

大容量固有値解析の一手法

伊藤哲次

(大崎研究室)

§ 1. はじめに

サブスペース反復法 (Subspace Iteration Method) は、大自由度の固有値解析、特にマトリクスバンド幅が大きい場合には、非常に有効な計算法と考えられている^{1)~3)}。さらに、原点をシフト (origin-shift) させる過程を計算に加えることにより、解の収束度を高めると同時に、計算に用いられる試行ベクトル (trial-vectors) の数もバンド幅の大きさに応じて変更することができる。それゆえ、計算スピードが改善され、かつ計算に必要なコンピュータのコア容量も少なくてすむため、現在最も有効な固有値解析手法となっている。

本報告書では、原点シフト過程を含めたサブスペース法の計算方法を述べると共に、解の収束率を高め、計算スピードを速めるための最適な試行ベクトルの数、原点シフトの位置、各原点シフトごとの反復計算の回数を推定する方法を述べる。

§ 2. 原点シフト付きサブスペース反復法

一般に、構造解析を対象とした無減衰の自由振動方程式より導かれる固有値解析用の式は次のように書ける。

$$K\Phi = M\Phi\Lambda \quad \cdots(1)$$

ここで、

K : $n \times n$ 対称剛性マトリクス

M : $n \times n$ 対称質量マトリクス

Λ : $m \times m$ 対角マトリクス (対角項は m 個の固有値)

Φ : $n \times m$ マトリクス (m 個の固有ベクトル)

n : 自由度数

2.1 サブスペース反復法

サブスペース反復法による固有値解析手順を述べる。式(1)に逆反復法 (Inverse Iteration) を適用すると、

$$KU_j = M\bar{U}_{j-1}A_{j-1} \quad \cdots(2)$$

ここで、

U_j : (j) 反復計算時の試行ベクトル ($n \times m$)

$\bar{U}_{j-1}$: ($j-1$) 反復計算時の直交化された試行ベクトル ($n \times m$)

A_{j-1} : ($j-1$) 反復計算時の近似固有値から成る対角マトリクス ($m \times m$)

m : 試行ベクトル数 (またはサブスペースの大きさ)

すなわち、($j-1$) 反復計算時の $\bar{U}_{j-1}$, A_{j-1} を用いて、(j) 時の U_j を求めることで、次の2つの計算過程に分類される。

$$P_j = M\bar{U}_{j-1} \quad \cdots(3)$$

$$KU_j = P_j \quad \cdots(4)$$

ここで、 A_{j-1} を式(3)より省略したのは、 A_{j-1} の対角項がベクトルに乗る係数にすぎないためである。

得られた U_j を用いて、

$$K_j^* = U_j^T K U_j \quad \cdots(5)$$

$$M_j^* = U_j^T M U_j \quad \cdots(6)$$

を計算し、次式よりサブスペースにおける固有値解析を行なう。

$$K_j^* X_j = M_j^* X_j \Lambda_j \quad \cdots(7)$$

式(7)の固有値解析は、普通 m を比較的小さな値に採るため、容易に計算できる。

さらに、式(7)より求められた固有ベクトル X_j を用いて、 U_j を次式より直交性のある試行ベクトルとする。

$$\bar{U}_j = U_j X_j \quad \cdots(8)$$

以上、式(2)から(8)までの過程が、サブスペース反復法の計算手順である。反復計算を重ねるうちに、 m 個の固有値ならびに固有ベクトルが小さな固有値から順次収束する。

2.2 原点シフト

ここでは、サブスペース反復法に原点シフトを付け加えてみよう。

サブスペース反復法による試行ベクトルの収束率 r_k

は,

$$r_k = \left| \frac{\lambda_k}{\lambda_{m+1}} \right| \quad \dots\dots(9)$$

$$k=1, 2, \dots, m$$

ここで, λ_k (固有値) は次のように仮定する.

$$|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_m| \leq |\lambda_{m+1}|$$

それゆえ, 固有値の大きさの違いにより収束率も大きく異なる. すなわち, 試行ベクトルの内で, 小さな固有値 ($|\lambda_1|$) に対応するベクトルは数回の反復計算で収束するが, 大きな固有値 ($|\lambda_m|$) に対応するベクトルは相当数の反復計算を行わないう限り収束しないということである. このように収束率に違いがあることは, 効率的な計算の観点からみれば好ましい現象とはいえない.

