Vol.2015-HPC-149 No.10
2015/6/26

有理数計算による対称行列の固有値問題における特性多項式の因子探し

寒川　光[1,a)]

概要：現在開発中の「有理数計算プログラミング環境」は，有理算術演算と浮動小数点演算をひとつのプログラムで使用することができる．本稿ではこの環境の例題として，対称行列の固有値の多重度解析を扱う．対称行列を有理算術演算によってフロベニウス変換して正確な特性多項式を得て，これに浮動小数点計算によって得られた近似固有値を組み合わせて当てはめることで因子多項式を探し出すプログラムを解説する．これは自前で作る数式処理システムの因数分解機能を代替するプログラムである．このプログラムに対し，有理算術演算と浮動小数点演算の特長を活かしたチューニングにより，1 桁近い高速化を実現する HPC プログラミングの新しい教材を作成したので報告する．

キーワード：有理算術演算，対称固有値問題，特性多項式，ヘッセンベルグ行列，フロベニウス標準形，因数分解, 根と係数の関係公式

A searching program for divisors of characteristic polynomial in symmetric eigenvalue problem by using rational arithmetic

Hikaru Samukawa[1,a)]

Abstract: We are developing a rational arithmetic programming environment in which floating-point arithmetic and rational arithmetic can be used togather. In this paper, we show a numerical example which can be achieved by using both floating-point arithmetic and rational arithmetic. In a multiplicity analysis of a symmetric eigenvalue problem, an exact characteristic polynomial is obtained by transforming symmetric matrix to Frobenius form in rational arithmetic, then search its divisors by substituting approximated eigenvalues obtained in floating-point arithmetic into Vieta's formula. This program can replace the factoring function of formula manipulation system. Roughly ten times performance gain is achieved to the original program by combining features of the rational arithmetic and floating-point arithmetic. We are aiming to learn to use the two arithmetics differently based on thier performance features in HPC programming education.

Keywords: rational arithmetic, symmetric eigenvalue problem, characteristic polynomial, Hessenberg matrix, Frobenius canonical form, factorization, Vieta's formula

1. はじめに

数値計算の多くは浮動小数点演算によって行われており，計算の効率は高く，融通がきくが，信頼性

[1] 早稲田大学理工学術院
Waseda University, Shinjuku-ku, Tokyo 169–8555, Japan
[a)] samukawa@sic.shibaura-it.ac.jp

は低い．数値計算と対照的な方法が数式処理で，信頼性は高いが柔軟性に欠け，計算効率も低い．我々は，浮動小数点演算を用いる数値計算のプログラミング環境に，有理算術演算を加えることで，浮動小数点計算で生じた丸め誤差に起因する問題を解決する目的で使用できる「有理数計算プログラミング環境」を開発中である．有理数の分母・分子を約 30 万桁まで動的に拡張可能な多桁の自然数を格納できる配列で保持し，有理数の四則演算ごとに分母・分子を既約分数にする．演算結果は丸めないので計算誤差は介入せず，桁溢れしないかぎり，計算は正確である．変数型 `rational` を C++ によるプログラミング環境に追加するだけで，有理数型の変数と浮動小数点型の変数は共存できる[*1]．

この環境は次の使用を目的としている [2].

- 丸め誤差の理論を学ぶ前の学生を対象とした数学とプログラミングの教育
- 数値シミュレーションで用いられる，浮動小数点演算による数値計算で発生した精度に関係する問題を分析するツール
- 計算機援用証明（Computer Assisted Proof）
- ベンチマークの検収など，計算機システムの稼働確認

本稿で紹介する解析は，「有理数計算プログラミング環境」の必要機能の調査および作動確認を兼ねて，大学の学部学生で力学知識を使用しない HPC プログラミングの教材を開発することを目的とした．浮動小数点計算では解明困難な対称行列の固有値の多重度の解析を題材にする．まず，有理算術演算で特性多項式を正確に求める．**浮動小数点数は数学的には有理数**である．したがって，倍精度浮動小数点演算で生成された係数行列は正確に有理行列に変換できる．有理行列の全行列要素の分母の最小公倍数（Least Common Multiple, LCM）を求めて掛けることで，有理行列は整数行列に変換できる．整数行列の特性多項式は，最高次の係数が 1 か -1 の整式となるので，次数と固有値の数が一致する．これを利用して，浮動小数点演算で求めた近似固有値から逆算して，因数分解された形の特性多項式の因子（divisor）を探す．探索は指数時間アルゴリズムであり，次数が高くなると計算量が爆発する．したがって，チューニングの題材には永く使用できる．ここでプログラミングに使用する数学の道具は，線形代数の授業で習う**基本変換**（elementary transformation）と，高校時代からおなじみの**根と係数の関係**公式（Vieta's formula）だけなので，大学の初年度，2 年次の学生の HPC プログラミング演習の教材としても使用できる（固体力学，量子力学などの専門性の高い力学の知識が不要）．

次章でプログラミング環境の概要について述べる．3 章で対称行列の重複固有値の解析を有理数計算で行うアルゴリズムついて述べる．4 章で数値実験の結果を示す．5 章でまとめる．

