Chapter 1

序章

1.1 なぜ固有値テンプレートなのか？

大規模な科学技術計算では、回路の周波数応答から地震時の建物の応答、分子のエネルギーレベルに至るまで、行列の固有値と固有ベクトルを求める必要とすることが多い。固有値問題を定式化する数学的手法は数多くあり、それらを数値的に求める方法はさらに多く存在する。この本はその時々の固有値問題に応じた最適な数値解法をみつける一つの指針となればという思いから作成した。

現在の最新技術をもってすれば多くの固有値問題、なかでも行列サイズが小規模から中規模の密行列に対する優れた手法が存在する。これらのアルゴリズムは MATLAB [319][1]のようなプログラミング環境や、LAPACK [12] のようなライブラリや、その他多くの商用あるいは公開パッケージで入手可能となっている。しかしながら、超大規模かつ (特に) 疎行列固有値問題については、唯一の最適な方法は存在しない。問題サイズ、行列の数学的特性に依存した複雑な手法、効率と精度のトレードオフのことを考え、与えられた問題に対する最良の方法を見つけようとすると、専門家でも苦労するし、一般のユーザは立ち往生する。優れたオンライン・サーチ機能やソフトウェア貯蔵サイトが存在する。際だっているのは GAMS (Guide to Available Mathematical Software)[2]や NETLIB[3]である。これらを使えばライブラリ名や、サブルーチン名や、キーワード、そして数値計算のテーマ別分類にもとづいた検索ができる。しかしこれらも、ある特定の問題に対してどの方法が最適であるかについての指針を与えてくれるわけではない。結果として、本書の著者やこの分野の専門家が、一体どのアルゴリズムを選択したらよいのかというアドバイスを求められることになる。このような現状を鑑み、われわれの知っていることができるだけ広くに行き渡るようこの本に収めようと決めた。

本書の全体構成は以下のようになっている。まず、どの計算法が読者にとって最適かを導くための**決定木** (decision tree) を考えだした。木の一つ一つのノード（節）には、問題の数学的構造、どんな量を求めたいのか、行列に対して施す演算にかかるコストといった質問が用意されている。決定木の葉の部分には、推奨されるアルゴリズムや入手可能なソフトウェアへのポインタといった**テンプレート** (template) が示される。テンプレートはアルゴリズムに帯する高水準の記述で、どのように動くのか、効率や精度をコントロールするためにどんなパラメータを調整できるのか、出力をどのように解釈したらよいのかといった理解の助けとなる。この構造については、以下でより詳細に扱う。

本書のもともとのモデルは “*Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*” [41] であった。“*Templates*” 本（本書と共通の著者が数名いる）では線形方程式を解くためになるべく簡単な反復法を選べるよう、決定木やテンプレートが用意されている。本来、固有値問題は線形方程式より

[1]MATLAB は The MathWorks, Inc. の登録商標です。
[2]`http://gams.nist.gov`, developed by NIST (National Institute of Standards and Technology).
[3]`http://www.netlib.org`

も難解な問題であるから、扱う固有値問題の種類によってはわれわれの理解しているレベルとは大きく違ってくる。さらに、アルゴリズムはもっとずっと複雑である。そこで、以前の “ Templates” 本とは対照的に、それぞれのテンプレートに対して、共通のスタイルや共通のプログラミング言語による完全な実装を示そうとはしていない。そのかわりに、それぞれのアルゴリズムの高水準なテンプレートを示し、プログラミング言語が何であれ、入手可能で優良なパブリック・ドメインの実装の有りかを紹介する。これらのプログラムはすべて、われわれが本書で ETHOME と呼んでいるウェブサイト（URL:`http://www.netlib.org/etemplates/.`）を通じてアクセスできる。このウェブサイトは、よりよいアルゴリズムが新たに入手可能になる度に更新していくつもりである。

本書で扱われる固有問題はたいへん幅の広いものである。広く知られていて、十分に研究し尽くされたアルゴリズムもあれば、一方でこれから集中的に研究する必要があるものもある。ゆえに、紹介のスタイルには違いが生じてくる。よく知られた領域でブラックボックス化された優れた解法の説明は簡潔かつ基本的に非エキスパート向けに書かれている。しかし、その逆に最近の研究成果をレビューするような領域もある。

ほとんどの節で、その節の著者をはっきりと示す。これは貢献という理由だけでなく、テキストに対する個人的な責任を反映させるためでもある。すべての節には十分な参考文献を用意し、固有値問題についての詳細知識と専門性の溢れる肥沃な世界への入り口としての役目を担わせている。