ところで, いま原点をある値 μ にシフトすれば, 式(9)の収束率は,

$$r_k = \left| \frac{\lambda_k - \mu}{\lambda_{m+1} - \mu} \right| \quad \dots\dots(10)$$

のように表わされる. それゆえ, 図-1 に示すように, もし原点を $\mu = (\lambda_1 + \lambda_m)/2$ の点にシフトさせると, 全体的な収束率は大きく向上する. このように原点シフトを導入することは, 試行ベクトル内の各ベクトルの収束率を色々変化させることができ, 計算上非常に有効な手段となる.

2.3 LDL^T 分解と Sturm sequence チェック

いま, 原点シフトが計算過程で何度か繰り返し行なわれると仮定し, ある原点シフト (μ_i) 後の新しい座標軸における固有値を $\bar{\lambda}_k$ とすると,

$$\bar{\lambda}_k = \lambda_k - \mu_i \quad \dots\dots(11)$$

また, 式(1)も,

$$\bar{K}_i \Phi = M \Phi \bar{A} \quad \dots\dots(12)$$

ここで,

$$\bar{K}_i = K - \mu_i M \quad \dots\dots(13)$$

のように変形され, 式(12)を解くことになる. そして, 式(4)も次式のように書き改まる.

$$\bar{K}_i U_j = P_j \quad \dots\dots(14)$$

式(14)より U_j を求める場合, $\bar{K}_i$ を予め原点シフト後に LDL^T 分解しておく,

$$\bar{K}_i = L_i D_i L_i^T \quad \dots\dots(15)$$

ここで,

L_i : 対角項が1の下三角マトリクス

D_i : 対角マトリクス

各反復計算時には, 前進消去過程 ($L_i Z_j = P_j$) と後退代入過程 ($D_i L_i^T U_j = Z_j$) のみの計算をすれば良い.

ところで, 式(15)の LDL^T 分解より得られる D_i の対

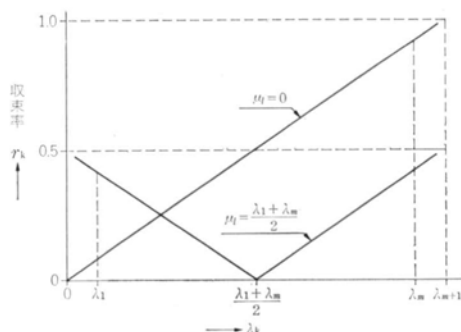


図-1 固有値の収束率

角項は, 貴重な情報を携えている. すなわち, 対角項がマイナス値となる項数と, μ_i より小さい固有値の数が等しくなるという Sturm sequence 性である.

それゆえ, 計算過程で得られた固有値のチェック (抜けた固有値はないか, あるいは余計な誤った固有値を求めているか) ができ, 非常に便利である.

2.4 Gram-Schmidt の直交性

式(10)で示すように, 試行ベクトルの収束率は均等でなく, その内のある固有ベクトルは速やかに収束するが, 別のものは緩やかに収束するという現象が生ずる. 収束した固有ベクトルを試行ベクトルから消去することにより, 新たな未収束の固有ベクトルを試行ベクトル内に含むことができる. それゆえ, 順次収束した固有ベクトル成分を試行ベクトルから消去することにより, 計算効率が大幅に向上する.

Gram-Schmidt の直交性を適用することにより, この計算処理は容易に行なうことができる. すなわち, ある反復計算後の試行ベクトル $\bar{U}_j$ に対して, それまでに収束した固有ベクトル群を Φ_j とし,

$$\Phi_j = [\phi_1, \phi_2, \dots, \phi_p] \quad \dots\dots(16)$$

$\bar{U}_j$ より Φ_j 成分を削除する式は,

$$\bar{U}_j = \bar{U}_j - \Phi_j B_j \quad \dots\dots(17)$$

ここで,

$$B_j = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pm} \end{bmatrix} = \Phi_j^T M \bar{U}_j \quad \dots\dots(18)$$

$$\Phi_j^T M \Phi_j = I \quad (\text{単位マトリクス}) \quad \dots\dots(19)$$

と表わされる.

