Ill－condition行列の固有値計算に関する誤差改善法

田　口　　　功

要　　約

行列の固有値を電子計算機を用いて計算すると誤差が生じる。誤差の評価は、条件数で評価することができる。本論文においては文献(2)による多項式の誤差限界を表す式をもとに、文献(1)における等間隔極配置に対するIll-condition行列の誤差改善法を考えた。n次元の相異なる実数の固有値をもつ行列に対して、誤差限界式は固有値を$\lambda_k(k=1\sim n)$とし、特性方程式の係数を$a_i\,(i=0\sim n)$とすると$|\Delta\lambda_k| \leqq 2^{-t}\sum_{i=0}^{n}|a_{i+1}|\cdot|\lambda_k|^i/|f'(\lambda_k)|$となる。誤差限界式を応用し、固有値の平行移動及びK倍移動を行なうことにより固有値の誤差を減少させる方法を考察した。

1．はじめに

従来から行列の固有値を計算する方法は多数考えられているにもかかわらず、計算機を用いた場合に固有値誤差をどのようにして減少させるかはあまり考えられていない。Ill-condition行列かそうでないかは一般的にはcondition-number（条件数）によって判断されていた。特に行列の次元が大きくなると、理論上の固有値と計算結果が非常に異なり誤差が多く実固有値が複素数に変わってしまう場合も起こる（文献(1)）。本稿では一般的に知られている多項式の誤差限界の式を応用し、固有値のK倍移動および平行移動により誤差を減少させる方法を考察した。すなわち、等間隔に配置された実固有値に対して、0を中心に配置されるように平行移動し、次にK倍移動を行なうと誤差は減少する事を示した。

2．固有値の誤差限界と平行移動について

2.1　固有値の誤差限界について

一般に行列の固有値を計算機を用いて求める場合、入力データは誤差を含む。本稿においては問題を扱いやすくするために、計算機の演算桁数が十分大きく、さらに相異なる実根を持つ場合の誤差を少なくするための検討を行なった。

最初に、行列の固有方程式の係数の変化が固有値に与える影響について考える。固有方程式

$$f(\lambda)=\lambda^n+a_n\lambda^{n-1}+a_{n-1}\lambda^{n-2}+\cdots+a_2\lambda+a_1 \tag{2-1}$$

の各係数がわずかに変化した多項式を

$$\overline{f}(\lambda)=\lambda^n+\overline{a}_n\lambda^{n-1}+\overline{a}_{n-1}\lambda^{n-2}+\cdots+\overline{a}_2\lambda+a_1 \tag{2-2}$$

とする。$f(\lambda)$ の任意の固有値を ξ、これに対応する $\bar{f}(\lambda)$ の固有値を $\bar{\xi}$ とする。ここで、$\bar{a}_{n+1}=a_{n+1}=1$ とし、$\Delta a_{n+1}=0$ とする。

$$\Delta a_{k+1}=\bar{a}_{k+1}-a_{k+1} \qquad (2\text{-}3)$$

$$\Delta \xi=\bar{\xi}-\xi \qquad (2\text{-}4)$$

とおく（$k=0\sim n$）。仮定から、

$$\begin{aligned}\bar{f}(\bar{\xi})&=0\\&=\bar{f}(\bar{\xi})-f(\xi)\\&=\sum_{k=0}^{n}\{(a_{k+1}+\Delta a_{k+1})(\xi+\Delta\xi)^k\\&\quad-a_{k+1}\xi^k\}\\&=\sum_{k=0}^{n}\{a_{k+1}(\xi+\Delta\xi)^k+\\&\quad\Delta a_{k+1}(\xi+\Delta\xi)^k-a_{k+1}\xi^k\}\\&=f(\xi+\Delta\xi)-f(\xi)+\sum_{k=0}^{n}\Delta a_{k+1}\\&\quad\bar{\xi}^k\\&=\sum_{k=0}^{n}f^{(k)}(\xi)\Delta\xi^k/k!+\sum_{k=0}^{n}\\&\quad\Delta a_{k+1}\bar{\xi}^k\end{aligned} \qquad (2\text{-}5)$$

となる。（2 - 5）式の右辺第2項は、係数の摂動による固有方程式 $f(\lambda)$ の $\bar{\xi}$ における摂動と考えられる。そこで、

$$\Delta f(\bar{\xi})=\sum_{k=0}^{n}\Delta a_{k+1}\bar{\xi}^k \qquad (2\text{-}6)$$

とおくと、固有値 ξ が受ける全体の摂動 $\Delta\xi$ は、

$$\sum_{k=0}^{n}f^{(k)}(\xi)\Delta\xi^k/k!+\Delta f(\bar{\xi})=0 \qquad (2\text{-}7)$$

と満足しなければならない。固有方程式の各係数が2進法 t 桁に丸められ、入力されると考えれば、

$$|\Delta a_{k+1}|\leqq 2^{-t}|a_{k+1}| \qquad (2\text{-}8)$$

となるから、関数値の誤差限界は、

$$\Delta f(\bar{\xi})\leqq 2^{-t}\sum_{k=0}^{n}|a_{k+1}|\,|\bar{\xi}|^k \qquad (2\text{-}9)$$

