) 進行測量。該地點是一個古老的礦渣場，其地形讓人想起裡約糖帽。跨越孤山頂部測量了兩個電阻率剖面。另一種輪廓是在山周圍以或多或少恆定的高度實現的。雖然這不是作為電極網格的密集採樣，但它應該足以獲得粗糙的圖像。

除了電極之外，還測量了一些地形點並將其放入文件中點.xyz。所以我們使用建立一個新項目 \$ bertNew3DTopo acucar.ohm >伯特設定檔

並添加行TOPOPOINTS=點.xyz到模型。如果我們現在打電話 \$bert bert.cfg 網格

我們看到網格沒有顯示山丘，因為地形覆蓋了電極高程。因此，我們必須將電極定義（第 3-230 行）新增至地形檔案中，然後查看山丘（圖 17 左）。然而，由於點密度，電極線顯示為銳利邊緣，這並非真正的事實，但在這種情況下就足夠了。

在其他情況下，我們可能有數位高程模型。為了在同一範例中展示這一點，我們透過在 2m 間距的規則網格上對可用點進行三次插值來建立一個範例。為了避免電極之間的插值錯誤 (Udphuay 等人，2011，圖 10)，我們為三個輪廓創建了一個多邊形檔案 poly.xyz<sup>17</sup> 號並介紹它 TOPOPOLY=poly.xyz。圖 17 顯示了兩種變體的表面網格。鋒利的邊緣現在消失了。

最後，反演結果如圖 18 所示。當然，剖面之間和模型邊界處的資料覆蓋率較低。因此，模型或多或少受到平滑度約束的內插法。

為了創建適當的 3D 網格，可能需要在進行反演之前使用網格選項。而不是通常的細化參數帕拉迪克斯要細化表面的網格，您可以考慮使用三角形表面的參數：表面質量（三角形網格的質量，請參閱二維反演），SURFACEMAXTRI-SIZE（最大元素大小）和表面光滑。常有人觀察到醜陋

---

<sup>17</sup> 號多個多邊形以空線分隔。

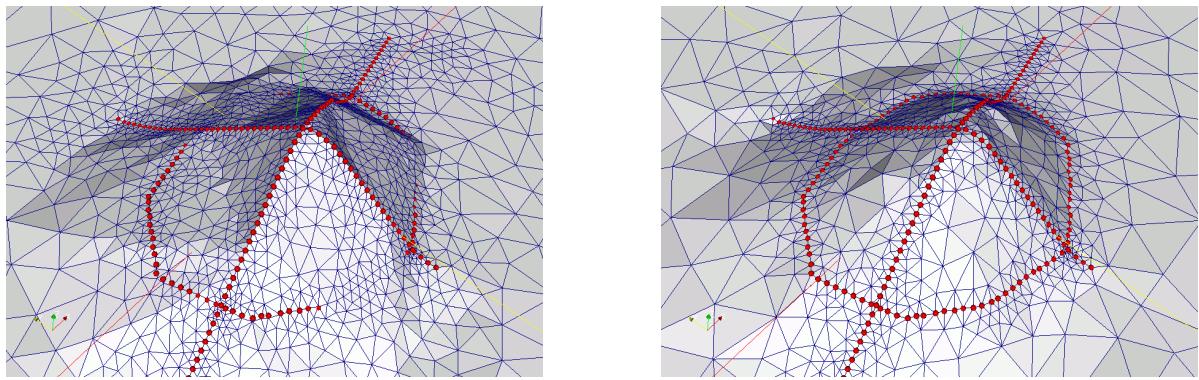


圖 17：逐點地形資訊（左）和數位元素的表面網格  
vation 模型（右），電極顯示為紅點。

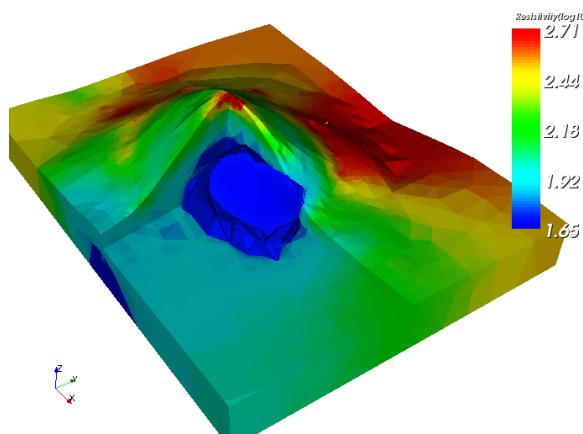


圖 18：3dtopo 情況的反演結果。

參數域邊界處的形狀四面體和對應的粗離散化。這可以透過指定來克服盧普特根=1,即循環調用網格生成器以逐步提高網格品質。

#### 4.3. 3D 跨孔測量

跨孔測量當然可以應用在三維。中的例子例/反演/3d 由蘇黎世聯邦理工學院的 J. Doetsch 提出。資料檔案3dhh歐姆包含飽和區 ( $d=4\text{-}10\text{m}$ ) 4 個鑽孔之間的 753 個數據，是監測實驗的一部分。我們使用以下命令建立一個 cfg 文件

```
資料檔=3dhh.ohm
維度=3
準邊界=15          # 在電極周圍獲得更多空間
PARAGEOMETRY="polyFlatWorld $DATAFILE"
```

透過使用Z權重=0.2我們可以增強主要的分層結構。然後反演收斂，預設值下降到大約  $\chi^2=1$ 。圖 19 顯示了最終結果。

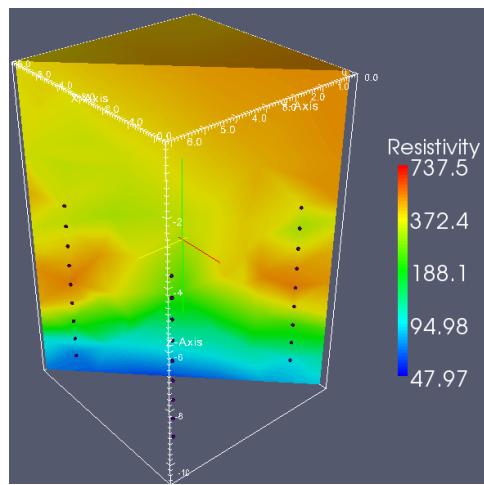


圖 19：3D 跨孔情況的反演結果。

#### 4.3.1. 地形和埋置電極

如果給了表面下的高度和深度，我們就不能使用創建曲面/創建ParaMesh 和聚扁平世界。在這種情況下，我們建議建立一個包含表面電極（帶有拓撲或帶有零和拓撲檔案）的資料檔案以及包含真實電極的地下電極清單。正常的 PLC 生成是透過第一個完成的，然後使用添加後者聚添加VIP。重要的是，必須設定電極的順序，以便首先出現表面電極，然後出現埋地電極。使用以下命令引入生成的腳本 參數分析=myscript.sh如下圖所示：

```
cp bert.cfg 表面.cfg          # 製作包含所有選項的副本
echo DATAFILE=datafile-without-electrodes.dat >> surface.cfg bert
surface.cfg 域 # 建立 mesh/mesh.poly polyAddVIP -m -99 -f 鑽孔電
極.xyz mesh/mesh.poly
```

最後，所有電極都以正確的順序進入 PLC，一切都應該很好。

#### 4.4. 封閉的 3D 幾何形狀

封閉幾何圖形實際上比開放幾何圖形更容易，因為我們不需要網格延伸和兩個不同的區域。然而，由於整個邊界是諾伊曼型的，我們必須確保兩個在開放情況下不需要的附加條件：

- 電流不可能無限消失，因此我們必須使用偶極源，例如通過參考電流節點。
- 由於邊界值問題中僅存在導數，因此我們必須使正向解唯一，例如透過添加參考電位節點，其電位強制為零。

## 坦克 3D 模型

在模型槽中進行 ERT 測量的例子有很多，可以是圓柱形 (Garre 等人，2010) 或立方體 (Bechtold 等人，2012；Persson 等人，2015)。在漢諾威聯邦地質和自然資源研究所 (BGR)，創建了一個圓柱形模型罐 (Falcon-Suarez 等人，2016)，以便用來自鋸末或爐渣堆的材料進行滲透實驗。柱子直徑30厘米，高80厘米。在垂直距離為5公分的5個環中，每個環中安裝了24根2公分長的鋼電極。偶極子-偶極子測量已應用於所有環，產生了 320 個數據。此範例位於範例/反轉/3dtank。

由於無法從檔案中自動偵測參數化，因此我們必須建立網格輸入，即 PLC網格/網格.聚使用腳本手動操作。有一個多邊形工具 多創建立方體建立一個單位立方體。隨著選項-Z相反，它會創建一個單位圓柱體，必須對其進行適當的縮放。然後我們將電極作為點放入<sup>18</sup>帶有標記-99。我們插入兩個帶有標記 -999 和 -1000 的附加節點，用於當前參考和潛在參考。所以腳本是這樣寫的：

```
MESH=mesh/mesh # PLC名稱
polyCreateCube -v -Z -s 48 -m 2 $MESH # 建立具有 48 段的單元圓柱體 polyTranslate -z -0.5
$MESH
# 移動它使得頂部為零
多尺度 -x 0.3 -y 0.3 -z 0.8 $MESH
# 縮放至半徑 0.15 和高度 0.8
貓咪土壤_column.dat | head -n 82 |tail -n 80 > elec.xyz # 萃取電極polyAddVIP -m -99 -f
elec.xyz $MESH # 將電極加入網格中polyAddVIP -m -999 -x 0 -y 0 -z 0 $MESH #目前參考節點
polyAddVIP -m -1000 -x 0 -y 0 -z -0.8 $MESH # 潛在參考節點 polyConvert -V -o $MESH-poly
$MESH # 轉換為 vtk 以載入至 paraview
```

我們建立一個空的 cfg 檔案（或使用 bertNew3DTopo），其中包含以下行

```
資料檔=soil_column.dat 繩
度=3
地形=1
氣缸=1      # 確保閉合幾何
```

我們以以下形式新增 PLC 腳本參數幾何學變數使得網格/網格.聚 將由它創建。

參數分析=mymesh.sh

可以透過品質改進來實現進一步的細化（優質）因為它相當粗糙，局部細化（悖論）或最大像元大小 (Paramax 細胞尺寸)。為了獲得準確的結果，我們對 1cm 的主網格和以大約 32000 個節點結尾的二次形函數進行了細化。

```
PRIMDX=0.01
PRIMP2MESH=1
```

圖 20 顯示了從網格輸入通過參數網格到最終結果的過程。

<sup>18</sup>由於與柱尺寸相比，電極沒有顯示出顯著的延伸，因此我們不將點放在表面但將其向內移動 1 厘米，請參閱 Rücker 和 Günther (2011) 來優化最佳點。

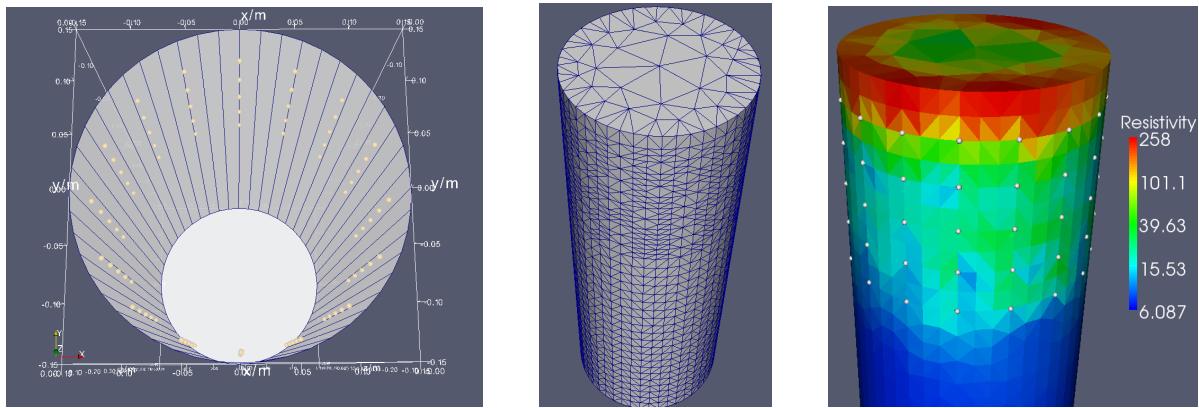


圖20：土柱的PLC（左）、參數網格（中）與反演結果（右）  
實驗。

對於立柱實驗，z 方向由於重力而不同。因此，在三角柱網格上運行反演並透過以下方式規範垂直方向可能會有所幫助：ZWEIGHT。請注意，這樣的腳本是透過帕拉梅什 應附有參數幾何學為初級電位產生外部 PLC 的腳本。

#### 重要提示：

我們計算了電極位置，使它們恰好落在圓柱棱鏡的邊界上（有一些捕捉以避免舍入誤差）。對於不同數量的稜柱邊，很容易發生電極位置落在 PLC 外部或非常接近內部的情況。第一種情況會導致錯誤，而後者會在這些點上導致不必要的精細離散化。一般來說，Rücker 和 Günther (2011) 討論了兩種替代方案：

- 電極從來都不是無限小的點（意味著無限的電流密度），而是具有實際尺寸。如果電極尺寸與其距離相比很大，則可以透過完整電極模型 (CEM) 來模擬真實形狀，如 Rücker 和 Günther (2011) 所描述的那樣。儘管這很容易建模，但這意味著在反演中創建不同網格時需要付出努力。
- 另一種方法是將電極放入內部。Rücker 和 Günther (2011) 模擬了一個立方模型儲罐，發現沿著電極軸約 60% 的替換點可將誤差降至最低。

## 5. 納入先驗訊息

通常還有關於地下的附加資訊。與結果比較相比，始終優先考慮納入反演過程。

## 5.1.結構性限制

Günther 和 Rücker (2006) 提出了一種更通用的最小化方法，允許模型單元之間的每個邊界具有任意權重。Rücker (2010) 更詳細地解釋了數值方法。在存在已知的不連續性的情況下，可以將其設為零，從而允許（但不強制）電阻率的任意跳躍。此類約束可用於 2D (Bazin 和 Pfaffhuber, 2013) 和 3D Doetsch 等人。 (2012) 案例。

Rücker (2010) 也提出的範例位於 (範例/反轉/反轉/- 2dstruct) 由 K-UTec GmbH Sondershausen (T. Schicht) 測量並友善提供。研究的目的是利用電阻率和折射地震進行基岩探測。速度結構顯示為非常清晰的 2 層情況。因此折射研究的結果（層邊界）可以作為結構資訊。

文件基岩.xz 包含 xz 對形式的邊界走向。我們現在使用以下命令將此文件包含到設定檔中介面選項。為了比較有結構和沒有結構的結果，我們調用

```
$bertNew2D bedrock.dat>伯特設定檔 $  
bert bert.cfg 全部顯示  
$迴聲介面=bedrock.xz>>伯特設定檔 $bert  
bert.cfg 全部顯示
```

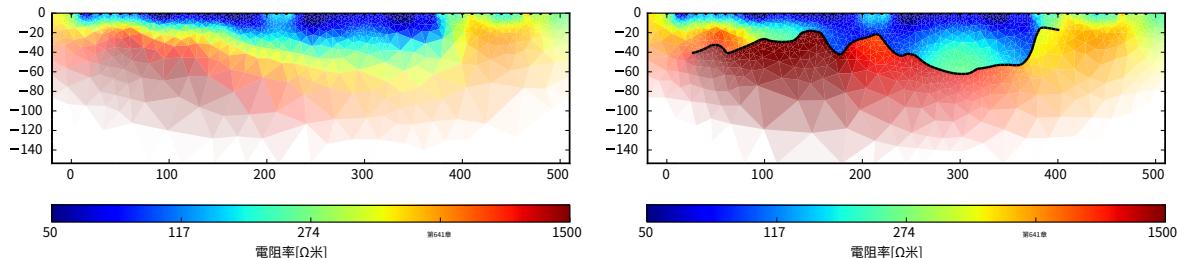


圖 21：沒有（左）和有（右）結構資訊的電阻率分佈。

圖 21 顯示了不帶結構資訊和結構資訊的地下圖像。明顯地添加附加資訊可以使地下影像更加清晰。在大多數位置，邊界處的電阻率反差很大。然而，在我在那裏這些位置，要么速度受到影響，要么折射結果不明確。  
不同

塞韋拉 介面可能存在於介面文件，以空白行分隔。結構約束也可能來自指示某一層的鑽孔描述，圖拉爾這會導致幾條小線延伸到側面，這取決於橫向代表深度。

森塔蒂鑽孔的活力。三維界面是三角測量的點列表並作為面放入模型中。

## 5.2.透過水下範例進行區域特定控制

在前面的例子中，我們幾乎將模型分成了先驗知識已知的兩個部分。很多時候，地下的不同部分（即區域）需要不同的處理。這可以是地質單元或人工物體，例如地下的鑽孔或電纜。弗萊希格等人。 (2010) 僅使用單一區域來反轉非常稀疏的資料集。多奇等人。 (2010) 使用鑽孔作為單一區域，透過跨孔 ERT 對含水層進行成像。科西亞等人。 (2011) 將其嵌入大型 3D 模型中以監測含水層動態。

Rücker (2010) 更詳細地描述了區域背後的技術。區域是具有恆定標記的網格的一部分，它們的行為可以透過區域文件輕鬆描述。可以透過呼叫查看該標記的分佈bert cfg 顯示標記。預設情況下，有一個嵌入建模區域（標記 1）的反演區域（標記 2），其電阻率不是反演的一部分，而是透過從反演區域延伸而用於正演計算。我們建議對此背景區域使用最低的數字，因為這是預設行為，以防在非諾伊曼體中檢測到多個區域。