このようにして, 収束した固有ベクトル成分を試行ベクトルより消去することは可能であるが, 実際の計算では計算機の丸め誤差により, 完全に消去することはできない. また, 原点近傍ですでに収束したような固有ベク

トルについてはその収束率が非常に良いため、丸め誤差で残された成分がすぐに成長する。それゆえ、Gram-Schmidtの直交性を利用した消去作業は反復計算の度に行なう必要がある。なお、消去作業に必要な計算ステップ数の全体の計算に占める割合は、わずかである。

2.5 収束チェック

試行ベクトルが収束したか否かを判定するため、以下のチェックを行なう。

$$e_k = \left| \frac{\bar{\lambda}_k^j - \bar{\lambda}_k^{j-1}}{\bar{\lambda}_k^j} \right| < e_{tol} \quad \dots\dots (20)$$

ここで、

$\bar{\lambda}_k^j$: (j) 反復計算後の k 試行ベクトルの近似固有値

e_{tol} : 収束検定用許容値 (たとえば, 10^{-6})

e_k : k 試行ベクトルの誤差

2.6 新しい原点の求め方

ところで、2.2ですでに述べたように、原点をシフトさせることにより試行ベクトルの収束率は全体的に向上する。ただし、新しい原点をどこへシフトさせたら良いかを定めることは容易なことでない。以下に、1つの方法を述べる。

はじめに、次のような計算状態を設定する (図-2 参照)。

- (i) 試行ベクトル数を m 個とする。
- (ii) $(i-1)$ 個の固有ベクトルは、すでに収束済み。
- (iii) 原点シフトを行なう前の反復計算時に、さらに q 個の固有ベクトルが収束した。

このような状態では、 $(m-q)$ 個の試行ベクトルに対応する近似固有値 $\bar{\lambda}_k (k=i+q \sim i+m-1)$ より、新しい原点の位置を推定することになる。ところで、これらの内で最大の固有値 $\bar{\lambda}_{i+m-1}$ は当然収束率が悪いことが予想されるため、値の信頼性も薄い。それゆえ、 $\bar{\lambda}_{i+m-1}$ は以下の計算には用いない。

試行ベクトルの収束率については式(10)で示されているが、対応する固有値の収束率は Rayleigh 商に相当することから、

$$R_k = \left(\frac{\lambda_k - \mu_l}{\lambda_{i+m-2} - \mu_l} \right)^2 \quad \dots\dots (21)$$

