

大型線性方程組解法

作者：許進超

作者簡介：許進超是美國賓州州立大學數學系的威拉曼數學講座教授（Verne M. Willaman Professor of Mathematics），同時也是該校的計算數學與應用中心主任。他是數值偏微分方程快速算法的專家，在開發、設計和分析解決大型方程組快速算法有許多開創性的研究。

重點摘要

- 如何找線性代數方程組的解？
這在數學與應用中是一個非常基本的問題，作者在本文討論求解的數值方法。
- 從最古老的高斯消去法到現代的多層網格法，作者在文中演示如何以數學技術研發大規模線性方程組的有效方法。
- 作者也有對於電腦硬體的開發與數學演算法的開發之間作出比較與探討。

引言

本文要討論的主題是解線性方程組的數學技術。先舉個我童年經歷中的一個簡單例子。

在我還小的時候，外公給我出了一道算術題：「魚頭 3 寸，魚身是頭尾之和，魚尾是頭身之半，魚長幾何？」這是一道簡單的線性代數習題。設魚頭、魚身和魚尾的長度分別是 x_1 、 x_2 和 x_3 ，根據題設的條件，我們可以建立如下等量關係：



← x_3 * — x_2 * — x_1 →

$$\begin{aligned} x_1 &= 3 \\ x_1 - x_2 + x_3 &= 0 \\ x_1 + x_2 - 2x_3 &= 0 \end{aligned}$$

通過求解上述線性方程組，我們就可以求得魚長為 $x_1 + x_2 + x_3 = 3 + 9 + 6 = 18$ 。用字母表示未知量，使得客觀世界中的未知量可以用字母來表示，然後依據一些規律找出數量之間的等量關係，列成方程組。如果這些方程關於未知量是線性的，我們稱之為線性代數方程組（linear algebraic system）。一般說來，含有 n 個未知數和 n 個線性代數方程式可以寫成

$$1 \quad \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$



九章算術 卷八：方程。（維基文庫）

大多數實際問題的數學應用都可以歸結到解上述線性代數方程組。例如，將微分方程組的數值離散化（numerical discretization）和對非線性代數方程組的做線性逼近（linear approximation），可以得到這種型式的線性代數方程組。因此求解線性方程組的數值算法是科學計算最基本的研究課題。高效的數值求解算法在現代工程學、物理學、化學、經濟學和計算機科學等科學研究中扮演著基礎而重要的角色。

最古老也是最常用的求解線性方程組方法是以數學家高斯（1777 年 4 月～1855 年 2 月）而命名的高斯消去法（Gaussian elimination），但它最早可見於成書於約公元前 150 年左右的中國古籍《九章算術》。高斯消去法的核心想法是通過方程之間的相互置換、消除得到上三角或者下三角方程組，然後以代入法完成求解。對於一個 $n \times n$ 的矩陣，高斯消去法所需的浮點運算次數（FLOPs）為 $\frac{2}{3}n^3 + O(n^2)$ ，當 n 非常大的時候，這個算法因為過於耗時而變得不可用。

在超級電腦領域裡，最為人們所關注的是對全球超級電腦系統作詳細介紹與進行排名的「TOP500 計畫」。因為高斯消去法在代數方程求解過程的重要性，電腦系統的排名就是通過高性能 LINPACK（HPL）^① 以高斯消去法求解 n 階稠密 ^② 線性代數方程組所花時間來確認的。如果測得系統計算時間為 T ，那麼該電腦系統的實際計算峰值為 $\frac{2n^3 + O(n^2)}{3T}$ ，測試結果以每秒浮點運算次數給出。

TOP500 計畫自 1993 年開始

實施起，最強大的超級電腦基本是由美國和日本角逐，在最近的十年內，中國逐漸位列並領先於該榜單中。2018 年 11 月列於前四名的依序是：IBM 為美國能源部旗下橡樹嶺國家實驗室（ORNL）開發建造的超級電腦高峰（Summit），IBM 為美國能源部旗下的勞倫斯利佛摩國家實驗室（LLNL）建造的雪樂（Sierra），中國國家超級計算中心（無錫）的神威·太湖之光和中國國家超級計算中心（廣州）的天河二號。高峰的計算峰值性能為 200 PFLOPs，也就是實際每秒進行 200×10^{15} 次浮點運算。於是，如果求解 10^7 階稠密線性代數方程組，就大概需要一小時；如果求解方程組階數 10^8 ，就大概需要 40 天；如果求解方程組階數 10^9 ，就大概需要 114 年。從這個角度來看，無論電腦性能如何提升，為了解決大規模計算問題，我們必須研究比高斯消去法更為高效的數值算法。

我們可以從兩個角度來研發求解代數方程更為高效的數值算法。第一：我們在高斯消去法的基礎上做改進。第二：我們用完全不同的方法，即迭代法（iterative method）。

- ① 編註：它是在數位電腦上執行數值線性代數計算的軟體資料庫。
- ② 編註：指大多數係數都是非零的。



高峰超級電腦。（Flickr, OLCF at ORNL）



天河二號超級電腦。（維基）

高斯消去法的改進

如何提高高斯消去法的效率，是科學計算中大家一直關注的一個重要問題。我們在這裡簡單介紹兩個途徑：施特拉森演算法（Strassen algorithm）以及稀疏矩陣演算法（sparse matrix algorithm）。

施特拉森演算法

1969 年，施特拉森（Volker Strassen）提出的利用分治（divide-and-conquer）思想結合特別的組合方式得到的快速矩陣相乘算法。我們以 2×2 矩陣為例，展示特殊的組合方式如何減少一次乘法運算。考慮兩個 2×2 矩陣的乘法：

$$\begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

於是我們得到： $r = ae + bg$ ， $s = af + bh$ ， $t = ce + dg$ ， $u = cf + dh$ 。但是，通過巧妙的組合，我們可以得到：

$$\begin{aligned} r &= p_5 + p_4 + p_6 - p_2 \\ s &= p_1 + p_2 \\ t &= p_3 + p_4 \\ u &= p_1 + p_5 - p_3 - p_7 \end{aligned}$$

其中：

$$\begin{aligned} p_1 &= a \cdot (f - h) \\ p_2 &= (a + b) \cdot h \\ p_3 &= (c + d) \cdot e \\ p_4 &= d \cdot (g - e) \\ p_5 &= (a + d) \cdot (e + h) \\ p_6 &= (b - d) \cdot (g + h) \\ p_7 &= (a - c) \cdot (e + f) \end{aligned}$$

如果將 a, b, \dots, h 都想像成 $\frac{n}{2} \times \frac{n}{2}$ 的矩陣，那麼這裡的乘法都是矩陣乘法，於是我們得到分而治