## 區域文件

對反轉的主要控制是由所謂的區域文件使用關鍵字完成的RE-GIONFILE (和延時區域文件用於時間步長反演)。當然，還有很多選擇，例如LAMBDA、ZWEIGHT、上界、下界可以直接在cfg檔案中針對整個反演域（無論是一個區域還是多個區域）進行設置，但可以針對不同的區域單獨進行調整。對於更嚴格的描述，我們參考 GIMLi 教程，附錄 C。

```
# 無啟動 Ctype MC 0    zWeight Trans lBound uBound
    100   1     1   0.2      紀錄   50     1000
  1   30   0     0.2 1      紀錄   10     200
# 沒有單一開始 2 1
    100
# 無背景
- 1 1
# 區域間
0 2 0.2
```

對於所有區域，數字 No 也可以是 \*，並且應該位於令牌清單的前面。其他標記定義編號標記的值。

開始起始電阻率

C型約束類型 (1/-平滑度一階/二階，0-最小長度，10/20 混合一階/二階  
零階)

MC模型控制 (個體 lambda 相對於全局 lambda)

z重量垂直邊界的約束權重 (僅適用於Ctype=1)

反式模型變換，預設為log(LU)，其他為lin或tan

綁定電阻率下限 (Trans=log)

優邦德電阻率上限 (Trans=log)

單身的該區域經過處理，電阻率恆定

使固定此區域具有已知的電導率，可用於正向計算，但不用於  
反轉

背景該區域是背景並充滿了鄰近區域的電阻率

對於前向計算，對於沒有區域檔案的非諾依曼域，預設最低數字是背景

區域間兩個區域預設是解耦的（約束權重為零），但可以耦合

懇求<sup>19</sup>

在上面的範例中，有兩個具有不同起始值、有效範圍和限制的主要區域。另一個區域被視為恆定並且與區域 0 弱耦合。最後，有一個背景區域。

## 湖案

湖泊資料集由漢諾威萊布尼茨應用地球物理研究所（W. Südekum 和 T. Günther）測量，位於例/反演/區域/湖。目的是描繪奧斯納布呂克附近 Feldungel 湖下方的沉積結構。電極已從一岸沿著湖底散佈到另一岸。間距為2m，對 Wenner-alpha 和 Wenner-beta 進行測量並合併。電極位置（0 到 -2.6m 高度）和電阻是輸入檔。由於湖泊電阻率是一個獨特的已知體（無論是其幾何形狀還是  $22.5 \Omega m$  的電阻率），因此將對其進行不同的處理。

1. 我們從地形狀況開始並產生網格劃分輸入 bert bert.cfg 域
2. 結果我們得到了 poly 文件網格/網格聚我們複製到 mymesh.poly<sup>20</sup>
3. 我們需要透過左岸和右岸之間的一條邊（高度為零的最內點）來添加水面。Poly 檔案的視圖顯示它們由點 3 和 138 表示。`151 3 138 -1` (number n1 n2 edgemarker) 並將第 152 行中的邊數從 151 增加到 152
4. 最後，我們透過附加線在湖中某處添加區域標記，標記為 1 (未倒置) `2 50 -1 1 0.0` (number xy 標記 maxTriangleSize) 並將區域數量從 2 增加到 3。

我們透過引入在反演中使用這個改變的 poly 文件參數幾何=mymesh.poly 到 cfg 檔中。這意味著 mesh 目錄中的 mesh.poly 是透過複製 mymesh.poly 來建立的。或者，我們可以在此處放置一個生成幾何體的 bash 腳本或 Python 腳本 (mesh/mesh.poly)。

圖22顯示了呼叫後輸入PLC的一部分伯特 bert.cfg showpoly。像往常一樣，我們有帶有標記 1 的外部區域和標記 2 的內部區域，此外我們還有用於湖泊的標記 3。預設情況下，外部區域是背景區域，以防有多個區域在沒有區域檔案的情況下可用。

我們使用以下選項

<sup>19</sup>您可以使用先前的密鑰區域間也是如此，但由於不同的破折號有多種，這是錯誤的  
我們推薦使用的關鍵表區域間。

<sup>20</sup>見三角頁 <https://www.cs.cmu.edu/~quake/triangle.research.html> 用於文件描述

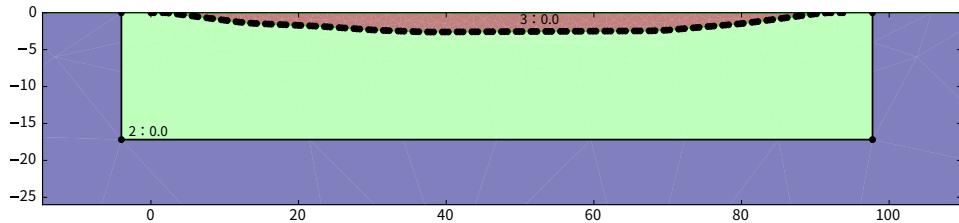


圖 22：湖案例的輸入 PLC 表示。

```

Z權重=0.2      # 增強層狀（沉積物）結構
覆蓋錯誤=1      # 不使用文件中的測量誤差（樂觀），但是：
輸入等級=2      # 2% 以上
輸入電壓=20e-6 # 20 微伏特 cMin=15

```

最大c=500

且反演收斂於  $1 < \chi_2 < 2$ 。圖 23 顯示了所得的電阻率分佈。湖泊沉積物通常顯示出非常低的電阻率，甚至低於測量值  $22.5 \Omega\text{m}$ 。

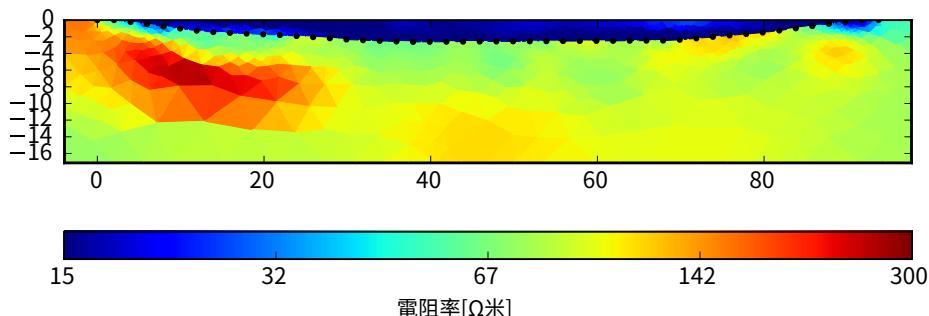


圖 23：使用自動解耦的兩個區域的水情況反轉結果。

那裡 giions 會自動解耦（它們之間沒有平滑度約束），如範例所示。圖 24a 顯示了使用正常處理的反演結果（透過將湖泊的標記設為 2 或使用多平面世界作為參數幾何），IE 最後的前任 地區 整個模型的平滑約束。我們看到很多與湖底有關的結構和不切實際的高水電阻率。現在我們開始治療

REGIONFILE=區域.control

我們首先將水區域（標記 3）定義為單參數區域，將地下定義為正常反演區域，具有增強垂直結構的平滑度約束，範圍為  $10-1000 \Omega\text{m}$ ，起始電阻率為  $100 \Omega\text{m}$ ：

```

# 沒有開始 Trans zWeight Ctype lBound uBound 2
100 log 0.11101000
# 沒有單一開始 3 1
22.5

```

```
# 無背景
```

```
1 1
```

圖 24b 顯示了結果。主圖像類似，但海底的振盪消失了。底部電阻率的增加消失了，揭示了更多關於更深層的資訊。然而，湖泊的電阻率得到的值約為  $16 \Omega\text{m}$ ，低於在側面測量的預期值  $22.5 \Omega\text{m}$ ，儘管我們將其用作起始值。

我們可以透過電阻率上限和下限擴展區域 3 的線來縮小湖泊的電阻率值：

```
# 無單一起始 Trans lBound uBound 3 1
```

```
22.5 log 22 23
```

雖然單一區域由一個未知數表示，但也可以將其與固定值相關聯，使其成為背景區域（即將其從反演中完全刪除）。

```
# 沒有修復
```

```
3 22.5
```

雖然正常背景（區域 1）是從反演模型中用鄰近電阻率填充（延長）的，但該區域是用固定值填充的。

結果如圖 24c 所示，也很好地擬合了數據。兩個結果顯然是等價的，在這個全空間問題中，湖泊電阻率無法獨立取得，必須添加額外的資訊。

或者，我們可以假設湖底沉積物的電阻率與水非常相似。因此，我們再次刪除修復並在兩個區域之間引入區域間約束：

```
# 沒有單一開始 3 1
```

```
22.5
```

```
# 區域間
```

```
2 3 0.5
```

因此，其他解耦區域透過平滑約束連接，其強度比正常弱 2 倍。結果與數據等效，如圖 24d 所示。它與圖 24c 非常相似，因此支持測量的電導率。

另一方面，從我們的簡單探針並不清楚水的電阻率是否真的恆定。因此我們可以將水視為具有適當上限和下限的正常區域。為了使我們的目標結構位於水層中，我們將該區域的模型控制（相對正則化參數）增加了 2 倍。

```
# 未啟動 Trans zWeight Ctype lBound uBound MC 2
```

25	紀錄	0.1	1	10	1000	1	
3	22.5	日誌	0.01	1	10	30	3

```
# 區域間
```

```
2 3 0.5
```

然而，結果（圖24e）與其他結果並沒有太大不同，但證明了區域方法的靈活性。

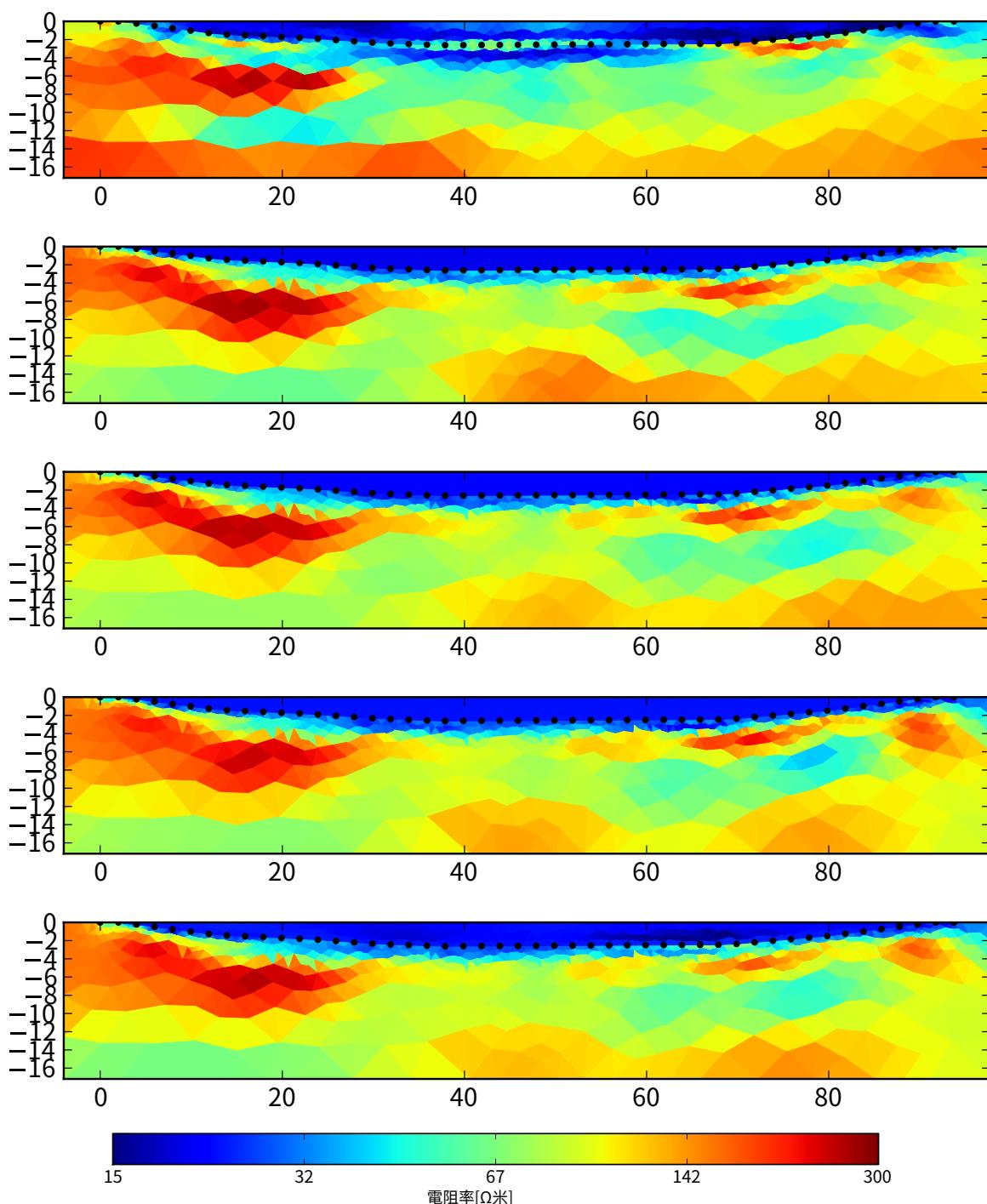


圖 24：不同選項的反演結果（由上至下）：  
 a) 處處平滑，  
 b) 湖泊為單一參數區域，c) 湖泊為固定區域，d) 湖泊與湖底之間的區域間約束，e) 與 d 類似，但水體可變。

## 6. 延時 ERT

我們經常對正在進行的物理過程感興趣，並使用 ERT 來監控實驗。可以透過將檔案名稱寫入文字檔案並將其傳遞來處理任意數量的後續資料集TIMESTEPS=檔名。

### 6.1. 策略

有大量不同的策略用於進行延時反演，因此有大量的關聯鍵。

1. 最簡單的是獨立反轉，可以使用簡單的 shell 腳本輕鬆完成，只需覆蓋數據檔。然而，經常會出現偽影，尤其是在查看變化（比率）時。
2. 另一種變體是比率或商反演，即計算數據比率並將其反演為視電阻率（Schütze 等，2002）。然而，這種方法僅對小對比度有效，因為它沒有考慮靈敏度分佈作為真實模型的函數。BERT1預設使用這種方式，這裡仍然可以使用比率步長=1。
3. 有一類基於參考模型的方案，其中對每個幀都進行了完全最小化，但模型僅限於採用第一個（預設）或前一個（延時步長模型=1）框架作為參考。這是最通用的方案，因為所使用的測量陣列甚至電極位置都會隨著時間而改變。因此它是預設方法。
4. LaBrecque 和 Yang (2001) 提出的特殊方案是所謂的差分反演。它基於以下觀察：所有時間步驟中都存在大量系統誤差。因此，如果TIMELAPSEREMOVEMISFIT=1，數據 $d_k$ 的 $k_{th}$ 幀透過第一幀的失配進行校正 ( $d_0$ ) 這樣
$$\| d_k - \text{調頻}_k ) - d_0 + \text{調頻}_0 \|$$
在模型差異正規化旁邊最小化 $\bar{m}_k - \bar{m}_0$ （已經在最後一個方案中）。如果存在已知的系統錯誤，我們建議這樣做。如果僅存在系統誤差，這只會增加誤差，從而增加平滑度。無論如何，使用相同的陣列。
5. 從反演的角度來看，最嚴格的方法是時間完全離散化同時反演。這是使用區塊矩陣有效完成的，但到目前為止僅在 Python 管理員中可用。

### 6.2. 處理和選項

正在進行的研究的一個問題是依賴時間的誤差模型。常態倒數資料的分析可能可以克服有效平滑度不同的問題。

通常，各個資料檔案中的陣列可以不同。然而，要使用差分反演，它們必須相同。為了避免透過過濾規則剝離數據，我們建議使用 減持=1。因此，數據被預處理成數據立方體（數字與數量）。

時間）。過濾規則以外的不存在資料或測量值保留在檔案中，但會因大誤差而降低權重。

為了定義延時策略，使用上述按鍵。預設情況下，對第一幀進行參考反轉，除了延時步長模型=1確定時間步長的正規化。如果使用拉布雷克法（不同的反演）延時移除移動=1。

延時反演可能需要很長的計算時間。如果變化很小，則可以透過使用省略主要計算工作，即雅可比矩陣的重新計算快速延遲=1，但僅在相同數組的情況下有效。

由於模型差異通常具有不同的範圍和形狀，因此其正則化可能不同於靜態反演。鑰匙拉姆達延時拍攝定義正規化強度，延時約束約束的類型（0、1、2、10 或 20，請參閱第 2.1 節）。儘管由於存在大量偽影，零階約束（無平滑度）通常會阻礙靜態反演，但它對於延時問題可能很有趣，因為平滑度不會系統地改變模型空間。更一般地，區域文件（參見第 5.2 節）可以應用於時間步反演，具體如下：TIMELAPSREGIONFILE。

反演的結果包含在二進位檔案中模型 Abs.bmat (絕對值) 和 modelDiff.bmat (相對差異%)。對於 2D 情況，它們可以用作參數，以及指定時間步長的數字，位於顯示/mkpdf顯示模型或其偏差的命令，例如

\$bert cfg 顯示模型Abs.bmat 5 顯示第 5

個時間步長的模型，\$bert cfg 顯示

modelDiff.bmat 5

顯示與基本模型（預設）或先前模型 (TIMELAPS-ESTEPMODEL=1) 的相對差異。請注意，在這些情況下c最小/c最大必須給出值。省略時間步數將顯示基準模型（目標展示）或建立多頁 pdf（目標mkpdf）。在 2D 和 3D 中，對於每個時間步 ia 檔案 dcinv.result i.vtk 都會被創建，以便在 Paraview 中，如果載入群組，它們會自動載入為時間步 dcinv.結果..vtk。vtk 檔案包含相對於基線的絕對值和模型比率。