で表わされる。しかるに、実際の計算では真の固有値 λ_k は不明であるため、式(21)に代わって次式の収束率を用いることにする。

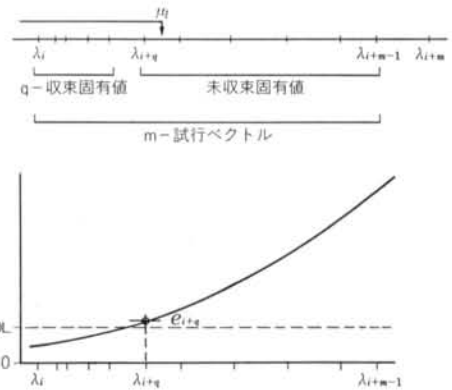


図-2 原点シフト

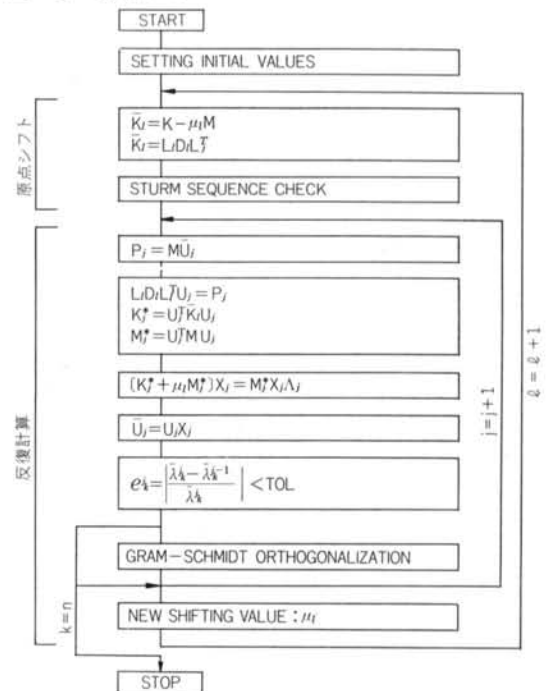


図-3 原点シフト付サブスペース反復法の計算フローチャート

$$\bar{R}_k = \left(\frac{\bar{\lambda}_k - \mu_l}{\bar{\lambda}_{i+m-2} - \mu_l} \right)^2 \quad \dots\dots (22)$$

そこでいま、新しい原点にシフト後、未だ収束していない最小の固有値 $\bar{\lambda}_{i+q}$ が S 回の反復計算中に収束すると仮定すると、

$$\left(\frac{\bar{\lambda}_{i+q} - \mu_l}{\bar{\lambda}_{i+m-2} - \mu_l} \right)^{2S} < \frac{e_{tol}}{e_{i+q}} \quad \dots\dots (23)$$

の条件を満足するように新しい原点 μ_l が定められる。

式(23)より μ_l を求めると、

$$\mu_l < \frac{\bar{\lambda}_{i+q} + \theta \bar{\lambda}_{i+m-2}}{1 + \theta} \quad \dots\dots (24)$$

ここで,

$$\theta = \left(\frac{e_{tol}}{e_{i+q}} \right)^{\frac{1}{2S}} \quad \dots\dots(25)$$

μ_i はできるだけ大きな値が望ましいから、式(24)の右辺の値をとる。一方、 θ を求める式(25)に代わって、よりコンサーバティブな次式を用いた方が計算ロジックとして好ましいことが、幾通りかの試算で判明した。

$$\theta = \left(\frac{e_{tol}}{e_{i+q}} \right)^{\frac{1}{2S-1}} \quad \dots\dots(26)$$

以上、式(24)、(26)より μ_i は求まるが、試行ベクトル数 m が小さく ($m=3\sim 5$)、かつ同時に収束する固有ベクトルが複数 ($q=1\sim 3$) の場合、 $\bar{\lambda}_{i+m-2}$ の代わりに $\bar{\lambda}_{i+m-1}$ を式(24)で用いざるを得ないこともある。

なお、図-3に原点シフト付サブスペース反復法のフローチャートを示す。

§ 3. 最適な試行ベクトル数および原点シフト回数

前章では、原点シフト付サブスペース反復法について説明をした。しかしながら、与えられた問題（自由度の大きさ、バンド幅の大きさ）に対して、どれ位の試行ベクトル数を探ったら良いか、あるいは原点シフトをどの程度の頻度で行なったら良いかは明らかでない。これらの判断を適切に行なわないと、折角の解析手法もそのメリットを十分に発揮できなくなる。それゆえ、この章では与えられた問題に対して最適な試行ベクトル数、および原点シフトごとに行なう反復計算の最適回数をシミュレーション解析より導き出すことにしよう。

3.1 シミュレーション解析手法

ここで行なうシミュレーション解析とは、実際の固有値解析に代わって、いま固有値の分布状態を仮定し、それに対して前章で述べた計算アルゴリズムを適用し、固有値が求められる過程を追跡することである。実際の計算過程では真の固有値は求められず、近似値しか用いられないことを考慮して、シミュレーション解析では固有値の近似値を次のように仮定した。

$$\bar{\lambda}_k^j = \lambda_k (1 + \varepsilon_k^j) \quad \dots\dots(27)$$

$$\varepsilon_k^j = \varepsilon_k^{j-1} \left(\frac{\lambda_k - \mu_i}{\bar{\lambda}_{i+m} - \mu_i} \right)^2 \quad \dots\dots(28)$$

