

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5011545号
(P5011545)

(45) 発行日 平成24年8月29日(2012.8.29)

(24) 登録日 平成24年6月15日(2012.6.15)

(51) Int.Cl. F I
G06F 17/16 (2006.01) G O 6 F 17/16 K
G06T 1/00 (2006.01) G O 6 T 1/00 A

請求項の数 23 (全 54 頁)

| | | | |
|---|------------------------------|-----------|------------------------------------|
| (21) 出願番号 | 特願2007-549031 (P2007-549031) | (73) 特許権者 | 504132272 |
| (86) (22) 出願日 | 平成18年9月21日(2006.9.21) | | 国立大学法人京都大学 |
| (86) 国際出願番号 | PCT/JP2006/318713 | | 京都府京都市左京区吉田本町36番地1 |
| (87) 国際公開番号 | W02007/066445 | (74) 代理人 | 100115749 |
| (87) 国際公開日 | 平成19年6月14日(2007.6.14) | | 弁理士 谷川 英和 |
| 審査請求日 | 平成21年8月7日(2009.8.7) | (72) 発明者 | 中村 佳正 |
| (31) 優先権主張番号 | 特願2005-351089 (P2005-351089) | | 京都府京都市左京区吉田本町 国立大学法人京都大学大学院情報学研究科内 |
| (32) 優先日 | 平成17年12月5日(2005.12.5) | (72) 発明者 | 譽田 太朗 |
| (33) 優先権主張国 | 日本国(JP) | | 京都府京都市左京区吉田本町 国立大学法人京都大学大学院情報学研究科内 |
| 特許法第30条第1項適用 2005年9月25日発行日本応用数学会論文誌にて発表 | | (72) 発明者 | 岩▲崎▼ 雅史 |
| | | | 京都府京都市左京区吉田本町 国立大学法人京都大学大学院情報学研究科内 |

最終頁に続く

(54) 【発明の名称】 特異値分解装置、及び特異値分解方法

(57) 【特許請求の範囲】

【請求項1】

2重対角行列 B が記憶される対角行列記憶部と、
 前記 2重対角行列 B が 2個の 2重対角行列に分割され、その 2重対角行列が 2個の 2重対角行列に分割される処理が、分割後の各 2重対角行列があらかじめ決められた大きさ以下となるまで繰り返され、当該あらかじめ決められた大きさ以下の各 2重対角行列に対して特異値分解が行われた結果である、前記各 2重対角行列の特異値と、前記各 2重対角行列の特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部と、

前記 2重対角行列 B の特異値が記憶される特異値記憶部と、
 各 2重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の 2重対角行列の特異値と、分割元の 2重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2重対角行列の特異値と、行列要素とを算出する処理を、2重対角行列 B の少なくとも 1個の特異値を算出するまで繰り返し、前記 2重対角行列 B の少なくとも 1個の特異値を前記特異値記憶部に蓄積する特異値算出部と、

前記 2重対角行列 B の特異ベクトルが記憶される特異ベクトル記憶部と、
 前記対角行列記憶部から 2重対角行列 B を読み出し、前記特異値記憶部から前記 2重対角行列 B の特異値を読み出し、2重対角行列 B とその特異値とから、ツイスト分解法を用いて 2重対角行列 B の少なくとも 1個の特異ベクトルを算出して前記特異ベクトル記憶部に

蓄積する特異ベクトル算出部と、を備えた特異値分解装置。

【請求項 2】

前記対角行列記憶部には、前記あらかじめ決められた大きさ以下の各 2 重対角行列も記憶され、

前記あらかじめ決められた大きさ以下の各 2 重対角行列を前記対角行列記憶部から読み出し、前記各 2 重対角行列に対して特異値分解を行い、前記各 2 重対角行列の特異値と、前記各 2 重対角行列の特異ベクトルとを算出し、当該特異値と、当該特異ベクトルからなる左右直交行列の一部の要素である行列要素とを前記特異値分解記憶部に蓄積する特異値分解部をさらに備えた請求項 1 記載の特異値分解装置。

【請求項 3】

2 重対角行列 B が記憶される対角行列記憶部と、
前記対角行列記憶部から前記 2 重対角行列 B を読み出し、当該 2 重対角行列 B を 2 個の 2 重対角行列に分割して前記対角行列記憶部に蓄積し、その 2 重対角行列を 2 個の 2 重対角行列に分割して前記対角行列記憶部に蓄積する処理を、分割後の各 2 重対角行列があらかじめ決められた大きさ以下となるまで繰り返す行列分割部と、

前記あらかじめ決められた大きさ以下の各 2 重対角行列を前記対角行列記憶部から読み出し、前記各 2 重対角行列に対して特異値分解を行い、前記各 2 重対角行列の特異値と、前記各 2 重対角行列の特異ベクトルとを算出する特異値分解部と、

前記特異値分解部によって特異値分解された特異値と、特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部と、

前記 2 重対角行列 B の特異値が記憶される特異値記憶部と、

各 2 重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の少なくとも 1 個の特異値を算出するまで繰り返し、前記 2 重対角行列 B の少なくとも 1 個の特異値を前記特異値記憶部に蓄積する特異値算出部と、

前記 2 重対角行列 B の特異ベクトルが記憶される特異ベクトル記憶部と、

前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の少なくとも 1 個の特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する特異ベクトル算出部と、を備えた特異値分解装置。

【請求項 4】

前記特異ベクトル算出部は、q d 型ツイスト分解法により特異ベクトルを算出する、請求項 3 記載の特異値分解装置。

【請求項 5】

前記特異ベクトル算出部は、L V 型ツイスト分解法により特異ベクトルを算出する、請求項 3 記載の特異値分解装置。

【請求項 6】

前記特異ベクトル算出部は、
前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、前記 2 重対角行列 B の各要素に関してミウラ変換、d L V v 型変換、逆ミウラ変換を行うことによって、前記 2 重対角行列 B を上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$ にコレスキー分解するコレスキー分解部と、

前記上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$ の各要素と、前記 2 重対角行列 B の特異値とを用いて一方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第 1 特異ベクトル算出部と、

前記第 1 特異ベクトル算出部が算出した一方の左右直交行列を構成する特異ベクトルと、前記 2 重対角行列 B の特異値と、前記 2 重対角行列 B とを用いて他方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第 2 特異ベクトル算出

10

20

30

40

50

部と、をさらに備えた、請求項 5 記載の特異値分解装置。

【請求項 7】

前記コレスキー分解部は、複数のコレスキー分解手段を備え、
前記複数のコレスキー分解手段が、前記 2 重対角行列 B をコレスキー分解する処理を並列実行する、請求項 6 記載の特異値分解装置。

【請求項 8】

前記第 1 特異ベクトル算出部は、複数の第 1 特異ベクトル算出手段を備え、
前記複数の第 1 特異ベクトル算出手段が、特異ベクトルを算出する処理を並列実行する、
請求項 6 または請求項 7 記載の特異値分解装置。

【請求項 9】

前記第 2 特異ベクトル算出部は、複数の第 2 特異ベクトル算出手段を備え、
前記複数の第 2 特異ベクトル算出手段が、特異ベクトルを算出する処理を並列実行する、
請求項 6 から請求項 8 のいずれか記載の特異値分解装置。

【請求項 10】

前記特異値算出部は、複数の特異値算出手段を備え、
前記複数の特異値算出手段が、分割元の 2 重対角行列の特異値と、行列要素とを算出する
処理を並列実行する、請求項 3 から請求項 9 のいずれか記載の特異値分解装置。

【請求項 11】

前記特異値分解部は、複数の特異値分解手段を備え、
前記複数の特異値分解手段が、2 重対角行列に対して特異値分解を行う処理を並列実行す
る、請求項 3 から請求項 10 のいずれか記載の特異値分解装置。

【請求項 12】

行列 A が記憶される行列記憶部と、
前記行列 A を前記行列記憶部から読み出し、前記行列 A を 2 重対角化した前記 2 重対角行
列 B を算出して前記対角行列記憶部に蓄積する対角化部と、をさらに備えた請求項 3 から
請求項 11 のいずれか特異値分解装置。

【請求項 13】

前記行列分割部は、2 重対角行列を略半分の 2 個の 2 重対角行列に分割する、請求項 3 から
請求項 12 のいずれか記載の特異値分解装置。

【請求項 14】

前記特異値算出部は、2 重対角行列 B の全ての特異値を算出する、請求項 3 から請求項 1
3 のいずれか記載の特異値分解装置。

【請求項 15】

前記特異ベクトル算出部は、2 重対角行列 B の全ての特異ベクトルを算出する、請求項 1
4 記載の特異値分解装置。

【請求項 16】

前記行列 A は、2 次元画像 j ($j = 1, \dots, m$ 、 m は 3 以上の整数) から抽出された特徴
点 i ($i = 1, \dots, n$ 、 n は 2 以上の整数) の座標 (x_i^j, y_i^j) から構成される行
列である、請求項 12 記載の特異値分解装置。

【請求項 17】

前記行列 A は、索引語の重みを要素とするベクトルであって、検索対象となる文書を示す
ベクトルである d_1, \dots, d_n (n は 2 以上の整数) を各列に有する行列である索引語文
書行列である、請求項 12 記載の特異値分解装置。

【請求項 18】

2 重対角行列 B が記憶される対角行列記憶部と、前記 2 重対角行列 B が 2 個の 2 重対角行
列に分割され、その 2 重対角行列が 2 個の 2 重対角行列に分割される処理が、分割後の各
2 重対角行列があらかじめ決められた大きさ以下となるまで繰り返され、当該あらかじめ
決められた大きさ以下の各 2 重対角行列に対して特異値分解が行われた結果である、前記
各 2 重対角行列の特異値と、前記各 2 重対角行列の特異ベクトルからなる左右直交行列の
一部の要素である行列要素とが記憶される特異値分解記憶部と、前記 2 重対角行列 B の特

10

20

30

40

50

異値が記憶される特異値記憶部と、特異値算出部と、前記 2 重対角行列 B の特異ベクトルが記憶される特異ベクトル記憶部と、特異ベクトル算出部とを備えた特異値分解装置で用いられる特異値分解方法であって、

前記特異値算出部が、各 2 重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の少なくとも 1 個の特異値を算出するまで繰り返し、前記 2 重対角行列 B の少なくとも 1 個の特異値を前記特異値記憶部に蓄積する特異値算出ステップと、

前記特異ベクトル算出部が、前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の少なくとも 1 個の特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する特異ベクトル算出ステップと、を備えた特異値分解方法。

10

【請求項 19】

2 重対角行列 B が記憶される対角行列記憶部と、行列分割部と、特異値分解部と、前記特異値分解部によって特異値分解された特異値と、特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部と、前記 2 重対角行列 B の特異値が記憶される特異値記憶部と、特異値算出部と、前記 2 重対角行列 B の特異ベクトルが記憶される特異ベクトル記憶部と、特異ベクトル算出部と、を備えた特異値分解装置で用いられる特異値分解方法であって、

20

前記行列分割部が、前記対角行列記憶部から前記 2 重対角行列 B を読み出し、当該 2 重対角行列 B を 2 個の 2 重対角行列に分割して前記対角行列記憶部に蓄積し、その 2 重対角行列を 2 個の 2 重対角行列に分割して前記対角行列記憶部に蓄積する処理を、分割後の各 2 重対角行列があらかじめ決められた大きさ以下となるまで繰り返す行列分割ステップと、前記特異値分解部が、前記あらかじめ決められた大きさ以下の各 2 重対角行列を前記対角行列記憶部から読み出し、前記各 2 重対角行列に対して特異値分解を行い、前記各 2 重対角行列の特異値と、前記各 2 重対角行列の特異ベクトルとを算出する特異値分解ステップと、

前記特異値算出部が、各 2 重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の少なくとも 1 個の特異値を算出するまで繰り返し、前記 2 重対角行列 B の少なくとも 1 個の特異値を前記特異値記憶部に蓄積する特異値算出ステップと、

30

前記特異ベクトル算出部が、前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の少なくとも 1 個の特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する特異ベクトル算出ステップと、を備えた特異値分解方法。

40

【請求項 20】

前記特異ベクトル算出部は、コレスキー分解部と、第 1 特異ベクトル算出部と、第 2 特異ベクトル算出部と、をさらに備えており、

前記特異ベクトル算出ステップは、

前記コレスキー分解部が、前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、前記 2 重対角行列 B の各要素に関してミウラ変換、d L V v 型変換、逆ミウラ変換を行うことによって、前記 2 重対角行列 B を上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$ にコレスキー分解するコレスキー分解ステップと、

前記第 1 特異ベクトル算出部が、前記上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$

50

・ 1) の各要素と、前記 2 重対角行列 B の特異値とを用いて一方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第 1 特異ベクトル算出ステップと、

前記第 2 特異ベクトル算出部が、前記第 1 特異ベクトル算出ステップで算出した一方の左右直交行列を構成する特異ベクトルと、前記 2 重対角行列 B の特異値と、前記 2 重対角行列 B とを用いて他方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第 2 特異ベクトル算出ステップと、をさらに備えた、請求項 19 記載の特異値分解方法。

【請求項 21】

コンピュータに、

2 重対角行列 B が 2 個の 2 重対角行列に分割され、その 2 重対角行列が 2 個の 2 重対角行列に分割される処理が、分割後の各 2 重対角行列があらかじめ決められた大きさ以下となるまで繰り返され、当該あらかじめ決められた大きさ以下の各 2 重対角行列に対して特異値分解が行われた結果である、前記各 2 重対角行列の特異値と、前記各 2 重対角行列の特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部から、各 2 重対角行列の特異値と、行列要素とを読み出し、前記特異値と、前記行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の少なくとも 1 個の特異値を算出するまで繰り返し、前記 2 重対角行列 B の少なくとも 1 個の特異値を特異値記憶部に蓄積する特異値算出ステップと、

前記 2 重対角行列 B が記憶される対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の少なくとも 1 個の特異ベクトルを算出して特異ベクトル記憶部に蓄積する特異ベクトル算出ステップと、を実現させるためのプログラム。

【請求項 22】

コンピュータに、

2 重対角行列 B が記憶される対角行列記憶部から前記 2 重対角行列 B を読み出し、当該 2 重対角行列 B を 2 個の 2 重対角行列に分割して前記対角行列記憶部に蓄積し、その 2 重対角行列を 2 個の 2 重対角行列に分割して前記対角行列記憶部に蓄積する処理を、分割後の各 2 重対角行列があらかじめ決められた大きさ以下となるまで繰り返す行列分割ステップと、

前記あらかじめ決められた大きさ以下の各 2 重対角行列を前記対角行列記憶部から読み出し、前記各 2 重対角行列に対して特異値分解を行い、前記各 2 重対角行列の特異値と、前記各 2 重対角行列の特異ベクトルとを算出し、特異値分解された特異値と、特異ベクトルからなる左右直交行列の一部の要素である行列要素とを特異値分解記憶部に蓄積する特異値分解ステップと、

各 2 重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の少なくとも 1 個の特異値を算出するまで繰り返し、前記 2 重対角行列 B の少なくとも 1 個の特異値を特異値記憶部に蓄積する特異値算出ステップと、

前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の少なくとも 1 個の特異ベクトルを算出して特異ベクトル記憶部に蓄積する特異ベクトル算出ステップと、を実行させるためのプログラム。

【請求項 23】

前記特異ベクトル算出ステップは、

10

20

30

40

50

前記対角行列記憶部から2重対角行列Bを読み出し、前記特異値記憶部から前記2重対角行列Bの特異値を読み出し、前記2重対角行列Bの各要素に関してミウラ変換、dLVV型変換、逆ミウラ変換を行うことによって、前記2重対角行列Bを上2重対角行列 $B^{(+)}$ 及び下2重対角行列 $B^{(-)}$ にコレスキー分解するコレスキー分解ステップと、前記上2重対角行列 $B^{(+)}$ 及び下2重対角行列 $B^{(-)}$ の各要素と、前記2重対角行列Bの特異値とを用いて一方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第1特異ベクトル算出ステップと、前記第1特異ベクトル算出ステップで算出した一方の左右直交行列を構成する特異ベクトルと、前記2重対角行列Bの特異値と、前記2重対角行列Bとを用いて他方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第2特異ベクトル算出ステップと、をさらに備えた、請求項22記載のプログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、特異値分解を行う特異値分解装置等に関する。

【背景技術】

【0002】

特異値分解(SVD: Singular Value Decomposition)はデータ処理の中心的な行列演算として、画像処理やデータ検索等の多くの分野に応用されている。

20

なお、特異値分解の従来例としては、例えば、下記の非特許文献1のものなどが知られている。

【非特許文献1】Ming Gu and Stanley C. Eisenstat, 「A Divide-and-Conquer Algorithm for the Bidiagonal SVD」、SIAM Journal on Matrix Analysis and Applications, Vol. 16, No. 1, pp. 79-92 (1995)

【発明の開示】

【発明が解決しようとする課題】

【0003】

近年、これらの応用分野におけるデータ量の増大などに伴って、高速・高精度な特異値分解が求められている。また、特異値分解を並列処理することができる特異値分解装置等の開発が望まれていた。

30

本発明は、上記状況の下になされたものであり、並列性に優れた高速、高精度な特異値分解を実行可能な特異値分解装置等を提供することを目的とする。

【課題を解決するための手段】

【0004】

上記目的を達成するため、本発明による特異値分解装置は、2重対角行列Bが記憶される対角行列記憶部と、前記2重対角行列Bが2個の2重対角行列に分割され、その2重対角行列が2個の2重対角行列に分割される処理が、分割後の各2重対角行列があらかじめ決められた大きさ以下となるまで繰り返され、当該あらかじめ決められた大きさ以下の各2重対角行列に対して特異値分解が行われた結果である、前記各2重対角行列の特異値と、前記各2重対角行列の特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部と、前記2重対角行列Bの特異値が記憶される特異値記憶部と、各2重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の2重対角行列の特異値と、分割元の2重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の2重対角行列の特異値と、行列要素とを算出する処理を、2重対角行列Bの少なくとも1個の特異値を算出するまで繰り返し、前記2重対角行列Bの少なくとも1個の特異値を前記特異値記憶部に蓄積する特異値算出部と、前記2重対角行列Bの特異ベクトルが記憶される特異ベ

40

50

クトル記憶部と、前記対角行列記憶部から2重対角行列Bを読み出し、前記特異値記憶部から前記2重対角行列Bの特異値を読み出し、2重対角行列Bとその特異値とから、ツイスト分解法を用いて2重対角行列Bの少なくとも1個の特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する特異ベクトル算出部と、を備えたものである。

【0005】

このような構成により、まず特異値を算出し、その特異値に基づいてツイスト分解法を用いて特異ベクトルを算出することによって、高速で高精度な特異値分解を実現することができる。また、並列性にも優れている。

【0006】

また、本発明による特異値分解装置では、前記対角行列記憶部には、前記あらかじめ決められた大きさ以下の各2重対角行列も記憶され、前記あらかじめ決められた大きさ以下の各2重対角行列を前記対角行列記憶部から読み出し、前記各2重対角行列に対して特異値分解を行い、前記各2重対角行列の特異値と、前記各2重対角行列の特異ベクトルとを算出し、当該特異値と、当該特異ベクトルからなる左右直交行列の一部の要素である行列要素とを前記特異値分解記憶部に蓄積する特異値分解部をさらに備えてもよい。