之思想，並且在只考慮乘法的情況下，我們得到 $n \times n$ 矩陣乘法運算量 $T(n)$ 的遞迴（recursion）：

$$T(n) = 7T\left(\frac{n}{2}\right) + \mathcal{O}(n^2)$$

從而，我們將一般的矩陣與矩陣相乘 $\mathcal{O}(n^3)$ 的工作量降到 $\mathcal{O}(n^{\log_2 7}) \approx \mathcal{O}(n^{2.807})$ 。進一步的計算發現，施特拉森演算法所需的加法次數也不超過 $\mathcal{O}(n^{2.807})$ 。

施特拉森演算法在文獻中不斷改進，至今為止，文獻中最好的結果是需要 $\mathcal{O}(n^{2.372})$ 的浮點運算次數。這個工作量仍然太大（而且還涉及穩定性問題）。不難看到，最快的演算法不會少於 $\mathcal{O}(n^2)$ 浮點運算次數，但是否存在演算法達到這個下界，還是一個未解問題。從理論上講，求解一個一般的線性方程組最少需要多少工作量是一個懸而未決，非常有意義的數學問題。

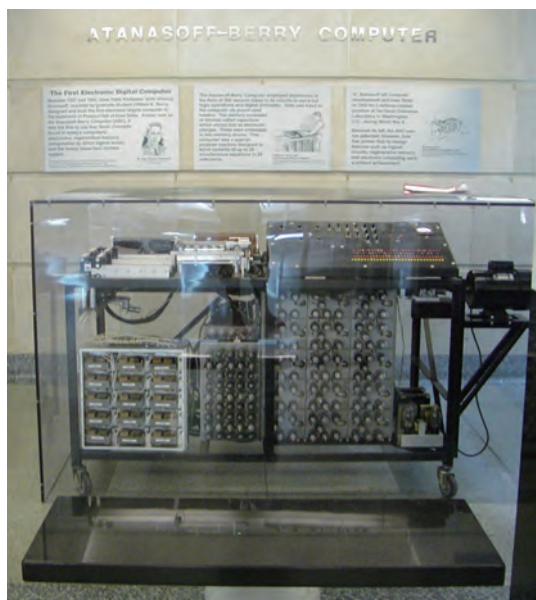
稀疏矩陣演算法

求解線性方程組，甚至是一般數學問題的一種有效途徑就是充分利用問題的特殊性。例如，當一個線性方程組 $Ax = b$ 的矩陣 A 的非零元（entry）的個數只有 $\mathcal{O}(n)$ 時，我們稱這樣的 A 為稀疏矩陣，對應的線性方程組為稀疏的。稀疏性在目前的科學與工程實踐中大量出現，比如採用有限元（finite element）或者有限差分（finite difference）得到的偏微分方程的離散方程組一般都是稀疏方程組。利用高斯消去時帶寬（bandwidth）的變化特點，通過特殊的技巧對未知量進行重新排序以減少矩陣的帶寬，這將極大的減少高斯消去法的計算量。然

而，如何最大可能的減少一個一般稀疏矩陣的帶寬本身是一個 NP-完全（NP-Complete，非決定性多項式時間完全）問題。

高性能計算：硬體與演算法的改進

1937 年，世界上第一台電腦（阿塔納索夫 / 貝瑞電腦〔Atanasoff-Berry Computer〕，簡稱 ABC 電腦）被開始設計，不可程式化，僅僅用來求解線性代數方程組，並於 1942 年成功進行了測試。從 20 世紀 60、70 年代開始，電腦技術得到飛速發展，極大的推動了生產力的發展，直到現在乃至更遠將來的每一天都在革新人類生活、工作和研究的方方面面。一個對電腦技術發展速度的預測是著名的摩爾定律（Moore's law），它由英特爾（Intel）創始人之一戈登·摩爾（Gordon Moore）提出，通用



阿塔納索夫 / 貝瑞電腦。（維基·Manop）

的表達是：積體電路（integrated circuit）上可容納的電晶體（transistor）數目，約每隔 18 個月便增加一倍。也就是說相同價格購買的電腦，性能每隔 18 個月增加一倍。

在很多科學計算領域，演算法的發展已經可以匹敵硬體技術發展的摩爾定律。例如，在大規模線性方程組的求解上，電腦誕生之後的 30 多年間，從高斯消去法到多層網格法（multigrid method），求解線性代數方程組的算法效率提升了 1600 多萬倍，這與同期硬體速度的提升是一致的。事實和理論皆證明，對於很多實際問題，「演算法的進展擊敗摩爾定律」。^③

迭代法

與直接法^④不同，迭代法採取逐次逼近的方法來求解線性方程組。即從某一個初始值出發，按照一

^③ 編註：“Progress in Algorithms Beats Moore's Law”，出自美國總統科技顧問委員會（PCAST）2010 年向總統歐巴馬和國會的報告。

^④ 編註：直接法就是由通過有限多步驟的運算，找到精確解的方法，如高斯消去法。



戈登·摩爾。（維基）

定的計算公式（迭代公式），構造一個向量的無窮序列，其極限才是方程的精確解。簡單說來，我們可以把迭代法的發展分成三類。第一類為定常迭代法（stationary iterative method），例如雅可比法（Jacobi method）、高斯 / 賽德爾法（Gauss-Seidel method）、逐次超鬆弛法（successive over-relaxation, SOR）以及這些方法的改進與加速形式。這些迭代法容易實現，但是只能對於一些特殊的矩陣保證收斂。隨著科學技術的發展，需要求解的線性方程組的規模越來越大，矩陣的類型越來越廣，此時這些方法的收斂速度一般太慢，甚至不收斂。其中比較有效的逐次超鬆弛方法依賴於參數的選取，但一般無法得到最優參數，從而限制了該方法。第二類為克雷洛夫子空間迭代法（Krylov subspace iteration method）。子空間方法具有儲存量小，計算量小且易於平行計算（parallel computing）等優點，非常適合於平行求解大型稀疏線性方程組。克雷洛夫子空間迭代法的工作原理是基於初始殘量（initial residue）生成一個克雷洛夫子空間，每一步在子空間中求一個極小化問題得到一個精確解的近似。例如求解對稱正定（symmetric positive definite, SPD）矩陣的共軛梯度（conjugate gradient, CG）法，求解對稱不定矩陣的最小殘量法（MINRES）和求解一般非對稱矩陣的廣義最小殘量法（GMRES）。第三類用於加速克雷洛夫子空間迭代法的收斂性的一些預條件方法。如果適當的選擇預條件子（preconditioner），迭代會在很少的步數就達到足夠的精度。多層網格法與區域分割法（domain decomposition method），或是更一般的子空間校正法（method of subspace correction）

[13]，是構建預條件子的最有效方法之一。作為示範，接下來我們將簡要討論布蘭堡 / 帕司柴克 / 許（Bramble-Pasciak-Xu, BPX）預條件子和希普特麥爾 / 許（Hiptmair-Xu, HX）預條件子在偏微分方程數值解的應用。