### 6.3.跨孔延遲測量

讓我們回到跨孔案例 2dxh（請參閱第 3.2 節）並解壓縮 2dxh-timelapse.zip 中的時間資料檔。2006 年，BGS 在 8 號鑽孔中注入了高鹽示踪劑，每 40 分鐘測量 36 個資料集，這樣就覆蓋了一整天。後續檔案名稱為 01.dat、02.dat...，並組裝在 timesteps.txt 中。

透過包括TIMESTEPS=timesteps.txt並打電話bert bert.cfg 計算再次。結果，我們獲得了幾個被 ParaView 識別為時間步長的 vtk 文件，從而可以輕鬆地滾動時間。

圖 25 顯示了一些選定的時間步長，可以看到示踪劑流向左邊界。請注意，這些只是用於展示 BERT 工作原理的初步結果。透過更複雜的延時策略，可以更準確地追蹤監控過程。

大多數情況下，人們對變化（比率）感興趣並想要測試幾種演算法，這是針對圖 26 中的不同設定完成的。

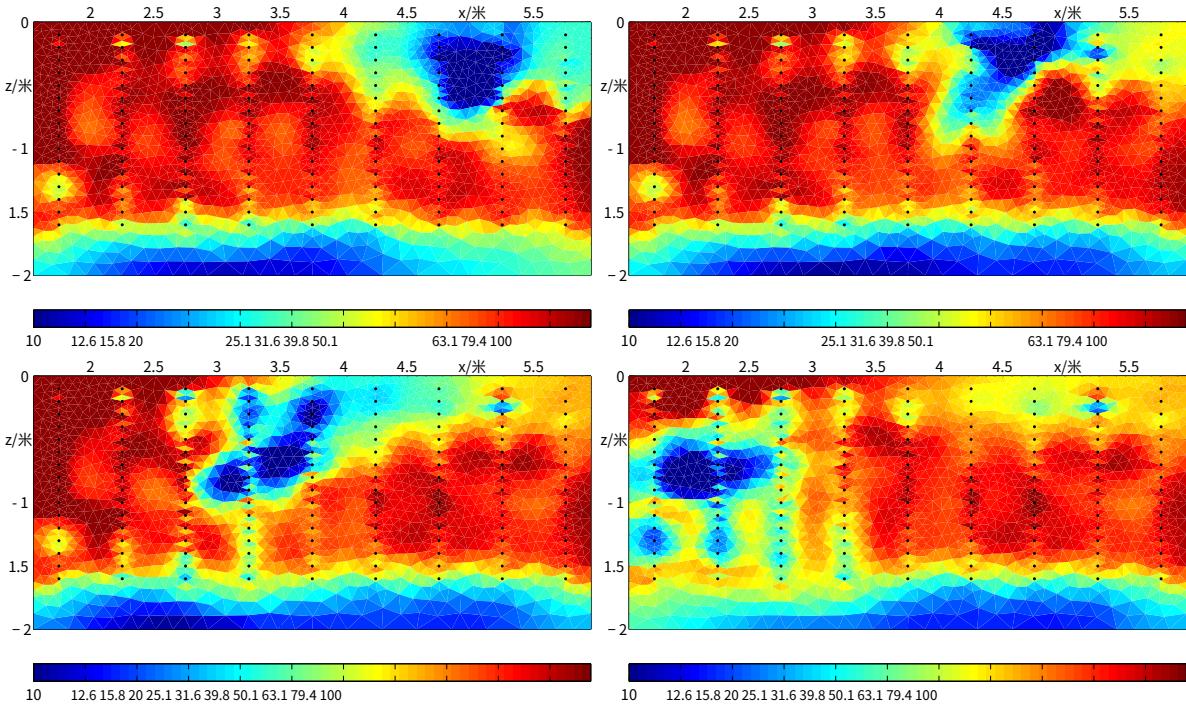


圖25：3小時（左上）、7小時（右上）、12小時（左下）反演結果  
以及示蹤劑注射後 16.5 小時（右下）。

從正則化階數為0、1和2的反演（第3-5行）來看，經典平滑度約束（垂直權重略有下降的第一/第二）表現出最大的影響，但在示蹤劑上方和下方也表現出最大的偽影。模型差異的最小長度正則化不會顯示此類偽影，但會在電極處增加，從而中斷示蹤劑形狀。如果將（各向同性）平滑度和純偏差結合（最後一行），則觀察到的偽影最少，但示蹤劑的形狀仍然中斷。進一步的調整可能會產生更好的圖像，但是事先並不清楚哪種方法最好以及現實是什麼樣子。

#### 6.4. 土柱測量

我們回到 4.4 節中的土柱範例。灌溉一定量的水後，每2小時測量一次完整的資料集並納入時間序列文件。由於變化相對較小，我們來看看檔案 diff.i.vtk 相對於初始電阻率的相對差異。圖 27 顯示了 5 個選定的時間步長。我們可以看到水鋒侵入，但在某個階段，柱子再次變乾。

#### 致謝

我們要感謝所有提供了非常有指導意義的人：d Donner（以前的 TU Berlin 與使用者故事：Folker Freiberg）、Markus Furche 和 Ulla Noell Schicht（K-UTec GmbH (BGR Hannover)、Thomas Sondershausen）、Niels Hoffmann（以前的 Kuras（英國地質調查 HAWK 哥廷根）、奧利佛和伊拉里亞·科局）、Joseph Doetsch（蘇黎世聯邦理工學院））  
西亞（Oliver 和 Ilaria Coscia）（前

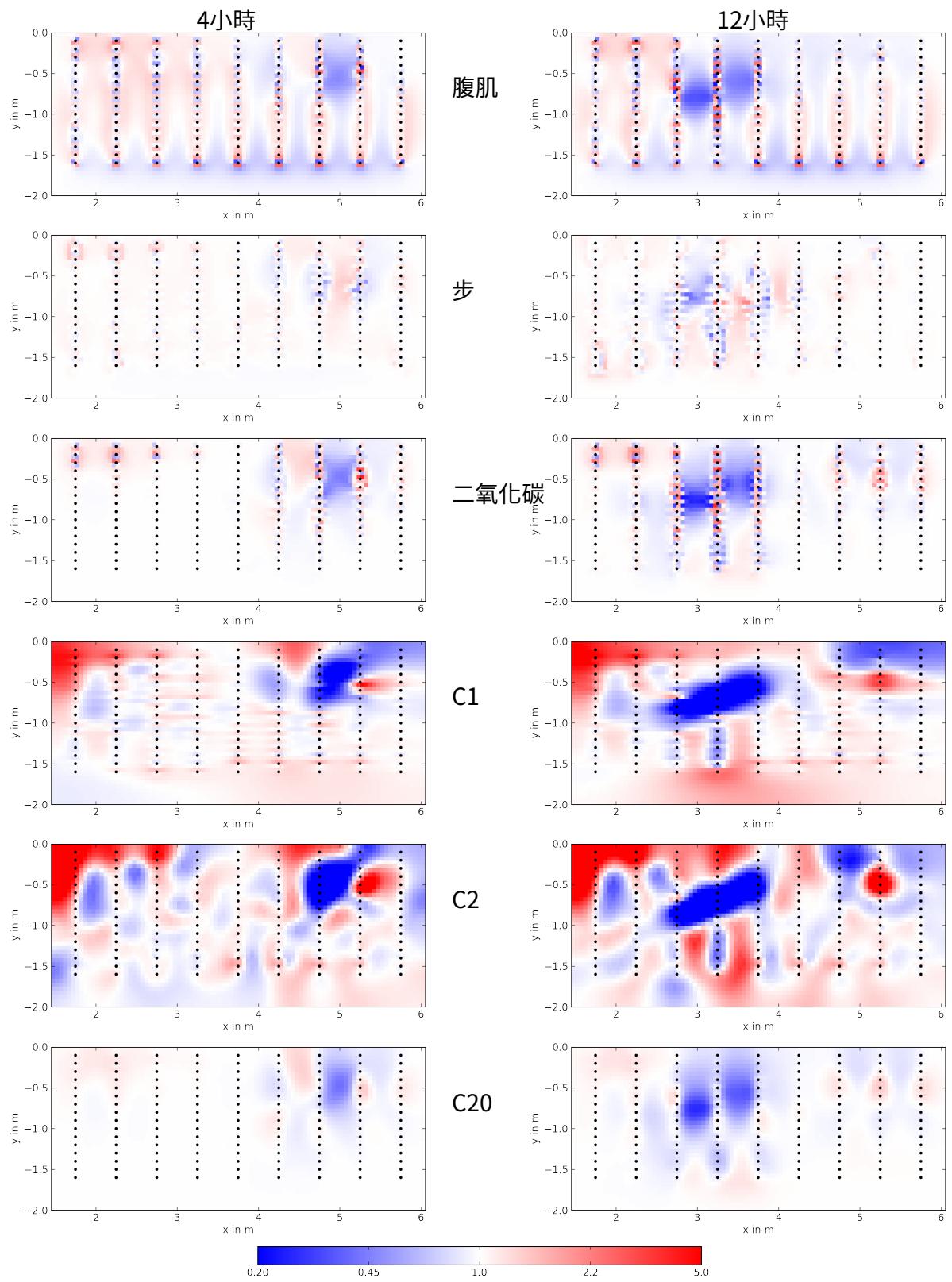


圖 26：時間步長 4 小時（左）和 12 小時（右）的電阻率比以及反演  
 方案：單獨反演、逐步約束和使用約束階數 0、1、2 和 0+2 的差分反演

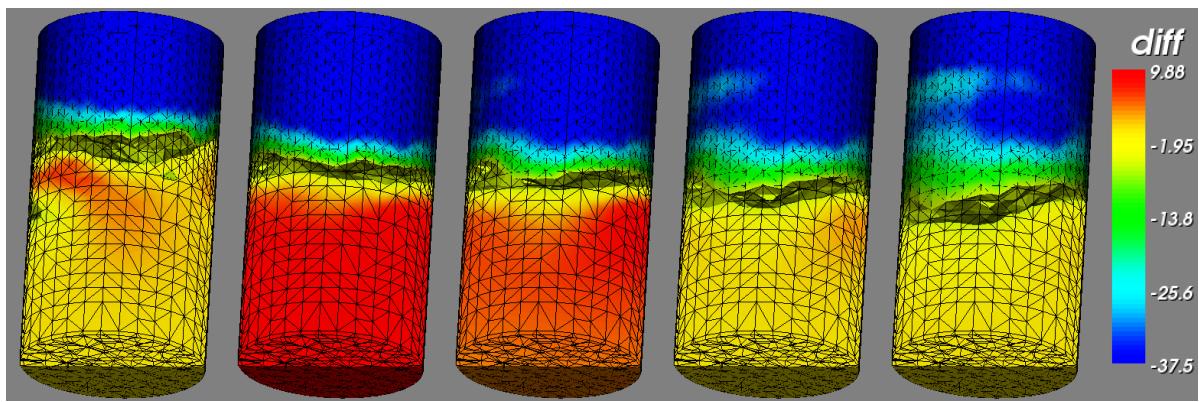


圖 27：在約 2、4、  
灌溉後6、10和16小時。

蘇黎世聯邦理工學院），Sarah Garre（列日大學，Gembloux）。此外，我們感謝 BERT 的所有使用者和測試人員，正是他們使該軟體成為了現在這樣一個強大的專家工具。

## 參考

- 巴贊，S. 和普法夫胡貝爾，A. (2013)。透過電阻率斷層掃描繪製快速黏土圖  
phy 在結構約束下。應用地球物理學雜誌，98：280–287。
- Bechtold, M.、Vanderborght, J.、Weihermüller, L.、Herbst, M.、Günther, T.、Ippisch, O.、Kastel, R. 與 Vereecken, H. (2012)。蒸發條件下三維異質實驗室土壤的向上傳輸。滲流區雜誌，11 (2)。
- 克萊爾布特，JF 和繆爾，F. (1973)。具有不穩定資料的穩健建模。地球物理學，38 (1) : 826-844。
- Coscia, I.、Greenhalgh, S.、Linde, N.、Doetsch, J.、Marescot, L.、Günther, T. 和 Green, A. (2011)。河川-含水層耦合系統的 3D 孔間視電阻率靜態反演與監測。地球物理學，76 (2) : G49-59。
- Doetsch, J.、Coscia, I.、Greenhalgh, S.、Linde, N.、Green, A. 與 Günther, T. (2010)。這電阻率成像中的鑽孔流體效應。地球物理學，75(4) : F107–F114。印刷中。
- Doetsch, J.、Linde, N.、Pessognelli, M.、Green, AG 與 Günther, T. (2012)。約束 3D 電阻層析成像與 GPR 反射資料可改善含水層特性。應用地球物理學雜誌，78:68 – 76。
- Falcon-Suarez, I.、Juncosa-Rivera, R.、Vardon, P.、Ramlmair, D.、Günter, T.、Noell, U. 和 德爾加多-馬丁，J. (2016)。來自龐特維德拉（西班牙）石材工業的壓實花崗岩鋸末的水動力行為：實驗和建模。環境地球科學，75(5)。

- Flechsig, C.、Fabig, T.、Rücker, C. 與 Schütze, C. (2010)。地電調查  
Cheb 盆地/W-波希米亞：一種評估近地表電導率結構的方法。螺柱。地球物理學。喬德，54：  
417-437。
- Garre, S.、Günther, T.、Diels, J. 與 Vanderborght, J. (2012)。評估實驗設計  
等高樹籬間作系統土壤濕度監測的 ERT。滲流區雜誌，11 (4)。
- Garre, S.、Koestel, J.、Günther, T.、Javaux, M.、Vanderborght, J. 與 Vereeken, H. (2010)。  
使用 ERT 在兩種不同土壤中觀察到的異質傳輸過程的比較。滲流區雜誌，9 (2)：207-212。
- Geuzaine, C. 和 Remacle, J.-F. (2009)。Gmsh：三維有限元素網格產生器  
具有內建預處理和後處理設施的產生器。國際工程數值方法雜誌，79：1309–1331。
- Günter, T.、Martin, T. 與 Rücker, C. (2016)。使用 sip 現場資料進行頻譜反演  
皮吉姆利/伯特。在第四屆誘導極化國際研討會。
- 岡瑟, T. (2002-2007)。DC2d/InvRes - 直流二維反轉和分辨率。電阻-  
ity.net 製作, <http://dc2dinvres.resistivity.net>。
- 岡瑟, T. (2003-2008)。DC3d/InvRes - 直流 3d 反演和分辨率。電阻-  
ity.net 製作, <http://dc3dinvres.resistivity.net>。
- 岡瑟, T. (2004)。2D/3D 重建的反演方法與解析度分析  
直流測量的電阻率結構。博士論文，弗萊貝格礦業技術大學。可以在<http://fridolin.tu-freiberg.de>。
- Günther, T. 與 Martin, T. (2016)。頻域頻譜二維反演  
來自礦渣堆的誘發極化數據。應用地球物理學雜誌，在新聞。公認。
- Günther, T. 與 Rücker, C. (2006)。引入結構資訊的通用方法  
- 從約束到聯合反演。在分機。摘要，EAGE 近地表地球物理研討會。3.-6.9.06, 赫爾辛基(芬蘭)  
。
- Günther, T.、Rücker, C. 與 Spitzer, K. (2006)。直流電阻率的 3D 建模和反演  
包含地形的資料 - 第二部分：反演。地球物理學。J. 國際。, 166 (2) : 506–517。
- 庫拉斯, O., 普里查德, J., 梅爾德魯姆, PI, 錢伯斯, JE, 威爾金森, PB, 奧美, RD,  
和 Wealthall, GP (2009)。使用自動延時電阻率斷層掃描 (ALERT) 監控液壓過程。Compte  
Rendus Geosciences - 水文地球物理學特刊, 341 (10-11) : 868–885。
- LaBrecque, DJ 與 Yang, X. (2001)。ERT 資料的差分反演：快速反演  
3-D 原位監測方法。J. 環境。工程師。地球物理學。, 6:83。
- 馬丁, T. 和岡瑟, T. (2013)。用於真菌檢測的複電阻率斷層掃描 (CRT)  
和直立的橡樹。歐洲森林研究雜誌, 132 (5) : 1-12。

Persson, M.、Dahlin, T. 與 Günther, T. (2015)。觀察瓶蓋中的溶質傳輸  
在實驗室實驗中使用影像分析和電阻率斷層掃描的細管條紋。滲流區雜誌, 14 (5) : 11p。

Ronczka, M.、Rücker, C. 與 Günther, T. (2015)。長電極電動勢數值研究  
電阻率斷層掃描—準確度、靈敏度和解析度。地球物理學, 80 (6) : E317-E328。

呂克, C. (2010)。使用非結構化的先進電阻率建模和反演  
離散化。博士論文, 萊比錫大學。在準備。

Rücker, C. 與 Günther, T. (2011)。使用完整的有限 ERT 電極模擬  
電極模型。地球物理學, 76(4) : F227-F238。

Rücker, C.、Günther, T. 與 Spitzer, K. (2006)。直流電阻率的 3D 建模和反演  
資料合併地形 - 第一部分：建模。地球物理學。J. 國際。, 166 (2) : 495-505。

Schütze, C.、Friedel, S. 與 Jacobs, F. (2002)。三維傳輸偵測  
使用地電商斷層掃描在多孔含水層中進行流程。歐元。J. of Env. 和恩金。地球物理學, 7 :  
3-19。

休楚克, JR (1996)。Triangle : 設計 2D 品質網格產生器和 Delaunay  
三角儀。Lin, MC 和 Manocha, D. 編輯, 應用計算幾何 : 走向幾何工程, 第 1148 卷計算機科學  
講義, 第 203-222 頁。施普林格出版社。來自第一屆 ACM 應用計算幾何研討會。

斯 H. (2002-2008)。TetGen - 品質受限的四面體網格產生器。維爾斯特拉斯  
研究所, 柏林, <http://www.tetgen.org>。

斯, H. (2015)。TetGen, 基於 Delaunay 的高品質四面體網格產生器。ACM 傳輸  
數學軟體上的操作, 41 (2) : 1-36。

Udphuay, S.、Günther, T.、Everett, M.、Warden, R. 與 Briaud, J.-L.。 (2011)。三-  
密集文化訊號中極端海岸地形的三維電阻率斷層掃描：應用於歷史諾曼第遺址的懸崖穩定性評  
估。地球物理學。J. 國際。, 185 : 201-220。

## 一、安裝

### A.1. 適用於 Windows 使用者的 BERT

BERT 在 Windows 平台上得到了成功應用，但是，對於更大的問題，Linux 上的效能可能會更  
好。儘管建置實際上變得像在 Linux 下一樣簡單，但它可以作為帶有二進位檔案以及所有範例和  
文件的安裝程式輕鬆分發。

由於 BERT 是在命令列上控制的，因此 Windows 使用者需要一個與 Linux bash 相容的命令  
shell。最近，最小系統MSYS（最小系統）被帶到了

新版本 2 包括 64 位元支援和套件管理器 (pacman)。請從以下位置下載安裝程序 <https://msys2.github.io>。

安裝 BERT，假設在 d:\軟體\BERT。那麼在 shell 中工作時必須知道這個路徑，或透過更改環境變數小路在下面系統控制

- 系統 - 環境變量或在 shell 中輸入 \$匯出 PATH=\$PATH:/d/software/BERT<sup>21</sup>

後一個命令也可以透過將其插入啟動時自動調用 \$(HOME)/.bashrc 主目錄中的檔案。另外，變數 Python 路徑 用於告訴 python 在哪裡查找模組 (pygimli 和 pybert - 不包括這些名稱)，因此應將其設定為 d:\軟體\BERT 在系統控製或使用 \$匯出 PYTHONPATH=/d/software/BERT.