ここで、

λ_k : 真の固有値

ε_k^j : (j) 反復計算時の収束誤差で、初期値は $\varepsilon_k^0 = 1$

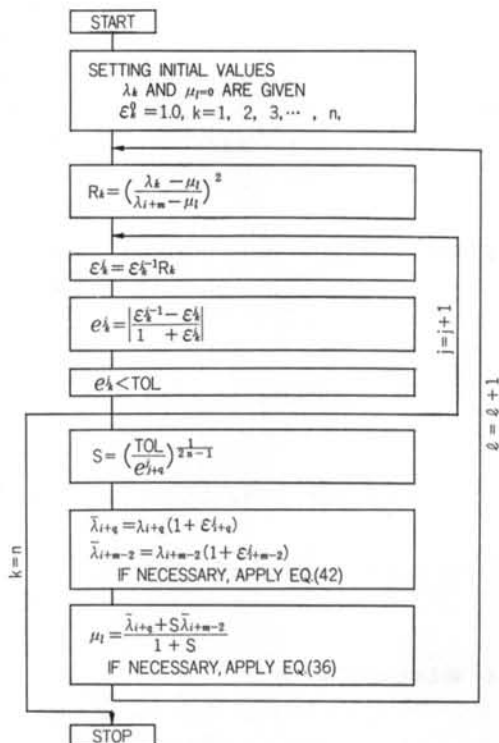


図-4 シミュレーション解析フローチャートと仮定する

なお、図-4にシミュレーション解析のフローチャートを示す。

ところで、このようなシミュレーション解析が実際の固有値解析を十分に模擬できるか否かを調べるため、両解析より得られた結果の比較を次のようにして行なった。固有値問題は図-5に示す簡単なフレームとし、実際の固有値の分布は図-6中の白丸に示すとおりである。一方、シミュレーション解析では固有値の分布を図-6中の黒丸に示すように直線分布とした。両解析を用いて、20個の固有値を求めるために必要な原点シフト回数および反復計算回数、さらに全オペレーション数を求め、表-1にそれらの結果を示した。この表では、試行ベクトルを幾つか変えた場合についての結果が比較されている。この表から分かるように、両解析結果は大変良く合っており、シミュレーション解析が実際の解析を模擬できることを裏付けている。

3.2 計算結果

ここでは、シミュレーション解析を用いて最適な試行ベクトル数、および原点シフトごとの最適反復計算回数を求めるため、以下に示す項目をパラメータに採って計算を行なった。

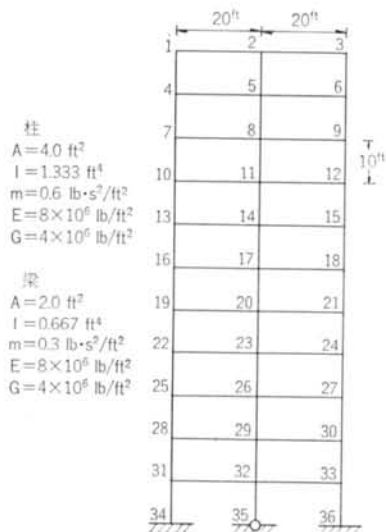


図-5 2次元フレーム

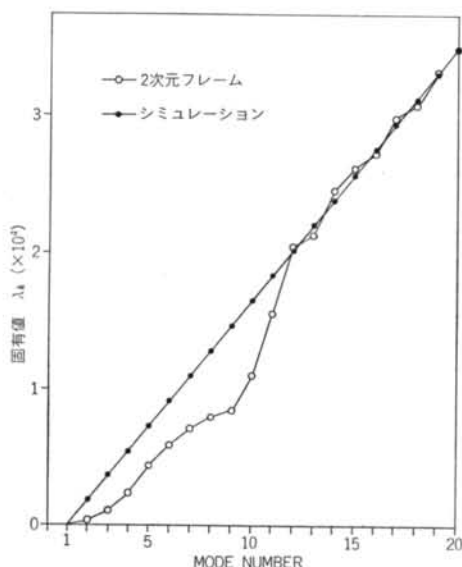


図-6 固有値の分布

①剛性マトリクスの平均バンド幅注1): NB

$NB=10, 20, 30, 40, 60, 100, 200, 400, 600, 1000$

②試行ベクトル数: NV

$NV=3, 4, 5, 6, 8, 10, 12, 15, 20, 25, 30, 40$

③原点シフトごとの反復計算回数: NI

$NI=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 20, 25, 30, 40, \infty$