【0007】

このような構成により、特異値を算出する際に、あらかじめ決められた大きさ以下の各2重対角行列に対する特異値分解を、特異値分解装置において行うことができる。

【0008】

また、本発明による特異値分解装置は、2重対角行列Bが記憶される対角行列記憶部と、前記対角行列記憶部から前記2重対角行列Bを読み出し、当該2重対角行列Bを2個の2重対角行列に分割して前記対角行列記憶部に蓄積し、その2重対角行列を2個の2重対角行列に分割して前記対角行列記憶部に蓄積する処理を、分割後の各2重対角行列があらかじめ決められた大きさ以下となるまで繰り返す行列分割部と、前記あらかじめ決められた大きさ以下の各2重対角行列を前記対角行列記憶部から読み出し、前記各2重対角行列に対して特異値分解を行い、前記各2重対角行列の特異値と、前記各2重対角行列の特異ベクトルとを算出する特異値分解部と、前記特異値分解部によって特異値分解された特異値と、特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部と、前記2重対角行列Bの特異値が記憶される特異値記憶部と、各2重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の2重対角行列の特異値と、分割元の2重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の2重対角行列の特異値と、行列要素とを算出する処理を、2重対角行列Bの少なくとも1個の特異値を算出するまで繰り返し、前記2重対角行列Bの少なくとも1個の特異値を前記特異値記憶部に蓄積する特異値算出部と、前記2重対角行列Bの特異ベクトルが記憶される特異ベクトル記憶部と、前記対角行列記憶部から2重対角行列Bを読み出し、前記特異値記憶部から前記2重対角行列Bの特異値を読み出し、2重対角行列Bとその特異値とから、ツイスト分解法を用いて2重対角行列Bの少なくとも1個の特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する特異ベクトル算出部と、を備えたものである。

【0009】

このような構成により、まず特異値を算出し、その特異値に基づいてツイスト分解法を用いて特異ベクトルを算出することによって、高速で高精度な特異値分解を実現することができる。また、並列性にも優れている。さらに、全ての特異値や特異ベクトルを算出する必要がない場合には、必要な範囲で特異値や特異ベクトルを算出することができ、処理負荷を軽減することができる。

【0010】

また、本発明による特異値分解装置では、前記特異ベクトル算出部は、qd型ツイスト分解法により特異ベクトルを算出してもよい。

このような構成により、高速で高精度な特異値分解を実現することができる。また、並列性にも優れている。

10

20

30

40

50

【 0 0 1 1 】

また、本発明による特異値分解装置では、前記特異ベクトル算出部は、L V型ツイスト分解法により特異ベクトルを算出してもよい。

このような構成により、数値安定的に特異ベクトルの算出を行うことができる。

【 0 0 1 2 】

また、本発明による特異値分解装置では、前記特異ベクトル算出部は、前記対角行列記憶部から2重対角行列Bを読み出し、前記特異値記憶部から前記2重対角行列Bの特異値を読み出し、前記2重対角行列Bの各要素に関してミウラ変換、d L V v型変換、逆ミウラ変換を行うことによって、前記2重対角行列Bを上2重対角行列 $B^{(+)}$ 及び下2重対角行列 $B^{(-)}$ にコレスキー分解するコレスキー分解部と、前記上2重対角行列 $B^{(+)}$ 及び下2重対角行列 $B^{(-)}$ の各要素と、前記2重対角行列Bの特異値とを用いて一方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第1特異ベクトル算出部と、前記第1特異ベクトル算出部が算出した一方の左右直交行列を構成する特異ベクトルと、前記2重対角行列Bの特異値と、前記2重対角行列Bとを用いて他方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第2特異ベクトル算出部と、をさらに備えてもよい。

このような構成により、コレスキー分解の処理において、複数の補助変数を用いることによって、数値安定的に特異ベクトルの算出を行うことができる。

【 0 0 1 3 】

また、本発明による特異値分解装置では、前記コレスキー分解部は、複数のコレスキー分解手段を備え、前記複数のコレスキー分解手段が、前記2重対角行列Bをコレスキー分解する処理を並列実行してもよい。

このような構成により、コレスキー分解の処理を短時間で行うことができる。

【 0 0 1 4 】

また、本発明による特異値分解装置では、前記第1特異ベクトル算出部は、複数の第1特異ベクトル算出手段を備え、前記複数の第1特異ベクトル算出手段が、特異ベクトルを算出する処理を並列実行してもよい。

このような構成により、特異ベクトルを算出する処理を短時間で行うことができる。

【 0 0 1 5 】

また、本発明による特異値分解装置では、前記第2特異ベクトル算出部は、複数の第2特異ベクトル算出手段を備え、前記複数の第2特異ベクトル算出手段が、特異ベクトルを算出する処理を並列実行してもよい。

このような構成により、特異ベクトルを算出する処理を短時間で行うことができる。

【 0 0 1 6 】

また、本発明による特異値分解装置では、前記特異値算出部は、複数の特異値算出手段を備え、前記複数の特異値算出手段が、分割元の2重対角行列の特異値と、行列要素とを算出する処理を並列実行してもよい。

このような構成により、特異値を算出する処理を短時間で行うことができる。

【 0 0 1 7 】

また、本発明による特異値分解装置では、前記特異値分解部は、複数の特異値分解手段を備え、前記複数の特異値分解手段が、2重対角行列に対して特異値分解を行う処理を並列実行してもよい。

このような構成により、特異値分解する処理を短時間で行うことができる。

【 0 0 1 8 】

また、本発明による特異値分解装置では、行列Aが記憶される行列記憶部と、前記行列Aを前記行列記憶部から読み出し、前記行列Aを2重対角化した前記2重対角行列Bを算出して前記対角行列記憶部に蓄積する対角化部と、をさらに備えてもよい。

このような構成により、任意の行列Aについて特異値を算出することができる。また、2重対角行列Bの特異ベクトルを用いることにより、行列Aの特異ベクトルを算出することもできる。

10

20

30

40

50

【 0 0 1 9 】

また、本発明による特異値分解装置では、前記行列分割部は、2重対角行列を略半分の2個の2重対角行列に分割してもよい。

このような構成により、並列処理を適切に行うことができる。

【 発明の効果 】

【 0 0 2 0 】

本発明による特異値分解装置等によれば、高速で高精度な特異値分解を行うことができる。また、並列性にも優れている。

【 発明を実施するための最良の形態 】

【 0 0 2 1 】

以下、本発明による特異値分解装置について、実施の形態を用いて説明する。なお、以下の実施の形態において、同じ符号を付した構成要素及びステップは同一または相当するものであり、再度の説明を省略することがある。

【 0 0 2 2 】

(実施の形態1)

本発明の実施の形態1による特異値分解装置について、図面を参照しながら説明する。

図1は、本実施の形態による特異値分解装置1の構成を示すブロック図である。図1において、本実施の形態による特異値分解装置1は、行列記憶部11と、対角化部12と、対角行列記憶部13と、行列分割部14と、特異値分解部15と、特異値分解記憶部16と、特異値算出部17と、特異値記憶部18と、特異ベクトル算出部19と、特異ベクトル記憶部20とを備える。

【 0 0 2 3 】

行列記憶部11では、任意の行列Aが記憶される。この行列Aは、各要素が実数である実行列である。なお、行列Aが記憶されているとは、行列Aを示すデータが記憶されている、という意味である。後述する記憶部においても同様である。行列記憶部11は、所定の記録媒体(例えば、半導体メモリや磁気ディスク、光ディスクなど)によって実現される。行列記憶部11での記憶は、RAM等における一時的な記憶でもよく、あるいは、長期的な記憶でもよい。行列記憶部11に行列Aが記憶される過程は問わない。例えば、記録媒体を介して行列Aが行列記憶部11で記憶されるようになってよく、通信回線を介して送信された行列Aが行列記憶部11で記憶されるようになってよく、あるいは、キーボードやマウス等の入力デバイスを介して入力された行列Aが行列記憶部11で記憶されるようになってよく、あるいは、

【 0 0 2 4 】

対角化部12は、行列Aを行列記憶部11から読み出し、その読み出した行列Aを2重対角化した2重対角行列Bを算出する。そして、対角化部12は、その算出した2重対角行列Bを対角行列記憶部13に蓄積する。対角化部12は、例えば、ハウスホルダー(Householder)変換を必要なだけ繰り返し行う方法や、その他の2重対角化法を用いて、行列Aを2重対角化する。ここで、2重対角行列Bは、上2重対角行列であってもよく、下2重対角行列であってもよい。本実施の形態では、2重対角行列Bが上2重対角行列である場合について説明する。

【 0 0 2 5 】

対角行列記憶部13では、2重対角行列Bが記憶される。対角行列記憶部13は、所定の記録媒体(例えば、半導体メモリや磁気ディスク、光ディスクなど)によって実現される。対角行列記憶部13での記憶は、RAM等における一時的な記憶でもよく、あるいは、長期的な記憶でもよい。

【 0 0 2 6 】

行列分割部14は、対角行列記憶部13から2重対角行列Bを読み出し、その2重対角行列Bを2個の2重対角行列に分割して対角行列記憶部13に蓄積する。行列分割部14は、その2重対角行列を2個の2重対角行列に分割して対角行列記憶部13に蓄積する処理を、分割後の各2重対角行列があらかじめ決められた大きさ以下となるまで再帰的に繰

10

20

30

40

50

り返す。

【 0 0 2 7 】

特異値分解部 1 5 は、あらかじめ決められた大きさ以下の各 2 重対角行列を対角行列記憶部 1 3 から読み出し、その各 2 重対角行列に対して特異値分解を行い、各 2 重対角行列の特異値と、その各 2 重対角行列の特異ベクトルを算出する。特異値分解部 1 5 は、例えば、2 分法と逆反復法を組み合わせた方法、MR³法、QRs 法等を用いて特異値分解を行ってもよい。ここで、MR³法において、例えば、dqds や pqds 等の qds 法によって特異値を算出してもよい。また、QRs 法による特異値分解を行う場合には、FORTRAN において提供されている DBDSQR を用いてもよい。また、qds 法によって特異値を算出する場合には、FORTRAN において提供されている DLASQ を

10

用いてもよい。これらの特異値分解の方法については、すでに公知であり、その詳細な説明を省略する。特異値分解部 1 5 は、算出した特異値を特異値分解記憶部 1 6 に蓄積する。また、特異値分解部 1 5 は、算出した特異ベクトルからなる左右直交行列の一部の要素である行列要素も特異値分解記憶部 1 6 に蓄積する。特異ベクトルからなる左右直交行列とは、特異ベクトルを各列とする左直交行列、及び特異ベクトルを各列とする右直交行列のことである。行列要素の詳細については後述する。

【 0 0 2 8 】

特異値分解記憶部 1 6 では、特異値分解部によって特異値分解された特異値と、前述の行列要素とが記憶される。また、特異値分解記憶部 1 6 では、特異値算出部 1 7 によって算出された特異値や行列要素も記憶される。特異値分解記憶部 1 6 は、所定の記録媒体（例えば、半導体メモリや磁気ディスク、光ディスクなど）によって実現されうる。特異値分解記憶部 1 6 での記憶は、RAM 等における一時的な記憶でもよく、あるいは、長期的な記憶でもよい。

20

【 0 0 2 9 】

特異値算出部 1 7 は、特異値分解部 1 5 によって算出された特異値と、行列要素とを特異値分解記憶部 1 6 から読み出し、その特異値と行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して特異値分解記憶部 1 6 に蓄積する。特異値算出部 1 7 は、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の特異値を算出するまで再帰的に繰り返す。そして、特異値算出部 1 7 は、算出した 2 重対角行列 B の特異値を特異値記憶部 1 8 に蓄積する。

30

【 0 0 3 0 】

特異値記憶部 1 8 では、2 重対角行列 B の特異値が記憶される。特異値記憶部 1 8 は、所定の記録媒体（例えば、半導体メモリや磁気ディスク、光ディスクなど）によって実現されうる。特異値記憶部 1 8 での記憶は、RAM 等における一時的な記憶でもよく、あるいは、長期的な記憶でもよい。

【 0 0 3 1 】

特異ベクトル算出部 1 9 は、対角行列記憶部 1 3 から 2 重対角行列 B を読み出し、特異値記憶部 1 8 から 2 重対角行列 B の特異値を読み出す。そして、特異ベクトル算出部 1 9 は、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の特異ベクトルを算出して特異ベクトル記憶部 2 0 に蓄積する。特異ベクトル算出部 1 9 は、それらの処理を実行する、コレスキー分解部 2 1 と、第 1 特異ベクトル算出部 2 2 と、第 2 特異ベクトル算出部 2 3 とを備える。特異ベクトル算出部 1 9 は、LV 型ツイスト分解法により特異ベクトルを算出してもよく、あるいは、qd 型ツイスト分解法により特異ベクトルを算出してもよい。本実施の形態では、前者の場合について説明する。

40

【 0 0 3 2 】

コレスキー分解部 2 1 は、対角行列記憶部 1 3 から 2 重対角行列 B を読み出し、特異値記憶部 1 8 から 2 重対角行列 B の特異値を読み出す。そして、コレスキー分解部 2 1 は、2 重対角行列 B の各要素に関してミウラ変換、dLVv 型変換、逆ミウラ変換を行うこと

50

コレスキー分解する。この処理の詳細については後述する。

【 0 0 3 3 】

第 1 特異ベクトル算出部 2 2 は、コレスキー分解部 2 1 が算出した上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$ の各要素と、2 重対角行列 B の特異値とを用いて一方の左右直交行列を構成する特異ベクトルを算出して特異ベクトル記憶部 2 0 に蓄積する。

【 0 0 3 4 】

第 2 特異ベクトル算出部 2 3 は、特異値記憶部 1 8 から 2 重対角行列 B の特異値を読み出す。そして、第 2 特異ベクトル算出部 2 3 は、第 1 特異ベクトル算出部 2 2 が算出した一方の左右直交行列を構成する特異ベクトルと、2 重対角行列 B の特異値と、2 重対角行列 B とを用いて他方の左右直交行列を構成する特異ベクトルを算出して特異ベクトル記憶部 2 0 に蓄積する。このように、第 1 特異ベクトル算出部 2 2 と、第 2 特異ベクトル算出部 2 3 とによって、左右直交行列のそれぞれが算出されることになる。

【 0 0 3 5 】

特異ベクトル記憶部 2 0 では、2 重対角行列 B の特異ベクトルが記憶される。特異ベクトル記憶部 2 0 は、所定の記録媒体（例えば、半導体メモリや磁気ディスク、光ディスクなど）によって実現されうる。特異ベクトル記憶部 2 0 での記憶は、RAM 等における一時的な記憶でもよく、あるいは、長期的な記憶でもよい。

【 0 0 3 6 】

なお、行列記憶部 1 1、対角行列記憶部 1 3、特異値分解記憶部 1 6、特異値記憶部 1 8、特異ベクトル記憶部 2 0 の任意の 2 以上の記憶部は、同一の記録媒体によって実現されてもよく、あるいは、別々の記録媒体によって実現されてもよい。前者の場合には、例えば、行列 A の記憶されている領域が行列記憶部 1 1 となり、2 重対角行列 B 等の記憶されている領域が対角行列記憶部 1 3 となる。

【 0 0 3 7 】

また、行列記憶部 1 1、対角行列記憶部 1 3、特異値分解記憶部 1 6、特異値記憶部 1 8、特異ベクトル記憶部 2 0 の各記憶部は、2 以上の記録媒体から構成されてもよい。

【 0 0 3 8 】

次に、本実施の形態による特異値分解装置 1 の動作について、図 2 のフローチャートを用いて説明する。

(ステップ S 1 0 1) 対角化部 1 2 は、行列記憶部 1 1 で記憶されている行列 A を読み出し、その行列 A を 2 重対角化して 2 重対角行列 B を算出して対角行列記憶部 1 3 に蓄積する。

【 0 0 3 9 】

(ステップ S 1 0 2) 行列分割部 1 4、特異値分解部 1 5、特異値算出部 1 7 によって 2 重対角行列 B の特異値が算出され、特異値記憶部 1 8 に蓄積される。この処理の詳細については後述する。

【 0 0 4 0 】

(ステップ S 1 0 3) 特異ベクトル算出部 1 9 は、対角行列記憶部 1 3 から 2 重対角行列 B を読み出し、特異値記憶部 1 8 から 2 重対角行列 B の特異値を読み出し、2 重対角行列 B の特異ベクトルを算出して特異ベクトル記憶部 2 0 に蓄積する。この処理の詳細については後述する。

【 0 0 4 1 】

このようにして、2 重対角行列 B の特異値分解が終了する。ここで、行列 A の特異値は 2 重対角行列 B の特異値に等しいため、行列 A の特異値も算出されたことになる。また、後述するように、所定の変換を行うことによって、行列 A の特異ベクトルも 2 重対角行列 B の特異ベクトルから容易に算出することができる。

【 0 0 4 2 】

次に、図 2 のフローチャートのステップ S 1 0 2 の処理について、図 3 のフローチャートを用いて説明する。

10

20

30

40

50

(ステップS201) 行列分割部14は、対角行列記憶部13から2重対角行列Bを読み出し、その2重対角行列Bを2個の2重対角行列に分割して対角行列記憶部13に蓄積する。行列分割部14は、その2重対角行列を2個の2重対角行列に分割する処理を、分割後の各2重対角行列があらかじめ決められた大きさ以下となるまで繰り返す。

【0043】

(ステップS202) 特異値分解部15は、対角行列記憶部13で記憶されているあらかじめ決められた大きさ以下の各2重対角行列について、特異値分解を行う。特異値分解部15は、特異値分解の結果である特異値と、特異ベクトルからなる左右直交行列の一部の要素である行列要素とを特異値分解記憶部16に蓄積する。

【0044】

(ステップS203) 特異値算出部17は、2重対角行列の特異値と、行列要素とを特異値分解記憶部16から読み出し、分割元の2重対角行列の特異値と、分割元の2重対角行列の行列要素とを算出して特異値分解記憶部16に蓄積する。特異値算出部17は、その分割元の2重対角行列の特異値と、行列要素とを算出する処理を、2重対角行列Bの特異値を算出するまで繰り返し、2重対角行列Bの特異値を特異値記憶部18に蓄積する。このようにして、特異値を算出する処理が終了する。

【0045】

次に、図2のフローチャートのステップS103の処理について、図4のフローチャートを用いて説明する。

(ステップS301) コレスキー分解部21は、対角行列記憶部13から2重対角行列Bを読み出し、特異値記憶部18から2重対角行列Bの特異値を読み出す。そして、コレスキー分解部21は、2重対角行列Bの各要素に関してミウラ変換、dLVv型変換、逆ミウラ変換を行うことによって、2重対角行列Bを上2重対角行列 $B^{(+)}$ 及び下2重対角行列 $B^{(-)}$ にコレスキー分解する。

【0046】

(ステップS302) 第1特異ベクトル算出部22は、上2重対角行列 $B^{(+)}$ 及び下2重対角行列 $B^{(-)}$ の各要素を用いて、一方の左右直交行列を構成する特異ベクトルを算出する。本実施の形態では、第1特異ベクトル算出部22は、右直交行列を構成する右特異ベクトルを算出するものとする。

【0047】

(ステップS303) 第1特異ベクトル算出部22は、算出した特異ベクトルを正規化する。すなわち、第1特異ベクトル算出部22は、算出した特異ベクトルのノルムを算出し、特異ベクトルを算出したノルムで割ったものを最終的な特異ベクトルとして特異ベクトル記憶部20に蓄積する。

【0048】

(ステップS304) 第2特異ベクトル算出部23は、第1特異ベクトル算出部22が算出した特異ベクトルと、2重対角行列Bの特異値と、2重対角行列Bとを用いて、第1特異ベクトル算出部22が算出した特異ベクトルと異なる方の特異ベクトルを算出する。2重対角行列Bを特異値分解した結果が、第1特異ベクトル算出部22が算出した特異ベクトルからなる直交行列と、第2特異ベクトル算出部23が算出する特異ベクトルからなる直交行列と、特異値記憶部18が記憶している特異値であるため、その性質を用いて、第2特異ベクトル算出部23は、特異ベクトルを算出することができる。

【0049】

(ステップS305) 第2特異ベクトル算出部23は、算出した特異ベクトルを正規化する。すなわち、第2特異ベクトル算出部23は、算出した特異ベクトルのノルムを算出し、特異ベクトルを算出したノルムで割ったものを最終的な特異ベクトルとして特異ベクトル記憶部20に蓄積する。このようにして、2重対角行列Bを特異値分解する処理は終了となる。