在目錄中準備資料和設定文件，透過以下方式在 shell 中轉到該目錄 \$cd /c/data/profile1/試用並使用運行反演伯特 bert.cfg 全部請注意，如果您的 Python 發行版沒有設定路徑，您也需要這樣做，例如在 .bashrc 文件，例如

\$導出 PATH=\$PATH:/c/Software/Anaconda3

在較新的 Anaconda 版本上，可能會丟失一些 dll 文件，因此您還需要

\$導出 PATH=\$PATH:/c/Software/Anaconda3/Library/bin

當然，您也可以在系統控制中設定這些路徑，就像 BERT 的路徑一樣。

## A2。適用於 Linux 用戶的 BERT

安裝 BERT (以及必要的 pyGIMLi) 的最簡單方法是使用 conda，該系統可以方便地安裝二進位版本及其先決條件，並關心路徑變數以查找二進位和 Python 庫。它可以從 Anaconda Python 發行版或其輕量級替代品 miniconda 中使用。無論如何，我們建議創建一個單獨的環境來安裝它。透過這樣做，您可以輕鬆地在 BERT/pyGIMLi 或 Python 的多個版本之間切換，例如：

```
$conda config --新增頻道 gimli --新增頻道 conda-forge $conda 創建-n bert  
$源激活bert $conda安裝pybert 如需更新，只需調用 $conda 更新 -f pygimli  
pybert
```

當需要從原始程式碼建置 pyGIMLi 和 BERT 二進位時，可能會出現一些原因 (系統不相容、與其他模組版本衝突、緊急更新)。在這種情況下，請先按照以下位置提供的安裝程序進行操作：<https://www.pygimli.org/installation.html> 以及給予的說明 <https://www.gitlab.com/resistivity-net/bert>。建置 pyGIMLi 和 bert 後，需要新增二進位檔案 (bert/build/bin)、函式庫 (gimli/build/bin) 和 Python 模組的路徑，例如 (最好在 .bashrc 文件)：\$導出 PATH=\$PATH:\$HOME/src/bert/build/bin

\$匯出 LD 庫路徑=\$LD 庫路徑:\$HOME/src/gimli/build/bin

\$匯出 \$PYTHONPATH:\$HOME/src/gimli/gimli/python:\$HOME/src/bert/bert/python

---

<sup>21</sup>請注意，檔案名稱與 Windows 不同，即 /C 代替 C：並用斜線代替反斜線。

## B、版本歷史

### BERT 2.4 (pyGIMLi 1.3)

2.4.1 2023年1月19日 (第 1.3.1 頁) 切換到 Python 3.9，錯誤修復  
2.4.0 21.08.2022 1.3.0 頁) 改進 TDIP/FDIP

### BERT 2.3 (pyGIMLi 1.1/1.2，開發僅限於 IP 類別)

2.3.4 2022年2月18日 (第 1.2.6 頁) 避免檔案中的 P2 網格細化，TDIP/HIRIP (第 1.2.3 頁)  
2.3.3 2021年12月12日 pyGIMLi 更新，TDIP/HIRIP，修復網格錯誤 (第 1.2.2 頁) FDIP/  
2.3.2 2021 年 8 月 1 日 TDIP 更新  
2.3.1 2020年11月1日 (第 1.1.1 頁) 更新 FDIP/TDIP (Martin 等人，2020) ，Python 3.8 (第 1.1.0 頁)  
2.3.0 2020年6月6日 升級到新的 pyGIMLi 風格，錯誤修復

### BERT 2.2 (帶有 ERT 核心和 ERTManager 的 pyGIMLi 1.0)

2.2.12 02.11.2019 (pg 1.0.12.1) 主要是 pyGIMLi 修復，最後 pg 1.0 (pg 1.0.12) 改進了 TDIPdata 類  
2019年11月2日 10月16日  
2.2.10 01.04.2019 (第 1.0.11 頁) 改進 TDIPdata 類別 (第 1.0.10 頁)  
2.2.9 2019年1月4日 第一個 Python 3.7 版本 (第 1.0.9 頁) 新的 TDIP 類別、次要內容和修復 (第 1.0.7 頁) 轉換和顏色  
2.2.8 2018年10月30日  
2.2.7 2018年9月24日  
2.2.6 2018年2月13日 (pg 1.0.6 嘉年華版) : udf&res2dinv topo (pg 1.0.5) 相當未知的版本  
2.2.5 2018年1月12日  
2.2.4 2017年11月17日 (pg 1.0.4 GELMON 版) 次要內容 (pg 1.0.1) 主要教程更新和錯誤修復 (pg 1.0.0) 主要是 pyGIMLi 更改  
2.2.3 2017年9月15日  
2.2.2 2017年8月21日  
2.2.1 2017年7月21日 穩定版本適用於 Py 3.4、3.5 和 3.6 第一個版本，沒有 C++ 核心庫  
2.2.0 2017年6月16日 (全部在 GIMLi 中)

### BERT 2.1 (libbert 結束，增加了 Python 的使用)

2.1.4 2017年6月2日 帶有 C++ BERT 核心庫的最新版本第二萊比錫研討會版本  
2.1.3 2017年3月3日  
2.1.2 2017年1月19日 第一個 Python 3.5 建置 (從此處預設)  
2.1.1 2016年9月28日 穩定的 bug 修復後研討會版本 (僅限 Python 3.4) BERT Workshop Leipzig 版，新教程  
2.1.0 2016年9月21日

### BERT 2.0.x (原為 2.0RCx)

新的核心庫libbert，建立Python的可視化

2.0.19 17.08.2016 Günther&Martin (2016) 定義 FDIP 的論文不使用此版本 (3D FEM 中的錯誤)  
2.0.18 28.06.2016  
2.0.17 24.05.2016  
2.0.16 14.04.2016 中間未測試版本復活節彩蛋 (從 2RCx 重新命名為 2.0.x) GelMon 版本  
2.0.15 24.03.2016  
2.0.14 26.11.2015

2.0.13	08.10.2015	(第一個工作電阻率Python類) 從這裡只有64位元與WinPython3.4第一個64位元版本與
2.0.12	25.09.2015	Python3.4-64位元只有32位元版本與
2.0.11	14.09.2015	Python3.4-32位元還有Win64與自己的Python64位元
2.0.10	09.07.2015	
2.0.9	2015年5月18日	
2.0.8	2015年3月21日	第一個 FDIP (SIPData) 類
2.0.7	2014年12月16日	
2.0.6	2014年10月17日	
2.0.5	2014年6月3日	
2.0.4	2013年12月2日	
2.0.3	2013年8月8日	還有Python32位元的Win64
2.0.2	2013年4月17日	

## BERT 2 測試版

2.0b23	2013年8月6日	
2.0b22	2013年4月5日	還有 Win64 23.01.2013
2.0b21	2012年11月22日	還有Win64
2.0b19/20	2012年5月29日/30日	
2.0b18	2012年2月20日	
2.0b17	2012年2月13日	還有Win64
2.0b16	2012年1月10日	僅Win64
2.0b15	2011年10月20日	還有 Win64 15.11.2011
2.0b14	2011年8月17日	
2.0b13	2011年7月6日	
2.0b12	2011年6月8日	
2.0b11	2011年4月13日	
2.0b10	2011年2月7日	
2.0b8	2010年11月2日	
2.0b4-6	2010年8月5日-13日	
2.0b3	2010年7月28日	
2.0b1	2010年4月21日	

## BERT 1 (2004-2008-2013)

1.3.2	2013年4月5日	最終的Win 32位元版本也是（最
1.3.1	2013年11月22日	終）Win64版本沒有更多新功能
1.3.0	2012年5月30日	
1.2.5	2011年8月30日	也是第一個Win64版本Linux
1.2.4	2011年5月10日	
1.2.3	2011年4月21日	Linux
1.2.2	2011年1月11日	
1.2.1	2010年12月3日	
1.2.0	2010年10月28日	Linux
1.1.0	2010年6月25日	(BERT2 beta 版本也開始)
1.1.1	2010年10月27日	

1.0.3 2009 年 11 月 17 日

2006-2009 之前版本

## 版本1到版本2的主要變化

### 主要改進

- 更靈活地處理區域（不同的行為、限制和轉換）、區域間（解）耦合，因此包含先驗知識的更多可能性
- 更多的約束類型（零階、一階和二階以及它們的組合）和更穩定的各向異性正則化方式
- 更多元素類型，例如矩形、六面體或棱柱網格或不同元素的組合
- 使用不同的電極類型：節點、表面（CEM 電極）或無節點點（或不同類型的混合）
- 使用不同的 MIN/MAX 關鍵字輕鬆過濾數據
- 電阻率下/上邊界與資料無關（在 BERT1 中，它們也適用於  $\rho_A$ ）
- 改進的延時策略，例如完整參考模型或差異反演
- 使用 pyGIMLi（GIMLi 的 Python 締定）改進了直接視覺化
- 提高速度（得益於 BERTTHREADS 的分散式運算）和收斂性，更輕鬆的建構過程
- 由於更嚴格的 IP 方案，改進了相轉化
- ERT 管理器可實現基於腳本的輕鬆處理

### 命令的變化

primPot/插值⇒鍋 -以前，一次電位是分兩步驟計算和插值的。現在，除非 KEEPPRIMPOT=1 另有說明，否則位勢將動態插值到前向網格上。

正確的⇒篩選 -由於命令 Correct（在 calc 之前調用）會導致混亂，命令過濾器現在將原始資料轉換為所需的反演輸入，包括視電阻率、幾何因子和誤差。另外，過濾是使用 MIN/MAX 關鍵字完成的。對於延時問題，過濾器將對各個時間步長的資料數組進行均質化。在未來的版本中，如果資料允許，可以呼叫自動正常倒數分析。

計算/calcSensM⇒計算-不再需要 calcSensM 指令並完全被 calc 取代。預設重新計算雅可比行列式。

## 新指令

唯感-僅計算雅可比矩陣（用於靈敏度或分辨率研究）版本 -列印出版本

顯示、顯示網格、顯示資料、顯示錯誤、顯示擬合、mkpdf -不同東西的圖（在二維幾何的情況下）

mkpdf、mkmeshpdf -自動產生 pdf

## C. 文件和程序

創建的文件及其含義

文件類型：

- \*。聚 三角形 (2d) 或 tetgen (3d) PLC 格式 二進位網格 (房屋) 格式
- \*。生物管理系統 VTK 可視化工具包網格或多邊形格式 (paraview)
- \*。網 MEdit網格格式
- \*。向量 所有電極浮點電位矩陣的 ascii 向量
- \*。收集 量

目錄及其內容：

網	網格輸入 (mesh.poly) 和網格 (meshPara、meshSec、meshPrim、meshParaDoma PrimaryPot/pot(.s.bmat) 主要電位（已在輔助網格上） sens.bmat
結果*	敏感度列 (smatrix.*.) 或行 smatrixCol.* 儲存包含最重要檔案的結果目錄

專案或結果目錄中的檔案：

命令歷史記錄	bert 反轉日誌檔案執行的命令的歷史記錄
伯特日誌	
*。數據	具有視覺電阻率和誤差的過濾資料檔案最終結果為 vtk
dcinv.結果.vtk	文件
電阻率.向量	最終（預計）電阻率分佈 最終模型正向
響應向量	響應
覆蓋率.向量	覆蓋範圍（絕對靈敏度總和）各個迭代
模型*.向量	的模型向量 網格輸入 PLC
網格/網格.聚	
網格/網格ParaDomain	純參數網格（用於視覺化）
如果是調試模式 (SAVEALOT=1) 回應	
*.vector_	每次迭代前向模型的響應向量
fop-模型*.vtk	
linesearchPhi(D).向量	線性搜尋 向量圖
約束.矩陣	約束矩陣（具有 I、J、值的稀疏矩陣）

用於 BERT 的程式調用

反演和參數化：

bert - 主要可執行檔基於 cfg 檔案運行

bertNew2D/2DTopo/2DCirc - 2d 情況下的CFG 檔案產生器（平面、拓撲、圓形）  
 bertNew3D/3DTopo/3DCyl - 3d 情況下的CFG 檔案產生器（平面、拓撲、坦克）  
 dcinv - 實際反演程式（參見dcinv -h）  
 dcmod - 合成（例如初級）勢的正向建模（參見 dcmod -h） dcredit - 使用數值幾何因子過濾原始數據  
 網格建立和變更：

超深度	透過一維靈敏度估計適當的模型深度從 dat 檔案建立參數網格
創建ParaMesh	
建立輔助網格	根據參數建立輔助網格 從 xyz 點清單建立 3D
創建曲面	表面網格 透過周圍的方塊關閉 3D 表面網格
近曲面	
準備網格細化 (prepareMeshRefinement)	插入細化點
dctriangle	三角調用
網格轉換	在各種匯入格式之間轉換網格

Poly 工具 - 建立 PLC 物件：

polyCreateWorld	創造一個具有 2 個表面和內部邊界的 world
	在電極周圍創建世界（2個區域）=節點在電極周圍創建
聚自由世界	世界（2個區域）（無節點）在原點周圍創建（單位）
多創建立方體	立方體
聚翻譯	翻譯PLC
多尺度	規模PLC
多旋轉	旋轉PLC
多合併	將 2 個 PLC 合併為一個新的 PLC 添加點（例如電極）到 PLC 添加電極輪廓
聚添加VIP	
多添加輪廓	
聚精煉VIPS	透過局部細化來細化點 將 PLC 轉換為
聚集轉換	VTK 或 STL 格式 各種 PLC 功能
多腳本.sh	

## D. 選項及其預設值的完整列表

```

# 全球的      設定
#
# 版本=2.0.19
資料檔案=不存在。資料維度=3          # 數據      檔案名稱      (必需的)
# 問題的維度 (2d 為 2, 3d 為 3)
地形=0                                # 定義地形是否存在 (0 或 1)
拓撲點=                               # 有 additional coordinates for 3d 地形 (xyz) 的文件
拓撲聚=                               # 為 3d 地形添加附加多邊形的文件 (九月通過空白行)
氣缸=                                  # 幾何是封閉的
時間步=                               # 名稱為 additionaldatafilesfor timelapse inversion 的文件
參數幾何=                             # 腳本 (python 或 bash) 建立幾何輸入 (mesh/mesh.poly) 或文件
參數網格=                             # 網格或建立反轉網格的指令 (mesh/mesh.bms)
介面=                                  # 文件，其中包含描述 (2d) 介面的 x/z 列 (以空白行分隔)
ELECTRODENODES=1 # 電極以節點表示 (尚未解釋!) USEBERT1=0
# 使用先前產生的 BERT1 網格、電位和靈敏度 (BERT2)
KEEPPRIMPO=0                          # 將主勢保留在主網格上，其他 rwi 動態插值 (BERT2)
調試=0                                 # 列印更詳細的格式並儲存重要的臨時文件
儲存批次=0                            # 保存大量文件用於調試目的

#
# 數據      設定
#
OVERRIDEERROR=0 # 使用 INPUTERRLEVEL 和 INPUTERRVOLTAGE 覆蓋給定的錯誤 INPUTERRLEVEL=3 # 如果沒有給出錯誤，則輸入錯誤等級 (以百分比表示)
(dc 編輯它 -p 但仍然 -e)