1.2 対象とする読者

このテンプレートは固有値計算ツールを必要とする次の三つの読者層のニーズに応えられように意図している：

- 学生と教員：シンプルであるが一般に効果的なアルゴリズムを必要とし、また容易に説明や理解ができる形で最新の開発状況について知りたがっている者
- 一般のエンジニアや科学者：使いやすく、信頼性があり、さらに効率的なソフトウェアを渇望する者
- ハイ・パフォーマンス・コンピューティングのコミュニティ：その研究分野の中で、最大サイズかつ最大難易度のアプリケーションを解こうとするエンジニアや科学者。そういったユーザは処理速度や問題に対する信頼性を上げようとするためアルゴリズムの詳細に一刻も早くアクセスしたいと考えている

こんなに多岐にわたる人々のニーズにたった一冊の書物で応えるのは困難に思えるかもしれない。それにもかかわらず、テンプレートには、すべての人に様々な形で役立つよう、情報をまとめられると信じている。たとえば、学生には高水準のアルゴリズム記述から、その解法がどのように機能するものかという基本的な理解を得られること期待している。一般のエンジニアや科学者は決定木を見て、解かねばならない問題に適したソフトウェアへのポインタを見つけだすであろう。最後に、ハイ・パフォーマンス・コンピューティングのコミュニティは、最難関の問題を解く手助けとして、性能のチューニングや計算結果の精度に関するアドバイスとして利用できるだろう。

1.3 決定木を用いたテンプレートの選び方

固有値問題やそれに関連した数値アルゴリズムは次の特性によって分類される：

1. **問題の数学的特性**: 固有値問題はエルミート（実対称、自己随伴）なのか、あるいは非エルミートなのか？一つの行列に対する標準固有値問題であるのか、それとも二つの行列からなる一般固有値問題であるのか？これら以外の問も合わせて、第 2 章では固有値問題を六つの数学的特性によって分けている。これが決定木の最上位となる。第**??**章から第**??**章までの各章ではこれら六つを一つずつ扱う。

2. **要求されるスペクトル情報**: 一番小さな固有値だけがほしいのか、スペクトルの両端の二、三の固有値が必要なのか、スペクトル “内側” の固有値の部分集合なのか、それともほとんどすべての固有値がほしいのか？固有値以外に対応する固有ベクトルや、不変部分空間、それ以外の量がいるのか？どのくらいの精度で必要なのか？

3. **使用可能な演算とそのコスト**: 行列要素すべてを配列に格納できるのか、またそれらに対して相似変換を施すことは可能か？線形方程式を解く際に、その行列（またはシフト行列）に対して直接法が使えるのか、それとも反復法が使えるのか？ベクトルに行列をかけることしかできないのか、その転置行列もかけられるのか？これらの演算のうちいくつかが使えるとして、それらの相対的なコストはどれくらいか？

要求されるスペクトル情報、使用可能な演算やそれにかかるコストに基づいて、第**??**章から第**??**章までの最初の節で紹介するアルゴリズムから一つを選定するよう薦めている。ここで推薦されたものは表形式にまとめられていることもある。表**??**、**??**、そして**??**を見よ。

最新の技術をもってしても、ユーザが投げかける固有値問題すべてに対して効率的なアルゴリズムがあるわけではない。たとえば、ユーザが n 行 n 列の行列 A の複素平面上すべての固有値のうち、虚軸に一番近い固有値をすべてを求めたいが、ただ A に対してはベクトル積しか許されていないとする。この場合、効率的で（すなわち n 回よりもずっと少ない行列ーベクトル積で済む）しかも同時に信頼性のある（すなわちユーザの指定した虚軸からの距離内の全固有値を見つけることが保証されるか、あるいはほぼ保証される）アルゴリズムは存在しない。このような制限事項については本文中で議論する。

1.4 テンプレートとは何か？

テンプレートには次の情報のうちいくつかあるいはすべてが含まれている：

- アルゴリズムの高水準な記述
- 入力や予想される計算時間、スペース、その他必要なリソースを含むアルゴリズムが有効となる条件についての記述。競合する別アルゴリズムについても参照する
- そのアルゴリズムをいつ使うべきかだけでなく、プログラム改良の可能性、ユーザによるチューニング可能なパラメータについての記述
- 精度を評価する方法
- 簡単な場合と困難な場合の両方の数値例
- 数種の言語、あるいは数種のコンピュータアーキテクチャ向け、部分的あるいは完全に実装したプログラムへのポインタ
- 追加情報がほしい人のためにテキストや雑誌論文へのポインタ