なお、この後に、行列記憶部11が記憶している行列Aの特異ベクトルを算出する処理が行われてもよいが、ここでは省略している。

10

20

30

40

50

【 0 0 5 0 】

また、特異ベクトル算出部 19 が算出した特異ベクトルの精度を上げるために、図 5 で示されるように、特異ベクトルに関する処理を実行してもよく、あるいは、実行しなくてもよい。すなわち、図示しない逆反復法処理部は、特異ベクトル記憶部 20 から特異ベクトル算出部 19 が算出した特異ベクトルを読み出し、その特異ベクトルに対して逆反復法の処理を実行し、その結果の特異ベクトルを特異ベクトル記憶部 20 に蓄積する（ステップ S 4 0 1）。次に、図示しないグラムシュミット処理部は、高精度が必要かどうか判断する（ステップ S 4 0 2）。この判断は、あらかじめ高精度が必要かどうか設定されている記録媒体等から、その設定を読み出すことによって判断してもよい。そして、図示しないグラムシュミット処理部は、高精度が必要な場合には、特異ベクトル記憶部 20 から逆反復法の処理の実行された特異ベクトルを読み出し、その特異ベクトルに対してグラムシュミット法の処理を実行し、その結果の特異ベクトルを特異ベクトル記憶部 20 に蓄積する（ステップ S 4 0 3）。なお、逆反復法や、グラムシュミット法については、すでに公知であり、その詳細な説明を省略する。

10

【 0 0 5 1 】

次に、本実施の形態による特異値分解装置 1 の動作について、以下、より詳細に説明する。

[行列 A の対角化]

行列 A は、例えば、ハウスホルダー変換等を用いて、以下に示されるように 2 重対角化することができることが知られている。ここでは、上 2 重対角化する場合について示すが、下 2 重対角化も同様にして行うことができる。ここで、2 重対角行列 B は、行の数と列の数とが一致する正方行列である。

20

【 数 1 】

$$U_H^T A V_H = \begin{cases} (B \ 0) & (l_1 \leq l_2) \\ \begin{pmatrix} B \\ 0 \end{pmatrix} & (l_1 \geq l_2) \end{cases}, \quad B = \begin{pmatrix} b_1 & b_2 & & \\ & b_3 & \ddots & \\ & & \ddots & b_{2m-2} \\ \mathbf{0} & & & b_{2m-1} \end{pmatrix}$$

ただし、A は $l_1 \times l_2$ 実行列、 U_H, V_H は直交行列、 $m = \min(l_1, l_2)$ である。

30

【 0 0 5 2 】

したがって、対角化部 12 は、上記のようにして、行列記憶部 11 から行列 A を読み出し、2 重対角行列 B を算出することができる。その算出された 2 重対角行列は、対角行列記憶部 13 で記憶される（ステップ S 1 0 1）。

【 0 0 5 3 】

[2 重対角行列 B の分割]

2 重対角行列 B を分割する処理について説明する。まず、図 6 で示されるように、上 2 重対角行列であって、正方行列である B の右端に、全ての要素が 0 である列を加えたものを新たに上 2 重対角行列 B とする。なお、この新たな 2 重対角行列 B の特異値は、元の正方行列である 2 重対角行列 B の特異値と同じである。したがって、今後、この新たな $n \times (n + 1)$ 行列の特異値を求める処理について説明する。

40

【 0 0 5 4 】

図 7 で示されるように、 $n \times (n + 1)$ の上 2 重対角行列 B が与えられた場合に、それらを 2 個の上 2 重対角行列と、2 個の要素（図 7 では、 b_7, b_8 ）とに分けることができる。したがって、上 2 重対角行列 B は、以下のように分割されることになる。

【 数 2 】

$$B = \begin{pmatrix} B_1 & 0 \\ b_{2k-1} \mathbf{e}_k^T & b_{2k} \mathbf{e}_1^T \\ 0 & B_2 \end{pmatrix} \quad (\text{式 1})$$

50

【 0 0 5 5 】

ただし、上 2 重対角行列 B が $n \times (n + 1)$ 行列であるため、上 2 重対角行列 B_1 は、 $(k - 1) \times k$ 行列であり、上 2 重対角行列 B_2 は、 $(n - k) \times (n - k + 1)$ 行列である。 k は、 $1 < k < n$ となる整数である。 e_j は、適切な次元における j 番目の単位ベクトルである。ここで、並列処理を適切に実行するためには、 k を、 $n / 2$ を超えない最大の整数にとるか、あるいは、 $n / 2$ を下まわらない最小の整数にとることが好適である（この k の値を用いて行列を 2 個の行列に分割する場合を、「行列を略半分の 2 個の行列に分割する」と呼ぶことにする）。

【 数 3 】

$$k = \lfloor n/2 \rfloor$$

10

本実施の形態では、 k を上式のようにとるものとする。なお、 k の値は、前述のように、 $1 < k < n$ の範囲内で任意であることは言うまでもない。

【 0 0 5 6 】

したがって、行列分割部 1 4 は、上述のようにして、対角行列記憶部 1 3 が記憶している上 2 重対角行列 B を 2 個の上 2 重対角行列と、2 個の要素に分割することができ、その分割の処理を繰り返すことができる。図 8 は、図 3 のフローチャートのステップ S 2 0 1 における行列分割部 1 4 による行列を分割する処理を示すフローチャートである。

【 0 0 5 7 】

（ステップ S 5 0 1）行列分割部 1 4 は、カウンタ I を「1」に設定する。

20

（ステップ S 5 0 2）行列分割部 1 4 は、対角行列記憶部 1 3 から I 番目の分割をしていない 2 重対角行列を読み出し、その 2 重対角行列を 2 個の 2 重対角行列と、2 個の要素とに分割する。そして、行列分割部 1 4 は、分割した 2 個の 2 重対角行列と、2 個の要素とを対角行列記憶部 1 3 に蓄積する。

【 0 0 5 8 】

（ステップ S 5 0 3）行列分割部 1 4 は、 I 番目の分割を行っていない 2 重対角行列が、対角行列記憶部 1 3 で記憶されているかどうか判断する。そして、 I 番目の分割を行っていない 2 重対角行列が、対角行列記憶部 1 3 で記憶されている場合には、ステップ S 5 0 2 に戻り、そうでない場合には、ステップ S 5 0 4 に進む。

【 0 0 5 9 】

（ステップ S 5 0 4）行列分割部 1 4 は、 I 番目の分割を行った 2 重対角行列の大きさがあらかじめ決められている大きさ以下かどうか判断する。行列分割部 1 4 は、例えば、目的とする行列の大きさ（例えば、 25×26 など）の記憶されている図示しない記録媒体から、その目的とする行列の大きさを読み出し、対角行列記憶部 1 3 で記憶されている、 I 番目の分割後の 2 重対角行列がその大きさ以下であるかどうかを判断してもよい。そして、 I 番目の分割を行った 2 重対角行列の大きさがあらかじめ決められている大きさ以下である場合には、2 重対角行列を分割する処理は終了となり、そうでない場合には、ステップ S 5 0 5 に進む。

30

（ステップ S 5 0 5）行列分割部 1 4 は、カウンタ I を 1 だけインクリメントする。そして、ステップ S 5 0 2 に戻る。

40

【 0 0 6 0 】

なお、この図 8 のフローチャートでは、上述のように、行列分割部 1 4 が各行列を、略半分の 2 個の行列に分割するため、 I 番目の分割後の各 2 重対角行列の大きさがほとんど同じである場合について説明したが、行列分割部 1 4 が各行列を、略半分の 2 個の行列に分割しない場合には、行列分割部 1 4 は、分割後の各行列があらかじめ決められた大きさ以下となるように、分割を繰り返すものとする。

【 0 0 6 1 】

また、図 8 のフローチャートでは、ステップ S 5 0 4 において、行列分割部 1 4 が、対角行列記憶部 1 3 で記憶されている分割後の行列の大きさをあらかじめ決められた大きさと比較する処理を実行する場合について説明したが、これは一例であって、行列分割部 1

50

4は、ステップS504において、それ以外の処理を行ってもよい。例えば、上述のように、行列分割部14が各行列を、略半分の2個の行列に分割する場合には、元の2重対角行列Bの大きさを知ることができれば、何番目の分割で目的とする行列の大きさとなるのかを知ることができる。したがって、N番目（Nは1以上の整数）の分割で目的とする行列の大きさとなる場合には、ステップS504において、IがNであるかどうかを比較し、NでなければステップS505に進み、Nであれば一連の処理を終了するようにしてもよい。

【0062】

図9は、行列の分割について説明するための図である。まず、行列分割部14は、1番目の分割として、2重対角行列Bを、2重対角行列 B_1 と、2重対角行列 B_2 とに分割する（ステップS501, S502）。2重対角行列Bは、1個しかいないため、行列分割部14は、1番目の分割をしていない行列がないと判断する（ステップS503）。また、2重対角行列 B_1 等はあらかじめ決められた大きさ以下の行列でないとすると（ステップS504）、行列分割部14は、2番目の分割として、2重対角行列 B_1 を、2重対角行列 B_{11} と、2重対角行列 B_{12} とに分割する（ステップS505, S502）。この場合には、2番目の分割を行っていない2重対角行列 B_2 が存在するため、行列分割部14は、2重対角行列 B_2 も、2重対角行列 B_{21} と、2重対角行列 B_{22} とに分割する（ステップS503, S502）。このようにして、分割後の各2重対角行列が目的とする大きさ以下になるまで、行列を分割する処理が繰り返される。なお、図9では、2重対角行列以外の2個の要素については省略している。また、図9において、各2重対角行列は、列の数が行の数よりも1だけ大きい行列である。

【0063】

[分割された行列の特異値分解]

行列 B_i が $n \times (n+1)$ 行列であるとする、行列 B_i の特異値分解は、次のようになる。

【数4】

$$B_i = U_i (D_i \ 0) (V_i \ v_i)^T$$

【0064】

ここで、 D_i は、 $n \times n$ の対角行列である。 $(D_i \ 0)$ は、行列 D_i の右側に全ての要素が0の列が1つある行列である。 D_i の各対角成分は B_i の特異値である。また、 v_i は、行列 B_i の特異値分解における右直交行列の一番右側のベクトルである。 V_i は、行列 B_i の特異値分解における右直交行列の v_i をのぞいた行列である。 $(V_i \ v_i)^T$ は、全体として $(n+1) \times (n+1)$ の行列である。

【0065】

したがって、特異値分解部15は、対角行列記憶部13から分割後の各上2重対角行列を読み出し、上記のようにして、特異値分解を行う（ステップS202）。特異値分解の方法として、例えば、2分法と逆反復法を組み合わせた方法、MR³法、QRs法等を用いてもよいことは前述の通りである。図9で示されるように上2重対角行列の分割が行われた場合には、特異値分解部15は、各2重対角行列 B_{111} , B_{112} , B_{121} , B_{122} , …, B_{222} に対して特異値分解を行うことになる。なお、特異値分解部15は、特異値分解の結果得られた各特異値と、特異値分解の結果得られた左右直交行列の一部の要素である行列要素とを特異値分解記憶部16に蓄積する。ここで、行列要素が、左右直交行列のどの要素を含むのかについては後述する。

【0066】

[特異値の算出]

まず、図10で示されるように、分割された2個の2重対角行列 B_1 , B_2 から、分割元の2重対角行列 B_0 の特異値等を算出する処理について説明する。2重対角行列 B_1 , B_2 は、次のように特異値分解されていたとする。ここで、2重対角行列 B_1 , B_2 は、両者共に列の数が行の数よりも1だけ大きい行列であるとする。

【数 5】

$$B_1 = U_1(D_1 \ 0)(V_1 \ \mathbf{v}_1)^T$$

$$B_2 = U_2(D_2 \ 0)(V_2 \ \mathbf{v}_2)^T$$

【0067】

この場合に、分割元の2重対角行列 B_0 は、次のようになる。

【数 6】

$$B_0 = \begin{pmatrix} 0 & U_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & U_2 \end{pmatrix} \begin{pmatrix} b_{2k-1}\Psi_1 & b_{2k-1}\mathbf{l}_1 & b_{2k}\mathbf{f}_2 & b_{2k}\Phi_2 \\ 0 & D_1 & 0 & 0 \\ 0 & 0 & D_2 & 0 \end{pmatrix} \times \begin{pmatrix} \mathbf{v}_1 & V_1 & 0 & 0 \\ 0 & 0 & V_2 & \mathbf{v}_2 \end{pmatrix}^T \quad 10$$

ただし、 \mathbf{l}_1 は V_1 の最後の行である。 Ψ_1 は \mathbf{v}_1 の最後の要素である。

\mathbf{f}_2 は V_2 の最初の行である。 Φ_2 は \mathbf{v}_2 の最初の要素である。

【0068】

また、上式の b_{2k-1} 等は2重対角行列の分割の処理において説明した行列の要素である。ここで、Givens変換を行い、 b_{2k-2} を0にすると、次のようになる。

【数 7】

$$B_0 = \tilde{U}(M \ 0)(\tilde{V} \ \tilde{\mathbf{v}})^T \quad 20$$

【0069】

ただし、

【数 8】

$$\tilde{U} \equiv \begin{pmatrix} 0 & U_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & U_2 \end{pmatrix}, \quad M \equiv \begin{pmatrix} r_0 & b_{2k-1}\mathbf{l}_1 & b_{2k}\mathbf{f}_2 \\ 0 & D_1 & 0 \\ 0 & 0 & D_2 \end{pmatrix} \quad 30$$

$$\tilde{V} \equiv \begin{pmatrix} c_0\mathbf{v}_1 & V_1 & 0 \\ s_0\mathbf{v}_2 & 0 & V_2 \end{pmatrix}, \quad \tilde{\mathbf{v}} \equiv \begin{pmatrix} -s_0\mathbf{v}_1 \\ c_0\mathbf{v}_2 \end{pmatrix}$$

$$r_0 = \sqrt{(b_{2k-1}\Psi_1)^2 + (b_{2k}\Phi_2)^2}, \quad c_0 = \frac{b_{2k-1}\Psi_1}{r_0}, \quad s_0 = \frac{b_{2k}\Phi_2}{r_0}$$

である。したがって、行列 B_0 は、

【数 9】

$$\tilde{U}, \quad (\tilde{V} \ \tilde{\mathbf{v}})^T$$

によって、 $(M \ 0)$ に直交変換される。これらより、行列 B_0 の特異値分解を行うためには、行列 M の特異値分解を行えばよいことになる。ここで、 $n \times n$ の行列 M を次のよう
40

【数 10】

$$M = \begin{pmatrix} z_1 & z_2 & \cdots & z_n \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}$$

【0070】

上記の行列 M において、 z_i 、 d_j 以外の各要素は0である。ただし、 $i = 1, 2, \dots, n$ 、 $j = 2, 3, \dots, n$ である。この行列 M の特異値分解は、次の定理によって行うこ
50

とができる。

(定理 1)

【数 1 1】

M の特異値分解を $U\Sigma V^T$ とし、

$U = (u_1, \dots, u_n)$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $V = (v_1, \dots, v_n)$ とすると、

特異値 $\{\sigma_i\}_{i=1}^n$ は、

$$0 = d_1 < \sigma_1 < d_2 < \dots < d_n < \sigma_n < d_n + \|z\|_2$$

を満たす。また、特異値の方程式は、

$$f(\sigma_i) \equiv 1 + \sum_{j=1}^n \frac{z_j^2}{d_j^2 - \sigma_i^2} = 0 \quad (\text{式 2})$$

となる。特異ベクトルは、

$$u_i = \frac{\left(-1, \frac{d_2 z_2}{d_2^2 - \sigma_i^2}, \dots, \frac{d_n z_n}{d_n^2 - \sigma_i^2} \right)^T}{\sqrt{1 + \sum_{k=2}^n \frac{(d_k z_k)^2}{(d_k^2 - \sigma_i^2)^2}}} \quad (\text{式 3})$$

$$v_i = \frac{\left(\frac{z_1}{d_1^2 - \sigma_i^2}, \dots, \frac{z_n}{d_n^2 - \sigma_i^2} \right)^T}{\sqrt{\sum_{k=1}^n \frac{z_k^2}{(d_k^2 - \sigma_i^2)^2}}} \quad (\text{式 4})$$

となる。

【0 0 7 1】

ここで、計算機によって求めることができるのは、行列 M の真の特異値ではなく、誤差を含む近似値である。したがって、

【数 1 2】

$$\frac{z_k}{d_k^2 - \hat{\sigma}_i^2}, \frac{d_k z_k}{d_k^2 - \hat{\sigma}_i^2} \quad (\text{ただし、}\hat{\sigma}_i\text{は計算機による近似値})$$

は、真の値である

【数 1 3】

$$\frac{z_k}{d_k^2 - \sigma_i^2}, \frac{d_k z_k}{d_k^2 - \sigma_i^2}$$

と大きく異なる可能性がありうる。すなわち、上記のように計算した場合には、特異ベクトルの計算が数値不安定となることがある。この欠点は、次の定理によって克服できる。

【0 0 7 2】

(定理 2)

10

20

30

40

【数 1 4】

対角行列 $D = \text{diag}(d_1, \dots, d_n)$ が与えられ、

$0 \equiv d_1 < \hat{\sigma}_1 < d_2 < \dots < d_n < \hat{\sigma}_n$ を満たす集合 $\{\hat{\sigma}_i\}_{i=1}^n$ が与えられると、

特異値が $\{\hat{\sigma}_i\}_{i=1}^n$ である行列

$$\hat{M} = \begin{pmatrix} \hat{z}_1 & \hat{z}_2 & \cdots & \hat{z}_n \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}$$

10

が存在する。ベクトル $\hat{z} = (\hat{z}_1, \hat{z}_2, \dots, \hat{z}_n)^T$ は、次式で与えられる。

$$|\hat{z}_i| = \sqrt{(\hat{\sigma}_n^2 - d_i^2) \prod_{k=1}^{i-1} \frac{(\hat{\sigma}_k^2 - d_i^2)}{(d_k^2 - d_i^2)} \prod_{k=i}^{n-1} \frac{(\hat{\sigma}_k^2 - d_i^2)}{(d_{k+1}^2 - d_i^2)}} \quad (\text{式 5})$$

ここで、 \hat{z}_i の符号は任意である。

【0 0 7 3】

したがって、定理 1 を用いて行列 M の近似特異値を算出した後に、定理 2 を用いて、その近似特異値を真の特異値とする行列 M を再構成する。特異ベクトルは、上記の式 5 を用いて算出した

20

【数 1 5】

$$\hat{z}_i$$

と、上記の式 2 を用いて算出した

【数 1 6】

$$\hat{\sigma}_i$$

とを上記の式 3, 式 4 に代入することによって数値安定的に求めることができる。このようにして、次のように行列 M の特異値分解が求められたとする。

【数 1 7】

$$M = U_M \Sigma V_M^T$$

ただし、 $U_M = (\hat{u}_1, \dots, \hat{u}_n)$, $\Sigma = \text{diag}(\hat{\sigma}_1, \dots, \hat{\sigma}_n)$, $V_M = (\hat{v}_1, \dots, \hat{v}_n)$ とする。

\hat{u}_i, \hat{v}_i はそれぞれ、 $\hat{z}_i, \hat{\sigma}_i$ を用いて算出された u_i, v_i である。

【0 0 7 4】

すると、 B_0 の特異値分解は以下ようになる。

【数 1 8】

$$\begin{aligned} B_0 &= \tilde{U} \begin{pmatrix} M & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} & \tilde{v} \end{pmatrix}^T \\ &= \tilde{U} \begin{pmatrix} U_M \Sigma V_M^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} & \tilde{v} \end{pmatrix}^T \\ &= \tilde{U} U_M \begin{pmatrix} \Sigma & 0 \\ 0 & 1 \end{pmatrix}^T \begin{pmatrix} \tilde{V} & \tilde{v} \end{pmatrix}^T \\ &= U \begin{pmatrix} \Sigma & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} V_M^T & \tilde{v} \end{pmatrix}^T \\ &= U \begin{pmatrix} \Sigma & 0 \end{pmatrix} V^T \end{aligned}$$