```

```

# 輸入電壓=100e-6          # 輸入電壓幾何值      錯誤 (五)    如果沒有給出錯誤
KMIN=-9e99                 # 最低限度        因素      (BERT2)
K最大值=9e99                # 最大限度        幾何的    因素      (BERT2)
最小剩餘時間=0              # 最低限度        明顯的    電阻率 resistivity y (BERT2)
RMAX=9e99                   # 最大限度        明顯的    (BERT2)
IPMIN=-9e99                 # 最小 IP 值 (BERT2)
IP最大值=9e99                # 最大 IP 值 (BERT2)
最大錯誤率=9e99              # 最大誤差估計 (BERT2)
減持=0                      # 減少刪除資料的步驟以保留資料結構 (BERT2)
吸光值=0                     # 強制所有視電阻率為正

#
# 我的版本 設定
#
約束=1                      # 約束          類型 (1/2- 第一/第二長度     命令, 0-分鐘。長度, 10/20 混合)
拉姆達=20                    # 正規化
Z權重=1                      # 純ly 的權重    垂直梯度 (BERT2)
蘭姆達奧普特=0              # 透過使用 ing l-curve 來優化 lambda
CH1選項=0                    # 優化 lambda 使得 CH1=1 (BERT2)
區域文件=                    # 每個檔案都有對應的檔案反轉選項   地區 (BERT2)
塊模型型=0                  # 迭代 (L1) 重新加權模型粗略度
穩健資料=0                  # 迭代 (L1) 重新加權資料失配
下限=0.0                     # 降低 電阻率 resistivity y 邊界 (logarithmic barrier) (0.0 =
上限=0.0                     # 上層 邊界 deactivate)
重新計算雅可比=1 #           # 重新計算 雅各安 每個 迭代步驟 (預設 在 BERT2 中)
最大=20                      # 最大幅度 數位 代步驟
啟動模型=                    # 起始模型, 可以是一個值 (同類模型), 也可以是一個包含向量的名稱
RESOLUTIONFILE= # 帶有索引/位置的檔案來計算解析度 kernelsfor (BERT2) LAMBDADECREASE=1 # 在每次迭代中按因子減少lambda (例如0.8)
LOCALREGULARIZATION=0 # 局部正則化 (c 僅在應變模型更新執行個體型號廣告)

# SINGVALUE=1               # 電極電位值, 靈敏度 (BERT1)
靈敏度下降=0                # 降低雅可比值的公差 (sparsestorage)
SENSMATMAXMEM=2000 # 用於預先分配稀疏雅可比行列式的可用記憶體 (以 MB 為單位)
#
# 時間步長設定
#
拉姆達延時拍攝=          # stepa 速率 lambda 值對於時間步 (BERT2) 約束
TIME LAPSECONSTRAINT=1 #       類型 (請參閱約束) 區域文件
TIME LAPSEREGIONFILE= #       例如我在檔案中 (看)
TIME LAPSESTEPMODEL=0 # 使用前面的模型作為第一個 FASTTIME LAPSE=0 的參考
                           # 沒有 jacobian 重新計算 (trast 上的 和 為了 時間步長 (僅相同 射線) (BERT2)
比率步長=0                  # 很簡單 low-c 和 快速法
TIME LAPSEREMOVEMISFIT=0 # LaBreque 等人之後的差異。 (1996) (BERT2)

#
# 網 設定
#
Paramax單元尺寸=0          # 最大細胞體積 (m3) (DIMENSION=3);面積 (m2) (DIMENSION=2) for para
副邊長度=0                  # 透過正三角形/四面體的體積計算 PARAMAXCELLSIZE (更容易計算)
PRIMMAXCELLSIZE=0            # 最大細胞體積 (m3) (DIMENSION=3); prim 的面積 (m2) (DIMENSION=2)
參數深度=0                  # 參數域的最大深度, 以公尺為單位 (0 = 透過 1d cov 自動估計影像)
準邊界=5                     # 參數域中電極周圍的邊界 (百分比)
樣條邊界=0                  # 樣條圓邊界取代線性內插
EQUIDIST邊界=1              # equidistantrefined 電極之間的空間 (2d) (BERT2 中的預設值)
邊界=500                     # 參數域周圍邊界區域的大小
MESHGEN=tetgen               # 指令或位置 3d 網格產生器
耐受性=1e-12                 # tetgen 耐受限
TETGENPRESERVEBOUNDARY=0 # tetgen 應抑制邊界界面或線段的分割 LOOP TETGEN=0
                           # 由於重複 tetgen call, 使用 tetgen 本身可以稍微提高網格質量
參數=0.0                     # 參數網格的細化 (值>0.5 將被強制為 0.5)
網格=0                       # 生成規則 (四面體或六面體) 網格 (僅 TOPOGRAPHY=0、BERT2)
層數=10                      # forgrid 使用的層數 (僅適用於 GRID=1)
帕拉茲=1                     # 使用 forgrid 的分層知識 (僅適用於 GRID=1)
PRIMDX=0.1                   # 電極細化 (原始網格) (註: 相對於 2d, 3d 中的絕對值)
PRIMDX R=0.0                 # 電極細化 (初始網格) 中心方向驅動 (覆蓋 PRIMDX)
PARA2D 質量=33.0             # 參數網格 (從 25 (非常差) 到 35 (好))
PRIM2D 質量=33.4             # 主要網格 (從 25 (非常差) 到 35 (好))
PARA3D 質量=1.5              # 參數網格 (從 1.11 (好) 到 2 (壞))
PRIM3D 質量=1.2              # 主網格 (從 1.11 (好) 到 2 (壞))
表面質量=30                  # 質量 of graph icalsurfacegrid (從 20 (差) 到 35 (好))
表面最大尺寸=0.0             # 最大 三角形 區域 參數表面的光學表面
表面平滑=0                   # 提升 品質 的 網格
ICDROPTOL=0.0                # 如果數字 節點數 200k 跌落耐受力
                               # 請勿使用二次位勢方法 (即對於 CEM 網格)
總潛力=0                     # 省略 全球的 正向計算的細化 (ffine 參數)
                               # 在前向網格中使用二次形狀函數 (BERT2)
SECMESHREFINE=0              # 在前向網格中使用二次形狀函數 (BERT2)
SECP2MESH=0                  # 使用主要 p2 網格 (BERT2 中預設)
PRIMP2MESH=1

#
# 目錄 設定
#
MESHBASENAME=格             # 基本名稱 為了 網 文件
DIRMESHES=網格               # 目錄 姓名 為了 網 文件
DIRPOT=primaryPot            # 主電位和插值電位的目錄名稱
DIRPRIMPOT=電位 DIRINTERPOLPOT=內插 ubd 目錄名稱          # 为了 初級電位 (過時的)
DIRSENS=sensM                 # ubd 目錄名稱 姓名 對於插值 潛力 (已過時)
                               # 目錄 靈敏度 矩陣
舊的PRIMMESSHTYLE=0          # 建立主網格的替代方法 (僅供內部使用)

```

網  
控制)  
網  
憤怒)

LS

網格  
網格  
網格

網格)

```

# 數據 濾 選項
#
KMIN=-9e99 # 最低限度 幾何的 因素
K最大值=9e99 # 最大限度 幾何的 因素
最小剩餘時間=0 # 最低限度 明顯的 電阻率電阻率相
RMAX=9e99 # 最大限度 明顯的 電阻率電阻率相
IPMIN=-9e99 # 最低限度 明顯的 電阻率電阻率相
IP最大值=9e99 # 最大限度 明顯的 電阻率電阻率相
最大錯誤率=9e99 # 最大限度 誤差估計” 階段

#
# 繪圖 設定 (2d 地塊 和 皮特里帕奇)
#
PYTHON=蟒蛇 # 命令 或者 位置 Python 執行檔
PYTRIPATCH=pytri 補丁 # pytri patch 的指令與位置 (二維網格繪圖)
USECOVERAGE=1 # 使用 Alpha 著色的覆蓋範圍
顯示電極=1 # 將電沉積顯示為黑點
INTERPERC=3 # 使用 3% 百分位數 用於使用 pytri patch 進行模型顯示
最小c= # 最低限度 顏色 規模
最大c= # 最低限度 顏色 規模

```

永久有效的選項可以保存在文件中。伯特克在主目錄（特定於使用者）或所在資料夾中伯特位於（對於所有使用者）。這對於SENS-MATMAXMEM (建議使用可用 RAM 的 50-80% (以 MB 為單位) 、並行使用的 CPU 數量 (BERTTHREADS) 或 Python 執行檔 (PYTHON)。

## E. 使用者故事/操作方法

有時，使用者有非常特定的任務要求，主要是要合併的幾何形狀。我們通常透過一些 Python 或 bash 腳本提供解決方案，這些解決方案超出了通常的文件範圍，相應的輸入文件位於文檔/操作方法。其中一些實際上是解決方法，將透過新關鍵字整合到 BERT 2 的後續版本中，但仍可向使用者展示如何建立一般解決方案。其他的使用者故事需要相當複雜的方法，而這些方法永遠不會直接整合到 BERT 中。我們希望現有的故事告訴您如何透過組合現有的故事來創建解決方案。如果您有一個已經解決的有趣案例，或者無法輕易解決但許多用戶可能感興趣的案例，請隨時與我們聯繫以整合它。

請注意，這些解決方案主要涉及創建適當的網格，並以 shell 腳本、Python 腳本或兩者的組合形式編寫。因此，需要具備程式設計的基本知識才能理解它。在這裡，我們逐一介紹所使用的主要元件，有關完整腳本，請參閱指定目錄中的檔案。通常我們會跳過一些常規部分，例如庫的導入，通常

```

導入 pygiml 我作為控球後衛
導入numpy作為np
進口matplotlib.pyplot 作為 plt

```

有關可用函數的完整列表，請參閱 pyGIMLi 的文檔 (<http://www.pygimli.org>) ，Numpy/SciPy (<http://www.scipy.org>) 或 Matplotlib。要使用的主要資料類型是 pg.Vector，它與 np.array 相容。

### E.1。如何利用外部地形資訊進行二維反演

（使用 INPUTFILE 或使用 python createParaMesh 包含到 BERT 中）檔案：文檔/操作方法/2d 拓撲文件 -

任務：使用最後帶有地形的 unidata 格式檔案進行反演 問題：BERT 1  
網格工具尚不支援此功能 解決方案：必須將資料拆分並轉換為標準格式。TODO：產生僅 pygimli 的解決方案

之前，通常的建議是使用 DC2dInvRes 及其選項 Save-Ohm-File，它可以進行磁帶校正。現在我們在shell腳本中解決這個問題並直接呼叫Python。首先，我們提取電極和資料的數量，並寫入檔案的第一部分（電極+資料），以便 pygimli 讀取。此外，我們讀取地形列表並將其匯出到另一個文件：

```
infile =' 設定檔 -topo 。 dat
'datfile=' 設定檔 。達特'
outfile=' 個個人資料 .歐姆'拓樸
文件='拓樸 。 xz ' 讀取行<$
infile ne l=$ {線% # *}

行 ='head -n $ [ ne l + 3 ]           $內菲萊/尾巴      - n  1'
ndata=$ {線% # *}
頭-n $ [ ne l + $ndata + 4 ]           $內菲萊>$ 資料檔案
line ='head -n $ [ ne l + $ndata + 5 ] ntopo=$ {  $內菲萊/尾部-n 1'
線% # *}迴音 $nel $ndata $ntopo

尾 -n $ntopo $ infile>$ 拓樸文件
```

我們現在切換到 python 並讀取檔案並提取電極位置。我們計算沿輪廓的膠帶距離，並使用地形將其插值到電極位置。

數據=頁。資料容器 ('\$資料檔') [列印](#) ( 數據 )

```
x , z = np 。 loadtxt (' $t opo 檔案 ' , unpack=True )
tt = np 。 hstack (( 0 . , np . cumsum(np . sqrt(np . diff(x))**2+np 。 差異 (z) **2))) xe =
[ el 。 位置 () [ 0 ]       為了 艾爾在 數據 。 電極 ())
z = np 。 int e rp (xe ,   X , )
```

然而，通常電極位置具有磁帶距離，而不是真實的 x 座標。因此，我們必須考慮它們的高度，以使磁帶距離保持恆定。之後我們設定

```
x = xe [0]+np 。 hstack (( 0 , np . cumsum(np . sqrt(np . diff (xe))**2-np 。 差異 (z) **2))) de =
np 。 sqrt(np . diff(x)**2+np 。 差異 (z) **2) 為了 我在 範圍 ( 資料 . sensorCount () ) ：
    數據 。 setSensorPosition (i , pg . RVector3 (x [i] , 0 . 0 , z [i]))
```

數據 。儲存 ('\$outfile' , 'abmnuiR' , 'xz')

就是這樣！使用已建立的文件，我們可以進行非常經典的反演。

## E.2。給定 3D 座標的 2D 反演

(自動或透過 DIMENSION=2.5 包含在標準 bert 中？) 文件：doc/Howto/3d  
座標：mkcoord.py 、\_mk3dvtk.py

任務：使用 3d 座標對輪廓進行 2d 反演並建立適當的 vtk 檔案問題：透過不精確的位置擬合適當的線並將結果轉換回 3d

## 1. 讀取資料檔並透過電極擬合輪廓線

輸入可以是：

- 1a) 每個電極具有 3d 座標 (x,y,z) 的 2d 資料檔案 1b) 2d 平坦地  
球資料檔案和附加 GPS 位置文件

我認為後者是更一般的情況，其中一些（不一定是全部）電極的 GPS 位置位於單獨的檔案 gps.txt（輪廓距離、高度、北距和東距）。

```
tt , hh , yy , xx = np . loadtxt('gps.txt',unpack=True)
# 磁帶距離、高度、北向和東向
```

讀取並繪製座標後，我們發現由於 GPS 不準確而導致的典型誤差高達幾米，我們希望消除這種誤差。因此，我們透過使用 pygimli 函數 Harmfit 中實現的調和函數來擬合平滑曲線。根據所得的 x 和 y，我們計算沿著磁帶的距離 t，用於反演。

```
x = 傷害它 (xx , TT , nCo 效率 =10)[0] nCo 效率
y = 傷害它 (yy , TT , =10)[0] nCo 效率 =15)[0]
z = 危害 (hh , TT ,
t = np . hstack (( 0 . , np . cumsum(np . sqrt(np . diff(x))**2+np . 差異 (y) **2 )) ) # 2d di姿態
```

係數 nCoefficients 的數量定義了擬合曲線的複雜性，並且應該足夠大以描述過程，但不能太大以避免小規模振盪。擁有使用不同數字的經驗，可以輕鬆為任何數據提供良好的值。

```
plt . 圖 ( tt , hh , 'bx-' , tt , z , 'r - ')
plt . 圖 (xx , yy , 'bx-' , x , y , 'b - ')
```

圖 28 顯示如何近似座標。

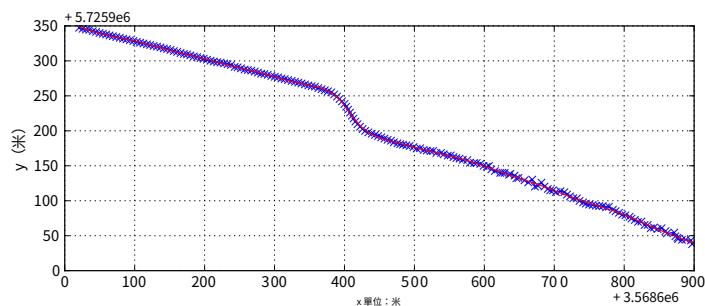
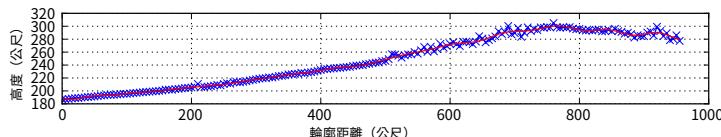


Figure 28: 米 確定的正面因素 昂斯（藍色）大約  
在 thky 平面  
米已更新的設定檔 e（紅色）阿隆 G 磁帶（t 操作）和

## 2a. 將職位投影到個人資料上

現在，我們載入資料容器並將（磁帶）座標 xe 插值到 2d 位置 t 和 z 上。然後，我們遍歷所有電極並將其 3d 位置更改變為 2d 位置，然後將資料儲存到 2d 檔案。

```
數據=頁。DataContainer('profile.dat') xe = [ pos[0]為了位置在數  
據。感測器位置 s () ] x2d = np 。int e rp ( xe ,  
TT , )  
z2d = np . int e rp ( xe , TT , )  
為了我在範圍 (資料.sensorCount () ) :  
    數據 。set tSensorPos iti on ( i , pg . RVector3 ( x2d [ i ] , 0.0 , z2d [ i ] ))  
data . save ( ' profile 2 d . ohm' , ' abmnu ' , ' xz ' )
```

此外，我們建立一個從 2d 到 3d 座標的映射以進行結果投影。

```
POS = np . vstack((x2d,x,y))np.  
savetxt(' pos .map' , POS.T)
```

## 2b. 計算 2d 和 3d 幾何因子併校正 u 或 R

我們實際上可以考慮到輪廓並不是嚴格筆直的。 ( $G_{2d} > G_{3d} \Rightarrow \rho_{2d} < \rho_{3d}$  對於常數  $\rho_A \Rightarrow$  正確的  $\rho_{2d} = \rho_{3d} * G_{3d} / G_{2d}$ ) 建立標準網格、細化並使用它們運行 dcmod 。

還沒做完

## 3. 進行二維反演並改變vtk檔案中的位置

使用標準工具對二維資料集進行二維反演：

```
bertNew2DTopo 輪廓 2 d 。歐姆>伯特. cfg  
# 添加一些其他有用的選項作為值邊界或任何 bert bert 。 cfgall
```

## 4. 將結果投影回 3d 域

因此，我們有一個帶有 2d 座標的 vtk 文件，需要將其轉換為 3d。我們讀取網格並將節點位置保存在變數 tn 和 zn 中：

```
網格=pg. Mesh('dcinv.result.vtk') tn = [n.位置 () [ 0 ]  
為了n在網。節點 () ] zn = [ n 。位置 () [ 1 ]為了n在  
網。節點 () ]
```

然後我們加載之前保存的地圖並將 tn 插值到真實的 x 和 y :

```
tt 、 xx 、 yy = np 。loadtxt ( ' pos .map' , unpack=True ) xn =  
np .積分 ( tn , tt , xx )  
yn = np .積分 ( tn , tt , 年 )
```