2. 有理算術演算とプログラミング環境

有理算術演算は計算機科学の黎明期からの研究テーマである [3].「有理数計算プログラミング環境」では，有理数を，分母と分子を多桁数で保持し，これに符号を付加することで表現する．四則演算は次のように行う．

$$r_1 \pm r_2 = \frac{b}{a} \pm \frac{d}{c} = \frac{bc \pm ad}{ac} \tag{1}$$

$$r_1 \times r_2 = \frac{b}{a} \times \frac{d}{c} = \frac{bd}{ac} \tag{2}$$

$$r_1 \div r_2 = \frac{b}{a} \div \frac{d}{c} = \frac{bc}{ad} \tag{3}$$

r_1, r_2 は有理数，a, b, c, d は多桁数である．

n 桁と n 桁の整数の積は $2n$ 桁になるので，式 (1) の ac，bc，ad などの整数の桁数は演算のたびに倍になる．そこで有理数を構成するときに，分母，分子の最大公約数（Greatest Common Divisor, GCD）で割り既約な分数とする．開発中のプログラミング環境は，上記の a, b, c, d などの多桁の数を **longint** 型で扱う多桁数の階層を持つ．

$$z = \sum_{i=0}^{l-1} d_i r^i \tag{4}$$

ここに d_i は各桁の数である．多桁の自然数 z を，基数を $r = 2^{32}$ として 32 ビット符号なし整数型の配列に格納し，桁数 l を整数型で保持する．零は桁数が零（$l = 0$）とする．配列長は 64 から始め，不足すると動的に倍，倍と拡張する可変長とした．最大長は 32,768 としている（10 進数で約 31 万桁）[*2]．多桁

[*1] 十進 BASIC は，数学教育を目的に，文教大学の白石教授が開発中のプログラミング言語であるが，これによって有理数計算を体験することができる [1]．十進 BASIC は 5 つの計算モードを持つ．十進 15 桁，十進 1000 桁，2 進（Intel x86 倍精度浮動小数点），2 進による複素数，有理数である．BASIC の言語規格により，数値は単一の表現に統一されるので，浮動小数点数と有理数を同時に使用することはできない．

[*2] $\log_{10} 4294967296 = 9.633\cdots$ である．計算可能な最大の数は，リコンパイルで変更可能である．

数の階層（longint クラス）では，式 (1)(2)(3) の左辺の演算の本体である四則演算の関数や，GCD 関数をもつ．

有理数階層（rational クラス）はその上に構築される．有理数は 2 つの `longint` 型の数と符号で保持される．式 (1)(2)(3) の有理数演算のエントリルーチン，有理数演算を有理数型の変数を用いて `z=x+y` のように記述するためのオペレータオーバーロードの定義のほか，`min`，`max`，`floor`，`ceil` などの有理数関数や，浮動小数点数を有理数に変換する関数 `Rdset`，多項式の除算などのユーティリティルーチンを提供する．

その上に有理数 BLAS（Basic Linear Algebra Subprograms）階層（rblas クラスとその並列版をサポートする prblas クラス）をもち，数値線形代数（NLA: numerical linear algebra）アプリケーションの並列化を容易にする [4]．本稿の例題では，ヘッセンベルグ変換を行う elmhes 関数やフロベニウス変換を行う hesfrb 関数を含む．

このプログラミング環境では，我々が通常数値計算で使用している C++のプログラミング言語に，変数型として `rational` 型が追加され，`rational` 型同士の演算を `z=x+y` のように記述することができる．浮動小数点数を正確に計算するためには，倍精度数の場合は関数 `Rdset` によって `rational` 型の変数に変換してから有理算術演算を用いる．この機能により，丸め誤差の発生しない正確な四則演算が，既存の数値計算プログラミングの延長として可能になる．

3. 実対称行列の固有値の多重度の解析のアルゴリズムと数値例

固有値問題は，次の線形同次方程式の n 個の未知数 λ を決定する．

$$A\boldsymbol{x} = \lambda\boldsymbol{x}. \tag{5}$$

本稿では，係数行列 A は**実対称**とする．行列 A は倍精度浮動小数点演算で作成され，有理数に変換される．固有方程式 (5) は次式が成立したとき，非自明解をもつ．

$$\det(A - \lambda I) = 0. \tag{6}$$

式 (6) は次のように展開される．

$$(-1)^n\lambda^n + p_{n-1}\lambda^{n-1} + p_{n-2}\lambda^{n-2} + \cdots + p_0 = 0 \tag{7}$$

これを係数行列 A の特性方程式（characteristic equation）と呼ぶ．左辺の多項式の全係数は，行列 A の要素 a_{ij} に対する乗算と加減算だけで得られるので，有理数計算で正確に求められる．行列 A の係数が整数の場合は，特性多項式の係数も整数である．

非線形方程式の解は一般には複素数であるが，特性方程式 (7) は実対称行列から生成されたので解は**実数**である [5]．方程式を解く段階で，数の範囲が有理数から代数的無理数に拡大される．

浮動小数点計算で全固有値を求める場合は，ヤコビ法，ギブンス法，ハウスホルダー法などが用いられ，この場合に得られる固有値の精度は数桁から 10 数桁である．これらの方法は，面内回転や鏡映変換などの無理数を扱う変換を用いるので，有理数計算で正確な解を求める目的では使えない．一方，ガウス消去法で用いる基本変換は，四則演算だけで構成されるので，有理算術演算に適している．

本章の 1 節で特性多項式を求めるアルゴリズムを述べる．2 節で数値例を示す．3 節で因子探しのアルゴリズムについて述べる．

3.1 特性多項式を求めるアルゴリズム

行列式 (6) を展開すれば特性多項式は得られるが，素朴に展開しても多重度は分からない．A をヘッセンベルグ行列 H に変換すると，多重度が高い固有値が存在する場合は，副対角項に現れる零要素からこれを判定できる．零要素があればヘッセンベルグ行列は小行列に分けられ，その後の計算は各ヘッセンベルグ小行列に対して独立に行える．