40

【0 0 7 5】

このように、対象となる行列を 2 個の副行列に分割し、その分割した後の各副行列について特異値分解を行う処理を再帰的に行うことによって特異値分解を行う方法は、分割統治法 (Divide and Conquer: D&C) と呼ばれている。

50

【 0 0 7 6 】

分割統治法では、特異ベクトルまで算出することになり、その特異ベクトルを算出する処理において行列の計算をしなければならぬため、非常に負荷の大きい処理となる。分割統治法において特異値分解をする場合には、例えば、計算時間の95%程度が特異ベクトルを算出するベクトル更新の処理（例えば、上述のように、行列Mの左右直交行列から、行列 B_0 の左右直交行列を算出するために行列を掛け合わせる処理）に費やされることもありうる。

【 0 0 7 7 】

一方、特異値のみを算出する場合には、上記処理を簡略化することができる。次に、その方法について説明する。上記の結果より、次のようになる。

【 数 1 9 】

$$\begin{aligned} B_0 &= \tilde{U}(M \ 0)(\tilde{V} \ \tilde{\mathbf{v}})^T \\ &= U(\Sigma \ 0)(\tilde{V}V_M \ \tilde{\mathbf{v}})^T \\ &= U(\Sigma \ 0)\left(\begin{pmatrix} c_0\mathbf{v}_1 & V_1 & 0 \\ s_0\mathbf{v}_2 & 0 & V_2 \end{pmatrix}V_M \begin{pmatrix} -s_0\mathbf{v}_1 \\ c_0\mathbf{v}_2 \end{pmatrix}\right)^T \end{aligned}$$

【 0 0 7 8 】

これより、

【 数 2 0 】

$$\mathbf{f} = (c_0\phi_1 \ \mathbf{f}_1 \ 0)V_M \quad (\text{式 6})$$

$$\mathbf{l} = (s_0\psi_2 \ 0 \ \mathbf{l}_2)V_M \quad (\text{式 7})$$

$$\phi = -s_0\phi_1 \quad (\text{式 8})$$

$$\psi = c_0\psi_2 \quad (\text{式 9})$$

ただし、 \mathbf{f}_1 は V_1 の最初の行、 ϕ_1 は \mathbf{v}_1 の最初の要素、 \mathbf{l}_2 は V_2 の最後の行、

\mathbf{l} は V の最後の行、 ψ は \mathbf{v} の最後の要素、 \mathbf{f} は V の最初の行、

ϕ は \mathbf{v} の最初の要素とする。

となる。ここで、 \mathbf{f} と ϕ とは、行列 B_0 を特異値分解した結果の右直交行列の最初の行の各要素である。また、 \mathbf{l} と ψ とは、行列 B_0 を特異値分解した結果の右直交行列の最後の行の各要素である。この右直交行列の最初の行の各要素と、最後の行の各要素とが行列要素となる。

【 0 0 7 9 】

上記の結果から、図10で示されるように、行列 B_1 、 B_2 から分割元の行列 B_0 を構成する場合に、行列 B_0 について、

【 数 2 1 】

$$\hat{\sigma}_i, \hat{z}_i, V_M, \mathbf{f}, \mathbf{l}, \phi, \psi$$

をそれぞれ算出することができる。このような処理を繰り返すことにより、最終的に、行列Aを2重対角化した2重対角行列Bの特異値を算出することができる。

【 0 0 8 0 】

したがって、特異値算出部17は、上述のようにして、特異値分解記憶部16から特異値と行列要素とを読み出し、対角行列記憶部13から行列の分割時に発生した2個の要素(b_{2k-1} 、 b_{2k})を読み出し、それらを用いることによって、分割元の2重対角行列の特異値と行列要素とを算出する。そして、それらを算出する処理を繰り返すことによって、最終的に2重対角行列Bの特異値を算出する。図11は、図3のフローチャートのステップS203における特異値算出部17が特異値を算出する処理を示すフローチャートである。

10

20

30

40

50

【 0 0 8 1 】

(ステップ S 6 0 1) 特異値算出部 1 7 は、カウンタ J を「 1 」に設定する。

(ステップ S 6 0 2) 特異値算出部 1 7 は、J 番目の特異値の算出は最後の特異値の算出であるかどうか判断する。ここで、最後の特異値の算出とは、2 重対角行列 B の特異値を算出することである。そして、最後の特異値の算出である場合には、ステップ S 6 0 6 に進み、そうでない場合には、ステップ S 6 0 3 に進む。

【 0 0 8 2 】

(ステップ S 6 0 3) 特異値算出部 1 7 は、分割元の 2 重対角行列の特異値等を算出する。この処理の詳細については後述する。

(ステップ S 6 0 4) 特異値算出部 1 7 は、J 番目の特異値の算出において、分割元の全ての 2 重対角行列の特異値等を算出したかどうか判断する。そして、J 番目の特異値の算出において、分割元の全ての 2 重対角行列の特異値等を算出した場合には、ステップ S 6 0 5 に進み、そうでない場合には、ステップ S 6 0 3 に戻る。

【 0 0 8 3 】

(ステップ S 6 0 5) 特異値算出部 1 7 は、カウンタ J を 1 だけインクリメントする。そして、ステップ S 6 0 2 に戻る。

(ステップ S 6 0 6) 特異値算出部 1 7 は、2 重対角行列 B の特異値を算出し、その算出した特異値を特異値記憶部 1 8 に蓄積する。このようにして、2 重対角行列 B の特異値を算出する一連の処理は終了となる。

【 0 0 8 4 】

図 1 2 は、図 1 1 のフローチャートにおけるステップ S 6 0 3 の詳細な処理を示すフローチャートである。

(ステップ S 7 0 1) 特異値算出部 1 7 は、分割元の行列の特異値を式 2 を用いて算出する。そして、特異値算出部 1 7 は、算出した特異値を特異値分解記憶部 1 6 に蓄積する。

【 0 0 8 5 】

(ステップ S 7 0 2) 特異値算出部 1 7 は、ステップ S 7 0 1 で算出した特異値を用いて、式 5 の z を算出する。

(ステップ S 7 0 3) 特異値算出部 1 7 は、ステップ S 7 0 1 で算出した特異値と、ステップ S 7 0 2 で算出した z とを用いて、式 4 の v_i を算出する。この v_i を算出することにより、 v_i を列ベクトルに有する V_M を算出したことになる。そして、特異値算出部 1 7 は、算出した V_M を特異値分解記憶部 1 6 に蓄積する。

【 0 0 8 6 】

(ステップ S 7 0 4) 特異値算出部 1 7 は、式 6 から式 9 を用いて、分割元の行列に関する行列要素を算出する。そして、特異値算出部 1 7 は、算出した分割元の行列要素を特異値分解記憶部 1 6 に蓄積する。このようにしてステップ S 6 0 3 の処理は終了となる。

【 0 0 8 7 】

図 1 3 は、特異値を算出する処理について説明するための図である。特異値分解記憶部 1 6 では、特異値分解部 1 5 によって行列 B_{111} , B_{112} , \dots , B_{222} が特異値分解された結果、すなわち、それらの特異値と、行列要素とが記憶されているものとする。まず、特異値算出部 1 7 は、特異値等の 1 番目の算出を開始する (ステップ S 6 0 1, S 6 0 2)。ここで、特異値等の 1 番目の算出とは、図 1 3 の一番下の行の行列の特異値と、行列要素とから、下から 2 番目の行の行列の特異値と、行列要素とを算出することである。特異値算出部 1 7 は、行列 B_{111} と、行列 B_{112} との各特異値、及び行列 B_{111} と、行列 B_{112} との各行列要素を特異値分解記憶部 1 6 から読み出す。また、行列 B_{111} を行列 B_{111} と、行列 B_{112} とに分解したときに発生した 2 個の要素を対角行列記憶部 1 3 から読み出す。そして、特異値算出部 1 7 は、それらの値を用いて、行列 B_{111} の特異値を算出して特異値分解記憶部 1 6 に蓄積する (ステップ S 7 0 1)。次に、特異値算出部 1 7 は、行列 B_{111} の特異値を用いて z の値を算出する (ステップ S 7 0 2)。特異値算出部 1 7 は、行列 B_{111} の特異値と、 z の値とを用いて直交行列 V_M を算出

10

20

30

40

50

して特異値分解記憶部 16 に蓄積する (ステップ S 7 0 3)。最後に、特異値算出部 17 は、行列 B_{11} の行列要素を算出して特異値分解記憶部 16 に蓄積する (ステップ S 7 0 4)。

【0088】

特異値等の 1 番目の算出はまだ終わっていないため (ステップ S 6 0 4)、特異値算出部 17 は、上記説明と同様にして、行列 B_{121} 、行列 B_{122} の特異値、行列要素等に基づいて、行列 B_{12} の特異値、行列要素を算出する (ステップ S 6 0 2, S 6 0 3)。このようにして、特異値等の 1 番目の算出が終了すると、特異値算出部 17 は、特異値等の 2 番目の算出、すなわち、行列 B_{11} 、行列 B_{12} の特異値、行列要素等に基づいて行列 B_1 の特異値、行列要素の算出を行う (ステップ S 6 0 4, S 6 0 5, S 6 0 2, S 6 0 3)。その後、特異値算出部 17 は、同様にして、行列 B_{21} 、行列 B_{22} の特異値、行列要素等に基づいて行列 B_2 の特異値、行列要素の算出を行う (ステップ S 6 0 4, S 6 0 3)。

10

【0089】

特異値算出部 17 が特異値等の 2 番目の算出を終了すると (ステップ S 6 0 4)、次の特異値等の算出は、2 重対角行列 B の特異値を求める最後の処理となるため (ステップ S 6 0 5, S 6 0 2)、特異値算出部 17 は、特異値の算出のみを行い、その算出した特異値を特異値記憶部 18 に蓄積する (ステップ S 6 0 6)。このようにして、特異値の算出が終了する。

【0090】

20

[特異値からの特異ベクトルの算出]

まず、行列 $B^T B$ を考える。ここで、行列 B は、前述の $n \times n$ の上 2 重対角行列である。この行列 $B^T B$ を次のように対角化したとする。

【数 2 2】

$$\Lambda = V^T B^T B V$$

ただし、 $\Lambda \equiv \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$, ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$)

$$V \equiv (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$$

λ_j は $B^T B$ の固有値であり、 \mathbf{x}_j は固有値 λ_j に対する固有ベクトルである。

30

【0091】

ここで、一般に次のことが成り立つ。

(1) 行列 $B^T B$ は対称な 3 重対角行列である。

(2) 行列 $B^T B$ の固有値は全て正であり、行列 B の特異値 σ_j ($j = 1, 2, \dots, m$) は、行列 $B^T B$ の固有値と次の関係を有する。

【数 2 3】

$$\sigma_j = \sqrt{\lambda_j}$$

(3) $V_B = V$ である。ただし、 V_B は行列 B の右直交行列である。したがって、行列 $B^T B$ の固有ベクトル \mathbf{x}_j は、行列 B の右特異ベクトルに等しい。

40

【0092】

したがって、行列 $B^T B$ の固有ベクトルを求めると、行列 B の右直交行列が求まることになる。さらに、行列 B の固有値分解を、

【数 2 4】

$$U_B \Sigma_B^T = B$$

とする。すると、右直交行列 V_B が求まり、特異値を対角成分に有する行列 Σ_B が求まることよって、

【数 2 5】

$$U_B = BV_B \Sigma^{-1}$$

から、左直交行列も求まることになり、行列 B が特異値分解される。したがって、行列 B の特異ベクトルを求めることは、行列 $B^T B = T_S$ の固有ベクトルを求めることに置き換えることができる。すなわち、次の方程式の固有ベクトル x_j を求めればよいことになる。

【数 2 6】

$$(B^T B - \sigma_j^2 I) x_j = \gamma_k e_k \quad (j=1,2,\dots,m)$$

ただし、行列 B は $m \times m$ 行列である。

e_k は、 k 番目の要素が 1 で他の要素が 0 のベクトルである。

(e_k は、単位行列 I の k 番目の列である)

10

【0 0 9 3】

本来であれば、上記式の右辺は 0 になるはずであるが、行列 B の特異値を算出するときに、いくらかの誤差を含むため、特異ベクトル x_j が真値であれば、上記のように右辺に残差項が存在する。

【0 0 9 4】

ここで、以下のようにコレスキー分解できたとする。

20

【数 2 7】

$$B^T B - \sigma_j^2 I = \begin{cases} (B^{(+1)})^T B^{(+1)} \\ (B^{(-1)})^T B^{(-1)} \end{cases}$$

$$B^{(n)} \equiv \begin{pmatrix} b_1^{(n)} & b_2^{(n)} & & \\ & b_3^{(n)} & \ddots & \\ & & \ddots & b_{2m-2}^{(n)} \\ \mathbf{0} & & & b_{2m-1}^{(n)} \end{pmatrix}, \quad (n=0,+1),$$

$$B^{(0)} \equiv B$$

30

$$B^{(-1)} \equiv \begin{pmatrix} b_1^{(-1)} & & & \mathbf{0} \\ b_2^{(-1)} & b_3^{(-1)} & & \\ & \ddots & \ddots & \\ & & b_{2m-2}^{(-1)} & b_{2m-1}^{(-1)} \end{pmatrix}$$

$$b_{2k-1}^{(n)} \equiv \xi_k^{(n)} \sqrt{q_k^{(n)}}, \quad b_{2k}^{(n)} \equiv \eta_k^{(n)} \sqrt{e_k^{(n)}},$$

$$(\xi_k^{(n)})^2 = 1, \quad (\eta_k^{(n)})^2 = 1, \quad (n=0,+1,-1),$$

$$\xi_k^{(1)} \eta_k^{(1)} = \xi_k^{(0)} \eta_k^{(0)}, \quad \xi_{k+1}^{(-1)} \eta_k^{(-1)} = \xi_k^{(0)} \eta_k^{(0)}$$

40

【0 0 9 5】

すると、次式のように書ける。

【数 2 8】

$$B^T B - \sigma_j^2 I = \begin{cases} LD^+ L^T \\ UD^- U^T \end{cases}$$

ただし、 L, U, D は以下の通りである。

$$L \equiv \begin{pmatrix} 1 & & & \mathbf{0} \\ L_1 & 1 & & \\ & \ddots & \ddots & \\ & & L_{m-1} & 1 \end{pmatrix}, \quad L_k \equiv \frac{b_{2k}^{(+1)}}{b_{2k-1}^{(+1)}} \quad 10$$

$$U \equiv \begin{pmatrix} 1 & U_1 & & \\ & 1 & \ddots & \\ & & \ddots & U_{m-1} \\ \mathbf{0} & & & 1 \end{pmatrix}, \quad U_k \equiv \frac{b_{2k}^{(-1)}}{b_{2k+1}^{(-1)}}$$

$$D^\pm \equiv \text{diag}(D_1^\pm, D_2^\pm, \dots, D_m^\pm), \quad D_k^\pm \equiv (b_{2k-1}^{(\pm)})^2$$

【0 0 9 6】

そして、次のようになる。

【数 2 9】

$$B^T B - \sigma_j^2 I = N_k D_k (N_k)^T \quad 20$$

$$N_k \equiv \begin{pmatrix} 1 & & & & & & \\ L_1 & 1 & & & & & \\ & \ddots & \ddots & & & & \\ & & L_{k-1} & 1 & U_k & & \\ & & & 1 & \ddots & & \\ & & & & \ddots & U_m & \\ & & & & & & 1 \end{pmatrix}, \quad 30$$

$$D_k \equiv \text{diag}(D_1^+, D_2^+, \dots, D_{k-1}^+, \gamma_k, D_{k+1}^-, \dots, D_m^-)$$

【0 0 9 7】

ここで、行列 N_k をツイスト行列と呼ぶ。また、

【数 3 0】

$$D_k \mathbf{e}_k = \gamma_k \mathbf{e}_k, \quad N_k \mathbf{e}_k = \mathbf{e}_k$$

であるので、 $(B^T B - \sigma_j^2 I) \mathbf{x}_j = \mathbf{e}_k$ は、

【数 3 1】

$$N_k^T \mathbf{x}_j = \mathbf{e}_k \quad 40$$

となる。この簡単な式を解くことによって特異ベクトルを算出することができる。具体的には、

【数 3 2】

$$x_j(\rho) = \begin{cases} 1, & \rho = k, \\ -L_\rho x_j(\rho+1), & \rho = k-1, k-2, \dots, 1, \\ -U_{\rho-1} x_j(\rho-1), & \rho = k+1, k+2, \dots, m \end{cases}$$

ただし、 $x_j(\rho)$ は x_j の ρ 番目の要素である。

のようにわずかな演算で特異ベクトルを算出することができる。なお、ある 0 について、 $D_{0+} = 0$ あるいは $D_{0-} = 0$ であったとしても、行列 B の (\quad, \quad) 成分である $b_{2-1}^{(0)}$ と、行列 B の $(\quad, \quad + 1)$ 成分である $b_{2}^{(0)}$ を用いて、

10

【数 3 3】

$$x_j(\rho_0) = \begin{cases} -\frac{b_{2\rho_0+1}^{(0)} b_{2\rho_0+2}^{(0)}}{b_{2\rho_0-1}^{(0)} b_{2\rho_0}^{(0)}} x_j(\rho_0+2), & \rho_0 < k, \\ -\frac{b_{2\rho_0-5}^{(0)} b_{2\rho_0-4}^{(0)}}{b_{2\rho_0-3}^{(0)} b_{2\rho_0-2}^{(0)}} x_j(\rho_0-2), & \rho_0 > k \end{cases}$$

と特異ベクトルを算出することができる（このように特異ベクトルを算出する処理を例外処理と呼ぶことにする）。なお、残差項のパラメータ k 及び k の値は、

【数 3 4】

$$\gamma_k \equiv q_k^{(+1)} + q_k^{(-1)} - (e_{k-1}^{(0)} + q_k^{(0)} - \sigma_j^2) \quad (\text{式 1 0})$$

20

の絶対値が最小となるように決定する。したがって、上述のコレスキー分解を求め、ツイスト行列 N_k を求めることができれば、特異ベクトルを求めることになる。そこで、次にコレスキー分解について説明する。

【0 0 9 8】

図 1 4 で示すように、 $B^{(0)T} B^{(0)} - \sigma_j^2 I$ をコレスキー分解することは、 $B^{(0)}$ に対応する $\{q_k^{(0)}, e_k^{(0)}\}$ から、 $B^{(\pm 1)}$ に対応する $\{q_k^{(\pm 1)}, e_k^{(\pm 1)}\}$ を求めることである。

【0 0 9 9】

(q d 型ツイスト分解法)

まず、従来から知られている q d 型ツイスト分解法で用いる q d 型変換について説明する (図 1 4 参照)。

【数 3 5】

$$\text{q d 型変換: } \{q_k^{(0)}, e_k^{(0)}\} \mapsto \{q_k^{(\pm 1)}, e_k^{(\pm 1)}\}$$

30

【0 1 0 0】

この変換は、さらに stationary q d with shift (stqds) 変換

【数 3 6】

$$\text{stqds 変換: } \{q_k^{(0)}, e_k^{(0)}\} \mapsto \{q_k^{(+1)}, e_k^{(+1)}\}$$

$$q_k^{(+1)} + e_{k-1}^{(+1)} = q_k^{(0)} + e_{k-1}^{(0)} - \sigma_j^2, \quad k = 1, 2, \dots, m,$$

$$q_k^{(+1)} e_k^{(+1)} = q_k^{(0)} e_k^{(0)}, \quad k = 1, 2, \dots, m-1,$$

$$e_0^{(+1)} \equiv 0, \quad e_0^{(0)} \equiv 0$$

及び reverse-time progressive q d with shift (rpqds) 変換

40

【数 3 7】

$$\begin{aligned} \text{r p q d s 変換} : \{q_k^{(0)}, e_k^{(0)}\} &\mapsto \{q_k^{(-1)}, e_k^{(-1)}\} \\ q_k^{(-1)} + e_k^{(-1)} &= q_k^{(0)} + e_{k-1}^{(0)} - \sigma_j^2, \quad k=1,2,\dots,m, \\ q_{k+1}^{(-1)} e_k^{(-1)} &= q_k^{(0)} e_k^{(0)}, \quad k=1,2,\dots,m-1, \\ e_0^{(0)} &\equiv 0, \quad e_m^{(-1)} \equiv 0 \end{aligned}$$