となる。また（2 - 7）式において $\Delta\xi$ の2次以上の項を無視すると、

$$f'(\xi)\ \Delta\xi/1!=-\Delta f(\bar{\xi}) \qquad (2\text{-}10)$$

となり、（2 - 9）式を用いると、

$$\begin{aligned}|\Delta\xi|&=|-\Delta f(\bar{\xi})/f'(\xi)|\\&\leqq 2^{-t}\sum_{k=0}^{n}|a_{k+1}|\,|\xi|^k/|f'\\&\quad(\xi)|\end{aligned} \qquad (2\text{-}11)$$

の関係が成立する。ここで、（2 -11）式は各固有値に対する誤差限界を示す。したがって、各固有値の有効桁数は、

$$\sum_{k=0}^{n}|a_{k+1}|\,|\xi|^k/|f'(\xi)| \qquad (2\text{-}12)$$

によって大きく変化する事となる（$k=0\sim n$）。

2．2　固有値の平行移動について

最初に実固有値 λ_1、λ_2、λ_3、…、λ_n に対して $\lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_n > 0$ の関係が成立し、移動後の固有値もすべて正の場合を考える。固有値の平行移動によって（2 - 1）式を微分した関係 $f'(\lambda_i)$（$i=1\sim n$）は変化しない。すなわち、（2 - 1）式に対し $f'(\lambda_i)$ は一般的に、$i=1\sim n$ に対して、

$$\begin{aligned}f'(\lambda_i)&=(\lambda_i-\lambda_1)\cdot(\lambda_i-\lambda_2)\cdot(\lambda_i-\\&\quad\lambda_3)\cdot\cdots(\lambda_i-\lambda_{i-1})\cdot\cdots(\lambda_i-\lambda_{i+1})\cdot\\&\quad\cdots(\lambda_i-\lambda_n)\end{aligned} \qquad (2\text{-}13)$$

のように、差積の形で表わすことができる。平行移動量を p とする。（2 -13）式の右辺の各項

は平行移動すると $(\lambda_i + p - \lambda_1 - p)\cdot(\lambda_i + p - \lambda_2 - p)\cdot(\lambda_i + p - \lambda_3 - p)\cdots(\lambda_i + p - \lambda_{i-1} - p)\cdots(\lambda_i + p - \lambda_{i+1} - p)\cdot\cdots(\lambda_i + p - \lambda_n - p)$ と表せるから各 λ_i に対してその値は変化しない。ここで、固有方程式の係数 a_i と $(i = 1 \sim n)$ 固有値 λ_i との関係を考えると次の関係が成立する。

$$\lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_n = -a_n$$
$$\lambda_1\lambda_2 + \lambda_2\lambda_3 + \cdots + \lambda_{n-1}\lambda_n = a_{n-1}$$
$$\lambda_1\lambda_2\lambda_3 + \lambda_2\lambda_3\lambda_4 + \cdots + \lambda_{n-2}\lambda_{n-1}\lambda_n = -a_{n-2}$$
$$\cdots\ \cdots$$
$$\lambda_1\lambda_2\lambda_3\cdots\lambda_{n-1}\lambda_n = (-1)^n a_1 \tag{2-14}$$

同様に、移動後の関係は、

$$\tilde{\lambda}_1 + \tilde{\lambda}_2 + \tilde{\lambda}_3 + \cdots + \tilde{\lambda}_n = -\tilde{a}_n$$
$$\tilde{\lambda}_1\tilde{\lambda}_2 + \tilde{\lambda}_2\tilde{\lambda}_3 + \cdots + \tilde{\lambda}_{n-1}\tilde{\lambda}_n = \tilde{a}_{n-1}$$
$$\tilde{\lambda}_1\tilde{\lambda}_2\tilde{\lambda}_3 + \tilde{\lambda}_2\tilde{\lambda}_3\tilde{\lambda}_4 + \cdots + \tilde{\lambda}_{n-2}\tilde{\lambda}_{n-1}\tilde{\lambda}_n = -\tilde{a}_{n-2}$$
$$\cdots\ \cdots$$
$$\tilde{\lambda}_1\tilde{\lambda}_2\tilde{\lambda}_3\cdots\tilde{\lambda}_{n-1}\tilde{\lambda}_n = (-1)^n \tilde{a}_1 \tag{2-15}$$

となる。さらに、移動量を $p\ (p > 0)$ とすると、

$$\lambda_1 - p = \tilde{\lambda}_1$$
$$\lambda_2 - p = \tilde{\lambda}_2$$
$$\cdots$$
$$\lambda_n - p = \tilde{\lambda}_n \tag{2-16}$$

が成立し、平行移動後に

$$\tilde{\lambda}_1 > \tilde{\lambda}_2 > \tilde{\lambda}_3 > \cdots > \tilde{\lambda}_{n-1} > \tilde{\lambda}_n > 0 \tag{2-17}$$