3.1.1 ヘッセンベルグ変換

次数 n の実対称行列 A に基本変換行列 R を $(n-2)$ 回，**相似変換**（similarity transformation）の形で掛ける[*3]．

$$H = R_{n-2}\cdots R_2R_1AR_1^{-1}R_2^{-1}\cdots R_{n-2}^{-1}. \tag{8}$$

次数 5 のヘッセンベルグ行列を示す．

[*3] $A\boldsymbol{x} = \lambda\boldsymbol{x}$ の両辺に H^{-1} を掛けると $H^{-1}A\boldsymbol{x} = \lambda H^{-1}\boldsymbol{x}$ になるが，$H^{-1}H = I = HH^{-1}$ なので

$$H^{-1}A\underbrace{HH^{-1}}_{=I}\boldsymbol{x} = \lambda H^{-1}\boldsymbol{x}$$

すなわち $(H^{-1}AH)H^{-1}\boldsymbol{x} = \lambda H^{-1}\boldsymbol{x}$ となる．$H^{-1}AH = B$，$H^{-1}\boldsymbol{x} = \boldsymbol{y}$ とおけば，$B\boldsymbol{y} = \lambda\boldsymbol{y}$ の固有方程式になる．λ は同じなので，**相似変換により固有値は保存**され，**固有ベクトルは H^{-1} が掛けられる**．

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ k_2 & h_{22} & h_{23} & h_{24} & h_{25} \\ & k_3 & h_{33} & h_{34} & h_{35} \\ & & k_4 & h_{44} & h_{45} \\ & & & k_5 & h_{55} \end{pmatrix}. \quad (9)$$

対角項の下の副対角項 $h_{s+1,s}$ を k_{s+1} で表す．ヘッセンベルグ行列への変換プログラムは，数値計算法のテキスト *Numerical Recipes* に記載されており，有理数計算でも同様のプログラムで実現できるため，アルゴリズムの詳細は割愛する [6].

行列が縮重していて，複数の重複固有値が存在する場合は，副対角項に零要素が現れる（証明は省略する）．$k_{r+1} = 0$ の場合は次のように書ける．

$$H = \begin{pmatrix} H_r & B_r \\ 0 & H_{n-r} \end{pmatrix},$$

H_r は次数 r，H_{n-r} は次数 $(n-r)$ で，どちらも上ヘッセンベルグ形である．行列式 (6) は次のように計算できる．

$$\det(H - \lambda I) = \det(H_r - \lambda I)\det(H_{n-r} - \lambda I),$$

つまり，$k_{r+1} = 0$ が検出されれば，後続の計算は 2 つの小行列に対して独立に進められる．

特性多項式を求めるために，上ヘッセンベルグ小行列 H_r と H_{n-r} をフロベニウス標準形 F に変換する．相似変換操作を $(n-1)$ 回繰り返して，次の行列 X が得られる．

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & x_0 \\ k_2 & 0 & 0 & 0 & x_1 \\ & k_3 & 0 & 0 & x_2 \\ & & k_4 & 0 & x_3 \\ & & & k_5 & x_4 \end{pmatrix}.$$

フロベニウス行列は対角行列を相似変換 $F = D^{-1}XD$ の形で掛けて得られる．ここに対角行列 D は $d_1 = 1$, $d_2 = k_2$, $d_3 = k_2k_3$, $d_4 = k_2k_3k_4$, $d_5 = k_2k_3k_4k_5$, である．

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 & p_0 \\ 1 & 0 & 0 & 0 & p_1 \\ & 1 & 0 & 0 & p_2 \\ & & 1 & 0 & p_3 \\ & & & 1 & p_4 \end{pmatrix}, \quad p_i = \left(\prod_{j=0}^{n-2-i} k_{n-j} \right) x_i. \quad (10)$$

次数 5 の特性多項式が $-\lambda^5 + p_4\lambda^4 + p_3\lambda^3 + p_2\lambda^2 + p_1\lambda + p_0$ として得られる．最高次の係数は “−1” であるが n が偶数の場合は “+1” とする．次数 n の特性多項式は式 (7) の左辺である．

ヘッセンベルグ行列を経由してフロベニウス行列を求める相似変換は，1 つの行列 T にまとめられ，これを用いて相似変換の形で $F = TAT^{-1}$ と記述できる．多重度が高い場合は F はブロック対角行列となる．

$$TAT^{-1} = \begin{pmatrix} F_1 & & & \\ & F_2 & & \\ & & \ddots & \\ & & & F_m \end{pmatrix}. \quad (11)$$

特性多項式は各フロベニウス小行列 F_i から得られる特性多項式の積になる[*4].

$$p(\lambda) = \prod_{i=1}^{m} p_i(\lambda) \quad (12)$$

有理数変数を導入することにより，これまでの浮動小数点数に対する数値計算のプログラムを融通することで正確な計算を実現することが，「有理数計算プログラミング環境」の目的である．