高斯 / 賽德爾法以及一般線性迭代法

以 3×3 線性方程組為例，高斯 / 賽德爾迭代公式 ($x^{m-1} \rightarrow x^m$) 如下：

$$\begin{aligned} a_{11}x_1^m + a_{12}x_2^{m-1} + a_{13}x_3^{m-1} &= b_1 \\ a_{21}x_1^m + a_{22}x_2^m + a_{23}x_3^{m-1} &= b_2 \\ a_{31}x_1^m + a_{32}x_2^m + a_{33}x_3^m &= b_3 \end{aligned}$$

我們現在來考慮下面這個特殊的 3×3 的簡單例子：

$$4 \quad A_\epsilon x = b$$

其中

$$\begin{aligned} A_\epsilon &= A_0 + \epsilon I \\ A_0 &= \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \\ 5 \quad b &= \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} \in \text{值域}(A_0) \end{aligned}$$

通過簡單計算，可以獲得 A_0 的三個特徵值 $3, 1, 0$ ，其零空間（null space）由向量 $e = (1, 1, 1)^T$ 張成（span）。可以看出，隨著 ϵ 的減小，矩陣 A_ϵ 逐漸變得奇異（singular）。

以 $x^0 = b$ 為初始值， $\|A_\epsilon x^m - b\| \leq 10^{-8}$ 為迭代停止條件，我們發現，如表 1 第二行所示，高斯 / 賽德爾法所需迭代步數隨著 ϵ 減小而急遽增長，令

人驚訝的是，當 ϵ 變到 0 的一霎那，迭代步數瞬間從上萬變成了 2！

ϵ	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	0
$A_\epsilon x = b$	18	100	852	6982	54470	2
$A_\epsilon \underline{x} = \underline{b}$	12	10	6	6	4	

表 1：高斯 / 賽德爾法應用到 4 與 6。

如何修正高斯 / 賽德爾法的這個奇怪的現象呢？我們這裡介紹一個簡單而且通用的方法。將近似奇異向量 (approximate singular vector) e 顯式引入到空間中，於是 $x = (x_1, x_2, x_3)^T \in \mathbb{R}^3$ 可以 (不唯一的) 表示為 $x = x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e = P \underline{x}$ ，其中擴充 (extended) 矩陣 $P: \mathbb{R}^3 \mapsto \mathbb{R}^4$ 為

$$P = (e_1, e_2, e_3, e) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

於是我們可以將 $A_\epsilon x = b$ 擴展為如下 4×4 的半正定 (positive semidefinite) 方程組：

$$\mathbf{6} \quad A_\epsilon \underline{x} = \underline{b}$$

其中

$$A_\epsilon = P^T A_\epsilon P = \begin{pmatrix} 1 + \epsilon & -1 & 0 & \epsilon \\ -1 & 2 + \epsilon & -1 & \epsilon \\ 0 & -1 & 1 + \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\epsilon \end{pmatrix}$$

$$\underline{b} = P^T b = \begin{pmatrix} -1 \\ -1 \\ 2 \\ 0 \end{pmatrix}$$

同樣以 $\underline{x}^0 = P^T b$ 為初始值， $\|A_\epsilon \underline{x}^m - \underline{b}\| \leq 10^{-8}$ 為迭代停止條件，我們利用高斯 / 賽德爾法求解擴充方程組 6，如表 1 第三行所示，高斯 / 賽德爾法對 ϵ 是一致收斂的。

為了理解這些不同的收斂行為，我們接下來簡要

討論在更一般的設定之下的線性迭代法，並給出一些收斂性分析。

給定一個有內積結構 (\cdot, \cdot) 的有限維向量空間 V 和對稱正定算子 $A: V \mapsto V$ ，亦即

$$(Au, v) = (u, Av), \quad (Aw, w) > 0$$

$$\forall u, v, w \in V, \quad w \neq 0$$

我們考慮下列線性方程組

$$\mathbf{7} \quad Au = f$$

的一般線性迭代法可以寫成

$$\mathbf{8} \quad u^m = u^{m-1} + B(f - Au^{m-1}), \quad m = 1, 2, \dots$$

其中 $B: V \mapsto V$ 是非奇異 (nonsingular) 的而且可以看成是 A 的近似逆矩陣 (approximate inverse matrix)。

我們注意到 (Au, v) 定義一個內積，而且也對任何算子 $T: V \mapsto V$ 誘導導出一個範數 $\|T\|_A$ ：

$$\|T\|_A^2 = \sup_{v \in V} \frac{(ATv, Tv)}{(Av, v)}$$

當

$$\mathbf{9} \quad \bar{B} = B' + B - B'AB$$

是 SPD，很容易證明迭代法 8 會收斂，而且還有

$$\mathbf{10} \quad \|I - BA\|_A^2 = 1 - \left(\sup_{\|v\|_A=1} (\bar{B}^{-1}v, v) \right)^{-1}$$

給定 $A \in \mathbb{R}^{n \times n}$ ，如果我們拆解 $A = D + L + U$ ，其中 D 是 A 的對角線部份，而 L 與 U 分別是 A 的嚴格下三角與嚴格上三角部分。則雅可比法與高斯 / 賽德爾法可以寫成 8 的形式有著

$$\mathbf{11} \quad B = \begin{cases} D^{-1} & \text{雅可比} \\ (D + L)^{-1} & \text{高斯 / 賽德爾} \end{cases}$$

如果 $A \in \mathbb{R}^{n \times n}$ 是 SPD，很容易證明高斯 / 賽德爾法對 $Ax = b$ 沒意外的會收斂。

事實上，我們注意到由拆解 $A = D + L + U$ ，運用等式 10，可以證明高斯 / 賽德爾法的收斂速率如下：

$$12 \quad \|I - BA\|_A^2 = 1 - \frac{1}{1 + c_0}$$

其中

$$c_0 = \sup_{v \neq 0} \frac{((LD^{-1}U)v, v)}{(v, v)_A}$$

作為上述分析的簡單應用，讓我們考慮 4。對於 4，拆解 $A_\epsilon = D_\epsilon + L + U$ 而推導得 $LD_\epsilon^{-1}U = \text{diag}(0, (1 + \epsilon)^{-1}, (2 + \epsilon)^{-1})$ ，以及

$$c_0 \geq \frac{((LD_\epsilon^{-1}U)e, e)}{(e, e)_{A_\epsilon}} \geq \frac{1}{4\epsilon}$$