の関係が明らかに成立するので、

$$|a_n| > |\tilde{a}_n|$$
$$|a_{n-1}| > |\tilde{a}_{n-1}|$$
$$\cdots\ \cdots$$
$$|a_1| > |\tilde{a}_1| \tag{2-18}$$

となる。(2-12) 式に対して $|a_{k+1}|$ $(k = 1 \sim n)$ の値はすべて減少する。また、$|f'(\xi)|$ の各固有値に対する値は変化なく $|\xi|^{n-1}$ に対する各固有値の値は減少するのですべての固有値に対して誤差限界は改善されると考えられる。

次に固有値 λ_1、λ_2、λ_3、…、λ_n に対して、$0 > \lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_n$ の関係が成り立ち平行移動後、$0 > \tilde{\lambda}_1 > \tilde{\lambda}_2 > \tilde{\lambda}_2 > \cdots > \tilde{\lambda}_n$ の関係が成立する場合を考える。$p > 0$ とする。すなわち、

$$\lambda_1 + p = \tilde{\lambda}_1$$
$$\lambda_2 + p = \tilde{\lambda}_2$$
$$\cdots$$
$$\lambda_n + p = \tilde{\lambda}_n \tag{2-19}$$

の関係を仮定する。(2-14) 式、(2-15) 式をもとに、(2-19) 式を考慮すると、(2-14) 式各式の符号と (2-15) 式の符号は変化しない。大きさのみ (2-19) 式から減少することがわかる。例えば a_n と $\tilde{a}_n$ の比較をすると、$np - a_n = -\tilde{a}_n$ の関係があるから $(n > 0,\ p > 0)$ $|\tilde{a}_n - a_n| = np$ となり np だけ減少する事になる。他の係数もそれぞれ p^2、p^3、…、p^n の割合で減少する事がわかる。この事は、最初に述べた正の固有値に対して p だけ平行に固有値を減少させた場合にも成立する。まとめ

ると、次の事が言える。すべての固有値が正(負)で移動後の固有値もすべて正（負）である場合、それらの固有値の絶対値が減少するように平行移動すれば誤差限界は改善できる。

次に、等間隔ではないが原点から十分離れた実固有値に対して、すなわち、平行移動前の固有値の絶対値が平行移動後のすべての固有値の絶対値より大きい場合について考える。この場合は移動後負の固有値が含まれてもよい。しかし、移動後の固有値の最大値が移動前の絶対値の最小値を越えない場合について考える。まず、すべての正なる実固有値 λ_1、λ_2、λ_3、…、λ_n に対して、(2 -14) 式、(2 -15) 式を考える。原点を 0 とし図 1 のような配置を考える。

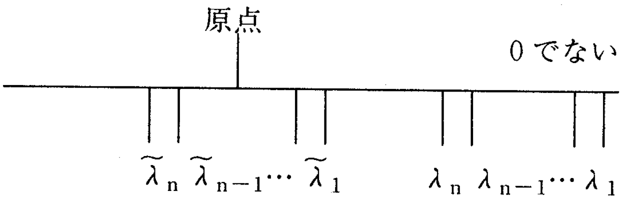

図　1

すると、(2 -14) 式の左辺の値はすべて正となる。(2 -15) 式の左辺の値は、$\tilde{\lambda}_n < 0$、$\tilde{\lambda}_{n-1} < 0$ とし他の平行移動後の固有値を 0 または正とすると正負の固有値の和または正負の固有値の積の和で表される。さらに、次のことを考える。$\tilde{\lambda}_n < 0$、$\tilde{\lambda}_{n-1} < 0$ とした固有値に対して絶対値は等しく符号を逆にした固有値 $\underline{\lambda}_n \underline{\lambda}_{n-1} \cdots \underline{\lambda}_1$ と $\tilde{\lambda}_1 \tilde{\lambda}_2 \cdots \tilde{\lambda}_n$ および $\lambda_1 \lambda_2 \cdots \lambda_n$ の 3 種類の配置に対する根と特性方程式の係数との関係について考える。平行移動前の $\lambda_1 \lambda_2 \cdots \lambda_n$ に対する係数を a_i、平行移動し正負の固有値を持つ係数を $\tilde{a}_i$ さらに 1 度平行移動しさらに負の固有値に対してその絶対値をとりすべて正の固有値にした係数を $\underline{a}_i$ とすると、$i = n \sim 1$ に対して、

$$|a_i| > |\underline{a}_i| \tag{2-20}$$

の関係が成立する。これはたがいにすべての固有値が正であるのでその絶対値の大きい固有値をもつ平行移動前の係数は固有値の和、または積の和で係数は表せるため当然そうなると考えられる。これは、前の議論から明かである。次に、$|\underline{a}_i|$ と $|\tilde{a}_i|$ の比較を考える。各固有値に対する絶対値は等しいが、$|\underline{a}_i|$ に対してはすべて正の固有値をもつが $|\tilde{a}_i|$ は、$\tilde{\lambda}_n < 0$、$\tilde{\lambda}_{n-1} < 0$ であるから