最後我們設定各個節點的位置並匯出檔案。

```
為了在在列舉中 (網格。節點 () ) :  
    名詞setPos(pg.RVector3(xn[i],yn[i],zn[i]))  
  
網 。 exportVTK('結果 3d.vtk')
```

然後結果顯示在圖 29 中。

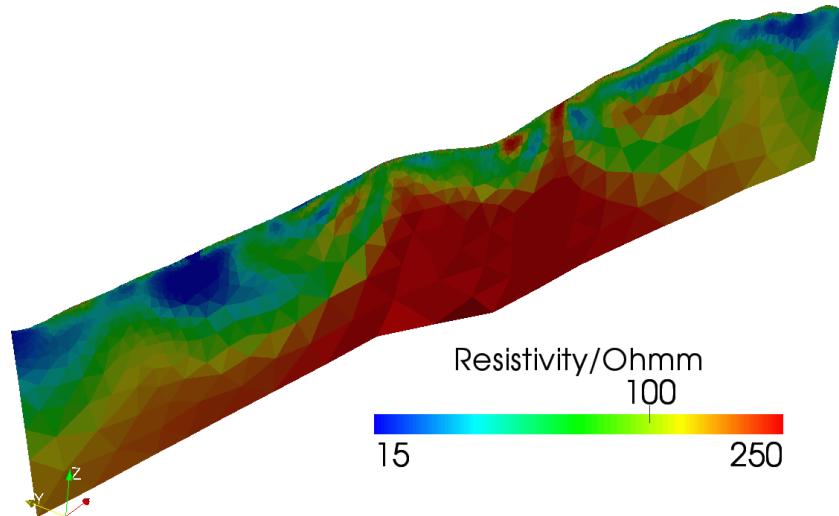


圖 29：以 3D 形式顯示的反演結果。

### E.3。使用者定義的規則二維網格

（使用任何 DX 關鍵字或 polyRectWorld 之類的腳本包含到 BERT 中）文檔/howto/  
常規 mesh2d

任務：規則矩形網格上的反演

問題：必要的邊界應由三角形組成，以保持節點數量較小解決方案：為參數建立規則的二維網格，並添加三角形邊界框，以便由於邊界條件而精確求解。

如下一個範例所示，建立了一個網格，並將其邊界整合到一個更大的盒子中，該盒子與常規網格合併並組合在一起。在 2D 中，這個任務非常簡單，並在 pygimli 函數appendTriangleBoundary 中實作。以下 python 腳本應該是自可讀的：

```
從 侏儒我。觀眾進口顯示網格
從 侏儒我。網格工具進口附加三角形邊界

數據=頁。DataContainer('gallery.dat') xmin = data.
感測器位置 (0) [ 0 ]
xmax=數據。感測器位置 (資料。感測器計數 () dx = (資料。感測器 - 1 )[0]
位置 (1)[ 0 ] - xmin ) / 2 . zmax = 10

NB = 2
x = pg . asvector (np . arange (xmin-dx*nb,xmax+dx*(nb+1),dx)) z = pg .
asvector (np . arange (- np . ceil (zmax / dx) ,1 . ) *dx)

網格=pg.網目 (2) 網目。 # 新的二維網格
建立二維網格 ( , , )
為了C在網。細胞 () :
```

```
C 。setMarker (2) # 將所有標記設為 2
```

```
mesh2 =appendTriangleBoundary(mesh, 50., 50.) # 主要的事情！顯示網格（網格2）
```

網格2。儲存（'網格.bms'）

我們現在透過指定包括網格PARAMESH=mesh.bms。圖 30 顯示了創建的混合單元網格和反演結果。

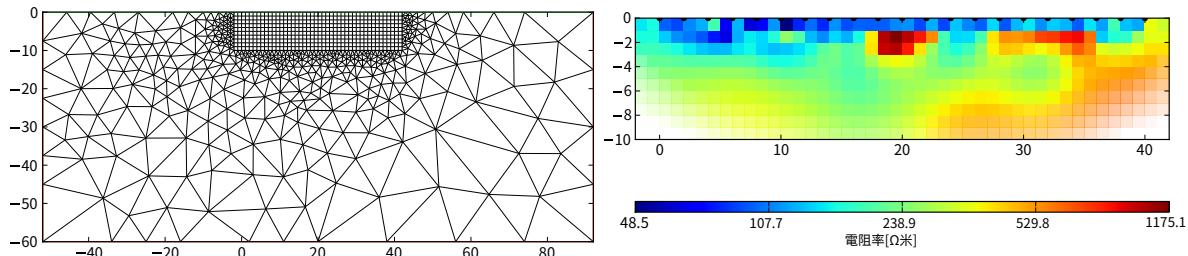


圖 30：建立混合網格（左）和使用常規網格的反演結果

如果 (i) 主要分層應該更好地成像或 (ii) 如果需要具有恆定尺寸的單元（例如用於跨孔測量），則規則網格可能比非結構化網格有利。

#### E.4。如何在 2D 域中反演 1d 電阻率

（整合常規二維網格後可選擇）文件/Howto/1d 垂直

任務：從垂直電極鏈反演電阻率資料 問題：Gimli 1D 正向算子不處理地下

源 解決方案：使用 2D (BERT) 正向算子和特殊的準一維網格

重要的部分是創建一個適合我們需求的網格。

我們選擇了

常規網格，可以使用 pygimli 函數 create2DMesh 輕鬆建立或

建立 3DMesh 並設定標記，以便所有矩形都獲得相同的標記，並且可以稍後使用

e 被視為一個單一區域。我們可以手動分配標記，但上面的 ns 已經包含典型的解決方案：  
功能 markerType (0、1、2 或 12) 命令指定，索引 (x 和 y) 隨軸增加標記。預設情況下，所  
哪個 有儲存格都會有

相同的阿克。我們使用索引 2 來使用常數 y 標記：

```
數據= pG 。DataContainer ('vest-ost.dat')
z1 = 數據 。感測器位置 (0) () [2] # 第一個電極的 z
z2 = 數據 。SensorPo sition (1) () [2] # 第二個電極的 z
zN = 數據 。感測器位置 (data .sensorCount () - 1) [2]
dz = z2 - z1 # 等距與電極距離
NB = 2 # 邊界元素的數量
nx2=20 # ha lf xcell 數量
# 向量上的位置,規則間距
xx = NP 啊 n葛 (- nX2, 恩克斯 * dz
ZZ= NP 啊 n葛 (z1 - n乙 * dz, z氮 + n乙 * dz, dz)
米陋= pG 鎔eA特中號譯D (XX , ZZ, 2)
```

請注意，create3DMesh 也存在類似情況。為了獲得適當的外部空間，我們將此網格插入一個大盒子中，並使用appendTriangleBoundary用三角形填滿中間空間：

```
mesh2 =appendTriangleBoundary(網格, 70., 70., 標記=0, isSubSurface=True) 列印(網格
2) 標記2=網格2。celMarker() showMesh(mesh2, data=marker2, Linear=True) mesh2。儲存('網
格/網格.bms')
```

使用 PARAMESH=mkmesh.py 應用的 python 腳本。此外，我們需要指定一個區域文件，將每個圖層視為恆定的單一區域，將外部視為背景。使用區域間約束定義層之間的平滑度約束：

```
# 無單一反式 lBound uBound
* 1 紀錄 1 300
# 不 背景
0 1
# 區域間
* * 1
```

圖為使用北海博爾庫姆島下方淡水-鹹水界面垂直埋置電極的資料集反演結果。由於資料擬合度約為 1%，我們推測岩性只是精細層理的粗略影像。在最後一層黏土層下面，開始出現電阻率低於  $1\Omega\text{m}$  的鹽水。

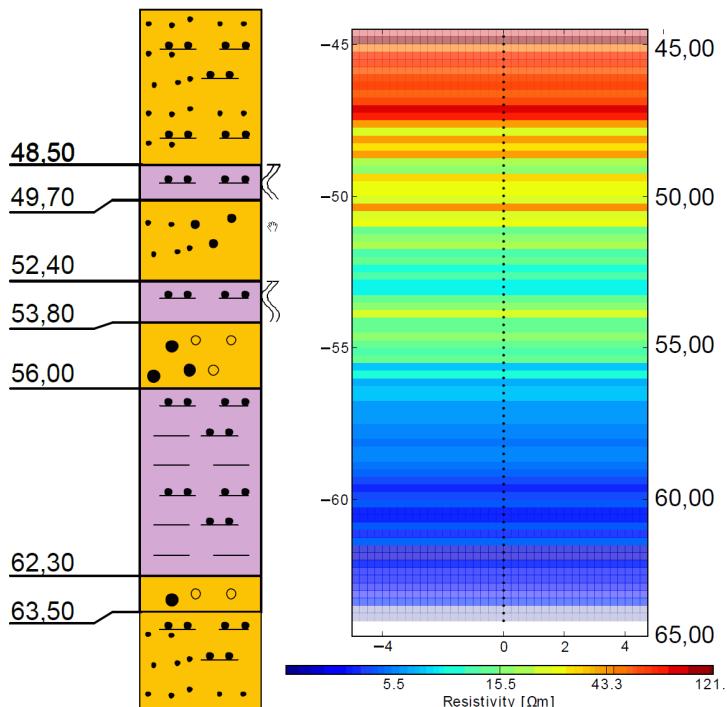


圖 31：反演結果（右）與岩性（左）

## E.5。使用者定義的常規 3D 網格反演

(由於 tetgen 臉部創建而不太穩定) 文檔/  
howto/常規 mesh3d

任務：使用常規3D（類似FD）離散化進行BERT 反演問題：創建四面體（反演部分）和六面體（外盒）的混合網格BERT 2 支援自然地或透過將每個六面體剖成5 個或5 個六面體來支援六面體有限元素計算6 個四面體。雖然第一個變體僅限於正交六面體，但後者可用於變形的規則網格，例如具有表面形貌的網格。對於無界域的反演，需要在反演域周圍設置外框，以確保邊界條件的正確性和準確性。如果網格定期延長，我們會得到非常難看的單元，其數值行為很差，並且在前向計算中會有大量節點。因此我們想用四面體來延長網格。可以使用內建函數在 pygimli 中建立規則網格。首先，建立向量（指定最小/最大/dx）並用於建立網格：

```
x = np .排列 (xmin-dx*nb,xmax+dx*(nb+1) , y = np 。排列  
(ymin-dx*最大y+dx*(nb+1) , z = np 。arange(-np . ceil(dx  
(zmax / dx ) , 1 .)*dx 網格 = pg 。網布 (3)
```

網。建立三維網格 (x 、y 、z)

第二個網格是透過將六面體 H2 細化為四面體而建立的：僅具有一個相鄰單元的外部網格邊界被設定為標記 1 以供稍後識別。

```
網格2 = pg .網格 (3) # 新的 3d 網格 mesh2 .  
createH2Mesh(網格)#精煉 列印mesh , mesh2 #  
顯示 為了乙在網格2。邊界 () : 節點/單元/邊界 數位
```

如果不是 (b .離開 Cell () 且 b.右側單元格 ()) :  
b.設定標記 (1)

```
網。保存 ('para') 網格。匯  
出VTK ('para')
```

最後我們提取網格的外邊界並將其儲存為poly檔：

```
聚 = pg .網布 (3)  
聚. createMeshByBoundaries(mesh2, mesh2.findBoundaryByMarker(1)) 多邊形。匯出  
AsTetgenPolyFile ('paraBoundary')
```

我們現在創建一個世界（帶有 FE 建模邊界標記的大盒子），將其與參數外邊界合併，並在中間添加一個孔標記（在 bash 中）：

```
polyCreateWorld -x100 -y100 -z50 世界 # 製作大盒子  
polyMerge world paraBoundary worldSurface # 可能需要一段時間 polyAddVIP -x 0 -y 0 -z -0.1  
worldSurface # 中間的孔標記
```

PLC 現在具有兩者的面，並且可以以中等質量進行網格劃分

```
tetgen -pazVACq2 worldSurface # 結果將是 worldSurface 。1.* meshconvert -vBDM -it  
-o worldBoundary 世界表面。1 # 轉換
```

現在，我們透過 -2 標記來提取生成的網格的外表面

```

世界邊界 = pg . Mesh('worldBoundary.bms')worldPoly =
pg 。網布 (3)
世界保利。 createMeshByBoundaries ( 世界邊界 ,
    世界邊界。 findBoundaryByMarker (-2, 0)) worldPoly 。
    exportAsTetgenPolyFile('worldBoundary.poly')

```

並獲得沒有內盒的外盒的三角表面網格。因此我們必須合併兩個表面網格

\$多合併世界邊界 paraBoundary allBoundary

請注意，後一個過程可能需要很長時間，因為必須檢查所有面對之間的交集。我們可以透過向 polyMerge 新增選項 -N 來避免這種情況，因為我們知道不存在交叉點。作為副作用，內部框邊界處的節點加倍並且必須被刪除，我們可以透過腳本建立這一點readmypolyfile.py：

```

聚 = pg 。網布 (3)
f=open('allBoundary.poly','r')第1行=f。
閱讀廣告行 ()
nnodes=int(第1行.split()[0])節
點=[]
為了我在範圍 (n個節點) :
    pos=f。重讀廣告行()。分裂()
    p=pg 。RVector3(浮點數(pos[1]),浮點數(pos[2]),浮點數(pos[3]))
    n=Poly。建立帶有檢查的節點(p)
        節點。追加(n)

第2行=f。閱讀廣告行()
nf ace=int(第2行.split()[0])為了
我在範圍 (nf ace) :
    bla=f。閱讀廣告行()
    指數=f。閱讀廣告行()。分裂()
    fa=聚.createTriangleFace(節點[int(ind[1])],節點[int(ind[2])],
                                節點[int(ind[3])],0)
    FA。設定標記(-2)#為了外邊界 (混合邊界條件)

F。關閉()
保利。exportAsTetgenPolyFile('測試.poly')

```

現在可以使用 tetgen -Y 對產生的 test.poly 進行網格劃分（保留面）：\$

tetgen -pazVACY -q2 測試

如果由於相交面而存在錯誤，則可以使用以下方法來識別它們 \$特根-d檢驗

\$meshconvert -V -it -o 錯誤的測試.1

我們觀察到，與 1.4.2 相比，tetgen 1.4.3 做得很好。檢查您使用的版本 (tetgen -h) 並確保您使用 1.4.3。如果 tetgen 網格正確，我們有兩個網格，必須合併它們

```

外層 = pg . Mesh('right gh t.bms')# 外框(world) 內=pg. Mesh('para.bms')#
內框(參數) 列印 (外、內)

```

```

# 將外部儲存格設定為 1，將內部儲存格設定為 2 為了C在外層。
細胞 () : c 。設定標記 (1)
    為了C在內。細胞 () :
        C 。設定標記 (2)

```

### 外層。複製單元格 (c)

```
列印 (外) 外。save('mesh') 外  
層。導出VTK ('網格')
```

我們就這樣吧。只需指定即可輕鬆將生成的 mesh.bms 用於反演 PARAMESH=網格.bms 在 cfg 檔中。整個腳本要么在 doall.sh 中調用各個 python 文件，要么在 makeall.sh 中，其中直接調用 python 命令。

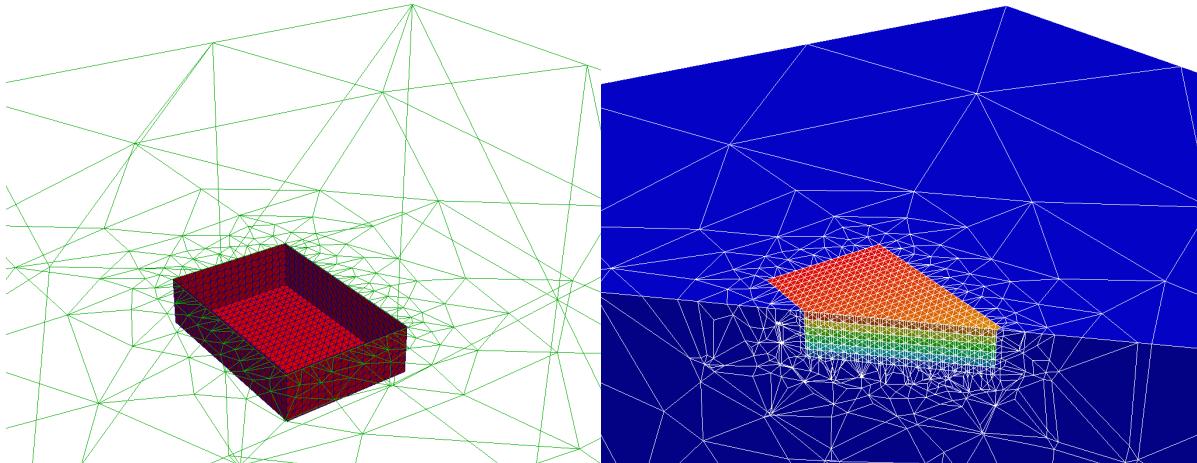


圖 32：內部 3D 網格（紅色面和藍色線）和外部框（綠色線）

請注意，內部的單元格具有標記 2，即在反轉中像往常一樣描述一個區域，但是對於四面體單元格，因為我們這樣劃分了每個六面體。為了避免這種情況，我們不會設定標記並保留 create3DGrid 中隨後增加的數字，以便每個四面體都是一個具有 5 個四面體的區域。透過在區域檔案中指定單一屬性

```
#不 單身的 反式  
* 1 紀錄  
#不 背景  
0 1  
#區域間  
* * 1
```

我們實際上對六面體進行反演，彼此之間具有平滑度約束。除了 REGIONFILE=區域.control 關鍵字，SECMESSREFINE=0 對於避免對已精煉的單元進行精煉很有用。

## E.6。3D 地形中 2D 剖面的 3D 反演與視覺化

文檔/操作方法/2d3dtopo

資料來源：Thomas Günther 和 Dave Tanner (漢諾威 LIAG)