ところで、この計算を行なう場合、固有値の分布状態を仮定しなければならない。実際の問題では固有値の分

注1) 平均バンド幅とは、 $K=LDL^T$ に分解後の L マトリクスの平均 non-zero バンド幅。

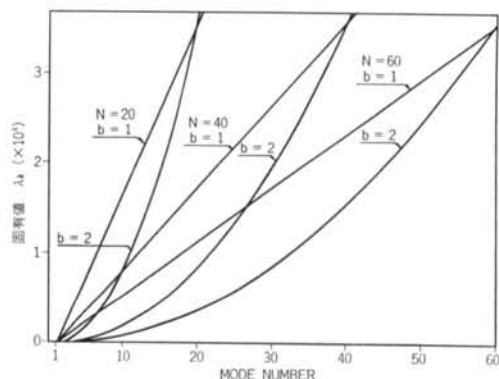


図-7 固有値の分布状態の仮定

試行ベクトル数	2次元フレーム解析		シミュレーション解析	
	シフト/反復回数/計算数	オペレーション数*	シフト/反復回数/計算数	オペレーション数*
$NV=3$	22/41	76×10^5	24/42	76×10^5
"=6	29/31	128×10^5	26/30	122×10^5
"=10	20/20	158×10^5	18/19	150×10^5
"=15	14/14	208×10^5	15/16	198×10^5

*オペレーション数は、20個の固有値・固有ベクトルを求めるために必要なもの

表-1 実解析(2次元フレーム)およびシミュレーション解析結果の比較(自由度=100)

布は複雑で、幾つかの固有値が密集していたり、またその反対で点在していたりもする。そして、局部的には固有値の分布状態は、解の収束率に密接に関係する。しかし、全体的な解の収束率から見れば、固有値の分布状態はそれほど重要な要因とはならないであろう。ここでは、固有値の分布状態を図-7に示すように、直線上に変化する場合($b=1$)と、放物線上に変化する場合($b=2$)とした。

また、計算結果は1つの固有値を求めるのに必要なオペレーション(乗除算)数 NOP を次式より求め、

$$NOP = \frac{NOS \times NOPS + NOI \times NOI}{N} \quad \dots\dots (29)$$

ここで、

N : 求めた全固有値数

NOS : 原点シフト回数

NOI : 反復計算回数

$NOPS$: 1回の原点シフトに要するオペレーション数

$NOPI$: 1回の反復計算に要するオペレーション数(表-2参照)

NOP の大小比較よりパラメータの最適値を見出すことにした。

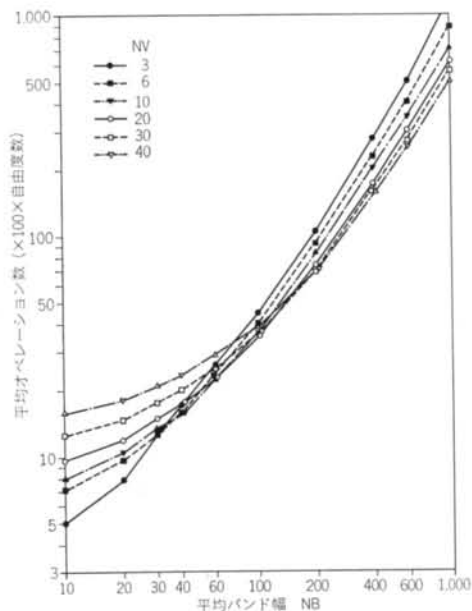


図-8(1) 1つの固有値を求めるのに要するオペレーション数 ($N=40, b=1$)