$$|\underline{a}_i| \geqq |\tilde{a}_i| \tag{2-21}$$

の関係が成立する。従って、(2 -20) 式および (2 -21) 式から、

$$|a_i| > |\tilde{a}_i| \tag{2-22}$$

の関係が成立する。この議論はすべての固有値が負である場合についても成立する。

次にすべての固有値が正で、等間隔に配置された固有値に対して、原点を中心に平行移動する場合を考える。原点に固有値が存在する場合は含まない場合より根に対する係数の絶対値は減少するから、含む場合に対して平行移動することを考えることにする。n が奇数の場合と、偶数の場合が考えられるので、二つの場合が考えられる。

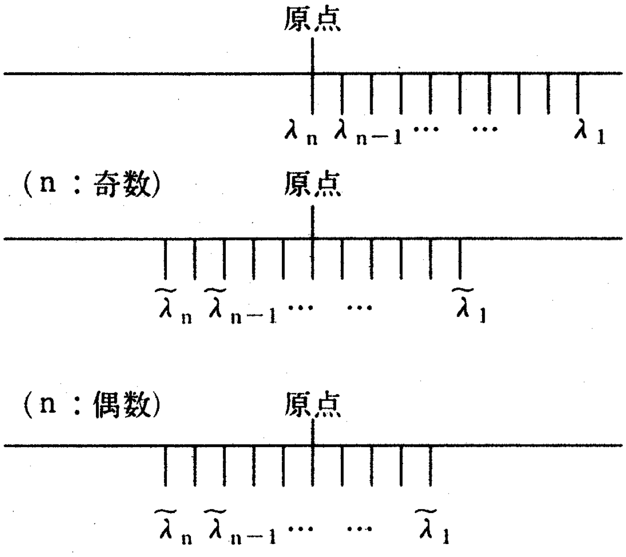

図　2

この場合、前の議論と同じく((2-20)、(2-21)、(2-22) 式) 平行移動した結果、特性方程式の係数の絶対値は減少する。これは、正の固有値に対しては平行移動前の固有値を必ず含み、さらに平行移動の結果負になる固有値の絶対値は平行移動前の残りの固有値に1対1に対応させて考えれば等しいか減少している。その結果特性方程式の係数の絶対値は必ず減少する。このことは、二つのパターンに対して、明かである。

3．固有値の等間隔配置における $f'(\lambda_i)$ の最小値について

固有値の等間隔配置における次元と $f'(\lambda_i)$ の最小値について考える。行列Aの固有値はすべて実数とし、重根を含まない、

$$\lambda_1 \neq \lambda_2 \neq \lambda_3 \cdots \neq \lambda_n \tag{3-1}$$

の関係が成立する場合を考える。行列Aの特性方程式は（$i=1\sim n$)、

$$f(\lambda_i)=(\lambda_i-\lambda_1)(\lambda_i-\lambda_2)(\lambda_i-\lambda_3)\cdots(\lambda_i-\lambda_n) \tag{3-2}$$

として表されるから、$f'(\lambda_i)$ は、

$$f'(\lambda_i)=\prod(\lambda_i-\lambda_j)\ (i\neq j) \tag{3-3}$$

の差積の形で表現される。10次元の場合を考える事とする。固有値の間隔をhとし、

$$\begin{aligned}\lambda_1-\lambda_2&=h\\ \lambda_2-\lambda_3&=h\\ &\vdots\\ \lambda_9-\lambda_{10}&=h\end{aligned} \tag{3-4}$$

とすると、(2－3) 式の使用で、

$$\begin{aligned}
f'(\lambda_1)&=9!\,h^9\\
f'(\lambda_2)&=-8!\,h^9\\
f'(\lambda_3)&=2!*7!\,h^9\\
f'(\lambda_4)&=-3!*6!\,h^9\\
f'(\lambda_5)&=4!*5!\,h^9\\
f'(\lambda_6)&=-5!*4!\,h^9\\
f'(\lambda_7)&=6!*3!\,h^9\\
f'(\lambda_8)&=-7!*2!\,h^9\\
f'(\lambda_9)&=8!*1!\,h^9\\
f'(\lambda_{10})&=-9!\,h^9
\end{aligned} \tag{3-5}$$

