実対称行列の連続な固有値分解追跡手法に関する数学的考察

1. 緒言

パラメータに依存する行列 $A(t)$ の固有値および固有ベクトルを連続的に追跡する問題は、信号処理、構造解析、量子力学における断熱発展など、多岐にわたる分野で現れる。標準的な数値計算アルゴリズム（例: QR法）は、各パラメータ点 t において独立に行列を対角化するため、固有値が接近または交差する点において、固有ベクトルの順序や符号の対応付けが不連続になる問題が生じる。

この問題に対処するため、固有値分解そのものを微分可能な多様体上の曲線とみなし、その時間発展を常微分方程式（ODE）として定式化するアプローチが存在する。このアプローチは、数値積分により、原理的に連続な固有対の軌道を得ることを可能にする。

本レポートでは、追跡対象が実対称行列 $A(t)$ である場合に限定し、このODEベースのアプローチを詳述する。具体的には、まず発展方程式を厳密に導出し、それを数値的に解くための連立一次方程式へと分解する。次に、実装に採用した**「積分後一括事後補正 (integrate-then-correct)」**方式について述べる。この方式は、まずODE積分によって連続だが数値誤差を含む「ガイド」となる軌道を生成し、その後、各時刻で別途計算した厳密な固有値分解を、このガイドに沿って並べ替えることで、最終的に連続性と代数的正確性の両方を満たす結果を得るものである。

2. 数学的定式化

本節では、実対称行列 $A(t)$ の固有値分解の時間発展を記述する常微分方程式を導出する。

2.1. 発展方程式の導出

追跡対象となる実対称行列 $A(t)$ の固有値分解は、全ての時刻 t において以下のように記述される。

$$A(t)=Q(t)\Lambda(t)Q(t)^T$$

ここで、$Q(t)$ は直交行列 ($Q(t)^TQ(t)=I$) であり、その列ベクトルは $A(t)$ の正規直交固有ベクトルである。$\Lambda(t)$ は対角行列であり、その対角成分 $\lambda_i(t)$ は $A(t)$ の固有値である。

この等式の両辺をパラメータ t で微分する。積の微分法則を適用すると、

$$\dot{A}=\dot{Q}\Lambda Q^T+Q\dot{\Lambda}Q^T+Q\Lambda\dot{Q}^T$$

となる。ここで、ドット記法は t に関する微分 ($\frac{d}{dt}$) を表す。

$Q(t)$ の直交性を維持するための運動学的拘束を導出する。恒等式 $Q^TQ=I$ の両辺を t で微分すると、

$$\frac{d}{dt}(Q^TQ)=\dot{Q}^TQ+Q^T\dot{Q}=0$$

ここで、補助的な行列 $H(t)$ を以下のように定義する。

$$H(t)\triangleq Q(t)T\dot{Q}(t)$$

この定義式の転置を取ると、$HT=(QT\dot{Q})T=\dot{Q}T(QT)T=\dot{Q}TQ$ となる。これを $\dot{Q}TQ+QT\dot{Q}=0$ に代入すると、

$$HT+H=0\Leftrightarrow HT=-H$$

となり、$H(t)$ は歪対称行列 (skew-symmetric matrix) でなければならないことが示される。

H の定義から、$\dot{Q}=QH$ および $\dot{Q}T=(QH)T=HTQT=-HQT$ が得られる。これらを $\dot{A}$ の式に代入すると、

$$\dot{A}=(QH)\Lambda QT+Q\dot{\Lambda}QT+Q\Lambda(-HQT)=Q(H\Lambda+\dot{\Lambda}-\Lambda H)QT$$

この式の左から QT を、右から Q を乗じ、$QTQ=I$ を用いて整理すると、最終的な発展方程式が得られる。

$$QT\dot{A}Q=H\Lambda+\dot{\Lambda}-\Lambda H$$

ここで、行列 $F(t)\triangleq Q(t)T\dot{A}(t)Q(t)$ と、行列の交換子 $[X,Y]\triangleq XY-YX$ を定義すると、方程式は以下のように簡潔に記述できる。

$$F=\dot{\Lambda}+[H,\Lambda]$$

2.2. 連立一次方程式への分解

上記の方程式は、未知の歪対称行列 H と未知の対角行列 $\dot{\Lambda}$ に関する線形方程式である。この方程式を、未知数の独立成分に関する一つの連立一次方程式として構成する。

H は歪対称行列であるため、その独立な未知数は厳密に上三角部分の $n(n-1)/2$ 個の成分 Hrc $(r<c)$ である。$\dot{\Lambda}$ は対角行列であるため、その独立な未知数は n 個の対角成分 $\dot{\lambda}r$ である。したがって、系の独立未知数の総数は $n(n-1)/2+n=n(n+1)/2$ 個となる。