3.2 数値例

正方形の断面をもつ四角柱で，表面の温度を与えられた場合の熱伝導問題を，断面領域で考える [7]. 図 1 に，内部のメッシュが 2×2 節点と 5×5 節点の場合を示す．メッシュ分割された (b) と (c) に 1 つの差分要素を示したが，上下左右の節点から温度勾配と熱伝導率に比例して熱が出入りする 5 点差分スキームを用いた．未知数は内部の節点にあるので，行列の行・列番号は，境界節点の番号を詰めて振られる．解析領域を $m \times m$ に分割すると，内部は $(m-1) \times (m-1)$ にメッシュ切りされ，行列のサイズは $n = (m-1) \times (m-1)$ になる．3×3 のメッシュでは内部は 2×2 になるので，行列の行・列番号では節点 6 が最初の行になり，節点 7 が 2 番目，節点 10 が 3 番目，節点 11 が 4 番目になる．

3.2.1 整数行列

熱伝導係数を 1 にすると次の行列が得られる．

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix} \quad (13)$$

[*4] 各特性多項式 $p_i(\lambda)$ は無平方（squarefree）である．

行列は物理形状の対称性のために縮重して，ヘッセンベルグ行列は次のようになる．

$$H = \begin{pmatrix} 4 & -2 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ & -2 & 4 & -1 \\ & & 0 & 4 \end{pmatrix} \qquad (14)$$

$k_4 = 0$ になるので，H は H_3 と H_1 に分かれる．大きいほうの行列から

$$H_3 = \begin{pmatrix} 4 & -2 & 0 \\ -1 & 4 & -1 \\ & -2 & 4 \end{pmatrix}, \quad X_3 = \begin{pmatrix} 0 & 0 & 24 \\ -1 & 0 & -22 \\ & -2 & 12 \end{pmatrix}$$

が得られ，特性多項式が得られ，因数分解できる．

$$p_3(\lambda) = 48 - 44\lambda + 12\lambda^2 - \lambda^3 = (2-\lambda)(4-\lambda)(6-\lambda).$$

小さいほうの小行列 H_1 から特性多項式 $p_1(\lambda) = -\lambda + 4$ が得られる．特性多項式が定義式 (6) から $\lambda^4 - 16\lambda^3 + 92\lambda^2 - 224\lambda + 192$ と得られても，それが $(2-\lambda)(4-\lambda)^2(6-\lambda)$ と 2 乗の項を含む形の因数分解を得るまでは重根の存在をいえない．

3.2.2 有理数行列

熱伝導係数を 1 から 0.1 にする．ただし 0.1 は倍精度浮動小数点数とするので，2 進数の丸め誤差の影響を受ける．倍精度浮動小数点数として生成された行列 A を有理数変換する．“0.4” は有理数では次の数値に変換される．

$$0.4 \fallingdotseq \frac{3,602,879,701,896,397}{9,007,199,254,740,992}. \qquad (15)$$

この分母と分子には，丸めのために (denominator) $\times$ $4 + 2 =$ (numerator) $\times 10$ の関係がある．2 進小数では，10 分の 1 が循環小数 $0.0\dot{0}01\dot{1}$ になるので，倍精度浮動小数点演算では丸めの影響は避けられない．行列要素の分母の最小公倍数は，36,028,797,018,963,968 と 17 桁の数になる．したがって最小公倍数（LCM）倍した行列の要素の桁数も 17 桁と大きくなるが，ヘッセンベルグ変換すると，熱伝導係数が 1 の場合と同様に，$k_4 = 0$ が現れる．

3.2.3 零固有値をもつ整数行列

熱伝導行列の境界条件を指定しないとグラフ・ラプラシアン行列（整数行列）が生成される．図 1 の (d) に分割数 $m = 4$ の場合を示した．節点数は $(m-2)^2 + 4(m-2)$ などになる．固有値の分布は次表のようになる．

λ_i					e_i	exact λ
λ_1					2.37e-16	0
λ_2	λ_3				0.43844	$0.5(5-\sqrt{17})$
λ_4					0.62771	$0.5(7-\sqrt{33})$
λ_5		λ_6	λ_7	λ_8	0.99999	1
λ_9					2.99999	3
λ_{10}	λ_{11}				4.56155	$0.5(5+\sqrt{17})$
λ_{12}					6.39228	$0.5(7+\sqrt{33})$
H_7	H_2	H_1	H_1	H_1		

多重度は 4 で，零固有値をもつが，浮動小数点計算では零固有値は正確に零としては得られず $e_1 = 2.37\cdots \times 10^{-16}$ となる．

3.3 因子探しの方法

整数行列の特性多項式の係数は整数で得られるが，固有値は代数的無理数になる．ここでは因数分解は，高校数学の因数分解と同様に「整式を整数の範囲で因数分解」することを考える．

図 1 の (c) のように 5×5 のメッシュに切ると，行列のサイズは $n = 25$ になる．ヘッセンベルグ行列は 5 つの小行列 $H_{13}, H_9, H_1, H_1, H_1$ に分かれる．H_9 から得られる特性多項式の係数は 6 桁に及ぶ．固有値の分布を次表に示す．

λ_i					e_i	exact λ
λ_1					.53589838	$2(2-\sqrt{3})$
λ_2	λ_3				1.2679491	$3-\sqrt{3}$
λ_4					2.0	2
λ_5	λ_6				2.2679491	$4-\sqrt{3}$
λ_7	λ_8				3.0	3
λ_9	λ_{10}				3.2679491	$5-\sqrt{3}$
λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	4.0	4
λ_{16}	λ_{17}				4.7320508	$3+\sqrt{3}$
λ_{18}	λ_{19}				5.0	5
λ_{20}	λ_{21}				5.7320508	$4+\sqrt{3}$
λ_{22}					6.0	6
λ_{23}	λ_{24}				6.7320508	$5+\sqrt{3}$
λ_{25}					7.4641016	$2(2+\sqrt{3})$
H_{13}	H_9	H_1	H_1	H_1		