と計算でき、規則正しくh間隔に固有値が設定されている場合一般的に

$$|f'(\lambda_i)|=|(i-1)*(n-i)!*h^{n-1}| \tag{3-6}$$

の関係が成立する。問題を簡単化するためにh＝1とした場合の $f'(\lambda_i)$ の最大値、最小値をn＝2～10、n＝20、n＝30について考え

る。(3-6) 式から$n=2$の時は、最小値=最大値=1となり、$n=3$の時は、$i=1$の時、

$$0!*2!=2 \tag{3-7}$$

$i=2$の時、

$$1!*1!=1 \tag{3-8}$$

$i=3$の時、

$$2!*0!=2 \tag{3-9}$$

となる。最大値は、$i=1$および$i=3$の時であり最小値は1となる。同様に、$n=4$の場合を計算すると、最大値は$i=1$の時と$i=4$の時でその値は6となり、最小値は$i=2$、$i=3$の時で、その値は2となる。この操作を繰り返すと$n=10$の時最大値は$i=1$の時および$i=10$の時でその絶対値はほぼ$3.62*10^5$となる。最小値は$i=5$、$i=6$の時で$2.88*10^3$となる。$n=20$の時は最小値は$i=10$、$i=11$の時で、その値は、$1.316*10^{12}$、$n=30$の時は$i=15$、$i=16$の時で、$1.71*10^{24}$となる。従って、次のことがいえる。最大値は$i=1$の時、および$i=n$の時、最小値は、nが偶数の時は$i=n/2$、および$i=(n/2)+1$、奇数の場合は$i=(n/2)+1$の時となる。

4. 固有値の$\widehat{k}$倍移動と誤差限界の改善

一般的に行列Aの固有方程式は、(2-1) 式で表すことができる。従って、その微分関数は、

$$f'(\lambda)=n\lambda^{n-1}+(n-1)a_n\lambda^{n-2}+(n-2)a_{n-1}\lambda^{n-3}+\cdots+a_2 \tag{4-1}$$

となる。ここで、相異なる固有値λ_i ($i=1\sim n$) に対して、$\widehat{k}$倍することを考える。すなわち、$\widetilde{\lambda}_i=\widehat{k}\lambda_i$の関係を仮定する。このとき、$\widehat{k}$倍された固有方程式を$F(\widetilde{\lambda}_i)$とすると、

$$F(\widetilde{\lambda}_i)=\widetilde{\lambda}_i^{n}+a_n\widehat{k}\,\widetilde{\lambda}_i^{n-1}+a_{n-1}\widehat{k}^{2}\widetilde{\lambda}_i^{n-2}+\cdots+a_2\widehat{k}^{n-1}\widetilde{\lambda}_i+a_1\widehat{k}^{n} \tag{4-2}$$

となる。(4-2) 式を$\widetilde{\lambda}_i$に関して微分すると、

$$F'(\widetilde{\lambda}_i)=n\,\widetilde{\lambda}_i^{n-1}+a_n\widehat{k}(n-1)\widetilde{\lambda}_i^{n-2}+\cdots+a_2\widehat{k}^{n-1} \tag{4-3}$$

となるから、(4-3) 式に対して、$\widetilde{\lambda}_i=\widehat{k}\lambda_i$の関係を代入すると、次の (4-4) 式が成り立つ。

$$F'(\widetilde{\lambda}_i)=\widehat{k}^{n-1}(f'(\lambda_i)) \tag{4-4}$$

ここで、2. で述べた誤差限界式 (2-11) 式に対して$\widehat{k}$倍移動した誤差を、$|\Delta\overline{\xi}|$とすると、

$$|\Delta\overline{\xi}|\leqq 2^{-t}\sum_{k=0}^{n}|\overline{a}_{k+1}||\overline{\xi}|^{k}/|F'(\overline{\xi})| \tag{4-5}$$

となる。ここで、$\overline{a}_{k+1}=a_{k+1}\widehat{k}^{n-k}$、$\overline{\xi}=\widehat{k}\xi$、および、(4-4) 式の関係を、(4-5) 式に代入すると、

$$|\Delta\overline{\xi}|\leqq 2^{-t}\sum_{k=0}^{n}|\overline{a}_{k+1}||\overline{\xi}|^{k}/|F'(\overline{\xi})|$$

$$=2^{-t}\sum|a_{k+1}\widehat{k}^{n-k}||\widehat{k}\xi|^{k}/\widehat{k}^{n-1}|f'(\xi)|$$

$$\leqq 2^{-t}\sum|a_{k+1}||\hat{k}^{n-k}||\hat{k}|^{k}|\xi|^{k}/|\hat{k}^{n-1}||f'(\xi)|$$

$$=2^{-t}\sum|a_{k+1}||\hat{k}||\xi|^{k}/|f'(\xi)| \qquad (4\text{-}7)$$

となる。ここで、$\hat{k}$倍移動によって、$|\hat{k}|<1$の条件が成立する場合には、誤差限界値は減少することになる。従って、2．で残された問題、すなわち、$\hat{k}$倍移動によって、原点付近に固有値を移動することにより誤差限界を減少させることができるということは、証明された。従って、固有値の平行移動および$\hat{k}$倍移動は、共に誤差限界を減少させることができる。すなわち、等間隔に配置された固有値に対して、$\hat{k}$倍移動（$|f'|$の最小値に注意しながら）により原点付近に固有値を集めること。さらには、平行移動により、原点に対して、対称的に固有値を配置させること（nが偶数の時、奇数の時を考えながら）により誤差限界あるいは、相対誤差を減少させる事ができるということになる。