因此，由 12，高斯 / 賽德爾法對於 4，即 $A_\epsilon x = b$ 的收斂速率是

$$\|I - (D + L)^{-1}A_\epsilon\|_{A_\epsilon}^2 \geq \frac{1}{1 + 4\epsilon}$$

這證明了當 ϵ 變小的時候，高斯 / 賽德爾法的收斂變慢。

我們稍後解釋為什麼高斯 / 賽德爾迭代法對 4 很有效。

共軛梯度法

作為解決 7 的非線性迭代法，給定 $u_0 \in V$ 和 $r_0 = f - Au_0$ ，共軛梯度 (CG) 法的第 m 次迭代由下式給出

$$u_m = \arg \min_{v_m \in u_0 + \mathcal{K}_m} \|u - v_m\|_A$$

其中

$\mathcal{K}_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}$, $m \geq 1$
結果是， u_m 可以通過一些三項遞迴關係推得，算法如下：對於 $m = 1, 2, \dots$ 直到收斂，做

$$1. \quad \alpha_m = \langle r_m, r_m \rangle / \langle Ap_m, p_m \rangle$$

$$u_{m+1} = u_m + \alpha_m p_m$$

$$r_{m+1} = r_m - \alpha_m Ap_m$$

$$2. \quad \beta_m = \langle r_{m+1}, r_{m+1} \rangle / \langle r_m, r_m \rangle$$

$$p_{m+1} = r_{m+1} + \beta_m p_m$$

CG 法的收斂速度可以由以下估計給出：

$$13 \quad \|u - u_m\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^m \|u - u_0\|_A$$

其中 $\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ 表示 A 的條件數 (condition number)。

從 CG 法的收斂結果可以看出，如果 $\kappa(A)$ 非常大，則 CG 法的收斂速度會很慢。因此，我們可以為 A 構建一個預條件子 B 使得 $\kappa(BA) \ll \kappa(A)$ 。預條件子應該是 SPD 且易於估值。那麼求 $Au = f$ 的解也就是等價於求 $BAu = Bf$ 的解。如果適當構建預條件子 B ，則用來解預處理 (preconditioned) 方程組 $BAu = Bf$ 的 CG 法收斂性可以大大提高。

FASP：輔助子空間預處理方法

這一節，我們討論構造預條件子的一種通用的方法，即輔助子空間預處理 (FASP: Fast Auxiliary Subspace Preconditioning) 法。我們發現把 4 擴充到 6，簡單的高斯 / 賽德爾法的收斂性得到本質的改進。我們現在把這個想法推廣到一般情形。

我們考慮 V 的擴展空間

$$\mathcal{V} = V_0 \times V_1 \times \dots \times V_J$$

並且定義

$$\Pi = (\Pi_0, \Pi_1, \dots, \Pi_J) : \mathcal{V} \mapsto V$$

我們假設以上算子 Π 是映成的，即是，任給一個 $u \in V$ 可以寫成 $u = \Pi \underline{u} = \sum_{i=0}^J \Pi_i u_i$ ，其中 $u_i \in V_i$ 。

於是，線性方程組 $Au = f$ 可以寫成如下擴展方程組：

$$\underline{A} \underline{u} = \underline{f} \quad (14)$$

其中 $\underline{f} = \Pi' f$ 以及 $\underline{A} = \Pi' A \Pi = (A_{ij})_{J \times J}$ ，即線性算子 A 擴展為

$$\underline{A} = (\Pi'_i A \Pi_j)_{J \times J} : \mathcal{V} \mapsto \mathcal{V} \quad (15)$$

需要注意對稱正定方程組對應的擴展方程組可以是半正定的，但是在對角線的算子 $\Pi'_i A \Pi_i$ 是正定的。因此可以直接對 $\underline{A} \underline{u} = \underline{f}$ 進行改進的雅可比法與改進的高斯 / 賽德爾迭代法。

讓 R_i 是 $A_{ii} = \Pi'_i A \Pi_i$ 逆矩陣的一個近似矩陣，那麼事實上，對 $\underline{A} \underline{u} = \underline{f}$ 應用改進的雅可比迭代法，即是對 $i = 0, \dots, J$ ，

$$\underline{u}_i^m = \underline{u}_i^{m-1} + R_i \left(\underline{f}_i - \sum_{j=0}^J A_{ij} \underline{u}_j^{m-1} \right) \quad (16)$$

容易證明，此方法等價於如下的平行子空間修正演算法 (parallel subspace correction algorithm)：

$$\underline{u}^m = \underline{u}^{m-1} + \sum_{i=0}^J \Pi_i R_i e_i \quad (17)$$

其中，對 $i = 0, \dots, J$ ，(在平行時)

$$e_i = \Pi'_i (f - Au^{m-1})$$

對 $\underline{A} \underline{u} = \underline{f}$ 應用改進的高斯 / 賽德爾迭代法，即對 $i = 0, \dots, J$ ，

$$\underline{u}_i^m = \underline{u}_i^{m-1} + R_i \left(\underline{f}_i - \sum_{j=0}^{i-1} A_{ij} \underline{u}_j^m - \sum_{j=i}^J A_{ij} \underline{u}_j^{m-1} \right) \quad (18)$$

容易證明，此方法等價於如下的逐次子空間修正演算法 (successive subspace correction algorithm)：

由 $u_{-1}^{m-1} = u^{m-1}$ 開始到得出 $u^m = u_j^{m-1}$ ，對 $i = 0, \dots, J$ ，如下：

$$\begin{aligned} e_i &= R_i \Pi'_i (f - Au_{i-1}^{m-1}) \\ \underline{u}_i^{m-1} &= \underline{u}_{i-1}^{m-1} + \Pi_i e_i \end{aligned} \quad (19)$$

如同 16 與 18，求 14 解的迭代法可以寫成如下一般的線性迭代法：對一些線性算子 $\underline{B} : \mathcal{V} \mapsto \mathcal{V}$ ，

$$\underline{u}^m = \underline{u}^{m-1} + \underline{B}(\underline{f} - \underline{A} \underline{u}^{m-1}) \quad (20)$$

容易證明迭代形式 20 等 8 價於加上

$$\underline{B} = \Pi \underline{B} \Pi' \quad (21)$$

而且還有

$$\langle B^{-1}v, v \rangle = \inf_{\Pi \underline{v} = v} \langle \underline{B}^{-1} \underline{v}, \underline{v} \rangle \quad (22)$$

22 的初步版本可於 [13] 中找到，而且可以直接推導得到以下的結果 [10,14]。

結合 22 與 10 我們得到

$$\begin{aligned} \|I - BA\|_A^2 &= 1 - \left(\sup_{\|v\|_A=1} \langle \bar{B}^{-1}v, v \rangle \right)^{-1} \\ &= 1 - \left(\sup_{\|v\|_A=1} \inf_{\Pi \underline{v} = v} \langle \bar{B}^{-1} \underline{v}, \underline{v} \rangle \right)^{-1} \end{aligned} \quad (23)$$

作為 23 的應用，讓我們估計範例 6 中擴展 4×4 矩陣高斯 / 賽德爾法的收斂速度。再次令 $A_\epsilon = D_\epsilon + L + U$ 給 6。則高斯 / 賽德爾法用於 6 意味著 $\underline{B} = (D_\epsilon + L)^{-1}$ 用於迭代法 20。因此根據 21，它等價於迭代法 8 用於 4，加上 $B = P \underline{B} P^T$ 。因此根據 23，我們得到

$$\|I - BA_\epsilon\|_{A_\epsilon}^2 = 1 - \frac{1}{1 + c_0} \quad (24)$$

其中

$$c_0 = \sup_{\|v\|_{A_\epsilon}=1} \inf_{P \underline{v} = v} \langle \underline{L} D_\epsilon^{-1} U \underline{v}, \underline{v} \rangle$$