多重度 5（$\lambda_{11} = \cdots = \lambda_{15}$）が存在する．この場合は H_9 から得られる多項式は H_{13} から得られる多項式を割り切るので，H_{13} の因子は 4 次多項式を分解して得られる．11 個の固有値が有理数で，14 個の固有値が無理数である．

表の第 2 列の浮動小数点計算で得られた近似値を眺めると，例えば 2 行目の 1.2679491 と 8 行目の 4.7320508 を加えれば 5.9999999 になることに気付

く．そこで，因数分解の代替として，浮動小数点計算で得られた倍精度の近似固有値を利用した因子探しプログラムを作成する演習問題が考えられる．

特性多項式が因数分解されて2次の因子（divisor）を持つとする．モニック2次方程式 “$x^2+rx+s=0$” に対する根と係数の関係公式は “$\alpha+\beta=-r$”, “$\alpha\beta=s$” なので，2つの候補 e_i と e_j を近似固有値から選び，$r=-(e_i+e_j)$ と $s=e_i\times e_j$ を作成し，

$$\mathrm{Floor}(|r|)-|r| \ \text{ or } \ \mathrm{Ceil}(|r|)-|r|$$
$$\text{and}$$
$$\mathrm{Floor}(|s|)-|s| \ \text{ or } \ \mathrm{Ceil}(|s|)-|s|$$

が許容誤差の範囲内にあるかどうかテストする[*5]．このテストを**整数性判定**と呼ぶことにする．範囲内であれば r と s を `rational` 型の変数に整数として取り出し，除算 $p_r(\lambda)\div(x^2+rx+s)$ を有理算術演算で試み，剰余が零ならこの2次多項式は特性多項式の因子であることが分かる．

公式（Vieta's formula）は，次数 $n=5$ のモニック多項式 $p_0x^5+p_1x^4+p_2x^3+p_3x^2+p_4x+p_5$ の場合は，根を α_1 から α_5 とすると次のようになる．

$$\begin{aligned}
p_0&=-1\\
p_1&=\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5\\
p_2&=-(\alpha_1\alpha_2+\alpha_1\alpha_3+\alpha_1\alpha_4+\cdots+\alpha_4\alpha_5)\\
p_3&=\alpha_1\alpha_2\alpha_3+\alpha_1\alpha_2\alpha_4+\alpha_1\alpha_2\alpha_5+\cdots+\alpha_3\alpha_4\alpha_5\\
p_4&=-(\alpha_1\alpha_2\alpha_3\alpha_4+\alpha_1\alpha_2\alpha_3\alpha_5+\cdots+\alpha_2\alpha_3\alpha_4\alpha_5)\\
p_5&=\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5
\end{aligned}$$

n 次多項式の r 次の係数は次のようになる．

$$p_r=(-1)^{(r+1)}\sum_{i1,i2,\cdots,ir}^{{}_nC_r}\alpha_{i1}\alpha_{i2}\cdots\alpha_{ir}\qquad(16)$$

倍精度浮動小数点数の有効桁数は16桁程度である．$p_1=\alpha_1+\alpha_2$ で $e_i=\alpha_i\pm\varepsilon_i$ ただし $\varepsilon_i\geq 0$ とすると，倍精度浮動小数点計算で得られる近似値を使って，倍精度演算で調べられる係数の桁数も16桁を超えられない．この理由を Floor 関数の場合で記すと，$p_1\leq e_i+e_j<p_1+1$ の範囲でなくては整数性は判定できないので，桁数の大きな係数に対しては，相対誤差 $\dfrac{\varepsilon_i}{e_i}$ も小さくしなくてはならないからである．したがって整数性判定には，求める係数の桁数に応じて近似固有値の精度を上げる必要がある．

[*5] 最高次の係数 $p_0=1$ の多項式をモニック多項式という．

有理行列の場合は，行列要素の分母の最小公倍数（LCM）を掛けて整数行列に変換し，桁数の多い特性多項式に対して，大きな固有値を用いて，因子探しを行う[*6]．この場合の整数性判定は，倍精度で得られた近似値を，有理算術演算による2分法によって精度改良する方法をとる[*7]．有理算術演算であるから精度改良は桁数の許す範囲で実現できる．精度改良を多倍精度演算で行うのがよいか，有理算術演算で行うのがよいかを考える上で興味深い題材である．

4. 因子探しプログラム

熱伝導係数が1で，分割数が12以下の整数行列の場合は，固有値が0.1から10の範囲に入り，**整数性の判定を倍精度浮動小数点演算**で可能なので，演習問題としても難易度は高くない．この場合の処理の概要は次のようになる．

浮動小数点計算で行列 A の固有値をヤコビ法で n 個求める．有理算術演算で $A\rightarrow H$ 変換を行う[*8]．以下，各ヘッセンベルグ小行列 H_k に対して

- 小行列 H_k をフロベニウス変換して特性多項式 $p(\lambda)$ を生成する（次数 `nsize`）．
- 既知の因子多項式がある場合はこれらで特性多項式 $p(\lambda)$ を割り，割り切れれば商多項式で $p(\lambda)$ を置き換える（`nsize` を小さくする）．
- 近似固有値の $p(\lambda)$ に対する包含（inclusion）を成立させる[*9]．$p(\lambda)$ が1次式なら，$p(a)\cdot p(b)<0$，n 次式の場合，$p(a_i)\cdot p(b_i)<0$ の区間 (a_i,b_i) を n 個見つける．
- 近似固有値 e_i の整数性を調べ，整数らしければ e_i を有理数型の整数 e として抽出し，1次式 $(-\lambda+e)$ で $p(\lambda)$ を割り，割り切れれば因子として保存し，商多項式で $p(\lambda)$ を置き換える
- 次数 $d=2,3,\ldots$ について以下の **2次以上の因子探し**を行う[*10]．