5．例　　題

例題1

等間隔に配置された11次元行列Aの固有値、20、19、18、…、10に対し、－5、－10、－15、…、－30と－5おきに平行移動した場合の、固有値計算を計算機でシミュレーションした結果を次に示す。行列Aをつくり、そのコンパニオン行列に対し移動を行わない場合と、－5間隔に固有値を移動させコンパニオン行列をつくり、そのコンパニオン行列の固有値を計算した結果を次に示す。この結果、すべての固有値を計算する場合、－15平行移動した時、誤差は最小になっていることがわかり、このとき、実際の固有値計算は、5、4、3、2、…、－5（原点に対して対称的）に対して固有値計算を行ない、全ての固有値に対し最後に＋15を加え、行列Aの固有値を算出したものである。その結果を以下に示す。この結果、原点に対して対称的に配置された固有値に対する誤差は最小となることがわかる。さらに、この場合にはすべての固有値に対して問題となるのが、$|f'|$であるが、最小値の$|f'(\lambda_i)|$の値が1より非常に大きくなるから、k倍移動を行なわなくとも平行移動のみで、精度は非常に向上していることがわかる。逆に、対称配置を離れるにしたがって誤差は増加するといえる。

（例題１の内容）

行列Ａの作成プログラム

```
»for i=1:11
  for j=1:11
   if i==j
     A(i,j)=20-i+1;
   else
     A(i,j)=0;
   end
  end
 end
»A

A =

  20     0     0     0     0     0     0     0     0     0     C
   0    19     0     0     0     0     0     0     0     0     C
   0     0    18     0     0     0     0     0     0     0     C
   0     0     0    17     0     0     0     0     0     0     C
   0     0     0     0    16     0     0     0     0     0     C
   0     0     0     0     0    15     0     0     0     0     C
   0     0     0     0     0     0    14     0     0     0     C
   0     0     0     0     0     0     0    13     0     0     C
   0     0     0     0     0     0     0     0    12     0     C
   0     0     0     0     0     0     0     0     0    11     C
   0     0     0     0     0     0     0     0     0     0    1C
```

行列Ａをコンパニオン行列に変換する作成プログラム

```
»ACOM=compan(poly(A))
»C1=rot90(ACOM)
»A1=rot90(C1)

A1 =

   1.0e+12 *

  Columns 1 through 4

                 0   0.00000000000100                  0                  C
                 0                  0   0.00000000000100                  C
                 0                  0                  0   0.0000000000010C
                 0                  0                  0                  C
                 0                  0                  0                  C
                 0                  0                  0                  C
                 0                  0                  0                  C
                 0                  0                  0                  C
                 0                  0                  0                  C
                 0                  0                  0                  C
  6.70442572800000  -5.15417077440000   1.79213978592000  -0.3720477135760C
```

```
Columns 5 through 8

                 0                 0                 0                 0
                 0                 0                 0                 0
                 0                 0                 0                 0
  0.00000000000100                 0                 0                 0
                 0  0.00000000000100                 0                 0
                 0                 0  0.00000000000100                 0
                 0                 0                 0  0.00000000000100
                 0                 0                 0                 0
                 0                 0                 0                 0
                 0                 0                 0                 0
  0.05124139350000 -0.00491646353000  0.00033534616500 -0.00001626177300

Columns 9 through 11

                 0                 0                 0
                 0                 0                 0
                 0                 0                 0
                 0                 0                 0
                 0                 0                 0
                 0                 0                 0
                 0                 0                 0
  0.00000000000100                 0                 0
                 0  0.00000000000100                 0
                 0                 0  0.00000000000100
  0.00000054945000 -0.00000001232000  0.00000000016500
```

平行移動、K倍移動のない場合の計算結果

```
»eig(A1)

ans =

  20.00000365349384
  18.99997527033916
  18.00007276109099
  16.99987864616618
  16.00012491309282
  14.99991966688947
  14.00003034621732
  12.99994990140100
  11.99999943793277
  11.00000035412950
   9.99999996050777
```

－５平行移動

```
»for i=1:11
  for j=1:11
   if i==j
     B(i,j)=20-i+1-5;
   else
     B(i,j)=0;
   end
  end
 end
»BCOM=compan(poly(B));
»BC1=rot90(BCOM);
»B1=rot90(BC1);
»eig(B1)+5

ans =

   19.99999997727729
   19.00000014142309
   17.99999961307119
   17.00000061333744
   15.99999937472204
   15.00000043028628
   13.99999979624646
   13.00000006604371
   11.99999998591611
   11.00000000177427
    9.99999999990208
```

－１０平行移動

```
»for i=1:11
  for j=1:11
   if i==j
     B(i,j)=20-i+1-10;
   else
     B(i,j)=0;
   end
  end
 end
»BCOM=compan(poly(B));
»BC1=rot90(BCOM);
»B1=rot90(BC1);
»eig(B1)+10

ans =

   10.00000000000000
   20.00000000005036
   18.99999999978161
   18.00000000035903
   16.99999999977177
   15.99999999994031
   15.00000000018574
   13.99999999987910
   13.00000000003710
   11.99999999999478
   11.00000000000022
```