現在，對任何 $v = (v_1, v_2, v_3)^T$ ，令

$$\underline{v} = -\frac{v_2+v_3}{2(1+\epsilon)}(1, 1, 1, -1)^T + (v_1, v_2, v_3, 0)^T$$

我們有 $P\underline{v} = v$ 與 $\langle D_\epsilon^{-1}U\underline{v}, U\underline{v} \rangle \leq \frac{1}{2(1+\epsilon)}\|v\|_{A_\epsilon}^2$ ，根據 24 導出 $\|I - BA_\epsilon\|_{A_\epsilon}^2 \leq \frac{1}{3}$ 。

如果應用改進的雅可比法到 6，也就是以迭代法 24 應用於 $\underline{B} = \underline{R}_\epsilon = \text{diag}(1, 1, 1, \epsilon^{-1})$ ，根據 21，我們直接計算得出

$$\begin{aligned} B_\epsilon &= \Pi B \Pi' = P B P^T = I + \epsilon^{-1} e e^T \\ B_\epsilon A_\epsilon &= A_0 + \epsilon I + e e^T \end{aligned}$$

因此 $\kappa(B_\epsilon A_\epsilon) \leq \frac{3+\epsilon}{1+\epsilon} \leq 3$ ，對任何 ϵ 都是一致的。

有限元方法及其離散方程組

我們將以一類的特殊問題：由有限元法來離散化的偏微分方程，展示更有效的迭代法如何被發展來解線性代數方程組。

有限元法例子

有限元法是科學計算中最常用的方法之一。本節我們討論三種常見的有限元離散化 (finite element discretization)。令 $\Omega \subset \mathbb{R}^3$ 是有界域， Ω 的邊界以 $\partial\Omega$ 表示。我們考慮以下列形式表述的三個不同的橢圓型問題

$$25 \quad (Du, Dv) + (u, v) = (f, v), \quad \forall v \in H(D; \Omega)$$

其中 D 可以是梯度 (gradient)、旋度 (curl)、或散度 (divergence) 算子， $H(D; \Omega)$ 的定義是

$$H(D; \Omega) = \{v : v \text{ 是平方可積且 } Dv \text{ 是平方可積}\}$$

有限元法從弱問題 25 出發，對 $H(D; \Omega)$ 施行有限維逼近：首先，對區域 Ω 施行滿足某種性質的網

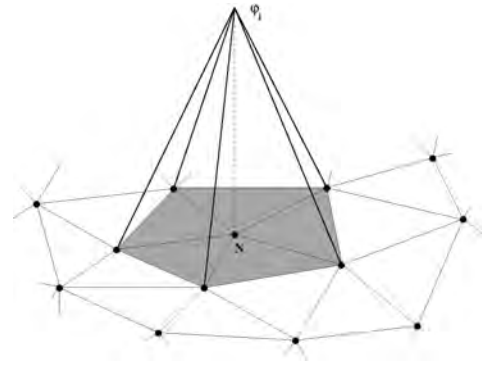


圖 1: 網格上的頂點 $\mathcal{N} \in \mathcal{V}$ 及其關聯的節點 / 拉格朗日 (node / Lagrange) 基函數 φ_i 。

格剖分 (grid decomposition)，記為 $\mathcal{T}_h = \{K\}$ ， h 表示網格的尺度 (grid scale)， K 是網格剖分 \mathcal{T}_h 中的一個單元 (element)；然後在每個單元內做某種局部逼近，最後需要注意的是，單元與單元之間的函數需要滿足某種連續性，而這個機制一般通過「自由度」(degree of freedom) 這個概念加以實現。比如以 $H(\text{grad}; \Omega)$ 的最低階有限元空間為例，即分段線性 (piecewise linear) 且整體連續的函數空間。這裡，單元內部是局部線性逼近，單元與單元之間自由度定義為節點 (node) 上的函數值。由於線性函數可以被節點值唯一確定，這個自由度的定義可以保證把局部函數「粘連」起來後整個函數是連續的。

下面我們介紹基函數 (basis function) φ_i ：它在第 i 個自由度上為 1，而在其它自由度上為 0。例如，對於分段線性連續函數空間，每個基函數看起來像一個帳篷 (見圖 1)。有了基函數，任何離散空間的函數可以表示為基函數的線性組合，即 $u_h = \sum_i c_i \varphi_i$ ，將測試函數 (test function) 取遍每個基函數，我們可以得到線性代數方程組 $Ac = b$ ，其中

$$26 \quad A = \left((\varphi_j, \varphi_i)_A \right), \quad b_i = (f, \varphi_i)$$

其中 $(u, v)_A = (Du, Dv) + (u, v)$ 。 A 稱為剛度矩陣 (stiffness matrix)， b 稱為負載 (load)。

$$\begin{array}{ccccccccccc}
\mathbb{R} & \longrightarrow & C^\infty(\Omega) & \xrightarrow{\text{grad}} & [C^\infty(\Omega)]^3 & \xrightarrow{\text{curl}} & [C^\infty(\Omega)]^3 & \xrightarrow{\text{div}} & C^\infty(\Omega) & \longrightarrow & 0 \\
& & \downarrow \Pi_h^{\text{grad}} & & \downarrow \Pi_h^{\text{curl}} & & \downarrow \Pi_h^{\text{div}} & & \downarrow \Pi_h^0 & & \\
\mathbb{R} & \longrightarrow & \mathcal{W}_h^0 & \xrightarrow{\text{grad}} & \mathcal{W}_h^1 & \xrightarrow{\text{curl}} & \mathcal{W}_h^2 & \xrightarrow{\text{div}} & \mathcal{W}_h^3 & \longrightarrow & 0
\end{array}$$

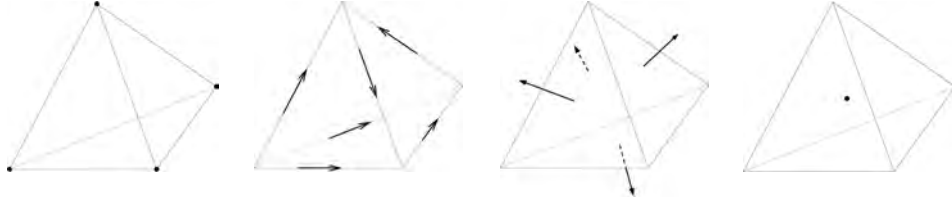


圖 2：連續與離散的德蘭姆序列（de Rham sequence）。

經過幾十年的發展，有限元方法已成功應用於彈性力學，流體力學，電磁學，材料科學，天文物理學和成像處理等領域。這些問題在數學理論和實際應用兩個方面都會帶來很多挑戰。一個自然的問題：什麼樣的離散是自然的？一般來說，一個好的離散方法應該符合物理和工程問題的底層結構。這個想法通常在各種情況下被稱為「保結構」。例如，在電磁現象的模擬中，電場自然的在一條路徑上被評估（整合）以給出電壓的改變；而對於擴散問題，在二維微元件上評估體積上的質量變化。在幾何學的語言中，這些物體被定義為微分 1- 形式和 2- 形式。為了在數值計算中模擬這些結構，很多研究工作考慮微分形式的離散，比如有限元外微積分（finite element exterior calculus）[4,1]。