[*6] 10分割では，H_{51},H_{41} と8つの H_1 の積に分かれるが，H_{41} から得られる次数41の特性多項式の定数項は660桁になる．

[*7] 2分法は高校の『数学 B』にある．

[*8] ヘッセンベルグ変換ルーチン elmhes や，フロベニウス変換ルーチン hesfrb は rblas クラスに用意されている．

[*9] 解を1つだけ含む区間 (a,b) を見つけることを「包含」と呼ぶ．

[*10] この処理は `while(d<=nsize)` ループで書かれ，内側に `for` ループによる「組合せループ」がある．「組合せループ」は n 個の固有値のうち包含が成立した `ncand` 個から d 個を選ぶ ${}_{ncand}C_d$ 個の「組合せ」の反復を行うが，因子多項式が見つかると `nsize` が次数 d だけ小さくなる．したがって，内側のループを終える場合と，継続する候補の組合わせを探す場合に分岐させて，すでにチェック済みの組合せは再チェックしないようにすると高速化できる．こ

– d 個の近似固有値から，根と係数の関係公式の 1 次項 $\sum_j^d e_j$ を求めて整数性を先行判定する．
– 先行判定に通過すれば，式 (16) によって他の係数を求め，それらの整数性を調べる．
– 全係数が整数と見なせれば，係数を有理数型の整数として抽出し，それらによって整多項式 $v(\lambda)$ を作り $p(\lambda)$ を割る．$p(\lambda) = v(\lambda)q(\lambda)$ なら割り切れて，$v(\lambda)$ を因子多項式とし保存し，商多項式 $q(\lambda)$ で $p(\lambda)$ を置き換える．

以上が整数行列に対する処理であるが，熱伝導係数 0.1 の浮動小数点行列を有理数変換した行列の場合は，全分母の LCM 倍した桁数の大きい整数行列を扱う．固有値も LCM 倍されるので，**近似固有値の精度を改良**する処理が加わる．改良された固有値の有効桁数は多いので，**整数性判定にも有理算術演算**を用いる必要がある．

4.1 計算結果

計算は，Linux ワークステーション（Intel の Xeon 2.60 GHz）の 1 スレッドで行った（乗算に FFT は使用しないで測定した）．

4.1.1 熱伝導行列

熱伝導係数が 1 の整数行列と 0.1 を 10 進小数と IEEE 754 標準の倍精度浮動小数点数で与えた場合の有理行列の因子探しを計算した．結果は 3 者とも同じ形に分解された．行列がヘッセンベルグ小行列の積に分かれる状態を表 1 に示した．表では $4-\lambda$ の 1 次式の因子が複数現れる場合を “3*1” のように表した．すべてのケースで $\lambda = 4$ の固有値が分割数に一致する多重度を持つ．内部分割が 5×5 以下は，整数の範囲での因数分解が 2 次以下の因子に分解される．6×6 の場合は 3 次式を含み，7×7 の場合は 4 次式を含む．因数分解が得られると，どの近似固有値がどの因子に繋がるかが分かるので，特性多項式の構成を考えることができる（固有値を数値として何 10 桁求めてみても，これは分からない）．表 1 の最下段から，9 分割以下，および 11 分割では 4 次多項式以下に分解されるので，べき根による公式を正確な解と考えれば，正確な固有値を得ることができる．10 分割は 5 次多項式以下に，12 分割は 6 次多項式以下に分解される．

10 分割を例に計算の進行を紹介する．ヘッセンベルグ行列 H は，H_{51}, H_{41} と 8 つの H_1 に分割される．H_{41} の特性多項式は H_1 の特性多項式で割り切れて 40 次式になり，8 つの 5 次式で割り切れる．${}_{40}C_5 = 658,008$ であるが，割り切れる組合せは，66,903 番目に現れる．`nsize` が減り ${}_{35}C_5 = 324,632$ で，割り切れる組合せは，36,980 番目に現れる．${}_{30}C_5 = 142,506$ で，割り切れる組合せは，18,032 番目に現れる．このように比較的少ない反復回数で因子が見つかる．

H_{51} の特性多項式は H_{41} の特性多項式で割り切れて 10 次式になり，2 つの 5 次式で割り切れる．小行列から得られる特性多項式の次数は大きく，近似固有値の精度も 80 桁以上が要求されるが，5 次の因子多項式で割り切れるので計算時間は短い．

4.1.2 グラフ・ラプラシアン行列

表 1（下）に m を 4 から 12 までの結果を示すが，$m = 7$ を例外として，多重度 4 は最後の 3 つの小行列と H_a に現れる．また H_a から得られる特性多項式の定数項は零である．熱伝導行列と比較すると，次数の高い因子のために計算時間が長くなる．