－１５平行移動
５～－５で計算

```
»for i=1:11
  for j=1:11
   if i==j
   B(i,j)=20-i+1-15;
   else
   B(i,j)=0;
   end
  end
 end
»BCOM=compan(poly(B));
»BC1=rot90(BCOM);
»B1=rot90(BC1);
»eig(B1)+15

ans =

   15.00000000000000
   20.00000000000000
   10.00000000000000
   18.99999999999999
   11.00000000000001
   18.00000000000001
   11.99999999999999
   17.00000000000000
   13.00000000000000
   16.00000000000000
   14.00000000000000
```

－２０平行移動
０～－１０で計算

```
»for i=1:11
  for j=1:11
   if i==j
   B(i,j)=20-i+1-20;
   else
   B(i,j)=0;
   end
  end
 end
»BCOM=compan(poly(B));
»BC1=rot90(BCOM);
»B1=rot90(BC1);
»eig(B1)+20
```

－２５平行移動
－５～－１５で計算

```
»for i=1:11
  for j=1:11
   if i==j
   B(i,j)=20-i+1-25;
   else
   B(i,j)=0;
   end
  end
 end
»BCOM=compan(poly(B));
»BC1=rot90(BCOM);
»B1=rot90(BC1);
»eig(B1)+25
```

－３０平行移動
－１０～－２０で計算

```
»for i=1:11
  for j=1:11
   if i==j
   B(i,j)=20-i+1-30;
   else
   B(i,j)=0;
   end
  end
 end
»BCOM=compan(poly(B));
»BC1=rot90(BCOM);
»B1=rot90(BC1);
»eig(B1)+30
```

ans =	ans =	ans =
20.00000000000000	10.00000002272271	9.99999634650613
9.99999999994964	10.99999985857691	11.00002472966097
11.00000000021839	12.00000038692881	11.99992723890876
11.99999999964097	12.99999938666256	13.00012135383405
13.00000000022823	14.00000062527796	13.99987508690710
14.00000000005969	14.99999956971372	15.00008033311054
14.99999999981426	16.00000020375354	15.99996965378268
16.00000000012091	16.99999993395629	17.00000500985989
16.99999999996290	18.00000001408389	18.00000056206722
18.00000000000522	18.99999999822573	18.99999964587050
18.99999999999978	20.00000000009792	20.00000003949224

例題２

行列Ａの固有値、2.0、1.9、…、1.0に対し平行移動のみを行ない、固有値を－2.5間隔で原点付近に移動し、それぞれ、－0.5～－1.5、－3.0～－4.0、－5.5～－6.5、で計算を行なった場合を次に示す。平行移動により、－0.5～－1.5において計算を行った場合には、移動を行わない場合よりやや精度が向上しているのがわかる。しかし、平行移動により原点から離れるにしたがって精度が悪化し、実固有値が複素数に変化していることがわかる。この例題により、等間隔に配置された固有値計算を行なう場合には、原点付近に固有値を集めたほうが、精度良く計算が行なわれることがわかる。さらに、原点から離れるにしたがって精度は減少し、実固有値が複素数にかわってしまう場合も起こることもわかる。

（例題２の内容）

行列Ａを作成し平行移動を行いコンパニオン行列の固有値を求めるプログラム

```
»for i=1:11
   for j=1:11
  if i==j
  A(i,j)=2-0.1*i+0.1;
  else
  A(i,j)=0;
   end
  end
 end
»acom=compan(poly(A));
»AC1=rot90(acom);
»A1=rot90(AC1);
»for k=0:5:20
  ACOM=compan(poly(A-0.5*eye(11)*k));
  AC1=rot90(ACOM);
  A1=rot90(AC1);
 eig(A1)+0.5*k
 end
```

１．０～２．０の範囲で固有値を計算（平行移動のない場合）

```
ans =

   1.99999993577565
   1.90000028709817
   1.79999982335635
   1.69999849453371
   1.60000476549158
   1.49999290779957
   1.40000634256921
   1.29999641129340
   1.20000126299450
   1.09999974702870
   1.00000002205915
```

－０．５～－１．５の範囲で固有値を計算

```
ans =

   0.99999999114994
   1.10000006094794
   1.19999981459600
   1.30000032749503
   1.39999962866490
   1.50000028197606
   1.59999985491280
   1.70000004996437
   1.79999998894796
   1.90000000142787
   1.99999999991713
```

－３．０～－４．０の範囲で固有値を計算

```
ans =

   1.00057033868204
   1.09579638461281
   1.22073458161783
   1.26700027554543
   1.45136203662654 + 0.02801223578217i
   1.45136203662654 - 0.02801223578217i
   1.62265940993745
   1.68840197504398
   1.80245973448861
   1.89963207859047
   2.00002114822830
```