方程式 $F=\dot{\Lambda}+[H,\Lambda]$ の各成分を書き下す。

対角成分 $(r=c)$:

$$Frr=\dot{\lambda}r+[H,\Lambda]rr=\dot{\lambda}r$$

(交換子の対角成分はゼロであるため)

非対角成分 (r偃=c):

$$Frc=[H,\Lambda]rc=(\lambda c-\lambda r)Hrc$$

($\dot{\Lambda}$ の非対角成分はゼロであるため)

これらの $n(n+1)/2$ 個の方程式を、未知数ベクトル $x=[\dot{\lambda}1,\dots,\dot{\lambda}n,H12,H13,\dots,Hn-1,n]T$ に関する連立一次方程式 $Mx=b$ の形に構成する。この系を数値的に解くことで、各時刻における微分係数 $(\dot{\Lambda},H)$ を一意に決定できる。

3. 実装手法：積分後一括事後補正

本実装では、数値積分の過程で生じる代数的な整合性からの乖離（ドリフト）を補正するため、以下の2段階のアプローチを採用した。

1. ステージ1: 常微分方程式の数値積分
   状態ベクトルを $y(t)=[\mathrm{vec}(Q(t))^T,\lambda_1(t),\ldots,\lambda_n(t)]^T$ と定義する。2.2節で導出した連立一次方程式を解くことで、各時刻 t における微分係数 $\dot{y}(t)$ を計算する関数を構築する。この関数を scipy.integrate.solve_ivp のような高次の常微分方程式ソルバーに渡し、指定された全時間区間にわたって積分を実行する。これにより、連続性は保証されるが、数値積分誤差の蓄積により $A(t)$偃$=Q(t)\Lambda(t)Q(t)^T$ となる時系列データ $(Q_{ode}(t_k),\Lambda_{ode}(t_k))$ が得られる。
2. ステージ2: 事後補正による代数的整合性の回復
   ステージ1で得られた軌道は、連続な「ガイド」としてのみ利用する。各評価時刻 t_k において、以下の手順で最終的な出力を得る。
   a. scipy.linalg.eigh を用い、真の行列 $A(t_k)$ から代数的に正確な固有値分解 $(Q_{exact}(t_k),\Lambda_{exact}(t_k))$ を計算する。この時点では、時刻間の連続性は保証されない。
   b. 順序対応付け**:** eigh の出力する固有値・固有ベクトルの順序を、ガイド $\Lambda_{ode}(t_k)$ の順序に合わせる。コスト行列 $C_{ij}=|\lambda_{ode,i}(t_k)-\lambda_{exact,j}(t_k)|$ を作成し、scipy.optimize.linear_sum_assignment (ハンガリー法) を適用して、総コストを最小化する最適な順序対応（置換）を決定する。この置換に従い、$Q_{exact}(t_k)$ の列と $\Lambda_{exact}(t_k)$ の対角成分を並べ替える。
   c. 符号対応付け**:** 並べ替え後の各固有ベクトル $q_{exact,j}$ について、対応するガイドの固有ベクトル $q_{ode,j}$ との内積を計算する。もし内積 $\langle q_{ode,j},q_{exact,j}\rangle<0$ であれば、$q_{exact,j}$ の符号を反転させる $(q_{exact,j}\leftarrow -q_{exact,j})$。

このプロセスを経て得られる最終的な時系列データは、各時刻で代数的に正確であり、かつ時刻間で連続性が維持されている。

4. 手法の特性

4.1. 利点

- 代数的正確性**:** 最終的な出力は、各評価時刻において $A(t)=Q(t)\Lambda(t)Q(t)^T$ を機械精度で満たす。
- 連続性**:** 常微分方程式の積分結果をガイドとして用いることで、固有値の交差や接近が生じる場合でも、固有対の連続的な対応付けが維持される。
- 実装の分離**:** 連続軌道の生成（積分）と代数的整合性の回復（補正）が明確に分離されているため、各部分の検証が容易である。

4.2. 欠点および留意事項

- 依存性**:** 本手法の成功は、ステージ1で生成されるガイド軌道の質に依存する。常微分方程式の数値積分が大きく発散したり、ドリフトが過度に蓄積したりすると、ステージ2での順序・符号のマッチングが失敗する可能性がある。
- 計算コスト**:** 積分と補正の2つのパスを必要とする。特に、評価時刻の点数が多い場合、各点で

の eigh の呼び出しが追加の計算コストとなる。

- **縮退への対処:** 2.2節で導出した連立一次方程式は、固有値の縮退 ($\lambda c=\lambda r$) が生じる点において、係数行列が悪条件となる。実装では、この系を numpy.linalg.lstsq を用いて解くことで、最小ノルム解を安定