

一般的固有値問題のためのアルゴリズム

日大 理工 戸川 隼人

問題1 与えられた正定値対称行列 M, C, K に対し

$$(\lambda^2 M + \lambda C + K) u = 0$$

を満たす固有値 λ , 固有ベクトル u を求める.問題2 $(\lambda^2 M + K) u = 0$ についての同様な問題.(1) $C = 0$ の場合に相当).

これらの問題に対し, 各種の解法が提案されているが, ここでは, Householder 変換により, 上記の M, C, K が, 5重対角化 (問題2の場合) または 7重対角化 (問題1の場合) できることを示す. このような前処理をしておけば, Gupta の方法^{注1)} などの計算時間は大幅に短縮できる. また QR 法に接続することも考えられる.

<記号> ^{細字体の}小文字 は スカラー, ^{太字体の}小文字 は 列ベクトル, T は 転置記号

大文字 は 行列 ($n \times n$)

注1 K.K. Gupta; Recent Advances in Numerical Analysis of Structural Eigenvalue Problems. in "Theory and Practice in Finite Element Structural Analysis" Univ. Tokyo Press

問題 2 の 5 重対角化の具体的な手順 ($k=1, 2, \dots, n-3$ の順に)

$$S = \sqrt{\sum_{j=k+1}^n m_{kj}^2}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \text{但し } x_i = \begin{cases} 0 & i \leq k \\ m_{ik} - S & i = k+1 \\ m_{ik} & i \geq k+2 \end{cases}$$

$$C_1 = x^T x$$

$$C_2 = 2 / C_1$$

$$w = C_2 M x$$

$$C_3 = (1 / C_1) x^T w$$

$$v = w - C_3 x$$

$$\text{新 } M = M - x v^T - v x^T$$

$$w = C_2 K x$$

$$C_3 = (1 / C_1) x^T w$$

$$v = w - C_3 x$$

$$\text{新 } K = K - x v^T - v x^T$$

$$g = C_2 T x$$

$$\text{新 } T = T - g x^T$$

$$S = \sqrt{\sum_{j=k+2}^n k_{kj}^2}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \text{但し } x_i = \begin{cases} 0 & (i \leq k+1) \\ k_{ik} - S & (i = k+2) \\ k_{ik} & (i \geq k+3) \end{cases}$$

$$c_1 = x^T x \quad c_2 = 2 / c_1$$

$$w = c_2 M x$$

$$c_3 = (1 / c_1) x^T w$$

$$v = w - c_3 x$$

$$\text{新 } M = M - x v^T - v x^T$$

$$w = c_2 K x$$

$$c_3 = (1 / c_1) x^T w$$

$$v = w - c_3 x$$

$$\text{新 } K = K - x v^T - v x^T$$

$$g = c_2 T x$$

$$\text{新 } T = T - g x^T$$

—— 別票 (Givens 流の変換法) ——

$$\left[\begin{array}{l} l = 1, 2, \dots, n-3 \text{ の順に} \\ \left[\begin{array}{l} k = l+3, l+4, \dots, n \text{ の順に} \\ \text{新 } M = P_{kl} M P_{kl}^T \\ \text{新 } K = P_{kl} K P_{kl}^T \end{array} \right. \end{array} \right.$$

但し

$$P_{kl} = \left[\begin{array}{cc|cc} I & & & 0 \\ & & & \\ \hline & & 1 & \\ 0 & & & \ddots \\ & & 0 & \alpha_k \beta_k \\ & & & & \ddots \\ & & & & & 1 \end{array} \right] \begin{array}{l} \leftarrow \text{ } l \text{ 行} \\ \leftarrow \text{ } k \text{ 行} \end{array}$$

ただし α_k, β_k は次式によって求める.

$$\Delta = \begin{vmatrix} m_{l+1,l} & m_{l+2,l} \\ k_{l+1,l} & k_{l+2,l} \end{vmatrix}$$

$$\Delta_1 = \begin{vmatrix} -m_{kl} & m_{l+2,l} \\ -k_{kl} & k_{l+2,l} \end{vmatrix}$$

$$\Delta_2 = \begin{vmatrix} m_{l+1,l} & -m_{kl} \\ k_{l+1,l} & -k_{kl} \end{vmatrix}$$

$$\alpha = \Delta_1 / \Delta$$

$$\beta = \Delta_2 / \Delta$$

古題1の7重対角化も同じ要領でできる.

(おわびと訂正) 研究集会当日, 算法 I として発表した方法には誤りがありました. つつしんで おわび申しあげ, 取り下げさせていただきます.