に分けられる。特異値 σ_j が既知であれば反復的な計算が不要なため、計算量を少なく抑えることができるが、常に数値安定性と精度が高い方法ではない。それは、s t q d s 変換、r p q d s 変換は、共に減算による桁落ちが発生する可能性があるからである。例えば、s t q d s 変換において、 $q_k^{(0)} + e_{k-1}^{(0)} - \sigma_j^2 \sim e_{k-1}^{(+1)}$ 10
 であれば、 $q_k^{(+1)}$ を求める際に、倍精度計算でもの有効数字がわずかに1桁になることもある。その場合には、 $q_k^{(0)} e_k^{(0)} / q_k^{(+1)}$ を計算すると誤差が生じる。つまり、 $e_k^{(+1)}$ が精度よく計算できないことになる。また、 $q_{k+1}^{(+1)}$ を求めるのに $e_k^{(+1)}$ が必要であり、 $e_k^{(+1)}$ を求めるのに $q_k^{(+1)}$ が必要であるといったように逐次的な計算が要求されるので、1箇所が発生した桁落ちによる誤差が波及し、さらなる誤差増大の可能性も秘めている。その結果、理論上は $q_k^{(+1)}$

0 であるが、誤差蓄積により $q_k^{(+1)} = 0$ となり、 $q_k^{(0)} e_k^{(0)} / q_k^{(+1)}$ の計算においてオーバーフローが起こるといった数値不安定な状況も想定される。 $B^{(0)}$ = B の成分 $\{b_{2k-1}^{(0)}, b_{2k}^{(0)}\}$ が与えられる、すなわち $\{q_k^{(0)}, e_k^{(0)}\}$ が与えられると、 σ_j^2 及び $e_{k-1}^{(+1)}$ が一意的に決まるので、この状況は避けることはできない。r p q d s 変換も同様の性質を持つため、実用的なレベルにまで達したとはいいがたい。L A P A C K において F O R T R A N ルーチン D S T E G R として改良版が公開されているものの欠点は完全に解決されてはいない。 20

【0 1 0 1】

(L V 型ツイスト分解法)

次に、L V 型ツイスト分解法で用いるミウラ変換、d L V v 型変換、逆ミウラ変換について説明する (図 1 4 参照) 。

【数 3 8】

$$\begin{aligned} \text{ミウラ変換} : \{q_k^{(0)}, e_k^{(0)}\} &\mapsto \{u_{2k-1}^{(0)}, u_{2k}^{(0)}\} \\ \text{s t d L V v 変換} : u_k^{(0)} &\mapsto u_k^{(+1)} \\ \text{r d L V v 変換} : u_k^{(0)} &\mapsto u_k^{(-1)} \\ \text{逆ミウラ変換} : \{u_{2k-1}^{(+1)}, u_{2k}^{(+1)}\} &\mapsto \{q_k^{(+1)}, e_k^{(+1)}\} \end{aligned}$$

【0 1 0 2】

まず、ミウラ変換について説明する。この変換は、次のように示される。

【数 3 9】

$$\begin{aligned} u_{2k-1}^{(0)} &= \frac{t_k^{(0)}}{\delta^{(0)}}, \quad k=1,2,\dots,m, \\ u_{2k}^{(0)} &= \frac{e_k^{(0)}}{t_k^{(0)}}, \quad k=1,2,\dots,m-1, \\ t_k^{(0)} &\equiv \frac{q_k^{(0)}}{\frac{1}{\delta^{(0)}} + u_{2k-2}^{(0)}} - 1, \end{aligned}$$

$$u_0^{(0)} \equiv 0,$$

ただし、 $\delta^{(0)}$ は任意である。

【0 1 0 3】

10

20

30

40

50

次に、d L V v型変換について説明する。この変換は、さらにstationary discrete Lotka - Volterra with variable step - size (std L V v)変換

【数40】

$$u_k^{(+1)} = u_k^{(0)} \times \frac{1 + \delta^{(0)} u_{k-1}^{(0)}}{1 + \delta^{(+1)} u_{k-1}^{(+1)}}, \quad k = 1, 2, \dots, 2m - 1,$$

$$u_0^{(0)} \equiv 0, \quad u_0^{(+1)} \equiv 0$$

及びreverse - time discrete Lotka - Volterra with variable step - size (rd L V v)変換

【数41】

$$u_k^{(-1)} = u_k^{(0)} \times \frac{1 + \delta^{(0)} u_{k-1}^{(0)}}{1 + \delta^{(-1)} u_{k+1}^{(-1)}}, \quad k = 1, 2, \dots, 2m - 1,$$

$$u_0^{(0)} \equiv 0, \quad u_{2m}^{(-1)} \equiv 0$$

に分けられる。ただし、

【数42】

$$\frac{1}{\delta^{(0)}} - \frac{1}{\delta^{(n)}} = \sigma_j^2, \quad (n = +1, -1)$$

を満たす範囲内で (± 1) は任意である。

【0104】

最後に、逆ミウラ変換について説明する。この変換は、次のように示される。

【数43】

$$q_k^{(\pm 1)} = \frac{(1 + \delta^{(\pm 1)} u_{2k-2}^{(\pm 1)}) \times (1 + \delta^{(\pm 1)} u_{2k-1}^{(\pm 1)})}{\delta^{(\pm 1)}}, \quad k = 1, 2, \dots, m,$$

$$e_k^{(\pm 1)} = \delta^{(\pm 1)} u_{2k-1}^{(\pm 1)} u_{2k}^{(\pm 1)}, \quad k = 1, 2, \dots, m - 1$$

$$u_0^{(\pm 1)} \equiv 0$$

【0105】

このようにして、q d型変換と同様に、コレスキー分解を行うことができる。q d型変換では見られない離散Lotka - Volterra型変換の大きな特徴は、任意パラメータを持つことである。すなわち、 $j^2 = 1 / (\delta^{(0)} - 1 / (\delta^{(\pm 1)}))$ を満たす範囲で $(\delta^{(n)})$ の値を任意に設定できる。 $(\delta^{(n)})$ を変動させると補助変数 $u_k^{(n)}$ の値も変化するが、桁落ちによる誤差や数値不安定が発生するかどうかは事前に判定できる。この判定は、if文によって実装されてもよい。この場合は、 $(\delta^{(n)})$ を再設定後に再度計算すればよい。また、 $u_k^{(\pm 1)}$ が求めれば逆ミウラ変換によって $q_k^{(\pm 1)}$ 及び $e_k^{(\pm 1)}$ が独立に計算されるので、誤差が伝播しないという性質を持つ。なお、逆ミウラ変換をミウラ変換、ミウラ変換を逆ミウラ変換と呼んでもよく、std L V v変換をs t L V v変換と呼んでもよく、rd L V v変換をr L V v変換と呼んでもよい。

【0106】

ここで、L V型ツイスト分解法による処理のより詳細な処理の一例について説明する。図15～図20は、L V型ツイスト分解法による処理の一例を示すフローチャートである。

【0107】

図15は、コレスキー分解の全体の処理の一例を示す図である。

(ステップS901)コレスキー分解部21は、ミウラ変換を行う。この処理の詳細については後述する。

(ステップS902)コレスキー分解部21は、 $1 / (\delta^{(\pm 1)})$ を $1 / (\delta^{(0)}) - j^2$ とする。

10

20

30

40

50

【0108】

(ステップS903) コレスキー分解部21は、後述するProcedure1の処理を実行する。

(ステップS904) コレスキー分解部21は、後述するProcedure2の処理を実行する。

【0109】

(ステップS905) コレスキー分解部21は、 $q_k^{(+1)}$, $e_k^{(+1)}$ がすでに算出されているかどうか判断する。そして、すでに算出されている場合には、コレスキー分解の一連の処理は終了となり、一方、算出されていない場合には、ステップS901に戻る。

10

【0110】

図16は、図15のフローチャートにおけるステップS903の処理の詳細を示すフローチャートである。

(ステップS1001) コレスキー分解部21は、 $q_k^{(+1)}$, $e_k^{(+1)}$ がすでに算出されているかどうか判断する。そして、すでに算出されている場合には、終了となり、算出されていない場合には、ステップS1002に進む。

(ステップS1002) コレスキー分解部21は、stdLVv変換等の処理を行う。この処理の詳細については後述する。

【0111】

図17は、図15のフローチャートにおけるステップS904の処理の詳細を示すフローチャートである。

20

(ステップS1101) コレスキー分解部21は、 $q_k^{(-1)}$, $e_k^{(-1)}$ がすでに算出されているかどうか判断する。そして、すでに算出されている場合には、終了となり、算出されていない場合には、ステップS1102に進む。

(ステップS1102) コレスキー分解部21は、rdLVv変換等の処理を行う。この処理の詳細については後述する。

【0112】

図18は、図15のフローチャートのステップS901の処理の詳細を示すフローチャートである。

(ステップS1201) コレスキー分解部21は、 $1/\lambda^{(0)}$ の値を決定する。この値は、前述のように任意に決定することができる。算出された特異値の小さい順に $\lambda_1, \lambda_2, \dots$ とする場合に、例えば、コレスキー分解部21は、 $1/\lambda^{(0)}$ の値を λ_1^2 よりも小さい値(例えば、「1」など)に設定し、その後、ステップS1203等で桁落ちの発生する可能性があると判断された場合に、 $1/\lambda^{(0)}$ の値を λ_j^2 と λ_{j+1}^2 の間 ($j = 1, 2, \dots$) に設定する、というように、 j を1ずつ大きくしながら $1/\lambda^{(0)}$ の値を設定していてもよい。なお、 $1/\lambda^{(0)}$ の値を決定する方法は、これに限定されないことは言うまでもない。

30

(ステップS1202) コレスキー分解部21は、 λ_1 を $q_1^{(0)} - 1/\lambda^{(0)}$ に設定する。

【0113】

(ステップS1203) コレスキー分解部21は、 λ_1 の絶対値が λ_2 よりも大きいかどうか判断する。そして、大きい場合には、ステップS1204に進み、そうでない場合には、ステップS1201に戻って、 $1/\lambda^{(0)}$ の値を決定する処理を再度実行する。なお、この λ_1 も任意に定めることができる。例えば、 ϵ としてマシン・イプシロンを用いてもよい。 λ_1 の値を大きくすると精度が向上し、 λ_1 の値を小さくすると精度が低下する。なお、このステップS1203で行っている処理は、桁落ちが発生する可能性について判断する処理である。 λ_1 の絶対値が λ_2 よりも大きくない場合には、桁落ちの発生する可能性があるとは判断されることになる。

40

【0114】

(ステップS1204) コレスキー分解部21は、 λ_2 を $q_1^{(0)} / (1 + \lambda_1^{(0)})$

50

$u_0^{(0)}$ に設定する。なお、前述のように $u_0^{(0)} = 0$ であるため、 2 は $q_1^{(0)}$ に設定されたことになる。

【0115】

(ステップS1205) コレスキー分解部21は、 $u_1^{(0)} (1 + u_0^{(0)})$ を 1 に設定する。ここで、 1 は、ステップS1202において $q_1^{(0)} - 1 / u_0^{(0)}$ に設定されているが、前述のように $u_0^{(0)} = 0$ であるため、 1 は、 $q_1^{(0)} / (1 + u_0^{(0)}) - 1 / u_0^{(0)}$ と等しく、これにミウラ変換を行うと $u_1^{(0)} = u_{2k-1}^{(0)} |_{k=1}$ となるからである。なお、 $u_0^{(0)} = 0$ から $(1 + u_0^{(0)}) = 1$ である。

【0116】

(ステップS1206) コレスキー分解部21は、カウンタ k を「1」に設定する。

(ステップS1207) コレスキー分解部21は、 1 を $e_k^{(0)} / 1$ に設定する。前述のように、 1 は $u_{2k-1}^{(0)}$ となるから、 $e_k^{(0)} / 1$ にミウラ変換を行うと、 1 は $u_{2k}^{(0)}$ と等しいことになる。

【0117】

(ステップS1208) コレスキー分解部21は、 2 を $1 + 1$ に設定する。ステップS1207の説明からわかるように、 2 は $1 + u_{2k}^{(0)}$ と等しいことになる。

【0118】

(ステップS1209) コレスキー分解部21は、 2 の絶対値が よりも大きいかどうか判断する。そして、大きい場合には、ステップS1210に進み、そうでない場合には、ステップS1201に戻って、 $1 / u_{2k}^{(0)}$ の値を決定する処理を再度実行する。なお、このステップS1209で行っている処理は、ステップS1203の処理と同様に、桁落ちが発生する可能性について判断する処理である。 2 の絶対値が よりも大きくない場合には、桁落ちの発生する可能性があるとは判断されることになる。

【0119】

(ステップS1210) コレスキー分解部21は、 1×2 を算出して $u_{2k}^{(0)} (1 + u_{2k-1}^{(0)})$ に設定する。前述のように、 1 は $u_{2k}^{(0)}$ と等しく、 2 は $q_k^{(0)} / (1 + u_{2k-2}^{(0)})$ であるため、 1×2 にミウラ変換を実行すると、 $u_{2k}^{(0)} (1 + u_{2k-1}^{(0)})$ に等しくなるからである。

【0120】

(ステップS1211) コレスキー分解部21は、 2 を $q_{k+1}^{(0)} / 2$ に設定する。前述のように、 2 は $1 + u_{2k}^{(0)}$ と等しいため、 2 は、 $q_{k+1}^{(0)} / (1 + u_{2k}^{(0)})$ となり、 2 における k の値を1だけインクリメントしたことになる。

【0121】

(ステップS1212) コレスキー分解部21は、 1 を $2 - 1 / u_{2k}^{(0)}$ に設定する。前述のように、 2 は $q_{k+1}^{(0)} / (1 + u_{2k}^{(0)})$ と等しいため、 1 は、 $q_{k+1}^{(0)} / (1 + u_{2k}^{(0)}) - 1 / u_{2k}^{(0)}$ となり、ミウラ変換を実行すると、 1 は、 $u_{2k+1}^{(0)}$ となる。したがって、 1 における k の値を1だけインクリメントしたことになる。

【0122】

(ステップS1213) コレスキー分解部21は、 1 の絶対値が よりも大きいかどうか判断する。そして、大きい場合には、ステップS1214に進み、そうでない場合には、ステップS1201に戻って、 $1 / u_{2k}^{(0)}$ の値を決定する処理を再度実行する。なお、このステップS1213で行っている処理は、ステップS1203の処理と同様に、桁落ちが発生する可能性について判断する処理である。 1 の絶対値が よりも大きくない場合には、桁落ちの発生する可能性があるとは判断されることになる。

【0123】

10

20

30

40

50

(ステップS1214) コレスキー分解部21は、 2×1 を算出して $u_{2k+1}^{(0)} (1 + u_{2k}^{(0)})$ に設定する。前述のように、 2 は $1 + u_{2k}^{(0)}$ と等しく、 1 は、 $u_{2k+1}^{(0)}$ に等しいからである。

【0124】

(ステップS1215) コレスキー分解部21は、カウンタ k を1だけインクリメントする。

(ステップS1216) コレスキー分解部21は、カウンタ k が m に等しいかどうか判断する。そして、 m に等しい場合には、一連の処理は終了となり、そうでない場合には、ステップS1207に戻る。

【0125】

このようにして、 $u_{2k+1}^{(0)} (1 + u_{2k}^{(0)})$ と、 $u_{2k}^{(0)} (1 + u_{2k-1}^{(0)})$ とが算出されることになる。これらの値は、コレスキー分解部21が有する図示しないメモリ等において一時的に記憶されてもよい。

【0126】

図19は、図16のフローチャートのステップS1002の処理の詳細を示すフローチャートである。

(ステップS1301) コレスキー分解部21は、 2 を $1 + u_0^{(+1)}$ に設定する。なお、前述のように $u_0^{(+1)} = 0$ であるため、 2 は1に設定されたことになる。

【0127】

(ステップS1302) コレスキー分解部21は、 1 を $u_1^{(0)} (1 + u_0^{(0)}) / (1 + u_0^{(+1)})$ に設定する。ここで、 $u_1^{(0)} (1 + u_0^{(0)})$ の値としては、ステップS1205で算出したものを用いる。なお、前述のように $u_0^{(+1)} = 0$ である。また、この 1 の式にstdLVv変換を実行すると、 1 は、 $u_1^{(+1)}$ に設定されたことになる。

【0128】

(ステップS1303) コレスキー分解部21は、カウンタ k を「1」に設定する。

(ステップS1304) コレスキー分解部21は、 1 を $1 + 1 / u_1^{(+1)}$ に設定する。

【0129】

(ステップS1305) コレスキー分解部21は、 1 の絶対値がより大きいかどうか判断する。そして、大きい場合には、ステップS1306に進み、そうでない場合には、図15のフローチャートのProcedure2(ステップS904)に進む。なお、このステップS1305で行っている処理は、ステップS1203の処理と同様に、桁落ちが発生する可能性について判断する処理である。 1 の絶対値がより大きくない場合には、桁落ちの発生する可能性があるとして判断されることになる。

【0130】

(ステップS1306) コレスキー分解部21は、 2 を $u_{2k}^{(0)} (1 + u_{2k-1}^{(0)}) / 1$ に設定する。ここで、 $u_{2k}^{(0)} (1 + u_{2k-1}^{(0)})$ の値としては、ステップS1210で算出したものを用いる。また、この 2 の式にstdLVv変換を実行すると、 2 は、 $u_{2k}^{(+1)}$ に設定されたことになる。

【0131】

(ステップS1307) コレスキー分解部21は、 1×2 を算出する。前述のように、 1 は $1 + 1 / u_1^{(+1)} = u_{2k-1}^{(+1)} + 1 / u_1^{(+1)}$ に等しく、 2 は $1 + u_{2k-2}^{(+1)}$ に等しいため、 1×2 に逆ミウラ変換を実行すると $q_k^{(+1)}$ に等しくなる。したがって、コレスキー分解部21は、 1×2 を算出して $q_k^{(+1)}$ に設定する。

【0132】

(ステップS1308) コレスキー分解部21は、 2 を $1 + 2$ に設定する。前述の

10

20

30

40

50

ように、 $u_{2k}^{(+1)}$ は $u_{2k-1}^{(+1)}$ と等しいため、 $u_{2k}^{(+1)}$ は $1 + u_{2k-1}^{(+1)}$ となる。したがって、 $u_{2k}^{(+1)}$ における k の値を 1 だけインクリメントしたことになる。

【0133】

(ステップ S1309) コレスキー分解部 21 は、 $u_{2k}^{(+1)}$ の絶対値が $u_{2k-1}^{(+1)}$ よりも大きいかどうか判断する。そして、大きい場合には、ステップ S1310 に進み、そうでない場合には、図 15 のフローチャートの Procedure 2 (ステップ S904) に進む。なお、このステップ S1309 で行っている処理は、ステップ S1203 の処理と同様に、桁落ちが発生する可能性について判断する処理である。 $u_{2k}^{(+1)}$ の絶対値が $u_{2k-1}^{(+1)}$ よりも大きくない場合には、桁落ちの発生する可能性があることが判断されることになる。

10

【0134】

(ステップ S1310) コレスキー分解部 21 は、 $u_{2k}^{(+1)}$ を算出する。前述のように、 $u_{2k-1}^{(+1)}$ は $u_{2k-2}^{(+1)}$ に等しく、 $u_{2k}^{(+1)}$ は $u_{2k-1}^{(+1)}$ と等しいため、 $u_{2k}^{(+1)}$ に逆ミウラ変換を実行すると、 $e_k^{(+1)}$ に等しくなる。したがって、コレスキー分解部 21 は、 $u_{2k}^{(+1)}$ を算出して $e_k^{(+1)}$ に設定する。