具體來說，我們考慮 $H(\text{curl}; \Omega)$ 、 $H(\text{div}; \Omega)$ 的有限元空間，它們分別可以作為電場和磁場的離散空間。圖 2 給出了最低階的相應單元，其具體構造如下：

對有限元空間 $H(\text{curl}; \Omega)$ ，每個單元 K 上的形狀函數空間（shape function space）是六維的而且自由度是每個單元邊上的切向積分（tangential integral），即是

$$\begin{aligned}
\mathcal{P} &= \{\alpha + \beta \times x : \alpha, \beta \in \mathbb{R}^3\} \\
\mathcal{N} &= \left\{ \int_e v \cdot t : \text{每個邊 } E \subset K \right\}
\end{aligned}$$

對有限元空間 $H(\text{div}; \Omega)$ ，每個單元 K 上的形狀函

數是四維的而且自由度是每個單元面上的法向積分（normal integral），即是

$$\begin{aligned}
\mathcal{P} &= \{\alpha + \beta x : \alpha \in \mathbb{R}^3, \beta \in \mathbb{R}\} \\
\mathcal{N} &= \left\{ \iint_F v \cdot n : \text{每個面 } F \subset K \right\}
\end{aligned}$$

通常 $H(\text{curl}; \Omega)$ 和 $H(\text{div}; \Omega)$ 的有限元空間分別稱為內德萊克空間（Nédélec space）和拉維亞爾 / 托馬斯空間（Raviart-Thomas space），分別在剖分單元稜（edge）上保持切向連續和剖分單元面上保持法向連續。這些有限元空間構成了一個離散的德蘭姆複形（complex）（圖 2）：通過微分算子（grad，curl 和 div）將一個空間映射到另一個空間。通過這種方式，可以在三角剖分上建立一個離散的德蘭姆上同調理論（cohomology theory）。拓樸信息忠實的反映在離散化中，這就是為什麼得到的數值算法是穩定的和保結構的幾何解釋。

從廣義相對論所描述各種各樣的問題，包括電磁學到彈性，晶體中的缺陷和錯位，以及時空的漣漪，微分複形都表現出內在的聯繫。這些來自幾何和拓樸的見解和工具也激勵了許多重要的數值數學和科學計算方便的進展。這些保結構方法遵循在離散網格和圖上發展代數和幾何結構的趨勢。這些扎根於現代幾何的發展在工程應用中取得了巨大的成功。幾何，物理，工程和計算之間的相互作用正在發揮越來越重要的作用，而離散方程組的快速求解法（solver）也是一個重要的研究方向。

離散化後，因此我們考慮如何求解方程組

$$27 \quad Au = f$$

其中 $A \in \mathbb{R}^{n \times n}$ ， $f \in \mathbb{R}^n$ 。

高斯 / 賽德爾法與共軛梯度法在有限元方程的應用

我們現在簡要討論高斯 / 賽德爾法和共軛梯度法在使用有限元法離散化 25 後的線性代數方程組 27 的應用。可以證明（見 [15]）

$$28 \quad \kappa(A) \lesssim h^{-2}$$

如果 [15] 對某些常數 c_0 與 c_1 ，

$$29 \quad 1 - c_0 h^2 \leq \|I - BA\|_A^2 \leq 1 - c_1 h^2$$

成立。也就是說，在一般的情況下，隨著問題規模的增加，即網格尺度 h 越來越小時，收斂速度 $1 - ch^2$ 會趨近於 1，收斂會越來越慢。

自 1950 年代起，見 [17]，大量研究所謂的 SOR 法（比較 11 對應於 $B = (\omega^{-1}D + L)^{-1}$ 來提高高斯 / 賽德爾法的表現。在 D 是梯度算子並且有限元網格是一致的情況下，可以選擇最佳的 $\omega \in (1, 2)$ 使得

$$30 \quad \|I - BA\|_A^2 \leq 1 - \tilde{\alpha}_1 h$$

這是對 29 的重大改進。

由於 13 和 28，應用於 25 的共軛梯度法收斂性一般是 $1 - \mathcal{O}(h)$ 的收斂速度。當最佳 SOR 方法作為預條件子時，對應的預處理共軛梯度法會有一個改進到 $1 - \mathcal{O}(\sqrt{h})$ 的收斂速度。

無論如何，像許多其他迭代法一樣，高斯 / 賽德爾和共軛梯度法的收斂性隨著 h 變小而惡化。相較之下，接下來要介紹的多層網格法可以證明對 h 是一致收斂。

多層網格法

我們現在將討論解一些以有限元離散化的偏微分方程解的多層網格法。為簡單起見，我們可以考慮定義域為 $\Omega = (0, 1)^d$ ， $1 \leq d \leq 3$ 。

我們假設三角剖分 \mathcal{T}_h 是由逐次細分的過程所建構。更確切的說，對某個 $J > 1$ ， $\mathcal{T}_h = \mathcal{T}_J$ ，而 \mathcal{T}_k 對 $k \leq J$ 是個準均勻（quasi-uniform）三角剖分的套序列（nested sequence）。也就是說， \mathcal{T}_k 由大小為 h_k 的單體（simplex） $\mathcal{T}_k = \{K_k^i\}$ 組成，使得 $\Omega = \cup_i K_k^i$ ，這裡準均勻性常數與 k 無關。我們進一步假設有一個與 k 無關的常數 $\gamma < 1$ ，使得 \mathcal{T}_k 的網格大小（mesh size） h_k 與 γ^{2k} 成比例。舉例來說，在二維時，給定最粗糙的初始三角剖分 \mathcal{T}_0 ，通過連接較粗網格的三角形邊緣中點得到較細的網格，這個做法是準均勻的。在這個例子中 $\gamma = 1/\sqrt{2}$ 。

現在我們定義

$$H_0^1(\Omega) = \{v \in H(\text{grad}; \Omega) : v|_{\partial\Omega} = 0\}$$

以及

$$V_k = \{v \in H_0^1(\Omega) : v|_{K_k^i} \in P_1(K_k^i), \forall K_k^i \in \mathcal{T}_k\}$$