10 分割の場合は，H_{52}，H_{23}，H_{18} と 3 つの H_1 に分割され，H_{18} の特性多項式が 2 つの 9 次の因子に分解される．H_{23} は分解されない．H_{52} の特性多項式は既知の因子のうち 1 次式，2 つの 9 次式，23 次式で割り切れて 10 次式になるが，1 次式（零固有値）と 4 次，5 次式に分解される．したがって計算時間は 23 次式まで次数を 1 次ずつ上げて行う整数性判定と，23 次式の係数の算出に大部分を費やしている[*11]．近似固有値の精度は，23 次の特性多項式の係数の最大が約 12 桁程度であるため，固有値（零から 7.77 の間）の精度改良も 30 桁程度で間にあう．しかし，因数分解されない特性多項式があるため，整数性判定の組合せループの反復回数が爆発して，計算時間が長くなる．

4.2 プログラムの高速化

有理算術演算と浮動小数点演算を使用して，数値式に素朴にプログラミングしたオリジナルのプログラムに対して，簡単なチューニングを施す．HPC プログラミングの教材としては，計算環境の下位の階

の分岐に，解答例には goto 文を用いた．HPC プログラミングでは，コンパイル後のアセンブラ・コードを読む機会もあるので，分岐命令を読めるようになることは，HPC では重要である．「計算機ハードウェアの理解できる唯一の言語は機械語」である [8].

[*11] $\sum_{r=2}^{n} {}_nC_r$ 回なので，23 次の全組合せは 8,388,584 回の反復を行う．指数時間アルゴリズムである．

層（longint クラス，rational クラス，rblas クラス）は与えられたものと考えて改良を加えずに，整数性判定，根と係数の関係公式のプログラミング，近似固有値の精度改良を対象とする．有理算術演算は，浮動小数点演算と比較するとチューニングマージンが大きいので，逐次処理の計算時間 10 分の 1 を目標としてチューニングを行うことで，有理算術演算の高速化技術と背景の数学を学ぶ．

4.2.1 整数性判定

有理数型で得られた候補固有値から 1 次の係数を求めて先行判定する部分で，演算に使用する桁数を少なくする．具体的には，固有値の小数点以下だけを抽出して倍精度浮動小数点数に変換して，浮動小数点演算によって判定する．

4.2.2 根と係数の関係

公式 (16) によって次数ごとに求めると乗算回数が多くなるので，図 2 の順序に改めた．アルゴリズムは再帰呼び出し（recursive call）を使う[*12]．

4.2.3 計測結果

グラフ・ラプラシアン行列で分割数が 10 と 11 の場合を示す（単位：秒）．

m	チューニングなし	チューニングあり
10	1,152	192
11	94,379	10,599

10 分割の ${}_{23}C_{23}$ と 11 分割の ${}_{28}C_{28}$ の違いは指数時間アルゴリズムであるため 30 倍以上あり計算時間に大きな差が現れる．

チューニングの効果は 10 倍に届かなかった．浮動小数点演算との大きな違いは，桁数を削減する効果にある．2 項目のチューニングを行ったが，精度改良した固有値の分母と分子の桁数を，2 分法そのままではなく，最良近似分数に変換すると高速化に効果があると考えられる．これを今後の課題としたい．このようにいろいろな例題を通して必要機能を洗い出し，プログラミング環境が装備すべき関数を充実させている．

5. まとめと今後の課題

本論文では，既存の数値計算プログラミングの環境に有理算術演算を含めることで実現可能な計算の例として，近似固有値を使用した対称行列の特性多項式の因数分解を代替する因子探しのプログラムを紹介した．倍精度浮動小数点計算で得られた近似値を，必要な精度に改善して，整数性をチェックする単純アルゴリズムであるが，プログラミング環境が正確な計算をサポートすると，ちょっとした工夫で数式処理システムの代替も可能なケースがあることが確認された．

「有理数計算プログラミング環境」の目的のひとつに「プログラミング演習を通して行う数学教育のツール」をあげられる．プログラミング環境は目的に特化した製品によって専用化されたシステムを使用する傾向が強くなってきており，近年，学生の汎用プログラミング言語離れが顕著である．数値計算は数式処理システムや MATLAB や SciLab のようなインタプリタ形式のスクリプト言語が利用される．これではなかなか HPC プログラミングに届かない．汎用プログラミング言語を授業で教えても，日常的なツールとして使う学生はいない（学生は使わないので忘れてしまう）．一方，言語そのものが高度になったので，即戦力になるようなプログラミング技術を授業で教えるには時間が足りない．本稿で紹介した有理算術演算を四則演算のレパートリーに加えたコンパイラ型のプログラミング言語なら，数学知識の習得にプログラミングを入れられるので，数値計算プログラミングの魅力を復活させられるかもしれない．現時点で保有している数学知識を，プログラミング技術で花開かせる経験を通して，HPC プログラミング好きになる学生を増やせればと考えた．

謝辞 本研究の一部は科学研究費補助金・基盤 (C) 課題番号 25330145「有理数計算ライブラリの並列化と誤差診断ツールの開発」から支援を頂いた．記して謝意を表す．

