

領域型メッシュレス法のための自動節点配置の高速化

齋藤歩研究室 20513490 荒木裕貴

2024 年 5 月 4 日

1 はじめに

自然現象及び工学的問題の多くが偏微分方程式の初期値・境界値問題で記述可能である。これらの問題は解析的に解くことはまず困難であるため、計算機によって数值的に説く必要がある。そこで有限要素法をはじめとした様々な数値解法が考案され、有限要素法を改良した解法である、対象となる領域に要素分割を行わない領域型メッシュレス法が開発されることになる。

領域型メッシュレス法は複数存在するが、そのどれもが領域上に配置された節点を使用し計算を行うため節点配置に計算結果が大きく依存する。そこで、対象領域に節点を一様分布させる自動配置法として PDS(Poisson Disk Sampling)を利用した配置法が考案された。しかし、本配置法は節点数の増加に伴い、節点配置時間が大幅に増加するという重大な欠点をもつ。それ故、より高速な自動節点配置法が望まれている。

そこで、本研究の目的として私は従来の PDS 法を用いた自動配置法を改良し、対象領域を分割した小領域内部で PDS 法を適用する自動節点配置法を提案する。また、従来法と提案法による節点配置の性能調査及び従来法と提案法による解の精度に及ぼす影響の調査を行う。

2 Collocation EFG Method

本章では問題設定と使用する数値解法について述べる。また本章中に現れる準備データは Table2.1 にまとめる。

2.1 2D Poisson 問題

本研究では 2 次元 Poisson 問題:

$$(*)1) \begin{cases} -\Delta u = p & \text{in } \Omega \\ u = \bar{u} & \text{on } \Gamma_D \\ q = \bar{q} & \text{on } \Gamma_N \end{cases} \quad (2.1)$$

を対象とする。但し、 p は単一閉曲線 $\partial\Omega (= \Gamma_D \cup \Gamma_N \text{ and } \Gamma_D \cap \Gamma_N = \emptyset)$ により囲まれた対象領域 Ω (Fig.2.1 参照) 内で定義された既知関数であり、 \bar{u} は Dirichlet 境界条件 Γ_D 上の既知関数、 \bar{q} は Neumann 境界条件 Γ_N 上の既知関数を表す。また、 \mathbf{n} は $\partial\Omega$ 上の外向き単位法線ベクトルであり、法線微係数 q は

$$q \equiv \frac{\partial u}{\partial n} \quad (2.2)$$

と表される。(2.1), (2.3) に対して任意の重み係数 w の付加により、(*)1 と等価になる連立方程式:

$$(*)2) \begin{cases} -\Delta u = p & \text{in } \Omega \\ q = \bar{q} & \text{on } \Gamma_N \end{cases} \quad (2.3)$$

を導出する。但し、 \mathbf{x} は領域 Ω 上の任意の点であり、 l は領域境界 $\partial\Omega$ 上の始点から領域境界 $\partial\Omega$ 上の \mathbf{x} までの線分である。(*)2 から直接弱形式を導く。

(2.5) の方程式に対して恒等式:

$w\Delta u = \nabla \cdot (w\nabla u) - \nabla w \cdot \nabla u$ と Green の定理より、

$$\iint_{\Omega} \nabla w \cdot \nabla u d^2\mathbf{x} = \iint_{\Omega} w f d^2\mathbf{x} + \int_{\Gamma_N} w \bar{q} dl \quad (2.4)$$

の弱形式が得られる [1]。 (2.6) より、(*)2 と等価な

$$(*)3 \left\{ \begin{array}{l} \forall \mathbf{w}, \mathbf{t}, \mathbf{w}|_{\Gamma_N} = 0 : \\ \iint_{\Omega} \nabla \mathbf{w} \cdot \nabla u d^2 \mathbf{x} = \iint_{\Omega} \mathbf{w} f d^2 \mathbf{x} + \int_{\Gamma_N} \mathbf{w} \bar{q} dl \\ q = \bar{q} \quad \text{on } \Gamma_D \end{array} \right.$$

が得られる。次節ではこの連立方程式 (※3) に対して離散化を行う。

2.2 離散化

離散化を行う前に、本数値解法で使用する MLS 近似の形状関数を導く。Fig.2.2 のように領域 Ω 内及び領域境界 $\partial\Omega$ 上に N 個の節点 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ を配置し、節点上で関数 $u(\mathbf{x})$ の値が u_1, u_2, \dots, u_N になるとする。 N 個の関数値を近似する関数を、

$$u^*(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) \quad (2.5)$$

と仮定する。但し、 $\mathbf{p}^T = [1, x, y]$ であり、 $\mathbf{a}(\mathbf{x})$ は、

$$J[\mathbf{a}] = \sum_{i=1}^N w_i(\mathbf{x}) [\mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) - u_i]^2$$

を最小にするように決定される。すなわち、

$$\mathbf{a}(\mathbf{x}) = \sum_{i=1}^N A^{-1}(\mathbf{x}) \mathbf{b}_i(\mathbf{x}) u_i \quad (2.6)$$

で与えられる。但し、3 次正方行列 $A(\mathbf{x})$ 、3 次元ベクトル $\mathbf{b}_i(\mathbf{x})$ 、 N 次元ベクトル \mathbf{u} はそれぞれ、

$$A(\mathbf{x}) = \sum_{i=1}^N w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}), \quad (2.7)$$

$$\mathbf{b}_i(\mathbf{x}) = w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i), \quad (2.8)$$

$$\mathbf{u} = [u_1, u_2, \dots, u_N]^T$$

である [3]。本研究では重み関数として、

$$w_i(\mathbf{x}) = \omega(|\mathbf{x} - \mathbf{x}_i|), \quad (2.9)$$

$$\omega(r) = \begin{cases} \frac{e^{-(r/C)^2} - e^{-(R/C)^2}}{1 - e^{-(R/C)^2}} & (r \leq R) \\ 0 & (r > R) \end{cases}$$

を採用する。但し、 R はサポート半径、 C は重み係数を表す。(2.9)、(2.10) に (2.11) を代入すると近似関数 $u^*(\mathbf{x})$ は、

$$u^*(\mathbf{x}) = \sum_{i=1}^N \phi_i(\mathbf{x}) u_i$$

となる。但し、 $\phi_i(\mathbf{x})$ は

$$\phi_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) A^{-1}(\mathbf{x}) \mathbf{b}_i(\mathbf{x})$$

で定義される形状関数を示す。

続いて、連立 1 次方程式を導出する。境界節点数を M として空間 V 、 W を N 次元空間、 M 次元空間と定義し、 $\{\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \dots, \psi_N(\mathbf{x})\}$ を空間 V の線形独立関数とする。

$$w(\mathbf{x}) = \sum_{i=1}^N \hat{w}_i \psi_i, \quad (2.10)$$

$$u(\mathbf{x}) = \sum_{i=1}^N \hat{u}_i \phi_i \quad (2.11)$$

と仮定すると、(2.6) から

$$(\hat{\mathbf{w}}, A\hat{\mathbf{u}} - \mathbf{f}) = 0 \quad (2.12)$$

を導出できる。但し、 $\hat{\mathbf{w}}$ 、 $\hat{\mathbf{u}}$ は定数であり、

$$\hat{\mathbf{w}} = \sum_{i=1}^N \hat{w}_i \mathbf{e}_i$$

$$\hat{\mathbf{u}} = \sum_{i=1}^N \hat{u}_i \mathbf{e}_i$$

である。また、

2.3 積分セル分割による数値積分

3 自動節点配置法

3.1 従来法

3.2 提案法

4 数値実験

5 結論