このような記述はアルゴリズムを効率的に使いこなすためには詳細情報が必要な反復法に使われる。われわれはあえて “ニューメリカルレシピ (numerical recipe)” と呼ぶ代わりに “テンプレート (template)” という言葉を使うことにした。これは、それらの手法が当てずっぽうに使うことができるものではないということ、むしろそうではなく、問題とアルゴリズムの両方の知識が問題を効率的に解くには必須であるということを強調するためである。小規模あるいは中規模の問題に最適な “ブラックボックス” 的な方法については、詳細をやや省略して紹介する。

1.5 本書の構成

本書は 11 章に分かれている。本書に現れる記号や略語については xvii ページをご覧いただきたい。そこに並べられた記法については、テキスト内で自由に使う。

第 1 章では執筆の動機、そして他章の概略について述べている。

第 2 章では本書で議論するすべての固有値問題についてざっと目を通す。もし本書を初めて読むのであれば必読である。なぜならば固有値問題に現れる基本用語や定義について書いてあり、さらにすべての固有値問題を六つのタイプに分類しているからである。これが決定木のいわば最上位である。第**??**章から第**??**章までは六つのタイプの固有値問題の詳細について議論する。

第**??**章では大規模固有値問題のほとんどのアルゴリズムに必要な二つの数学原理、部分空間上への射影、スペクトル変換の概要を説明する。

第**??**章はエルミート行列の固有値問題 $Ax = \lambda x$ を扱う。A はエルミート行列であり、$A = A^*$ を満たす（A が実行列の場合、これは A が対称ということである）。第**??**章から第**??**章までのいずれの章でも、まず最初にアルゴリズムの選定法と変換方法（小規模から中規模の密行列向けの標準的方法）について簡単に議論するためのアドバイスから始める。それからよく使われる反復法の一つ一つに対し、詳細の盛り込まれたテンプレートを用意する。一番わかりやすいだろうエルミート行列の固有値問題を例にとると、それにはべき乗法や、逆反復法、レーリー商反復法、部分空間反復法、ランチョス法、陰的リスタートランチョス法、帯ランチョス法、そしてヤコビ・ダビッドソン法がある。各章はそれらの安定性や精度の評価をもって終わる。

第**??**章では一般化エルミート行列の固有値問題 $Ax = \lambda Bx$ をとりあげる。$A = A^*$ かつ $B = B^*$（つまりエルミート行列）であり、さらに B は正定値である。第**??**章内の議論に類似する形でこの固有値問題解法についても述べられている。

第**??**章は特異値分解 $A = U\Sigma V^*$ についてである。特にこの分解の中でも、選択された箇所のみで計算を行う大規模疎行列に対する方法について強調して議論する。本章内の議論も第**??**章に合わせて行われる。

第**??**章では非エルミート行列の固有値問題について紹介する。つまり、$Ax = \lambda x$ で、A が一般の正方行列の場合である。本章の核となるのは、アーノルディ法、非エルミート・ランチョス法、ヤコビ・ダビッドソン法についての節である。これには陰的リスタート、ブロック、帯変形も含まれる。

第**??**章は一般化非エルミート行列の固有値問題 $Ax = \lambda Bx$ についてである。これには**正則な場合** (regular case) と**特異な場合** (singular case) の二つが含まれており、正則な場合は A, B が n 行 n 列で n 個の固有値を持っていて、A そして B の成分の連続関数になっている場合である。一方で特異な場合は、固有値が n 個より少ない場合や、固有値が不連続である場合、または（A も B も正方行列ではない場合）ひとつも固有値を持たない場合である。正則の場合、斜影ヤコビ・ダビッドソン法や有理クリロフ法、そして対称な不定対向けのランチョス法の特殊変形についても議論する。

第**??**章では非線形な固有値問題に二種類をとりあげる。まず、三つあるいはそれ以上の行列によって定義される多項式固有値問題についてとりあげる。それから、直交制約のある非線形固有値問題について議論を移す。

第 10 章ではすべてのアルゴリズムに共通の疎行列の表現や、逐次計算と並列計算における計算方法について説明する。これには標準的な疎行列格納形式の一覧や、高速な行列ーベクトル積のアルゴリズム（“BLAS”）であるとか、直接法による線形ソルバのまとめ、反復法による線形ソルバのまとめが含まれ、さらにどのように並列化を見いだすかについての簡単な議論も準備している。

最終章**??**章では最新の研究テーマでもあり、まだ inxeact-法ならびに前処理技法に焦点を当てる。反復法は、処理速度を改善したり、高速な収束を保証するため、前処理の過程に頼っているものも多い。

本書では、多くの重要なトピックを除外せざるを得なかった。付録では、本書では扱われなかったいくつかのテーマをリストアップし、それらのテーマをより深く追究したい読者のために参考文献も与えている。

（宮崎 佳典, 2004.10. 1, 2019. 8. XX）