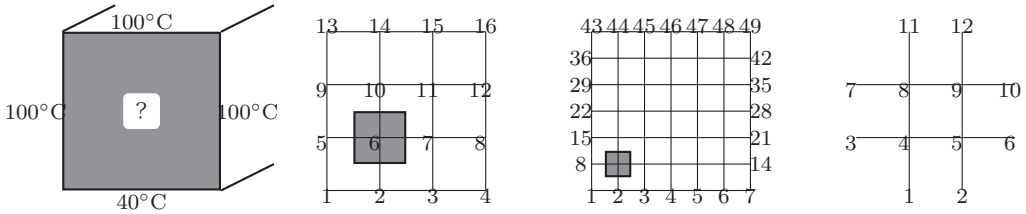
表 1 分割数 m と小行列サイズと因子多項式の最高次数 d
熱伝導行列（上）とグラフ・ラプラシアン行列（下）

m	3	4	5	6	7	8	9	10	11	12
n	9	16	25	36	49	64	81	100	121	144
H_a	5	9	13	19	25	33	41	51	55	73
H_b	3	3	9	13	19	25	33	41	37	61
H_c	1	3	3*1	4*1	5*1	6*1	7*1	8*1	15	10*1
H_d		1							7	
H_e									7*1	
d	2	2	2	3	4	3	4	5	4	6

m	4	5	6	7	8	9	10	11	12
n	12	21	32	43	60	77	96	117	140
H_a	7	12	18	25	33	42	52	63	75
H_b	2	6	7	15	14	32	23	42	34
H_c	3*1	3*1	4	1	10	3*1	18	9	28
H_d			3*1	1	3*1		3*1	3*1	3*1
H_e				2					
H_f				1					
d	2	4	7	10	14	18	23	28	34

[*12] 「有理数計算プログラミング環境」では，rational 型の変数は動的に配列要素を拡張する．プロシージャフレームに拡張されるローカル変数があると，再帰関数が機能しないので，稼働を確認する目的も含めてプログラミングした．プロシージャフレーム（またはアクティベーション・レコード）については教科書を参照されたい [8].

Vol.2015-HPC-149 No.10
2015/6/26



(a) 境界条件　(b) 解析領域 3×3 分割　(c) 解析領域 6×6 分割　(d) 4×4 分割

図 1　熱伝導行列とグラフ・ラプラシアン行列

$$p_2 = -(\underbrace{\alpha_1\alpha_2}_{1} + \underbrace{\alpha_1\alpha_3}_{9} + \underbrace{\alpha_1\alpha_4}_{13} + \underbrace{\alpha_1\alpha_5}_{15} + \underbrace{\alpha_2\alpha_3}_{16} + \underbrace{\alpha_2\alpha_4}_{20} + \underbrace{\alpha_2\alpha_5}_{22} + \underbrace{\alpha_3\alpha_4}_{23} + \underbrace{\alpha_3\alpha_5}_{25} + \underbrace{\alpha_4\alpha_5}_{26})$$

$$p_3 = \underbrace{\overbrace{\alpha_1\alpha_2}\,\alpha_3}_{2} + \underbrace{\overbrace{\alpha_1\alpha_2}\,\alpha_4}_{6} + \underbrace{\overbrace{\alpha_1\alpha_2}\,\alpha_5}_{8} + \underbrace{\overbrace{\alpha_1\alpha_3}\,\alpha_4}_{10} + \underbrace{\overbrace{\alpha_1\alpha_3}\,\alpha_5}_{12} + \underbrace{\overbrace{\alpha_1\alpha_4}\,\alpha_5}_{14} + \underbrace{\overbrace{\alpha_2\alpha_3}\,\alpha_4}_{17} + \underbrace{\overbrace{\alpha_2\alpha_3}\,\alpha_5}_{19} + \underbrace{\overbrace{\alpha_2\alpha_4}\,\alpha_5}_{21} + \underbrace{\overbrace{\alpha_3\alpha_4}\,\alpha_5}_{24}$$

$$p_4 = -(\underbrace{\overbrace{\alpha_1\alpha_2\alpha_3}\,\alpha_4}_{3} + \underbrace{\overbrace{\alpha_1\alpha_2\alpha_3}\,\alpha_5}_{5} + \underbrace{\overbrace{\alpha_1\alpha_2\alpha_4}\,\alpha_5}_{7} + \underbrace{\overbrace{\alpha_1\alpha_3\alpha_4}\,\alpha_5}_{11} + \underbrace{\overbrace{\alpha_2\alpha_3\alpha_4}\,\alpha_5}_{18})$$

$$p_5 = \underbrace{\overbrace{\alpha_1\alpha_2\alpha_3\alpha_4}\,\alpha_5}_{4}$$

図 2　根と係数の関係公式の計算順序

参考文献

[1] *http://hp.vector.co.jp/authors/VA008683/.*
[2] 寒川 光, 有理数計算による実対称行列の正確な 3 重対角化による固有値の高精度計算, *HPCS2013* **論文集**, pp. 11–22, 2013.
[3] D. Knuth., *The Art of Computer Programming, Volume 2, Third Edition*, Addison-Wesley, 1998, 有澤　誠，和田英一監訳：The Art of Computer Programming, Third Edition，株式会社アスキー，2004.
[4] 寒川 光, 有理数線形代数計算における有理数 blas の提案, *HPCS2014* **論文集**, pp. 57–64, 2014.
[5] J. H. Wilkinson, *Algebraic Eigenvalue Problem*, Oxford University Press, 1965.
[6] W. H. Press, Teukolsky S. A., Vetterling W. T., and B. P. Flannery, *Numerical Recipes in Fortran, second edition*, Cambridge University Press, 1992.
[7] 寒川 光, 藤野清次, 長嶋利夫, 高橋大介, *HPC* **プログラミング**—*ITText* **シリーズ**, オーム社, 2009.
[8] Patterson, D. and Hennessy, J. *Computer Organization & Design: The Hardware/Software Interface, fourth edition*, Morgan Kaufmann Publishers, Inc., 2011, 成田光彰訳: コンピュータの構成と設計–ハードウェアとソフトウェアのインターフェース–第 4 版,（上）（下），日経 BP 社, 2011.