【0135】

(ステップ S1311) コレスキー分解部 21 は、 $u_{2k+1}^{(0)}$ を $(1 + u_{2k}^{(0)}) / u_{2k+1}^{(0)}$ に設定する。ここで、 $u_{2k+1}^{(0)}$ の値としては、ステップ S1214 で算出したものを用いる。また、この $u_{2k+1}^{(0)}$ の式に stdLV 変換を実行すると、 $u_{2k+1}^{(+1)}$ は、 $u_{2k+1}^{(+1)}$ に設定されたことになる。したがって、 $u_{2k+1}^{(+1)}$ における k の値を 1 だけインクリメントしたことになる。

20

【0136】

(ステップ S1312) コレスキー分解部 21 は、カウンタ k を 1 だけインクリメントする。

(ステップ S1313) コレスキー分解部 21 は、カウンタ k が m に等しいかどうか判断する。そして、 m に等しい場合には、ステップ S1314 に進み、そうでない場合には、ステップ S1304 に戻る。

【0137】

(ステップ S1314) コレスキー分解部 21 は、 $q_1^{(+1)}$ を $(1 + 1 / u_{2k}^{(+1)})$ を算出する。これは、ステップ S1304 で $u_{2k}^{(+1)}$ を更新した後に、ステップ S1307 で $q_1^{(+1)}$ を計算することと等しい。したがって、コレスキー分解部 21 は、 $q_1^{(+1)}$ を算出して $q_1^{(+1)}$ に設定する。このようにして、ミウラ変換された結果に stdLV 変換と、逆ミウラ変換とを実行して、 $q_k^{(+1)}$ 、 $e_k^{(+1)}$ を算出する処理が終了する。これらの値は、コレスキー分解部 21 が有する図示しないメモリ等において一時的に記憶されてもよい。

30

【0138】

図 20 は、図 17 のフローチャートのステップ S1102 の処理の詳細を示すフローチャートである。

(ステップ S1401) コレスキー分解部 21 は、 $u_{2m-1}^{(0)}$ を $(1 + u_{2m-2}^{(0)}) / u_{2m-1}^{(0)}$ に設定する。ここで、 $u_{2m-1}^{(0)}$ の値としては、ステップ S1214 で算出したものを用いる。なお、前述のように $u_{2m}^{(-1)} = 0$ である。また、この $u_{2m-1}^{(0)}$ の式に rdLV 変換を実行すると、 $u_{2m-1}^{(-1)}$ は、 $u_{2m-1}^{(-1)}$ に設定されたことになる。

40

【0139】

(ステップ S1402) コレスキー分解部 21 は、カウンタ k を「 $m - 1$ 」に設定する。

(ステップ S1403) コレスキー分解部 21 は、 $q_1^{(-1)}$ を $(1 + 1 / u_{2m-1}^{(-1)})$ に設定する。

【0140】

50

(ステップS1404) コレスキー分解部21は、 α_1 の絶対値が α_2 よりも大きいかどうか判断する。そして、大きい場合には、ステップS1405に進み、そうでない場合には、図15のフローチャートのミウラ変換(ステップS901)に戻る。なお、このステップS1404で行っている処理は、ステップS1203の処理と同様に、桁落ちが発生する可能性について判断する処理である。 α_1 の絶対値が α_2 よりも大きくない場合には、桁落ちの発生する可能性があるとは判断されることになる。

【0141】

(ステップS1405) コレスキー分解部21は、 α_2 を $u_{2k}^{(0)}(1 + u_{2k-1}^{(0)}) / \alpha_1$ に設定する。ここで、 $u_{2k}^{(0)}(1 + u_{2k-1}^{(0)})$ の値としては、ステップS1210で算出したものを用いる。また、この式にrdLV変換を実行すると、 α_2 は、 $u_{2k}^{(-1)}$ に設定されたことになる。

10

【0142】

(ステップS1406) コレスキー分解部21は、 α_2 を $1 + \alpha_2$ に設定する。前述のように、 α_2 は $u_{2k}^{(-1)}$ と等しいため、 α_2 は $1 + u_{2k}^{(-1)}$ となる。

【0143】

(ステップS1407) コレスキー分解部21は、 α_2 の絶対値が α_1 よりも大きいかどうか判断する。そして、大きい場合には、ステップS1408に進み、そうでない場合には、図15のフローチャートのミウラ変換(ステップS901)に戻る。なお、このステップS1407で行っている処理は、ステップS1203の処理と同様に、桁落ちが発生する可能性について判断する処理である。 α_2 の絶対値が α_1 よりも大きくない場合には、桁落ちの発生する可能性があるとは判断されることになる。

20

【0144】

(ステップS1408) コレスキー分解部21は、 $\alpha_1 \times \alpha_2$ を算出する。前述のように、 α_1 は $1 + 1 / u_{2k+1}^{(-1)} = u_{2k+1}^{(-1)} + 1 / u_{2k+1}^{(-1)}$ に等しく、 α_2 は $1 + u_{2k}^{(-1)}$ に等しいため、 $\alpha_1 \times \alpha_2$ に逆ミウラ変換を実行すると $q_{k+1}^{(-1)}$ に等しくなる。したがって、コレスキー分解部21は、 $\alpha_1 \times \alpha_2$ を算出して $q_{k+1}^{(-1)}$ に設定する。

【0145】

(ステップS1409) コレスキー分解部21は、 α_1 を $u_{2k-1}^{(0)}(1 + u_{2k-2}^{(0)}) / \alpha_2$ に設定する。ここで、 $u_{2k-1}^{(0)}(1 + u_{2k-2}^{(0)})$ の値としては、ステップS1205, S1214で算出したものを用いる。また、この式にstdLV変換を実行すると、 α_1 は、 $u_{2k-1}^{(-1)}$ に設定されたことになる。したがって、 α_1 におけるkの値を1だけデクリメントしたことになる。

30

【0146】

(ステップS1410) コレスキー分解部21は、 $\alpha_1 \times \alpha_2$ を算出する。前述のように、 α_1 は $u_{2k-1}^{(-1)}$ に等しく、 α_2 は $u_{2k}^{(-1)}$ と等しいため、 $\alpha_1 \times \alpha_2$ に逆ミウラ変換を実行すると、 $e_k^{(-1)}$ に等しくなる。したがって、コレスキー分解部21は、 $\alpha_1 \times \alpha_2$ を算出して $e_k^{(-1)}$ に設定する。

40

【0147】

(ステップS1411) コレスキー分解部21は、カウンタkを1だけデクリメントする。

(ステップS1412) コレスキー分解部21は、カウンタkが0に等しいかどうか判断する。そして、0に等しい場合には、ステップS1413に進み、そうでない場合には、ステップS1403に戻る。

【0148】

(ステップS1413) コレスキー分解部21は、 $\alpha_1 + 1 / \alpha_1$ を算出する。これは、ステップS1403で α_1 を更新した後に、ステップS1408で $\alpha_1 \times \alpha_2$ を

50

計算することと等しい。この場合、 λ_2 は 1 だからである。したがって、コレスキー分解部 2 1 は、 $\lambda_1 + 1 / (\lambda_1 - 1)$ を算出して $q_1 (\lambda_1 - 1)$ に設定する。このようにして、ミウラ変換された結果に $r d L V v$ 変換と、逆ミウラ変換とを実行して、 $q_k (\lambda_1 - 1)$ 、 $e_k (\lambda_1 - 1)$ を算出する処理が終了する。これらの値は、コレスキー分解部 2 1 が有する図示しないメモリ等において一時的に記憶されてもよい。

【0149】

なお、図 15 のフローチャートの処理によって算出することができるのは 1 個の特異値に対応する特異ベクトルであるため、全ての特異値に対応する特異ベクトルを算出する場合には、コレスキー分解部 2 1 は、図 15 のフローチャートの処理を特異値の数だけ繰り返すことになる。

【0150】

メモリ消費を抑えるために、補助変数 $u_k (\lambda_1)$ のための配列は必ずしも用意する必要はない。一方、 $\lambda_1 + 1 / (\lambda_1 - 1) u_k (\lambda_1)$ のためのメモリ領域を確保し、ミウラ変換、 $d L V v$ 型変換、逆ミウラ変換のステップにまたがってこの値を利用することで、メモリ消費を抑え、計算量を低減することができる。これにより、誤差も低減される。

【0151】

ここで、誤差について説明する。人間が理想的な状況、すなわち、無限桁の計算をいくらでもできるとすると、 $q d$ 型ツイスト分解法であっても、 $L V$ 型ツイスト分解法であっても問題ない。しかしながら、コンピュータで計算を行う場合は注意が必要である。有限桁の計算しか行えないコンピュータ上では、数学的に正しい計算法を使ったとしても必ずしも正しい結果が得られる訳ではない。そればかりか、いつまでたっても計算が終了しないといった思わぬ数値的な問題が発生する場合もある。

【0152】

コンピュータ計算による誤差には、丸め誤差及び桁落ちによる誤差などが知られている。丸め誤差単独では、高々有効桁の最後の桁が真値と比べて異なる程度で大きな誤差にはならない。また、指数部が異なる 2 つの実数の加算、乗算、除算の少なくとも 1 つの演算を行えばやはり丸め誤差が生じるが、それ以上の誤差は発生しない。さらに、このような丸め誤差を発生するような操作が繰り返されても、丸めモードが $n e a r$ (四捨五入) ならば、一方的に切り上げられたり、あるいは切り捨てられたりして誤差が極端に蓄積することは少ない。よって、多くの数値計算法は加算、乗算、除算の少なくとも 1 つの演算によって発生する丸め誤差を特別注意することは少なく、 $d L V s$ ルーチンによる特異値計算でも結果的に丸め誤差は一様に増大しない。

【0153】

問題となるのは、同符号の実数の減算及び異符号の実数の加算により生じる、桁落ちである。桁落ちによる誤差で値が 0 となった後、その値による除算を行うと、0 が分母になるような不定形となり計算不可能となる。こうなるといつまでたっても計算が終了しない。減算 除算と計算する部分が $q d$ 型ツイスト分解法、 $L V$ 型ツイスト分解法の両方に存在するので、桁落ち誤差には十分に注意する必要がある。

【0154】

$L V$ 型ツイスト分解法では、上述の桁落ちによる誤差を含んでいるかどうかは減算によって得られた値が小さいかどうかで判断できる。 $q d$ 型ツイスト分解法の場合、桁落ち誤差を含むことが分かったとしても、それを回避することはできない。なぜならば、初期値として $\{q_k (\lambda_1), e_k (\lambda_1)\}$ が与えられると、 λ_j は一意的に決定され、 $\{q_k (\lambda_{\pm 1}), e_k (\lambda_{\pm 1})\}$ も一意的に導出されるためである。すなわち、任意パラメータを持たない自由度のない計算法であるためである。

【0155】

それに対して、 $L V$ 型ツイスト分解法は、自由に設定できるパラメータ $\lambda_1 (\lambda_1)$ を持つため、補助変数 $u_k (\lambda_1)$ の値を様々に変化させることができる (図 2 1 A、図 2 1 B 参照)。すなわち、様々な経路で $\{q_k (\lambda_{\pm 1}), e_k (\lambda_{\pm 1})\}$ を計算することができる。よって、桁落ちが発生する場合も回避できる。図 1 8 ~ 図 2 0 の条件判定によって桁落

10

20

30

40

50

この影響をチェックし、減算によって得られた値の絶対値が小さな数より大きいという条件が満たされなければ、パラメータ (0) の設定に戻るといものである。この処理は、条件が満たされるまで繰り返される。なお、精度よりも高速性を重視する場合は、数回条件が満たさなければ $(q_k^{(+1)} = 0$ あるいは $q_k^{(-1)} = 0$ ならば)、例外処理を行ってもよい。

【0156】

qd型ツイスト分解法あるいはLV型ツイスト分解法によってコレスキー分解をすることができると、上述のようにツイストされた行列 N_k を算出することができ、その行列 N_k の $N_k^T x_j = e_k$ を解くことによって、 x_j を算出することができる。ここで、

【数44】

$$x_j \leftarrow \frac{x_j}{\|x_j\|}$$

と x_j を置き換えることにより、この x_j を正規化する。このようにして、右特異ベクトル x_j を求めることができる。この右特異ベクトル x_j を用いて、 $V_B = (x_1, x_2, \dots, x_m)$ とすることにより、右直交行列 V_B を算出することができる。

【0157】

また、前述のように、

【数45】

$$U_B = BV_B \Sigma^{-1}$$

となるため、 V_B が求めれば、2重対角行列 B 、特異値を対角成分に有する行列は既知であるため、左直交行列を算出することができる。より具体的には、特異値 σ_j が0でない場合には、

【数46】

$$y_j = \frac{Bx_j}{\sigma_j}$$

となる。一方、特異値 σ_j が0である場合には、

【数47】

$$B^T y_j = 0$$

を解くことにより、 y_j を算出することができる。ここで、 x_j の場合と同様に、

【数48】

$$y_j \leftarrow \frac{y_j}{\|y_j\|}$$

と y_j を置き換えることにより、この y_j を正規化する。このようにして、左特異ベクトル y_j を求めることができる。この左特異ベクトル y_j を用いて、 $U_B = (y_1, y_2, \dots, y_m)$ とすることにより、左直交行列 U_B を算出することができる。このようにして、2重対角行列 B の特異値分解がなされる。なお、この処理の後に、逆反復法や、グラムシュミット法の処理を行ってもよいことは前述の通りである。また、ここでは右直交行列 V_B を先に算出する場合について説明したが、 BB^T の固有ベクトル、すなわち左直交行列 U_B を先に算出してもよい。

【0158】

また、2重対角行列 B の特異値分解を行うことができれば、次のように、行列 A の特異値分解も行われる。

10

20

30

40

【数 4 9】

$$\begin{aligned}
 A &= \begin{cases} U_H (B \ 0) V_H^T & (l_1 \leq l_2) \\ U_H \begin{pmatrix} B \\ 0 \end{pmatrix} V_H^T & (l_1 \geq l_2) \end{cases} \\
 &= \begin{cases} U_H (U_B \Sigma V_B^T \ 0) V_H^T & (l_1 \leq l_2) \\ U_H \begin{pmatrix} U_B \Sigma V_B^T \\ 0 \end{pmatrix} V_H^T & (l_1 \geq l_2) \end{cases} \\
 &= \begin{cases} U_H U_B (\Sigma \ 0) \begin{pmatrix} V_B & 0 \\ 0 & 1 \end{pmatrix}^T V_H^T & (l_1 \leq l_2) \\ U_H \begin{pmatrix} U_B & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V_B^T V_H^T & (l_1 \geq l_2) \end{cases} \\
 &= \begin{cases} U_A (\Sigma \ 0) V_A^T & (l_1 \leq l_2) \\ U_A \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V_A^T & (l_1 \geq l_2) \end{cases}
 \end{aligned}$$

10

【0 1 5 9】

20

特異ベクトル算出部 1 9 は、上述のようにして、特異値記憶部 1 8 が記憶している上 2 重対角行列 B の特異値と、対角行列記憶部 1 3 が記憶している上 2 重対角行列 B とを用いて、上 2 重対角行列 B の特異ベクトルを算出する。まず、コレスキー分解部 2 1 は、対角行列記憶部 1 3 から上 2 重対角行列 B を読み出し、特異値記憶部 1 8 から上 2 重対角行列 B の特異値を読み出す。そして、コレスキー分解部 2 1 は、図 2 2 のフローチャートで示されるように各変換を行い、コレスキー分解の処理を行う。ここでは、L V 型ツイスト分解法を用いる場合について説明する。

【0 1 6 0】

コレスキー分解部 2 1 は、まず、上 2 重対角行列 B の各要素の値から、 $q_k^{(0)}$ 、 $e_k^{(0)}$ を求める。そして、コレスキー分解部 2 1 は、ミウラ変換を実行することにより、 $u_1^{(0)}$ 、 $u_2^{(0)}$ 等を順次求めていく (ステップ S 8 0 1)。

30

【0 1 6 1】

次に、コレスキー分解部 2 1 は、ミウラ変換で得られた $u_k^{(0)}$ を用いて、s t d L V v 変換を実行することにより、 $u_1^{(+1)}$ 等を順次求めていく (ステップ S 8 0 2) 。また、コレスキー分解部 2 1 は、ミウラ変換で得られた $u_k^{(0)}$ を用いて、r d L V v 変換を実行することにより、 $u_{2m-1}^{(-1)}$ 等を順次求めていく (ステップ S 8 0 3) 。

【0 1 6 2】

最後に、コレスキー分解部 2 1 は、s t d L V v 変換で得られた $u_k^{(+1)}$ 及び r d L V v 変換で得られた $u_k^{(-1)}$ を用いて、逆ミウラ変換を実行することにより、 $q_1^{(\pm 1)}$ 、 $e_1^{(\pm 1)}$ 等を順次求めていく (ステップ S 8 0 4) 。 $q_k^{(\pm 1)}$ 及び $e_k^{(\pm 1)}$ が求まると、すなわち、上 2 重対角行列 $B^{(+1)}$ と、下 2 重対角行列 $B^{(-1)}$ が求まると、コレスキー分解部 2 1 は、それらを第 1 特異ベクトル算出部 2 2 に渡す。なお、コレスキー分解部 2 1 は、各特異値について、それぞれコレスキー分解を行うものとする。

40

【0 1 6 3】

なお、ここでは、説明の便宜上、図 2 2 のフローチャートを用いてコレスキー分解の処理を説明したが、図 1 5 ~ 図 2 0 のフローチャートで示されるように処理を行ってもよいことは言うまでもない。

【0 1 6 4】

50

第1特異ベクトル算出部22は、特異値記憶部18から特異値を読み出し、式10を用いて k の値を決定する。第1特異ベクトル参集部22は、その k の値を用いて、 q_k ($\neq 1$) 及び e_k ($\neq 1$) からツイスト行列 N_k を求め、 $N_k^T x_j = e_k$ を解くことによって、 x_j を算出する (ステップS302)。

【0165】

次に、第1特異ベクトル算出部22は、算出した x_j を正規化して、右特異ベクトル x_j を求めて特異ベクトル記憶部20に蓄積する (ステップS303)。そして、第1特異ベクトル算出部22は、算出した右特異ベクトル x_j を第2特異ベクトル算出部23に渡す。なお、第1特異ベクトル算出部22は、各特異値について、 x_j を算出する処理と、正規化する処理とを行うことによって、全ての右特異ベクトルを算出することができる。

10

【0166】

第2特異ベクトル算出部23は、上2重対角行列 B を対角行列記憶部13から読み出し、その上2重対角行列 B の特異値を特異値記憶部18から読み出す。そして、第2特異ベクトル算出部23は、第1特異ベクトル算出部22から受け取った右特異ベクトル x_j と、上2重対角行列 B と、特異値とを用いて、 $y_j = B x_j / \lambda_j$ 、あるいは、 $B^T y_j = 0$ を解くことによって、 y_j を算出する (ステップS304)。次に、第2特異ベクトル算出部23は、右特異ベクトル x_j の場合と同様に、算出した y_j を正規化して、左特異ベクトル y_j を求めて特異ベクトル記憶部20に蓄積する (ステップS305)。このようにして、特異ベクトルの算出が終了する。なお、第2特異ベクトル算出部23は、各特異値について、 y_j を算出する処理と、正規化する処理とを行うことによって、全ての左特異ベクトルを算出することができる。