其中 $P_1(K_k^i)$ 是單元 K_k^i 上的線性函數。

則我們會有套空間

$$V_0 \subset V_1 \subset \cdots \subset V_J = V \subset H_0^1(\Omega)$$

多層網格法：高斯 / 賽德爾法與雅可比法在擴展矩陣上的應用

多層網格法可以從許多不同的面相切入。這裡我

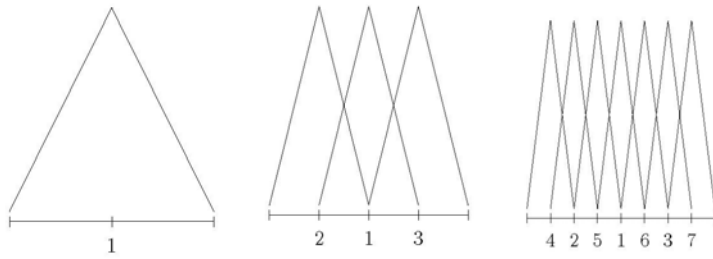


圖 3：一維多層基函數

們依循 [13] 和 [3]，從應用於只以半正定基架 (frame) (而不是基底) 表示的擴展剛度矩陣的高斯 / 賽德爾法或雅可比法，來介紹多層網格法。

以一維為例，我們考慮 $\Omega = (0, 1)$ 的三層均勻細分網格。不同層的有限元空間由圖 3 所示。如圖 3 所示，最末層有七個內點，而且由此網格上在 $x = 0, 1$ 歸零的連續分段線性函數組成的對應有限元空間，是由七個節點基函數 $\{\varphi_i^2, i = 1 : 7\}$ 組成。在通常的情況下，使用這七個基函數，我們會得到一個 7×7 大小的剛度矩陣。但是，這裡我們還要利用其他四個節點基函數，即 $\{\varphi_i^1, i = 1 : 3\}$ 和 $\{\varphi_1^0\}$ 來得到一個由 11 個線性獨立的函數組成所謂的基架 $\{\varphi_i^k, i = 1 : n_k, k = 0 : 2\}$ 。例如，不難看出

$$\varphi_1^0 = \frac{1}{2}\varphi_1^1 + \varphi_2^1 + \frac{1}{2}\varphi_3^1$$

這個想法可以直接推廣到更高的維度。亦即我們考慮的是擴展剛度矩陣

$$\mathbf{31} \quad \mathcal{A} = \left((\varphi_j^\ell, \varphi_i^k)_A \right) \in \mathbb{R}^{N \times N}, \quad N = \sum_{k=0}^J n_k$$

這裡 φ_i^k 為 V_k 上第 i 個基函數， n_k 是指 V_k 的維度。

結果是應用到 [26] 中的原始剛度矩陣 A 的高斯 / 賽德爾法的收斂速度遠慢於應用到 [31] 中的擴展剛度矩陣 \mathcal{A} 。這兩種不同方法的比較可以由經二維計算的表 2 中看出。

Size	G-S for A	G-S for \mathcal{A}	V + PCG for A
225	266/0.04s	13/0.004s	5/0.003s
961	910/0.26s	13/0.007s	5/0.005s
3,969	3,044/2.40s	13/0.021s	5/0.016s
16,129	9,869/31.45s	13/0.08s	5/0.06s
65,025	30,226/347.38s	13/0.30s	5/0.22s

表 2： $\|A x - b\|/\|b\| \leq 10^{-6}$ 的迭代次數。

如上所述，應用於擴展方程組的高斯 / 賽德爾法在數學上等價於典型的多重網格法在每一層以高斯 / 賽德爾法作柔化 (smoother)，見 [13]。

因為 $V_k \subset V_J$ ，存在 $\Pi_k \in \mathbb{R}^{n_J \times n_k}$ 使得

$$(\varphi_1^k, \dots, \varphi_{n_k}^k) = (\varphi_1^J, \dots, \varphi_{n_J}^J) \Pi_k$$

不難看出 [31] 中的擴展剛度矩陣 \mathcal{A} 與 [26] 中的原始剛度矩陣 A 的關係如下：

$$\mathcal{A} = \Pi^T A \Pi, \quad \Pi = (\Pi_0, \Pi_1, \dots, \Pi_J)$$

$$\text{亦即 } \mathcal{A} = (A_{ij})_{J \times J}, \quad \text{其中 } A_{ij} = \Pi_i^T A \Pi_j$$

令 D_k 是 A_{kk} 的對角矩陣。取

$$\mathcal{B} = \mathcal{D}^{-1} = \text{diag}(D_0^{-1}, D_1^{-1}, \dots, D_J^{-1})$$

從 [21] 的觀點，我們得到下列原始方程組的預條件子：

$$\mathbf{32} \quad B = \sum_{k=0}^J \Pi_k D_k^{-1} \Pi_k^T$$

即 BPX 預條件子，見 [12] 與 [2]。BPX 預條件子是二階橢圓形問題的高效簡便的預條件子，現已成為科學計算中最基本的算法之一。

等價範數與 BPX 預條件子

這裡我們給出 BPX 預條件子 [32] 的簡要理論分析。我們從以下範數的等價結果開始。

$$\mathbf{33} \quad \begin{aligned} \|v\|_1^2 &\approx \sum_{k=0}^J \|(Q_k - Q_{k-1})v\|_1^2 \\ &\approx \sum_{k=0}^J h_k^{-2} \|(Q_k - Q_{k-1})v\|_0^2, \quad \forall v \in V_J \end{aligned}$$

其中 Q_k 為 V_k 上的 L^2 投影，而且 Q_{-1} 為零算子。

[33] 的初步版本可見於 [12]，其證明可見於 [13]。

我們注意到 33 可寫成以下的等價形式：

$$\langle -\Delta v, v \rangle \approx \sum_{k=0}^J h_k^{-2} ((Q_k - Q_{k-1})v, v), \quad \forall v \in V_J$$

因此，在譜等價的意義上

$$-\Delta \approx X \equiv \sum_{k=0}^J h_k^{-2} (Q_k - Q_{k-1})$$

因為 $(Q_k - Q_{k-1})(Q_\ell - Q_{\ell-1}) = \delta_{k\ell}(Q_k - Q_{k-1})$ ，其中 $\delta_{k\ell} = 1$ 當 $k = \ell$ 而 $\delta_{k\ell} = 0$ 當 $k \neq \ell$ ，不難看出

$$X^{-s} \equiv \sum_{k=0}^J h_k^{2s} (Q_k - Q_{k-1}), \quad s \in [-1, 1]$$

特別是，

$$(-\Delta)^{-1} \approx X^{-1} \equiv \sum_{k=0}^J h_k^2 (Q_k - Q_{k-1})$$

因此

$$\begin{aligned} ((-\Delta)^{-1}f, f) &\approx (X^{-1}f, f) \\ &\equiv \sum_{k=0}^J h_k^2 ((Q_k - Q_{k-1})f, f) \\ &\approx \sum_{k=0}^J h_k^2 (Q_k f, f) \approx \sum_{k=0}^J (R_k Q_k f, f) \end{aligned} \quad 34$$