任務：從 3d DEM 產生 2d 和 3d 資料檔案並視覺化結果問題：2d 設定檔是本機 (捲尺) 和 DEM 全域解決方案：讀取設定檔、磁帶展開和 vtk 檔案操作

由於電極有限，真正的 3D 測量（電極網格）通常是令人望而卻步的，因此通常會測量偽 3D，即 2D 輪廓 - 不一定是平行和垂直的。對於地理配準，全站儀可產生數位高程模型以及由 2 個或更多點組成的多邊形的剖面走向。剖面的位置儲存在剖面檔案中（例如由 DC3dInvRes 使用），每個 2d 剖面有一行，如下所示：

```
filen am eof2 dfile x1 y1 x2 y2 . . .
```

首先我們讀入檔名和點：

```
XL、YL、檔案 = []、[]、[] fid  
= open('kilmore2011.pro')行在射  
為了 頻識別：  
文件。追加(行.split()[0])  
XL。追加(np.double(line.split()[1:2])) YL.附加  
(np.double(行.split()[2:2]))  
  
FIID。關閉()
```

接下來，我們讀取 DEM 並進行 Delaunay 三角剖分

```
進口matplotlib .三作為 mtri  
x ` y ` z = np 。 loadtxt('topo.xyz',unpack=True)tri = mtri.三角  
測量(x, y)  
interplin= mtri .線性 Tri 插值器 ( tri , )
```

我們建立一個新的 3D 網格，其中 DEM 點作為節點，三角測量作為單元：

```
為了x1 , y1 , z1郵遞區號 (x, y, z) :  
網。建立節點 (x1 , y1 , z1)
```

```
為了v在三.三角形：  
網。創建 eCell(pg.Triangle(mesh.node(int(v[0])),  
網。節點 (int(v[1])), 網格。節點 ( int (v [ 2 ]) )))
```

網。導出VTK ('拓樸.vtk')

對於每個剖面，我們需要將資料檔案中的磁帶座標捲動到地形上。因此，在使用 GIMLi 表面網格插值例程實現地形之前，我們沿著剖面路線創建一個密集採樣點列表（由 xl/y1 數組給出）並創建它的純節點網格。

```
le = np . hstack (( 0 . , np . cumsum(np . sqrt(np . diff ( xl )**2+np . 差值 ( 基 )**2 )))) xi =  
np 。 int erp(np . linspace(le[0],le[-1],nump), 樂, xl)  
yi = np . int erp (np . linspace ( le [ 0 ] , le [ -1 ] ,nump ) , zi = 樂, )  
interplin ( xi , yi_ )
```

對於所獲得的點列表，我們計算沿地形的磁帶距離，並從中插值所有三個座標。

```
dt = np . sqrt (np . diff ( xi )**2 + np . 差異 (一) **2 + np . 差異 (zi) **2 ) # 差異 = np . hstack (( 0 . , np . 英語  
cumsum( dt ))) # 磁帶距離 沿著輪廓  
數據 = b 。資料容器ERT (資料檔)  
te =[位置。X () 為了位置在數據 。感測器位置 s ()] ze = np 。  
積分 (te , 鈦, 子)  
xe = np . int erp ( te , ye 鈦, xi )  
= np . int erp ( te , 鈦, 義 )
```

對於二維反演，我們需要沿著剖面的方向。磁帶真實座標映射被儲存到稍後使用的某個暫存檔案。

```
x2d = np.hstack((0., np.cumsum(np.sqrt(np.diff(xe))**2+np.square(np.diff(ye))))全部=
np.vstack((te, xe, ye, ze))
NP.savetxt('positions/'+datfile.replace('.dat','.txyz'),ALL.T)
```

首先，我們要使用後一個座標產生一個有效的二維文件，準確地保存讀入的欄位：

為了我在範圍 (資料.sensorCount () )：  
    數據.setSensorPosition(i,pg.RVector3(x2d[i],0.0,zn[i]))

filename2d = '2dfiles / ' + datfile.替換('.dat','2d.dat')數據。儲存 (檔案名稱2d，資料.inputFormatString () )

接下來，我們將座標設定為真實位置，然後將資料新增到先前建立的空資料容器中。

為了我在範圍 (資料.sensorCount () )：  
    數據.setSensorPosition(i,pg.RVector3(xe[i],ye[i],ze[i]))

數據 3d。新增 (數據)

我們現在正在對各個 2d 檔案和 3d 檔案進行反轉。雖然 3d 檔案已經具有正確的 3d 座標，但我們必須對生成的 vtk 檔案進行反向轉換。假設我們的結果命名方式與二維資料檔相同，但副檔名為 vtk。我們讀入之前建立的vtk檔案和map檔案：

```
posfile = '位置/' + 資料檔.替換 ('.dat', '.txyz') t, x, y, z, x2d = np.loadtxt
(posfile, unpack=True)

vtk 檔案 = 資料檔.替換 ('.dat', '.vtk') mesh = pg.網
格 (vtk 檔)

xm = pg.x(網格.位置 ()) # 2d x zn = pg.x(網
格.位置 ()) # z
```

然後我們從 2d 座標插值到 3d 座標。我們還需要進行推斷，而不是 numpy.interp，因為網格通常超出電極。為此，我們編寫了一個簡單的插值-外推函數：

定義 myextrap (x, xp, )：  
    ” ” ” 麻木。int e 功能 和 線性 外插 ” ” ”  
    rp y = np 積分 (x, 經驗值)  
    y = np。其中 (x < xp[0] ,      yp[0]+(xp[1]-xp[0])\*(yp[0]-yp[1])/(xp[0]-xp[1]), 是)  
    y=np。其中 (x > xp[-1] , 回傳    yp[-1]+(xp[-1]-xp[-2])\*(yp[-1]-yp[-2])/(xp[-1]-xp[-2]) , y)

並使用此函數確定網格節點的 x 和 y 位置

```
xn = myextrap (xm, yn x2d, X)
= myextrap (xm, x2d, 是)
```

最後我們設定網格節點的位置並匯出vtk。

為了我在範圍 (網格.nodeCount () )：  
    網。節點(i).setPos(pg.RVector3(xn[i],yn[i],zn[i]))

```
網 。匯出VTK( '3 dvtk / ' + vtkfile )
```

現在3dvtk資料夾中的所有vtk檔案都可以載入到paraview中以形成柵欄圖。此外，還可以載入3d反演結果並與2d反演結果進行比較。如果地形沒有顯示大曲率，則可以使用原始（捲尺）檔案進行二維反演，並且反向轉換將添加地形：

```
xn = myextrap (xm, yn = t, X)
myextrap (xm, zn = zn + t, 是)
myextrap (tn,          t, z)
為了我在範圍 (網格.nodeCount () ) :
網 。節點(i)。setPos ( pg.RVector3 ( xn [ i ], yn[i], 錄[我]))
```

生成的vtk檔案可以在ParaView中一起查看，並且透過對檔案進行分組可以非常輕鬆地控制其視覺化。為了更方便，可以將網格連接到單一vtk中。最後，結果如圖33所示。

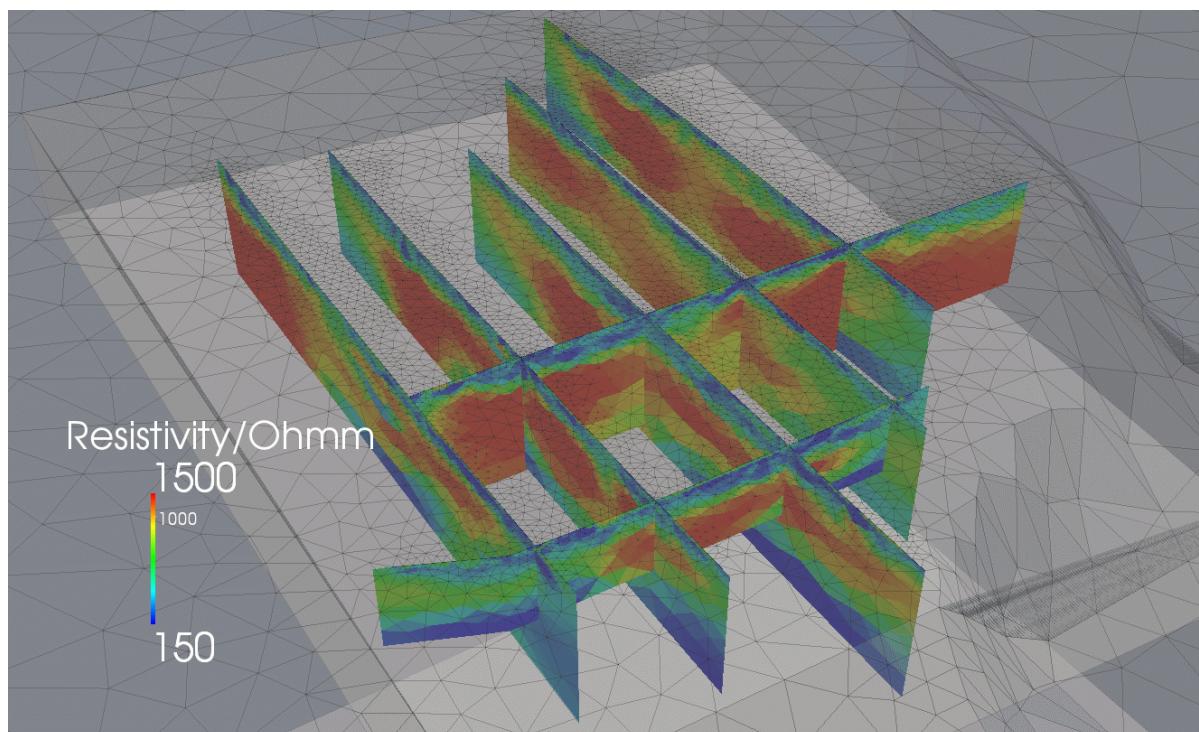


圖33：2d結果與3d表面網格的柵欄圖

## E.7。使用土柱的Hydrus3D網格進行正演計算

(請注意由於邊界節點重新編號而造成的限制) 文檔/操作方法/  
模擬Hydrus結果始發地：烏拉諾爾(Ulla Noell)(BGR漢諾威)

任務：使用Hydrus3D模擬中的離散化和電阻率向量問題：將非結構化網格邊界附加到  
(通常是規則的)四面體網格

隨著使用 ERT 監測水力過程，需要對給定的含水量和電阻率進行有效的正演計算。Hydrus3D 使用通常規則排列的四面體網格。為了進行適當的前向計算，需要遠離參數的邊界。進一步定期延長將導致節點數量大幅增加，從而導致運行時間大幅增加。因此，參數框被非結構化四面體包圍，其電阻率是使用反演中的參數延長來確定的。這個問題與創建常規 3D 網格進行反演的問題非常相似，不同之處在於 (a) 輸入已經是四面體網格，並且 (b) 必須定義輸出，以便可以直接調用使用 dcmod 的正向計算。根據(a)，建立網格並細化網格的第一部分被省略。相反，使用以下 pygimli 腳本匯入並儲存 Hydrus 網格。（使用 gzip -d 解壓縮 MESHTRIA.TXT.gz 以獲得 Hydrus3D 網格）

```

網格=pg. Mesh (3) # 新格 f = open
('MESHTRIA.TXT', 'r')
為了我在range(6):#讀取前6行
    第 1 行 = f 。閱讀廣告行 ()

nnodes = int (line 1 . split () [0]) # number ncells =   的 節點
int (line 1 . split () [1]) # and 列印          細胞
    n 個節點，   細胞
1號線 = F 。閱讀廣告行 ()
節點 = []
為了你在range ( nnodes ):# 讀取所有節點並追加在網
    位置 = f 。重讀廣告行 () 。split() # 否 。 ,x ,y ,z ->位置
    p = pg 。RVector3(float(pos[1]),float(pos[2]),float(pos[3])) n = 網格。建立節點
        (p)
    節點。追加 (n)

第 1 行=f 。讀取 line ()
第 1 行=f 。讀取行 () 個
單元格 =[]
為了字在range ( ncells ):# 讀取所有儲存格並追加在網
    位置 = f 。重讀廣告行 () 。分裂 ()
    i 、j 、k 、l=int(pos[1]) 、int(pos[2]) 、int(pos[3]) 、int(pos[4])
    c = 網格。建立四面體 (節點 [ i-1 ] 、節點 [ j-1 ] 、節點 [ k-1 ] 、節點 [ l-1 ]) 個單元格 。追加 (c)

F 。關閉 ()

列印網 # 列印網格屬性並匯出為 bms 和 vtk 網格。保存 ('水蛇')

網 。導出VTK ('水蛇')

結果，我們有了 Hydrus.bms 和 Hydrus.vtk，並且可以使用 ParaView 將其視覺化。就像在常規 3d 網格中一樣，其邊界會匯出為 vtk 檔案。

網 。建立鄰居資訊 () 為了乙在網 。邊
界 ()::
    如果不是 (b .離開 Cell () 且      b.右側單元格 ()) :
        b.設定標記 (1)

聚 = pg .網布 (3)

```

```
聚. createMeshByBoundaries (mesh, mesh.findBoundaryByMarker(1)) 多邊形。匯出  
AsTetgenPolyFile ('paraBoundary')
```

其餘部分與另一個範例相同，即- 第2部分：創建世界，網格以獲得邊界- 第3部分：將內盒和外盒的表面合併並網格在一起，只是各個單元的標記不是恆定為2（反轉）），但從1開始計數。

```
# 將內部網格合併到外部  
為了米、公分在列舉中(Inner.cells()): # 遍歷所有儲存格  
    節點=pg.標準向量節點()  
    為了在在列舉中(c.nodes()): # 如果不存在則新增節點  
        節點。Append(Outer.createNodeWithCheck(n.pos()))  
    Outer.append(Outer.createNodeWithCheck(n.pos()))創建 eCell( 節點 , m+1 );
```

進一步注意，外框中的單元保持標記0。)：

```
資源 = . 。 # 載入從文件  
指數 = NP . arange ( len ( res ) + 1 ) A =  
np 。 vstack ( (索引， NP 。 hstack ((0, 休息  
f = open('rho.map', 為了 'w' )  
排在A：  
F 。寫成('%d'%行[0] +' |t' + '%f' % 行 [1] + ' |)  
F 。關閉 ()
```

對 dcmod 的最終呼叫如下：

\$dcmod -v -S -a rho.map -s 資料檔 mesh.bms

## E.8。如何使用 Hydrus2D 模擬進行合成資料反演

文件/操作方法/Hydrus mesh2d 貢獻者：

Sarah Garre (魯汶大學)

任務：使用 Hydrus2D 網格/電阻率進行正向模擬和反演給定的三角形網格 (MESHTRIA.TXT) 顯示的地形

Garre 等人更詳細地解釋了背景和結果。 (2012) 。問題：給定的 Hydrus 網格沒有

1. 足夠遠的邊界以確保準確的解決方案
2. 適當的邊界條件 (地下3個外部邊界上的混合BC)

解決方案：

1. 以原生Hydrus2d格式讀取
2. 在網格周圍添加一個帶有適當BC的盒子
3. 外部空間進行三角剖分並與內部網格合併
4. 在pygimli中細化並進行前向計算
5. 使用表面/地下電極設定反演

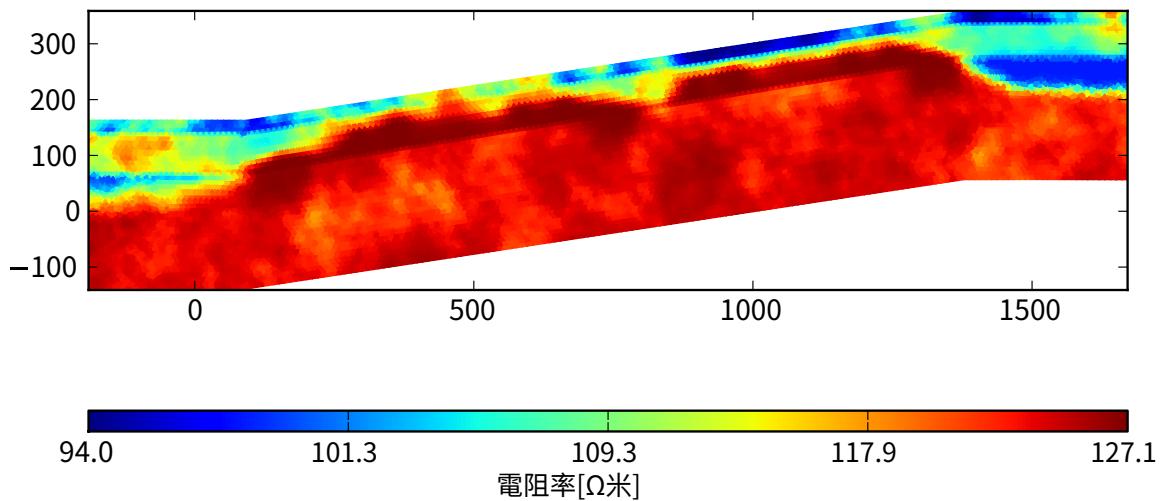


圖 34：透過 Hydrus2d 建模獲得的電阻率網格

圖 34 顯示了從水力模擬和阿奇定律應用中獲得的網格和給定電阻率分佈。

首先我們使用函數readHydrus2dMesh讀取MESHTRIA.TXT，參見pygimli/utils/mesh.py以程式碼為例，說明如何讀取 ASCII 檔案。然後我們建立相鄰訊息，將所有儲存格的標記從1開始編號。

(從 pygimli.viewer 導入 showMesh) 。

導入 pygimli 我作為控球後衛

```
網格 = readHydrus2dMesh ('MESHTRIA.TXT' 網  
格 .createNe ighbourInfos ()  
為了 我知道了在列舉中 (網格。細胞 ()) :  
C 。設定標記 (i + 1)
```