20

この後、行列記憶部11が記憶している行列 A の特異ベクトルを算出してもよい。その算出の方法は、前述の通りである。

【0167】

また、特異値算出部17が算出した特異値、特異ベクトル算出部19が算出した特異ベクトルを出力する図示しない出力部を特異値分解装置1が備えてもよい。ここで、図示しない出力部による出力は、例えば、特異値等の表示デバイス (例えば、CRTや液晶ディスプレイなど) への表示でもよく、特異値等の所定の機器への通信回線を介した送信でもよく、特異値等のプリンタによる印刷でもよく、特異値等の記録媒体への蓄積でもよい。なお、その出力部は、出力を行うデバイス (例えば、表示デバイスやプリンタなど) を含んでもよく、あるいは含まなくてもよい。また、その出力部は、ハードウェアによって実現されてもよく、あるいは、それらのデバイスを駆動するドライバ等のソフトウェアによって実現されてもよい。

30

【0168】

なお、ここでは、コレスキー分解部21がLV型ツイスト分解法によってコレスキー分解を行う場合について説明したが、コレスキー分解部21は、qd型ツイスト分解法によってコレスキー分解を行ってもよい。例えば、コレスキー分解部21は、算出された特異値の分布を見て、分布が密でない場合には、qd型ツイスト分解法を用いるようにしてもよい。

【0169】

40

また、上記説明では、行列 B が上2重対角行列である場合について説明したが、行列 B は下2重対角行列であっても、その行列 B を上2重対角行列に変換することによって特異値分解を同様に実行することができる。例えば、行列 C が下2重対角行列であるとすると、上2重対角行列 $B = C^T$ とすることができる。その上2重対角行列 B を、

$$B = U \quad V^T$$

と特異値分解できたとすると、

$$C = B^T = (U \quad V^T)^T = V \quad U^T$$

となる。したがって、下2重対角行列の特異値分解において、特異値は上2重対角行列 B の特異値と同じであり、特異ベクトルは、左右が逆になることがわかる。

【0170】

50

一方、下2重対角行列Cについても、行列分割部14等が下2重対角行列Cの分割を行い、その分割された下2重対角行列について特異値分解部15が特異値分解を行い、その特異値分解の結果を用いて特異値算出部17が下2重対角行列Cの特異値を算出するようにしてもよい。また、特異ベクトル算出部19において、下2重対角行列Cと、その特異値とを用いてコレスキー分解を行い、その特異値に対応する各特異ベクトルを算出するようにしてもよい。

【0171】

また、上記説明では、特異値算出部17が全ての特異値を算出する場合について説明したが、一部の特異値のみを算出するようにしてもよい。例えば、図13において、特異値算出部17は、行列 B_1 、 B_2 までは、全ての特異値を算出する必要があるが、行列 B_1 、 B_2 から上2重対角行列Bの特異値を算出する際に、必要な範囲で特異値を算出してよい。したがって、特異値算出部17は、2重対角行列Bの少なくとも1個の特異値を算出するものであってもよい。その場合には、不必要な特異値を算出することに伴う余分な処理を実行しなくてよいため、処理負荷を軽減することができる。このように、一部の特異値のみを算出する場合には、例えば、上2重対角行列Bの特異値を算出する処理に費やす計算時間を約 $(1 + k/m)/2$ 倍にすることができる。ここで、 m は行列サイズであり、 k は求める特異値の個数である。

【0172】

また、上記説明では、特異ベクトル算出部19が算出された特異値に対応する全ての特異ベクトルを算出する場合について説明したが、上記説明から明らかなように、特異ベクトルを算出する処理は、特異値ごとに行うことができる。したがって、特異ベクトル算出部19は、2重対角行列Bの少なくとも1個の特異ベクトルを算出するものであってもよい。このように、特異ベクトル算出部19は、必要な範囲で特異ベクトルを算出ことができ、不必要な特異ベクトルを算出することに伴う余分な処理を実行しなくてよいため、処理負荷を軽減することができる。

【0173】

次に、本実施の形態による特異値分解装置1における並列処理について説明する。

本実施の形態による特異値分解装置1では、特異値の計算及び特異ベクトルの計算において、並列的に処理を実行することができる。例えば、図23で示されるように、特異値分解装置1において、特異値分解部15、特異値算出部17、コレスキー分解部21、第1特異ベクトル算出部22、第2特異ベクトル算出部23において、それぞれ並列処理を行ってもよい。

【0174】

特異値分解部15は、複数の特異値分解手段15a、15bを備え、その複数の特異値分解手段15a、15bが、分割後の2重対角行列の特異値と、行列要素とを算出する処理を並列実行してもよい。例えば、図13において、特異値分解手段15aが行列 B_{11} 、 B_{112} 、 B_{121} 、 B_{121} の特異値分解を実行し、特異値分解手段15bが行列 B_{211} 、 B_{212} 、 B_{221} 、 B_{222} の特異値分解を実行してもよい。

【0175】

特異値算出部17は、複数の特異値算出手段17a、17bを備え、その複数の特異値算出手段17a、17bが、分割元の2重対角行列の特異値と、行列要素とを算出する処理を並列実行してもよい。例えば、図13において、特異値算出手段17aが行列 B_{11} 、 B_{12} 、 B_1 の特異値等を算出する処理を実行し、特異値算出手段17bが行列 B_{21} 、 B_{22} 、 B_2 の特異値等を算出する処理を実行してもよい。

【0176】

コレスキー分解部21は、複数のコレスキー分解手段21a、21bを備え、その複数のコレスキー分解手段21a、21bが、2重対角行列Bをコレスキー分解する処理を並列実行してもよい。例えば、コレスキー分解手段21aが半分の特異値についてコレスキー分解する処理を実行し、コレスキー分解手段21bが残りの半分の特異値についてコレスキー分解する処理を実行してもよい。

10

20

30

40

50

【 0 1 7 7 】

第 1 特異ベクトル算出部 2 2 は、複数の第 1 特異ベクトル算出手段 2 2 a、2 2 b を備え、その複数の第 1 特異ベクトル算出手段 2 2 a、2 2 b が、特異ベクトルを算出する処理を並列実行してもよい。例えば、第 1 特異ベクトル算出手段 2 2 a がコレスキー分解手段 2 1 a によってコレスキー分解された特異値に関して特異ベクトルを算出する処理を実行し、第 1 特異ベクトル算出手段 2 2 b がコレスキー分解手段 2 1 b によってコレスキー分解された特異値に関して特異ベクトルを算出する処理を実行してもよい。

【 0 1 7 8 】

第 2 特異ベクトル算出部 2 3 は、複数の第 2 特異ベクトル算出手段 2 3 a、2 3 b を備え、その複数の第 2 特異ベクトル算出手段 2 3 a、2 3 b が、特異ベクトルを算出する処理を並列実行してもよい。例えば、第 2 特異ベクトル算出手段 2 3 a が第 1 特異ベクトル算出手段 2 2 a によって算出された特異ベクトルに対応する特異ベクトルを算出する処理を実行し、第 2 特異ベクトル算出手段 2 3 b が第 1 特異ベクトル算出手段 2 2 b によって算出された特異ベクトルに対応する特異ベクトルを算出する処理を実行してもよい。

10

【 0 1 7 9 】

なお、特異値算出部 1 7 の複数の特異値算出手段 1 7 a、1 7 b は、図 1 0 で示される分割された 2 個の行列 B_1 、 B_2 の特異値と行列要素とから、分割元の行列 B_0 の特異値と行列要素とを算出する処理を、並列実行してもよい。以下、その処理について説明する。

【 0 1 8 0 】

まず、式 2 から、各特異値を求める処理については、並列実行可能なことがわかる。ここで、式 5 を次のように書き換えることにする。

20

【 数 5 0 】

$$|\hat{z}_i| = \sqrt{(\hat{\sigma}_i^2 - d_i^2) \prod_{k=1}^{i-1} \frac{(\hat{\sigma}_k^2 - d_i^2)}{(d_k^2 - d_i^2)} \prod_{k=i+1}^n \frac{(\hat{\sigma}_k^2 - d_i^2)}{(d_k^2 - d_i^2)}} \quad (\text{式 1 1})$$

【 0 1 8 1 】

この式 1 1 から、式 5 を計算するのに、一部の特異値に対応する式 1 1 の部分を計算しておき、他の特異値に対応する式 1 1 の部分を後から掛け合わせることによって、最終的に

30

【 数 5 1 】

$$|\hat{z}_i|$$

を算出できることがわかる。したがって、特異値算出手段 1 7 a、1 7 b のそれぞれは、まず、2 個の行列 B_1 、 B_2 の特異値と行列要素とを特異値分解記憶部 1 6 から読み出し、2 個の行列 B_1 、 B_2 の分割時に発生した 2 個の要素を対角行列記憶部 1 3 から読み出す。その読み出された行列要素と、2 個の要素とを用いて、 z_i の値を知ることができる。その後、各特異値算出手段 1 7 a、1 7 b は、式 2 を用いて、それぞれが担当する特異値を算出する。この処理は、並列実行することができる。

【 0 1 8 2 】

次に、特異値算出手段 1 7 a は、算出した特異値を用いて計算することができる式 1 1 の部分について計算する。また同様に、特異値算出手段 1 7 b も、算出した特異値を用いて計算することができる式 1 1 の部分について計算する。そして、特異値算出手段 1 7 a、1 7 b は、その計算した値を交換し、あらかじめ計算していた値と掛け合わせるによって、最終的に式 1 1 の値を計算することができる。このように、特異値算出手段 1 7 a、1 7 b は、式 1 1 を計算する処理も並列実行することができる。

40

【 0 1 8 3 】

次に、特異値算出手段 1 7 a、1 7 b は、式 4 を用いて、それぞれが算出した特異値に対応する特異ベクトルを計算する。このように、特異値算出手段 1 7 a、1 7 b は、右直交行列 V_M を計算する処理も並列実行することができる。

50

【 0 1 8 4 】

その後、特異値算出手段 1 7 a、あるいは特異値算出手段 1 7 b が、式 6 から式 9 を用いて行列 B_0 の行列要素を計算することによって、分割元の行列 B_0 の特異値と行列要素とを算出する処理が終了する。このように、分割された 2 個の行列 B_1 、 B_2 の特異値と行列要素とから、分割元の行列 B_0 の特異値と行列要素とを算出する処理を、特異値算出手段 1 7 a、1 7 b によって並列実行することができる。

【 0 1 8 5 】

上記のような各構成要素における並列処理において、各手段が特異値等の情報を格納するメモリを用いる場合に、複数の手段が同一のメモリ、すなわち共有メモリを使用してもよく、あるいは、各手段がそれぞれ別のメモリを使用してもよい。

10

【 0 1 8 6 】

なお、図 2 3 を用いて、特異値分解部 1 5 や特異値算出部 1 7 等が 2 個の手段によって並列処理を実行する場合について説明したが、特異値分解部 1 5 や特異値算出部 1 7 等が 3 以上の手段を備え、並列処理を実行してもよい。また、特異値分解部 1 5、特異値算出部 1 7、コレスキー分解部 2 1、第 1 特異ベクトル算出部 2 2、第 2 特異ベクトル算出部 2 3 のそれぞれにおいて並列処理が実行される場合について説明したが、いずれかの任意の 1 以上の部において、並列処理が実行されなくてもよい。例えば、特異値分解部 1 5 において並列処理が行われなくてもよい。

【 0 1 8 7 】

また、その並列処理は、1 の装置において、2 以上の CPU 等を用いて実行されてもよく、2 以上の装置において実行されてもよい。例えば、図 2 4 で示されるように、装置 A と、装置 B とがそれぞれ特異値分解手段 1 5 a、1 5 b を備え、各装置において、特異値分解の処理が並列実行されてもよい。この場合には、装置 A の特異値分解手段 1 5 a を備える特異値分解部 1 5 - 1 と、装置 B の特異値分解手段 1 5 b を備える特異値分解部 1 5 - 2 とによって特異値分解部 1 5 が構成されることになる。したがって、特異値分解装置 1 は、装置 A と、装置 B とからなるシステムを構成することになる。ここでは、特異値分解部 1 5 の並列処理について説明したが、その他の特異値算出部 1 7 やコレスキー分解部 2 1 等についても、2 以上の装置による並列処理を行ってもよい。

20

【 0 1 8 8 】

[2 次元画像から 3 次元へ復元する画像処理への応用]

30

次に、物体の 2 次元画像から 3 次元へ復元する画像処理に特異値分解を応用する場合について説明する。

【 0 1 8 9 】

複数の 2 次元画像から 3 次元復元を行う処理は、以下の各ステップによって行われる。

- (1) 2 次元画像から特徴点を抽出するステップ。
- (2) 特徴点データより形状 (元の物体の特徴点の 3 次元座標データ) 及び回転 (3 次元データから特徴点データへの変換) に関するデータを計算するステップ。
- (3) 形状及び回転のデータより可視化を行うステップ。

【 0 1 9 0 】

以下、上記 (1)、(2) の各ステップについて、図 2 5 のフローチャートを用いて説明する。ここで、2 次元画像は、例えば、スキャナやデジタルスチルカメラ、デジタルビデオカメラ等の光学的読み取り機器によって読み取られた 2 次元画像であってもよい。

40

【 0 1 9 1 】

ステップ S 5 0 0 1 において、2 次元画像 j ($j = 1, \dots, m$ 、 m は 3 以上の整数) から特徴点 i ($i = 1, \dots, n$ 、 n は 2 以上の整数) の座標 (x_i^j, y_i^j) を抽出する。取り扱う 2 次元画像は、弱中心射影画像であることが好ましい。このとき、次の式が成り立つ。

【数52】

$$\begin{pmatrix} x_i^j \\ y_i^j \end{pmatrix} = s_j \begin{pmatrix} \mathbf{r}_{1,j}^T \\ \mathbf{r}_{2,j}^T \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$

【0192】

ここで、 s_j は物体のスケールに相対する j 番目の画像のスケール、 $\mathbf{r}_{1,j}, \mathbf{r}_{2,j}$ はそれぞれ物体座標系に相対する j 番目のカメラ座標系の回転行列の1番目と2番目の行ベクトル、 $(X_i, Y_i, Z_i)^T$ は i 番目の点の3次元座標である。物体のスケールは1番目の画像のスケールと同じにし ($s_1 = 1$)、物体の座標系の姿勢は1番目の画像のカメラ座標系と同じにする ($\mathbf{r}_{1,1} = (1, 0, 0)^T, \mathbf{r}_{2,1} = (0, 1, 0)^T$)。 10

【数53】

$$D = MS$$

$$\text{ただし、} D = \begin{pmatrix} x_1^1 & \dots & x_i^1 & \dots & x_n^1 \\ \dots & \dots & \dots & \dots & \dots \\ x_1^j & \dots & x_i^j & \dots & x_n^j \\ \dots & \dots & \dots & \dots & \dots \\ x_1^m & \dots & x_i^m & \dots & x_n^m \\ y_1^1 & \dots & y_i^1 & \dots & y_n^1 \\ \dots & \dots & \dots & \dots & \dots \\ y_1^j & \dots & y_i^j & \dots & y_n^j \\ \dots & \dots & \dots & \dots & \dots \\ y_1^m & \dots & y_i^m & \dots & y_n^m \end{pmatrix}, \quad M = \begin{pmatrix} s_1 \mathbf{r}_{1,1}^T \\ \dots \\ s_j \mathbf{r}_{1,j}^T \\ \dots \\ s_m \mathbf{r}_{1,m}^T \\ s_1 \mathbf{r}_{2,1}^T \\ \dots \\ s_j \mathbf{r}_{2,j}^T \\ \dots \\ s_m \mathbf{r}_{2,m}^T \end{pmatrix},$$
20

$$S = \begin{pmatrix} X_1 & \dots & X_i & \dots & X_n \\ Y_1 & \dots & Y_i & \dots & Y_n \\ Z_1 & \dots & Z_i & \dots & Z_n \end{pmatrix}$$
30

【0193】

MとSの形から分かるように、Dのランクは3である。ここで、ステップ5001において、行列Dが与えられている。以下、回転に関するデータM及び形状Sを求める。

【0194】

そこで、行列Dの特異値分解

$$D = U \quad V^T$$

を考える。ここで、 λ は特異値を大小順に対角線上に並べたもので、U及びVはそれぞれ左直交行列、及び右直交行列である。この特異値分解として、前述の方法を用いることができる。すなわち、特異値分解装置1において、行列記憶部11が記憶している行列Aを、このたびの行列Dとすることにより、前述の説明のようにして、行列Aの特異値分解がなされることになる。なお、特異値分解装置1において、特異ベクトル記憶部20に蓄積されるのは、行列Dを変換した上2重対角行列Bの特異ベクトルであるので、その特異ベクトルを、前述の説明のようにして、行列Dの特異ベクトルに変換する必要がある。 40

【0195】

ここで、画像のデジタル誤差のため、ゼロでない特異値は3つ以上出てくる。しかし、4番目以降の特異値はノイズによるもので、最初の3つの特異値と比べて格段に小さい。そこで、ステップ5002では、最初の3つの特異値に対して特異ベクトルを計算する。採用する3個のベクトルをまとめると、次式となる。 50

$$D' = L'^{-1} R'^T = M' S'$$

ただし、 $M' = L' (\quad)^{-1/2}$ 、 $S' = (\quad)^{-1/2} R'^T$ 、 D' は $D - D'$ を最小にするランク3の行列である。

【0196】

次に、 D' から M 及び S を求めたいが、その組合せは唯一ではない。なぜなら、任意の正則行列 C が

$$D' = (M' C) (C^{-1} S')$$

を満たすからである。そこで、上式における C を $M = M' C$ を満たすように決める。 C は下記の式を満たす。

【数54】

$$MM^T = M' C C^T M'^T = \begin{pmatrix} 1 & \cdots & \cdots & 0 & \cdots & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & s_m^2 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 & \cdots & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & 0 & \cdots & \cdots & s_m^2 \end{pmatrix}$$

10

【0197】

$E = C C^T$ とすると、上式から E の6つの要素に関する $2m + 1$ 個の線形方程式が得られる。 $m = 3$ であるので、 E の要素を一意に決めることができる。ステップS5003において、行列 E を求める。

20

【0198】

次に、ステップS5004において、 E から C を求める。 C の自由度(9)は E の自由度(6)より多い。そこで、条件 $r_{1j} = (1, 0, 0)^T$ 、 $r_{2j} = (0, 1, 0)^T$ を加えれば、 C を決めることができる。このとき2つの解(Necker reversal)が出る。

次に、ステップS5005において、 $M = M' C$ 及び $S = C^{-1} S'$ より、回転に関するデータ M 及び形状 S が決まる。

【0199】

30

[文書検索への応用]

次に、文書検索処理に特異値分解を応用する場合について説明する。文書中からその文書の内容に関連する索引語を抽出し、索引語の重みを計算する処理の後、ベクトル空間モデルでは、この索引語の重みを要素とするベクトルで文書を表現する。ここで、検索対象となる文書を d_1, d_2, \dots, d_n とし、これら文書集合全体を通して全部で m 個の索引語 w_1, w_2, \dots, w_m があるとする。このとき、文書 d_j は、次のようなベクトルで表現されることになる。これを文書ベクトルと呼ぶ。

【数55】

$$d_j = \begin{pmatrix} d_{1j} \\ d_{2j} \\ \vdots \\ d_{mj} \end{pmatrix}$$

40

【0200】

ここで、 d_{ij} は索引語 w_i の文書 d_j における重みである。また、文書集合全体は、次のような $m \times n$ 行列 D によって表現することができる。

【数 5 6】

$$D = (\mathbf{d}_1 \quad \mathbf{d}_2 \quad \dots \quad \mathbf{d}_n) = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{pmatrix}$$

【0 2 0 1】

行列 D を索引語文書行列と呼ぶ。索引語文書行列の各列は文書に関する情報を表している文書ベクトルであるが、同様に、索引語文書行列の各行は索引語に関する情報を表しているベクトルであり、これを索引語ベクトルと呼ぶ。検索質問も、文書と同様に、索引語の重みを要素とするベクトルで表現することができる。検索質問文に含まれる索引語 w_i の重みを q_i とすると、検索質問ベクトル \mathbf{q} は次のように表されることになる。