－５．０～－６．５の範囲で固有値を計算

```
ans =

   0.86435863679879 + 0.13352116089059i
   0.86435863679879 - 0.13352116089059i
   1.07242463652313 + 0.36623037723150i
   1.07242463652313 - 0.36623037723150i
   1.41194778308727 + 0.48208854141554i
   1.41194778308727 - 0.48208854141554i
   1.77403197325424 + 0.43526715203324i
   1.77403197325424 - 0.43526715203324i
   2.04958457287962 + 0.25028642428915i
   2.04958457287962 - 0.25028642428915i
   2.15530479491395
```

例題３

ｋ倍移動の有効性を例題２の問題を使用しここで述べる。（３－６）式から、間隔 $h=0.1$、$n=11$ である。２章で述べた関係から、$|f'(\lambda_i)|$ の最小値は $i=6$ となるから、

$$\text{最小値}\{|f'(\lambda_6)|\}=5!\times(11-6)\times(0.1)^{10} \qquad (4\text{-}1)$$

となる。さらに、（３－４）式から

$$\text{最小値}\{|f'(\lambda_6)|\}=1/|k|^{10} \qquad (4\text{-}2)$$

となるので、$k=3.83$ となる。$k=3.83$ とした場合に、固有値は、ほぼ、7.66～3.83に配置されるから、平行移動で、すべての固有値を3.8減少させれば、3.86～0.03で計算されることになり、原点に集められる。この結果を次に示す。

(例題3の1内容)

K倍移動量＝3．83
K倍移動後−3．8平行移動
0．03〜3．86の範囲で固有値計算

```
»(eig(compan(poly(A*3.83-3.8*eye(11))))+3.8)/3.83

ans =

   2.00000000000441
   1.89999999997540
   1.80000000006137
   1.69999999991325
   1.60000000007385
   1.49999999996217
   1.40000000001116
   1.29999999999829
   1.20000000000011
   1.10000000000000
   1.00000000000000
```

平行移動、k倍移動両移動を行なわない場合より、精度は向上していることがわかる。例題2のように、精度が悪化するようなことはなくなっている。さらに、k＝3.83に対して、−5.74

(例題3の1内容)

K倍移動量＝3．83
K倍移動後−5．74平行移動
−1．91〜1．92の範囲で固有値計算

```
»(eig(compan(poly(A*3.83-5.74*eye(11))))+5.74)/3.83

ans =

   1.99999999999999
   1.00000000000000
   1.90000000000002
   1.10000000000000
   1.79999999999999
   1.20000000000000
   1.70000000000000
   1.30000000000000
   1.60000000000000
   1.40000000000000
   1.50000000000000
```

すべての固有値を平行移動し、1.92〜1.91の領域で固有値計算を行ない、求められた固有値に対して、+5.74を加え、その結果に対して、3.83で割り固有値計算を行なった場合を以下に示す。ほとんど誤差なく計算が行なわれ、k倍移動と平行移動の効果が現れた結果となっていることがわかる。すなわち、平行移動による固有値の原点付近の対称配置の有効性とk倍移動による｜f'｜の最小値選択の有効性が現れた結果、精度が向上したものと考えられる。

6．おわりに

行列Aの実固有値に対し等間隔に固有値が配置され、行列の要素にわずかな変動があった場合、計算機を用いて、固有値を計算すると、その値が大きく変動してしまう場合がある（文献(1)、文献(2)）。本論文においては、固有値のk倍移動と、平行移動を取り入れ、精度よく計算を行なうための方法を多項式の誤差限界式を利用し理論的に示した。最後に実際に、計算機で、シミュレーションによって、その有用性を明らかにした。

7．謝　辞

本研究にあたり、細かい部分にまで御指導いただいた本学教授岡本茂氏、千葉大学美多勉教授に、感謝いたします。

8．参考文献

(1) Wilkinson J.H：The Algebraic Eigenvalue Problem, OXFORD, pp.89－90, 1988年

(2) 牧之内三郎：数値解析、オーム社、pp.77－83、pp.150－152, 1975年

(3) 町田東一 他：マトリクスの固有値と対角化、東海大学出版会、P 36－P 39, 1990年

(4) 有本卓：数値解析(1)、コロナ社、pp.49－51、1981年

(5) 田口 功：行列の一次摂動に関する固有値の誤差に対する一考察、千葉敬愛短期大学紀要、第16号、1994年

ABSTRACT

A Method of Improving Errors of Eigenvalues of Ill－Condition Matrices.

Isao Taguchi

Errors occur in the numerical calculation of all eigenvalues of the matrix when using a computer. In general, error assessment determines the condition number. In this paper, the author discusses ill-conditioned matrices (1) using the limit of error discussed in Makinouchi(2). The limit of error for each eigenvalue is changed, and satisfies a relation of the form $|\Delta\lambda_k| \leqq 2^{-t}\sum_{i=0}^{n}|a_{i+1}|.|\lambda_k|^{i}/|f'(\lambda_k)|$. For the original matrix, each calculated eigenvalue, a parallel translation, and a K times translation are used to recalculate the numerical calculation. As a result, accuracy is further improved for each eigenvalue.