接下來，我們找到所有邊界節點並按順時針順序對它們進行排序：

```
x,y, b 節點 = [],[],[]  
為了乙 在網。邊界 ():  
我不是 b 。是否離開 Cell () b · 右側單元格 ():  
為了n在b.節點 ():  
如果不是在b 節點 :  
    b 節點。追加 ( n )  
    X 。追加 ( np . round ( n . pos () . x () * 1e3 ) / 1e3)  
    y 。追加 ( np . round ( n . pos () . y () * 1e3 ) / 1e3)  
  
xm = np .平均值 (x)  
ym = np .平均值 (y)  
角度 es = np .數組 ( np .角度 ( x - xm + ( y - ym ) * 1j ))  
指數 = NP .argsort (角度)
```

我們創建一個空網格，創建邊界節點並透過建立 ed 連接它們

格斯：

```
聚 = pg .網布 (2)  
為了我在範圍 (l en ( ind )):  
    聚. createNode(bNodes[ind[i]].pos())
```

```

為了ID在範圍 (0, poly.nodeCount () ) :
    聚. createEdge (聚. 節點 (id), 聚. 節點 ((id + 1)%
        聚. nodeCount (), pg. MARKERBOUNDHOMOGENNEUMANN )

```

我們現在提取四個角落來建造外盒。

```

y1b = min(y [x==min(x)]) y1t =
max(y [x==min(x)]) y2b = min(y
[x==max(x)]) y2t = max(y
[x==max(x)]) x2t = max(x
[y==max(y)]) xbound, ybound =
500。, 500。
n1 = 聚. createNode (pg.RVector3 (min(x) - xbound, y1t, 0.0))
n2 = 聚. createNode (pg.RVector3 (min(x) - xbound, y1b - ybound, 0.0)) n3 = 聚. createNode
(pg.RVector3 (max(x) + xbound, y2b - ybound, 0.0)) n4 = 聚。 createNode (pg.RVector3 (x2t +
xbound, max(y), 0.0))

```

四個角落相互連接並與內盒的兩個上角相連：

```

聚. createEdge (聚. n1, n2, pg. MARKERBUNDMIXED )
createEdge (聚. n2, n3, pg. MARKERBUNDMIXED )
createEdge (i1=int( n3, n4, pg. MARKERBUNDMIXED )
    NP。非零 ( (x [ind]==min(x) )&(y [索引] ==y1t ) )[0][0])
i2=int( NP。非零 ( (x [ind]==x2t )&( y [ind] == max(y) ) )[0][0])
聚.創建邊 (n1, 聚.節點 (i1), pg. MARKERBOUNDHOMOGENNEUMANN )
聚. createEdge(poly.node(i2), n4, PG. MARKERBOUNDHOMOGENNEUMANN )

```

我們可以將 Poly 檔案輸出為 vtk 並顯示它：

```

聚.
exportVTK('out.poly')showMesh(poly)

```

我們建立一個新的三角形對象，放入標記並產生網格：

```

三 = pg. TriangleWrapper (poly) # 新三角形物件 tri. addRegionMarkerTmp
(0, pg. RVector3 (網格. xmin () + 1.,
                    網。 ymin () + 1。 ), -1 )
三. setSwitches ('-pzeAfaq34') # plc, 0 索引, attribute, mesh2 = pg. 網布     品質     34
(2)
三.生成 (網格2)

```

最後我們將原始網格中的所有單元複製到獲得的網格中：

```

為了細胞在網。細胞 () :
    網格2。複製單元格 (單元格)

```

```

顯示網格 (網格2)
網格2。儲存 ('網格.bms')

```

由於正向計算需要更高的精度，我們決定透過在二次網格上使用總勢計算來實現：

```

網格3=pg. 網格 (2) 網格 3。建立
P2Mesh (網格2) 網格3。儲存
('meshFor.bms') 列印網格3

```

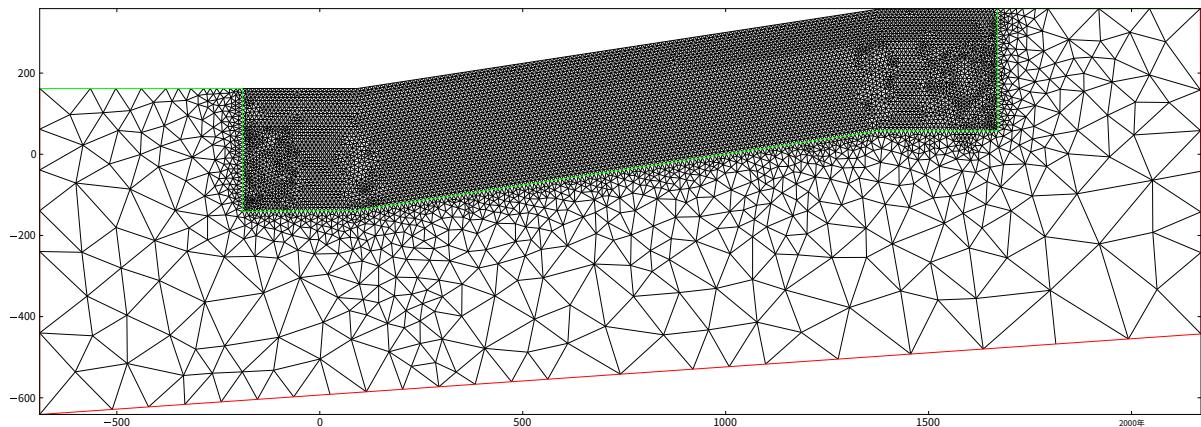


圖 35：輸入  $\text{fo}_r$  網格劃分（紅線和綠線）和組合網格

現在我們透過導入來了解實際 向後計算：電阻率資訊取自的 VTK 檔案：

```
vtk=pg。網布 (2)
VTK。importVTK(' 電阻 泰。VTK')
資源=np。數組 ( vtk. 導出 Data ('電阻率'))
```

電阻率向量出現在第一列（關 t 使用 std C++ 映射將數字（標記）res 放入映射檔中：鍵）中，並且

```
nr=np .排列 (l en ( res ))+ 1
resmap=np 。vstack ( (nr.T,  res .T)) .T
單元格映射 = pg .stdMapF F_()
為了鍵值在重新映射：
cellMap [鍵]=val
```

所得電阻率圖（標記 0）可獲得應用空白單元格，即建立的外部區域 y 延長中的單元格：電阻率 b

```
網格3。映射單元格屬性 ( 細胞圖)
網格3 。填充空單元格 ( m 伊什3.透過屬性找出 (0.0), -1.0 )
```

我們現在載入方案檔（te it 。沒有實際資料的資料），建立一個新的建模運算子並重新視為 R。計算 產生的 u A

```
數據= pG 。數據容器 (' Hydrus2d 。噓')
f=pg 。DC多電極Mo 戴爾林 (mesh3, 數據, True)
F 。計算 計算 (數據)
數據 。設定 ('r', 數據。u ())
數據 。保存 ('輸出.dat', 'abmn')
```

現在我們可以使用bert腳本對資料集進行反演。這裡的問題是我們有表面和地下電極，但 bert 尚未自動覆蓋（但很快就會覆蓋）。因此，我們創建一個僅包含表面電極的虛擬資料文件，將其用於網格生成，並在向其添加一些高斯雜訊後反轉合成資料：

```
ns el=36 # 表面電極數量
錯誤=1
```

```

輸入文件=輸出.dat
輸出檔=out0.dat
datfile=syn.數據
# 僅使用表面電極建立虛擬資料檔 echo $ns el
> $輸出檔案
頭 -n 2      $內菲萊/尾部-n 1>>$ 輸出文件
頭 -n $ [ ns el + 2 ] echo 0  $內菲萊/尾巴 - n $ns el/塔克>>$ 輸出文件
>>$ 輸出文件
# cr 吃 ECFG 錐刀 產生 網格
bertNew2D拓撲 $輸出檔案 > 伯特0.cfg
回波參數深度=300>>伯特0.cfg 回顯
PARADX=0.2>>伯特0.cfg
# 僅使用虛擬檔案 bert bert0 進行網格劃分。cfg 網格

# 無高斯雜訊的合成資料 dcedit -vSEN -e $ERR -u 0 -o $
datfile $ infile
# 建立正常的 cfg 檔案並在 bertNew2DTopo $ datfile 上
執行 nv ersi>伯特.cfg 回顯 LAMBDA=300>>伯特.cfg 回顯
ZWEIGHT=0.1>>伯特.cfg 回顯 BLOCKYMODEL=1>>伯特.
cfg

# cp 網格。bms mesh/ # 在 bert bert 上使用 Hydrus 網格進行 nve
rsi .cfg 罐計算

```

圖 36 顯示了所獲得的反演結果，該結果與合成模型接近。模型中的非常低的對比度和有限的電極數量阻礙了更好的重建。在現實異質性的情況下，這樣的結果只能透過延時方案來檢索，其中所有系統效應和小規模結構都將被抵消。

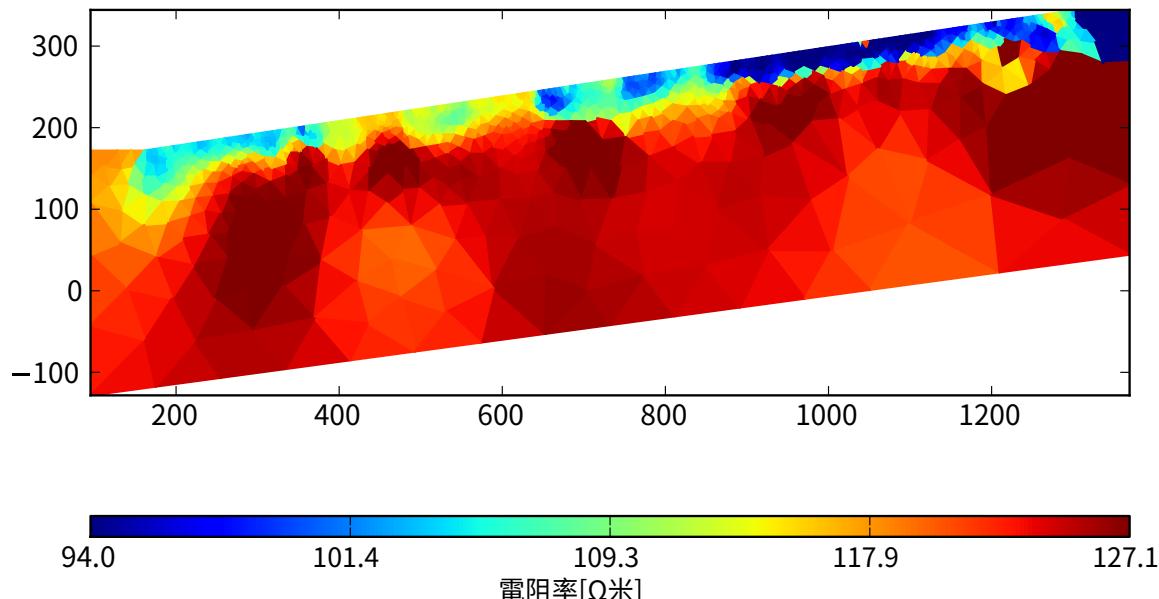


圖 36：得到的反演結果

## E.9。模擬傾斜鑽孔中的 CEM 環形電極

文件/操作方法/傾斜鑽孔 貢獻者：Laure

Beff (加州大學魯汶分校)

任務：使用 CEM-FEM 模擬環形電極的真實幾何因素 問題：定義包含環形電極的傾斜鑽孔工具的網格

鑽孔探頭是在某些位置帶有金屬環電極的圓柱體。圓柱體近似為規則稜柱（例如 6 段），可以透過以下方式建立

createCube -Z -s 6 立方體

建立一個以原點為中心的 1x1x1m 圓柱體。透過使用 -H 新增孔標記，在網格劃分中忽略圓柱體的空間。探測工具本身由一系列非電極和電極組成，它們隨後相互合併，並在平移到鑽孔位置後與具有典型邊界條件的大盒子（進行模擬的世界）合併：

```
polyCreateWorld -x $SIZE -y $SIZE -z $SIZE -m1 $MESH # 大盒子=世界
```

在某些長度儲存在標量和向量變數之前，例如

```
RAD=0.045#鑽孔半徑  
DIST=(0.103 0.152 0.202 0.252 0.302 0.353 0.100) # 電子編輯
```

首先，我們創建兩個不同圓柱體的原型並將其移動，使頂部為零並且它們具有正確的半徑

```
polyCreateCube -H -Z -s $SEG cyl # 建立（閉合）單位圓柱體 polyTranslate -z -0.5 cyl  
# 移動到上邊界=表面  
polyScale -x $RAD -y $RAD cyl # 縮放到正確的半徑
```

由於距離不同，它們的高度稍後會縮放，現在電極

```
cp cyl.聚電 0.聚 PolyScale -z # 單位電極  
$DRING elec 0 # 正確的高度
```

每個鑽孔 b 都是從圓柱體元素開始在原點建構的

```
cp 環 C.聚鑽孔. poly # 最上面的 piece
```

變數 z 保存當前深度，並使用到第一個電極的距離進行初始化，該距離透過使用從向量 PD 中檢索

```
z=${PD[ $b ]}/
```

每次添加元素後，透過使用 awk 添加單一長度來增加 z

```
z='迴聲 $z $DRING/awk' {列印 $1 + $2} '# '' # z 中的輸出
```

複製並翻譯標準電極以及電極標記 \$ELM。相關聯，稍後用於識別 CEM 電極（從 -10000 開始向下）。最後電極與鑽孔合併

```
cp 電 0.聚電聚polyAddVIP -B -m -$ELM elec #  
polyTranslate -z -$zelec polyMerge 鑽孔  
elec 鑽孔
```

電極之間的部件也進行類似的處理，即縮放、平移和合併。內循環

前進/0.。6/做

完畢

穿過電極，外部 (b) 穿過鑽孔工具。在與世界融合之前，電極會被平移到它們的位置。然而，鑽孔傾斜特定角度（也儲存向量 ROT）。繞 y 軸的旋轉（即 y 保持不變）是透過以下方式完成的

多旋轉 -R -y \$/旋轉[ \$b ]鑽孔 # -R 為 rad ( defau lt=deg )

最上面的面（6 個電極中的每一個）與地球表面相交，因此必須移動到 z=0。因此，將建立鑽孔多邊形檔案的副本，並使用 head 和 tail 命令相應地更改具有接近表面的點的線

```
頭-n1 鑽孔。聚>鑽孔。聚 新鑽孔。聚 # 保持 /尾巴 第1行，更改2+3  
頭-n3  
-n2/  
awk'列印 1美元"|" 2美元"|" 3美元"|" 0"|" 5美元"/>>新鑽孔。聚  
頭-n6鑽孔。聚/尾部-n3>>新鑽孔。poly # 保留 4-6 等。
```

問題在於，由於移動，最上段的矩形側面不再共面。一種解決方案可能是更複雜的定位。另一種方法是將所有矩形分割成定義為共面的三角形。這是透過將第一部分（初學者）轉換為常見的 STL 格式（使用三角形）然後使用 polyConvert 返回 POLY 來實現的：

多轉換-S 鑽孔。聚  
聚轉換-P 鑽孔。stl

不幸的是，STL 檔案中的孔標記遺失了，因此必須重新新增：

polyAddVIP -H -x 0 -y 0 -z -0.01 鑽孔

請注意，所有 PLC 都可以使用以下命令轉換為 VTK（並在 ParaView 中查看）

polyConvert -V 檔案。聚

當觀察PLC時，我們看到上片的下表面被解剖，而同一位置的相鄰電極的上表面仍然有6個節點。因此，我們使用選項 -O 建立所有電極圓柱體，以便省略頂面和底面。因此，我們有一個有效的網格輸入，它可以是使用 tetgen 的網格，並使用以下方法轉換為 BMS/MESH/VTK 格式檔案：

```
tetgen -pazVACq1.15 $MESH # 網格 $PLC，質量 =1.2 – 網格 。1.* meshconvert -vBDMV  
-it -o $MESH $MESH。1 # 轉換為BMS/MESH/VTK
```

為了實現高精度，我們使用 -p 進一步將網格轉換為二次元

網格轉換 -vBD -p -o \$/網/P \$MESH。生物管理系統

請注意，如果存在網格劃分錯誤，可以使用以下方法調查有問題的區域

tetgen -d me sh 文件

且產生的網格僅包含網格剖分或非共面元素。如果有額外的表面電極，可以使用 polyAddVIP 添加

polyAddVIP -fsurfaceelec 。xyz -m -99 \$MESH

它們應局部細化至  $a/10$ ，其中  $a$  是電極距離：

```
polyRefineVIPS -z -0.015 $MESH
```

整個腳本如下。

現在我們要使用腳本進行建模並研究點電極的差異。首先我們使用 -1 運行 dcmod 並應用方案文件

```
dcmod -v -1 -sdatafile 網格/meshP 。bms # 輸出數量。歐姆
```

請注意，雖然 dcmod 嘗試透過位置找到電極，但有一個優先級，首先搜尋 CEM 電極，然後搜尋節點。在不同類型之間距離較近的情況下，可能會發生錯誤關聯，導致「自由電極」。因此，我們建議始終在 Node 之前使用 CEM，並透過對資料進行排序來釋放電極。接下來，我們使用以下命令建立一個包含解析幾何因子的文件

```
dcedit -f "abmnk" -o geom 。資料檔案
```

最後，我們使用該文件用這些幾何因素「過濾」數值結果。

```
dcedit -vSB -c geom 。數據-oeleff。資料編號歐姆
```

正如建模的右=1/ $k$ 編號，所結果的  $\rho_A=k_{\text{安娜}} \cdot \text{右} = k_{\text{安娜}}/k$ 編號是電極效應。