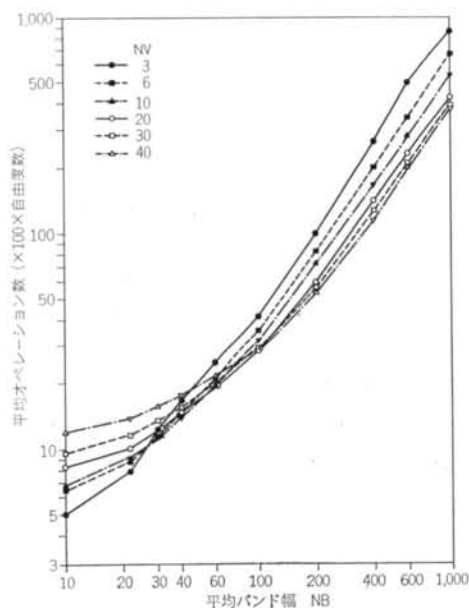


図-8(2) 1つの固有値を求めるのに要するオペレーション数 ($N=40, b=2$)

1) 最適試行ベクトル数

計算結果より、以下の結論が得られた。なお、計算結果例を図-8(1)、図-8(2)に示す。

(1) 平均バンド幅が20以下の場合、試行ベクトル数は3が最適である。

計 算 式	オペレーション数
$\bar{K}_i = K - \mu_i M$ $\bar{K}_i = L_i D_i L_i^T$	n $\frac{n}{2}(m^2 + m)$
$P_j = M \bar{U}_j$ $L_i D_i L_i^T U_j = P_j$ $K_j^* = U_j^T \bar{K}_i U_j$ $M_j^* = U_j^T M U_j$ $\bar{U}_{j+1} = U_j X_j$ $\bar{U}_{j+1} = \bar{U}_{j+1} - \sum_{i=1}^L \beta_{ij} \Phi_i$	np $2nmp$ $\frac{n}{2}p(p+3)$ $\frac{n}{2}p(p+3)$ np^2 $\frac{n}{2}N(2p+1)$

n : 自由度数
 m : 平均バンド幅 (NB)
 p : 試行ベクトル数 (NV)
 N : 求める固有値・固有ベクトル数

表-2 各計算に要するオペレーション数

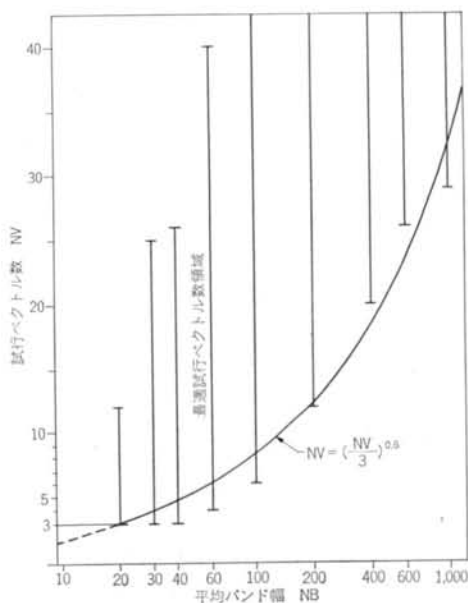


図-9 最適な試行ベクトル数領域

(2) 平均バンド幅が600以上の場合、試行ベクトル数は30～40が適当であろう。

(3) 計算結果をまとめ、平均バンド幅に対する最適な試行ベクトル数領域をプロットした結果(図-9)より、次に示すような最適試行ベクトル数を求めるための略算式を導びいた。