這裡 $R_k : V_k \mapsto V_k$ 是任意滿足

$$(R_k v_k, v_k) \approx h_k^2 (v_k, v_k), \quad \forall v_k \in V_k \quad 35$$

的線性算子。34 中的等價關係式導出 BPX 預條件子的算子形式如下：

$$B = \sum_{k=0}^J R_k Q_k \quad 36$$

如果我們選擇 R_k 使得

$$R_k v = \sum_{i=1}^{n_k} (\nabla \varphi_i^k, \nabla \varphi_i^k)^{-1} (v, \varphi_i^k) \varphi_i^k, \quad v \in V_k$$

然後可以證明 BPX 預條件子的矩陣形式 32 是 BPX

預條件子的算子形式 36 的矩陣表示。鑑於 33，我們看到 $\kappa(BA)$ 對網格參數 (mesh parameter) 是均勻有界的。

HX 預條件子

對於一般的模型問題 25，BPX 預條件子是應用於 $D = \text{gradient}$ 。對於 $D = \text{curl}$ 或 divergence ，希普特麥爾 (Ralf Hiptmair) 與筆者 [5] 給出了一系列最佳預條件子，稱為 HX 預條件子。HX 預條件子是電磁場方程最有效的快速求解方法之一。

我們先以 $D = \text{curl}$ 為例，注意到 $\text{curl}^* \text{curl}$ 算子有很大的零空間，德蘭姆序列的正合性 (exactness) 給出了零空間的精確的描述： gradient 的值域。因此，快速迭代法的重點是如何刻畫出微分算子的零空間。首先在連續函數空間上 $H(\text{curl}; \Omega)$ 有正則分解 $H(\text{curl}; \Omega) = (H(\text{grad}; \Omega))^3 + \text{grad}(\text{grad}; \Omega)$ 。在離散層面，根據圖 2，希普特麥爾與筆者給出了如下的離散正則分解：

$$\mathbf{E}_h = \tilde{\mathbf{E}}_h + \Pi_h^{\text{curl}} \Psi_h + \text{grad } \varphi_h \quad 37$$

並證明

$$\begin{aligned} h^{-1} \|\tilde{\mathbf{E}}_h\|_0 &\lesssim \|\mathbf{E}_h\|_{H(\text{curl})} \\ \|\Psi_h\|_1 &\lesssim \|\mathbf{E}_h\|_{H(\text{curl})} \\ \|\varphi_h\|_1 &\lesssim \|\mathbf{E}_h\|_{H(\text{curl})} \end{aligned}$$

也就是說任何內德萊克邊單元函數 $\mathbf{E}_h \in \mathcal{W}_h^1$ 可以被分解為一個高頻項 $\tilde{\mathbf{E}}_h \in \mathcal{W}_h^1$ ，一個向量型的拉格朗日單元 $\Psi_h \in (\mathcal{W}_h^0)^3$ 和一個拉格朗日單元的梯度 $\varphi_h \in \mathcal{W}_h^0$ 。其中，最後一項就是 curl 在內德萊克邊單元下的核空間。

根據 FASP 一般框架，考慮 $V_h = \mathcal{W}_h^1$ 的擴展

空間 $\mathcal{V}_h = \mathcal{W}_h^1 \times (\mathcal{W}_h^0)^3 \times \mathcal{W}_h^0$ ，而相應的 $\Pi = (\mathbf{I}, \Pi_h^{\text{curl}}, \text{grad})$ 。現在我們設

$$\underline{B}_h^{\text{curl}} = \text{diag}(S_h^{\text{curl}}, B_h^{\text{grad}}, B_h^{\text{grad}})$$

則根據 [21]，我們得到 HX 預條件子如下：

$$B_h^{\text{curl}} = S_h^{\text{curl}} + \Pi_h^{\text{curl}} B_h^{\text{grad}} (\Pi_h^{\text{curl}})^T + \text{grad } B_h^{\text{grad}} (\text{grad})^T$$

具體現實中，高頻部分可以由原始方程組的雅可比法或高斯 / 賽德爾法 S_h^{curl} 加以處理，而 $(\mathcal{W}_h^0)^3$ 和 \mathcal{W}_h^0 則需要調用 $H(\text{grad}; \Omega)$ 求解法 B_h^{grad} ，比如上文中介紹的 BPX 預條件子或者代數多層網格式 (AMG)。也就是說，三維 HX 預條件子一次迭代中需要調用四次布阿松 (Poisson) 求解法。我們在一般框架中給出了 HX 預條件子的有效性。因此， B_h^{curl} 是算子 $I + \text{curl}^* \text{curl}$ 的一個好的 (譜等價) 近似反矩陣。

同樣的想法可以推廣到 $D = \text{divergence}$ 以及一般的德蘭姆複形。例如在知道了如何對於 $D = \text{curl}$ 求解近似逆 (approximate inverse) 之後，對於 $D = \text{divergence}$ ，我們有

$$\begin{aligned} B_h^{\text{div}} &= S_h^{\text{div}} + \Pi_h^{\text{div}} B_h^{\text{curl}} (\Pi_h^{\text{div}})^T \\ &\quad + \text{curl } B_h^{\text{curl}} (\text{curl})^T \\ &= S_h^{\text{div}} + \Pi_h^{\text{div}} B_h^{\text{grad}} (\Pi_h^{\text{div}})^T \\ &\quad + \text{curl } S_h^{\text{curl}} (\text{curl})^T \\ &\quad + \text{curl } \Pi_h^{\text{curl}} B_h^{\text{grad}} (\Pi_h^{\text{curl}})^T (\text{curl})^T \end{aligned}$$

此時一次迭代需要調用六次布阿松求解法。

2008 年，美國能源部發布報告，介紹近年來應用數學和計算機科學領域中出現的十大突破性成就。HX 算法 [5] 作為其中唯一一個數值算法入選為十大突破性成就之一。該算法的有效性不太受問題規模和處理器負載的增長的影響，計算性能較傳統方

法有顯著改善。通過美國 LLNL 所做的大量的數值實驗證實，利用該新算法可以縮短計算時間一個到幾個數量級。新的算法可以處理幾何形狀比較複雜以及材料系數有大的間斷的問題。特別值得指出的是，此方法是一個有數學理論保證的適用於馬克斯威方程 (Maxwell's equation) 的適應於無結構網格式，可擴展的一般算法。

更複雜的應用例子

通過結合偏微分方程、函數和數值分析、以及上述 BPX 和 HX 預條件子的理論，我們可以為相當複雜的非線性偏微分方程組設計有效的離散化和預處理技術。像是流體結構相互作用 [16]、磁流體力學 (MHD) [6,7,9]、油藏數值模擬 [8,11]，所有這些應用都展示了使用 BPX 和 HX 預條件子作為基本組件的預處理技術的卓越性能。☺

本文出處

本文是許進超教授在 ICCM 思廉講座的講稿。本刊感謝許教授同意刊登。

本文參考資料請見〈數理人文資料網頁〉
<http://yaucenter.nctu.edu.tw/periodical.php>