$$NV = \left(\frac{NB}{3} \right)^{0.6} \quad \dots\dots(3)$$

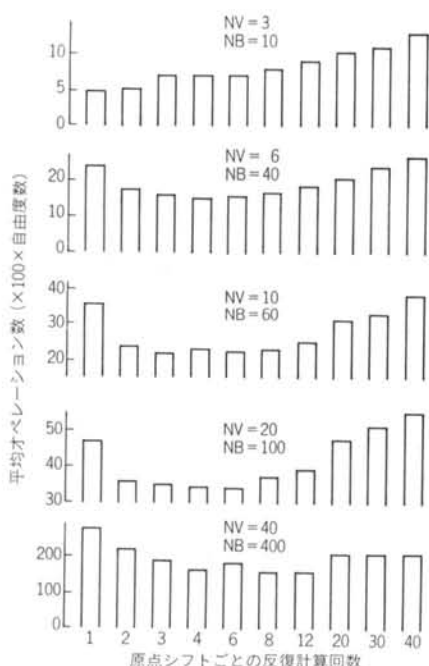


図-10(1) オペレーション数と反復計算回数 ($N=40$, $b=1$)
ただし, $NV \geq 3$

2) 原点シフトごとの最適反復計算回数

最適反復計算回数を求めることは容易でない。図-10(1), 図-10(2)にオペレーション数と反復計算回数の関係の一部を示す。計算結果より、以下に示すような最適反復計算回数を求めるための略算式を導いた。

(i) $NV \geq 4$:

$$NI = 2 \times \frac{NOPS}{NOPI} + 2 \quad \dots\dots (3)$$

(ii) $NV=3$:

$$NI=2$$

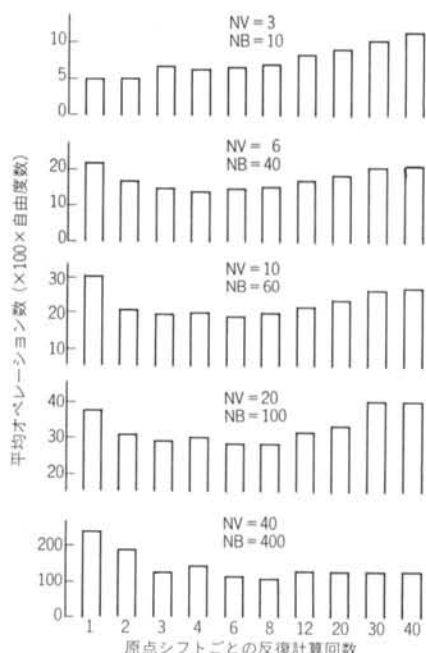


図-10(2) オペレーション数と反復計算回数 ($N=40$, $b=2$)

§ 4. おわりに

以上、大自由度の固有値解析を行なうための最も有効な方法であるサブスペース反復法、ならびにこの手法を実際に RUN する際に問題となる試行ベクトルの数、さらに原点シフトごとの反復計算回数について説明を行ってきた。サブスペース反復法は、与えられた問題のバンド幅に合わせて試行ベクトル数を変えることにより、どのようなサイズのバンド幅に対しても対応できる。それゆえ、小自由度から大自由度に至るまで、あらゆる種類の問題に適用できる最も便利な計算手法である。

最後に、この研究はカリフォルニア大学バークレー校の Wilson 教授と共同に行なったものである。

<参考文献>

- 1) K. J. Bathe & E. L. Wilson: "Numerical Methods in Finite Element Analysis" Prentice-Hall (1977)
- 2) K. J. Bathe & E. L. Wilson: "Large Eigenvalue Problem in Dynamic Analysis" Journal of Engineering Mechanics Division, ASCE, Vol. 98 (1972) pp. 1471~1485
- 3) A. Jennings & D. R. L. Orr: "Application of the Simultaneous Iteration Method to Undamped Vibration Problems" International Journal of Numerical Methods in Engineering, Vol. 3 (1971) pp. 13~24
- 4) A. Jennings: "Matrix Computation for Engineers and Scientists" Wiley (1977)
- 5) T. J. A. Agar & A. Jennings: "Progressive Simultaneous Inverse Iteration for Symmetric Linearized Eigenvalue

Problems" (To be published)

- 6) H. Rutishauser: "Computational Aspects of F.L.Bauer's Simultaneous Iteration Method" Numerische Mathematik, Vol. 13 (1969) pp. 4~13
- 7) E.L. Wilson: "CAL Computer Analysis Language for the Static and Dynamic Analysis of Structural Systems" Report UC SESM 77-2, Civil Engineering, University of California, Berkeley (1977)