10

【数 5 7】

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{pmatrix}$$

【0 2 0 2】

実際の文書検索においては、与えられた検索質問文と類似した文書を見つけ出す必要があるが、検索質問ベクトル \mathbf{q} と各文書ベクトル \mathbf{d}_j の間の類似度を計算することにより行う。ベクトル間の類似度の定義としてはさまざまなものが考えられるが、文書検索においてよく用いられているものはコサイン尺度（2つのベクトルのなす角度）または内積である。

20

【数 5 8】

$$\cos(\mathbf{d}_j, \mathbf{q}) = \frac{\mathbf{d}_j \cdot \mathbf{q}}{\|\mathbf{d}_j\| \|\mathbf{q}\|} = \frac{\sum_{i=1}^m d_{ij} q_i}{\sqrt{\sum_{i=1}^m d_{ij}^2} \sqrt{\sum_{i=1}^m q_i^2}} \quad (\text{コサイン尺度})$$

30

$$\mathbf{d}_j \cdot \mathbf{q} = \sum_{i=1}^m d_{ij} q_i \quad (\text{内積})$$

【0 2 0 3】

なお、ベクトルの長さが 1 に正規化（コサイン正規化）されている場合には、コサイン尺度と内積とは一致する。

【0 2 0 4】

図 2 6 は、本実施の形態による特異値分解装置 1 を利用した文書検索方法の一例を示すフローチャートである。

ステップ 6 0 0 1 において、質問ベクトル \mathbf{q} を受け取る。

40

【0 2 0 5】

ここでは、 D の近似行列 D_k を使った検索を考える。ベクトル空間モデルでは、検索質問ベクトル \mathbf{q} と索引語文書行列 D 中の各文書ベクトル \mathbf{d}_j の間の類似度を計算することにより検索を行うが、ここでは D の代わりに D_k を使う。ベクトル空間モデルでは、文書ベクトルの次元数は索引語の総数と等しい。したがって、検索対象となる文書が増えるに従い、文書ベクトルの次元数も増加する傾向にある。しかし、次元数が増加してくると、コンピュータのメモリによる制限や検索時間の増大などの問題が生じてくるばかりでなく、文書中に含まれる不必要な索引語がノイズ的な影響を及ぼし、検索精度を低下させてしまうという現象も起こってくる。潜在的意味インデキシング (latent semantic indexing; LSI) は、高次元の空間にある文書ベクトルを低次元の

50

空間へと射影することにより、検索精度の改善を図る技術である。高次元の空間では別々に扱われていた索引語が、低次元の空間では相互に関連を持ったものとして扱われる可能性もあるため、索引語の持つ意味や概念に基づく検索を行うことができる。たとえば、通常のベクトル空間モデルでは"car"という索引語と"automobile"という索引語はまったく別物であり、一方の索引語による質問ではもう片方の索引語を含んだ文書を検索することができない。しかし、低次元の空間ではこれらの意味的に関連した索引語は1つの次元に縮退することが期待できるため、"car"という検索質問によって"car"を含む文書ばかりでなく"automobile"を含む文書をも検索することが可能となる。潜在的意味インデキシングでは、特異値分解により高次元ベクトルの次元削減を行うが、これは基本的に多変量解析における主成分分析と等価である。

10

【0206】

ステップS6002において、kを選択する。ここでは、 $k < r$ なるkの値を選択する。 $r = \min(n, m)$ である。kの値は、予め与えられていてもよいし、計算ごとに選択可能であってもよい。

【0207】

次に、ステップS6003では、行列Dの特異値分解を行う。この特異値分解として、前述の方法を用いることができる。すなわち、特異値分解装置1において、行列記憶部11が記憶している行列Aを、このたびの行列Dとすることにより、前述の説明のようにして、行列Aの特異値分解がなされることになる。なお、特異値分解装置1において、特異ベクトル記憶部20に蓄積されるのは、行列Dを変換した上2重対角行列Bの特異ベクトルであるので、その特異ベクトルを、前述の説明のようにして、行列Dの特異ベクトルに変換する必要がある。また、この特異値分解では、計算された特異値のうち、大きい順に1番目からk番目までのk個の特異値に対してDの特異ベクトルを算出する。kは、ステップS6002で選択された値である。

20

【0208】

すなわち、

$$D_k = U_k V_k^T$$

なる U_k 及び V_k を計算する。ここで、 U_k は、最初のk個の左特異ベクトルのみから構成される $m \times k$ 行列であり、 V_k は、最初のk個の右特異ベクトルのみから構成される $n \times k$ 行列であり、 D_k は、最初のk個の特異値のみから構成される $k \times k$ 対角行列である。

30

【0209】

次に、ステップ6004において、行列 D_k と質問ベクトル q との類似度を計算する。いま、ベクトル e_j をn次元の単位ベクトルとすると、 D_k のj番目の文書ベクトルは $D_k e_j$ で表すことができる。文書ベクトル $D_k e_j$ と検索質問ベクトル q との間の類似度計算は、

【数59】

$$\begin{aligned} \cos(D_k e_j, q) &= \frac{(D_k e_j) \cdot q}{\|D_k e_j\| \|q\|} = \frac{(D_k e_j)^T q}{\|D_k e_j\| \|q\|} \\ &= \frac{(U_k \Sigma_k V_k^T e_j)^T q}{\|U_k \Sigma_k V_k^T e_j\| \|q\|} = \frac{e_j^T V_k \Sigma_k U_k^T q}{\|\Sigma_k V_k^T e_j\| \|q\|} \\ &= \frac{(\Sigma_k V_k^T e_j)^T (U_k^T q)}{\|\Sigma_k V_k^T e_j\| \|q\|} \end{aligned}$$

40

としてもよいが、別の定義を用いてもよい。上式では、 D_k を U_k 、 Σ_k 、 V_k から再構成する必要はなく特異値分解の結果から、直接、類似度を計算できることを示している。

上式の中に現われる $\Sigma_k V_k^T e_j$ は、

$$\Sigma_k V_k^T e_j = U_k^T D_k e_j$$

と書き直すことができる。この式の右辺は、近似行列 D_k におけるj番目の文書ベクトル

50

の基底 U_k のもとでの座標（文書の k 次元表現）を表している。同様に、上式の中の $U_k^T q$ は、検索質問ベクトル q の基底 U_k のもとでの座標（検索質問の k 次元表現）である。

【0210】

ステップ S6005 において、ステップ S6004 において計算された類似度を基準に、検索結果を出力する。

最後に、数値実験による性能の評価について説明する。ここでは、本実施の形態による特異値分解装置 1 の方法と、標準的な分割統治法と、シフト付き QR 法とを比較した。分割統治法としては、LAPACK で提供されている DBDSDC を用いた。QR 法としては、LAPACK で提供されている DBDSQR を用いた。実験で用いる LAPACK は ATLAS (Automatically Tuned Linear Algebra Software) で最適化した BLAS (Basic Linear Algebra Subprograms) を呼び出す。また、本実施の形態による特異値分解装置 1 の特異値分解部 15 では、QR 法を用いて特異値分解を行った。計算機は、キャッシュメモリ L1D: 64KB, L1I: 64KB, L2: 1024KB を搭載した Opteron 1.8GHz を用いた。

10

【0211】

図 27 は、計算時間を示す図である。特異値分解装置 1 による特異値分解が、他の特異値分解方法よりも非常に高速であることがわかる。特に、分割統治法 (D&C) と比較すると、ベクトル更新を省略した効果がわかる。

20

【0212】

図 28 は、異なる行列サイズの計算時間の変化を示す図である。特異値分解装置 1 による特異値分解は、ほぼ $O(n^2)$ という計算量を持つことがわかる。

なお、図 27 及び図 28 で示される計算時間の測定では、対角成分が 2.001, 非対角成分が 2.0 であり、近接特異値を持たない行列について計算を行った。

【0213】

図 29 は、計算制度を示す図である。特異値分解装置 1 による特異値の算出は、分割統治法と同程度の高い精度を有することがわかる。図 29 では、100 個のランダム行列を解き、精度を評価した。行列サイズは、1000 である。

【0214】

以上のように、本実施の形態による特異値分解装置 1 では、分割統治法を用いて特異値のみを算出するため、標準的な分割統治法で実行されるベクトル更新が必要なくなり、標準的な分割統治法よりも非常に高速である。また、特異値から特異ベクトルを算出する処理も、高速に処理することができる。さらに、QR 法では特異値を算出する段階の並列化が困難であるが、本実施の形態による特異値分解装置 1 では、特異値を算出する段階、及び特異値から特異ベクトルを算出する段階のそれぞれにおいて、本質的に高い並列性を持つ。また、本実施の形態による特異値分解装置 1 の精度は、分割統治法や QR 法とほぼ同等の精度が得られることがわかる。

30

【0215】

なお、本実施の形態による特異値分解装置 1 は、スタンドアロンの装置であってもよく、サーバクライアントシステムにおけるサーバ装置であってもよく、前述のように、複数の装置から構成されるシステムであってもよい。特異値分解装置 1 がサーバクライアントシステムにおけるサーバ装置である場合には、行列記憶部 11 で記憶される行列 A や、特異値記憶部 18 で記憶される特異値や特異ベクトル記憶部 20 で記憶される特異ベクトル等は、インターネットやイントラネット等の通信回線を介して送受信されてもよい。

40

【0216】

また、特異値分解装置 1 における一部の処理を、特異値分解装置 1 以外において行ってもよい。例えば、行列分解部 14 による 2 重対角行列 B を分割する処理、あるいは、特異値分解部 15 による特異値分解の処理を、特異値分解装置 1 において実行しなくてもよい。その場合には、例えば、2 重対角行列 B が分割された結果である、あらかじめ決められ

50

た大きさ以下の各2重対角行列が、入力デバイスや通信回線、記録媒体等を介して受け付けられ、対角行列記憶部13で記憶されていてもよい。また、例えば、そのあらかじめ決められた大きさ以下の各2重対角行列に対して特異値分解が行われた結果である、各2重対角行列の特異値と、各2重対角行列の特異ベクトルからなる左右直交行列の一部の要素である行列要素とが、入力デバイスや通信回線、記録媒体等を介して受け付けられ、特異値分解記憶部15で記憶されていてもよい。

【0217】

また、上記実施の形態において、各処理または各機能は、単一の装置または単一のシステムによって集中処理されることによって実現されてもよく、あるいは、複数の装置または複数のシステムによって分散処理されることによって実現されてもよい。

10

【0218】

また、上記実施の形態において、各構成要素は専用のハードウェアにより構成されてもよく、あるいは、ソフトウェアにより実現可能な構成要素については、プログラムを実行することによって実現されてもよい。例えば、ハードディスクや半導体メモリ等の記録媒体に記録されたソフトウェア・プログラムをCPU等のプログラム実行部が読み出して実行することによって、各構成要素が実現され得る。なお、上記実施の形態における特異値分解装置を実現するソフトウェアは、以下のようなプログラムである。つまり、このプログラムは、コンピュータに、2重対角行列Bが2個の2重対角行列に分割され、その2重対角行列が2個の2重対角行列に分割される処理が、分割後の各2重対角行列があらかじめ決められた大きさ以下となるまで繰り返され、当該あらかじめ決められた大きさ以下の各2重対角行列に対して特異値分解が行われた結果である、前記各2重対角行列の特異値と、前記各2重対角行列の特異ベクトルからなる左右直交行列の一部の要素である行列要素とが記憶される特異値分解記憶部から、各2重対角行列の特異値と、行列要素とを読み出し、前記特異値と、前記行列要素とから、分割元の2重対角行列の特異値と、分割元の2重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の2重対角行列の特異値と、行列要素とを算出する処理を、2重対角行列Bの少なくとも1個の特異値を算出するまで繰り返し、前記2重対角行列Bの少なくとも1個の特異値を特異値記憶部に蓄積する特異値算出ステップと、前記2重対角行列Bが記憶される対角行列記憶部から2重対角行列Bを読み出し、前記特異値記憶部から前記2重対角行列Bの特異値を読み出し、2重対角行列Bとその特異値とから、ツイスト分解法を用いて2重対角行列Bの少なくとも1個の特異ベクトルを算出して特異ベクトル記憶部に蓄積する特異ベクトル算出ステップと、を実現させるためのものである。

20

30

【0219】

また、このプログラムでは、コンピュータに、前記あらかじめ決められた大きさ以下の各2重対角行列も記憶される前記対角行列記憶部から、前記あらかじめ決められた大きさ以下の各2重対角行列を読み出し、前記各2重対角行列に対して特異値分解を行い、前記各2重対角行列の特異値と、前記各2重対角行列の特異ベクトルとを算出し、当該特異値と、当該特異ベクトルからなる左右直交行列の一部の要素である行列要素とを前記特異値分解記憶部に蓄積する特異値分解ステップをさらに実行させてもよい。

【0220】

また、上記実施の形態における特異値分解装置を実現するソフトウェアは、以下のようなプログラムである。つまり、このプログラムは、コンピュータに、2重対角行列Bが記憶される対角行列記憶部から前記2重対角行列Bを読み出し、当該2重対角行列Bを2個の2重対角行列に分割して前記対角行列記憶部に蓄積し、その2重対角行列を2個の2重対角行列に分割して前記対角行列記憶部に蓄積する処理を、分割後の各2重対角行列があらかじめ決められた大きさ以下となるまで繰り返す行列分割ステップと、前記あらかじめ決められた大きさ以下の各2重対角行列を前記対角行列記憶部から読み出し、前記各2重対角行列に対して特異値分解を行い、前記各2重対角行列の特異値と、前記各2重対角行列の特異ベクトルとを算出し、特異値分解された特異値と、特異ベクトルからなる左右直交行列の一部の要素である行列要素とを特異値分解記憶部に蓄積する特異値分解ステップ

40

50

と、各 2 重対角行列の特異値と、行列要素とを前記特異値分解記憶部から読み出し、前記特異値と、前記行列要素とから、分割元の 2 重対角行列の特異値と、分割元の 2 重対角行列の行列要素とを算出して前記特異値分解記憶部に蓄積し、その分割元の 2 重対角行列の特異値と、行列要素とを算出する処理を、2 重対角行列 B の少なくとも 1 個の特異値を算出するまで繰り返し、前記 2 重対角行列 B の少なくとも 1 個の特異値を特異値記憶部に蓄積する特異値算出ステップと、前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、2 重対角行列 B とその特異値とから、ツイスト分解法を用いて 2 重対角行列 B の少なくとも 1 個の特異ベクトルを算出して特異ベクトル記憶部に蓄積する特異ベクトル算出ステップと、を実行させるためのものである。

10

【0221】

また、このプログラムでは、前記特異ベクトル算出ステップは、前記対角行列記憶部から 2 重対角行列 B を読み出し、前記特異値記憶部から前記 2 重対角行列 B の特異値を読み出し、前記 2 重対角行列 B の各要素に関してミウラ変換、d L V v 型変換、逆ミウラ変換を行うことによって、前記 2 重対角行列 B を上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$ にコレスキー分解するコレスキー分解ステップと、前記上 2 重対角行列 $B^{(+1)}$ 及び下 2 重対角行列 $B^{(-1)}$ の各要素と、前記 2 重対角行列 B の特異値とを用いて一方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第 1 特異ベクトル算出ステップと、前記第 1 特異ベクトル算出ステップで算出した一方の左右直交行列を構成する特異ベクトルと、前記 2 重対角行列 B の特異値と、前記 2 重対角行列 B とを用いて他方の左右直交行列を構成する特異ベクトルを算出して前記特異ベクトル記憶部に蓄積する第 2 特異ベクトル算出ステップと、をさらに備えてもよい。

20

なお、上記プログラムにおいて、情報を蓄積する処理などでは、ハードウェアでしか行われない処理は少なくとも含まれない。

【0222】

また、このプログラムは、サーバなどからダウンロードされることによって実行されてもよく、所定の記録媒体（例えば、CD-ROM などの光ディスクや磁気ディスク、半導体メモリなど）に記録されたプログラムが読み出されることによって実行されてもよい。

【0223】

また、このプログラムを実行するコンピュータは、単数であってもよく、複数であってもよい。すなわち、集中処理を行ってもよく、あるいは分散処理を行ってもよい。

30

【0224】

図 30 は、上記プログラムを実行して、上記実施の形態による特異値分解装置 1 を実現するコンピュータの外観の一例を示す模式図である。上記実施の形態は、コンピュータハードウェア及びその上で実行されるコンピュータプログラムによって実現される。

【0225】

図 30 において、コンピュータシステム 100 は、CD-ROM (Compact Disk Read Only Memory) ドライブ 105、FD (Flexible Disk) ドライブ 106 を含むコンピュータ 101 と、キーボード 102 と、マウス 103 と、モニタ 104 とを備える。

40

【0226】

図 31 は、コンピュータシステムを示す図である。図 31 において、コンピュータ 101 は、CD-ROM ドライブ 105、FD ドライブ 106 に加えて、CPU (Central Processing Unit) 111 と、ブートアッププログラム等のプログラムを記憶するための ROM (Read Only Memory) 112 と、CPU 111 に接続され、アプリケーションプログラムの命令を一時的に記憶すると共に、一時記憶空間を提供する RAM (Random Access Memory) 113 と、アプリケーションプログラム、システムプログラム、及びデータを記憶するハードディスク 114 と、CPU 111、ROM 112 等を相互に接続するバス 115 とを備える。なお、コンピュータ 101 は、LAN への接続を提供する図示しないネットワークカードを含ん

50

でもよい。

【0227】

コンピュータシステム100に、上記実施の形態による特異値分解装置1の機能を実行させるプログラムは、CD-ROM121、またはFD122に記憶されて、CD-ROMドライブ105、またはFDドライブ106に挿入され、ハードディスク114に転送されてもよい。これに代えて、そのプログラムは、図示しないネットワークを介してコンピュータ101に送信され、ハードディスク114に記憶されてもよい。プログラムは実行の際にRAM113にロードされる。なお、プログラムは、CD-ROM121やFD122、またはネットワークから直接、ロードされてもよい。

【0228】

また、行列記憶部11、対角行列記憶部13、特異値分解記憶部16、特異値記憶部18、特異ベクトル記憶部20は、RAM113やハードディスク114によって実現されてもよい。

【0229】

プログラムは、コンピュータ101に、上記実施の形態による特異値分解装置1の機能を実行させるオペレーティングシステム(OS)、またはサードパーティプログラム等を必ずしも含まなくてもよい。プログラムは、制御された態様で適切な機能(モジュール)を呼び出し、所望の結果が得られるようにする命令の部分のみを含んでいてもよい。コンピュータシステム100がどのように動作するのかについては周知であり、詳細な説明を省略する。

【0230】

また、本発明は、以上の実施の形態に限定されることなく、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

また、本発明のほんのいくつかの典型的な実施例について上で詳細に説明したが、その典型的な実施例において、発明の利益と新規な技術から実質的にはずれることなく多くの変更が可能であることを当業者は容易に認識することができるであろう。したがって、そのようなすべての変更は、本発明の範囲に含まれるものである。

【産業上の利用可能性】

【0231】

以上のように、本発明による特異値分解装置等によれば、特異値分解を高速に処理することができ、画像処理や検索処理、その他の特異値分解を用いる処理を実行する装置等において有用である。

【図面の簡単な説明】

【0232】

【図1】本発明の実施の形態1による特異値分解装置の構成を示すブロック図

【図2】同実施の形態による特異値分解装置の動作を示すフローチャート

【図3】同実施の形態による特異値分解装置の動作を示すフローチャート

【図4】同実施の形態による特異値分解装置の動作を示すフローチャート

【図5】同実施の形態による特異値分解装置の動作を示すフローチャート

【図6】同実施の形態における上2重対角行列Bの分割について説明するための図

【図7】同実施の形態における上2重対角行列Bの分割について説明するための図

【図8】同実施の形態による特異値分解装置の動作を示すフローチャート

【図9】同実施の形態における上2重対角行列Bの分割について説明するための図

【図10】同実施の形態における上2重対角行列Bの特異値の算出について説明するための図

【図11】同実施の形態による特異値分解装置の動作を示すフローチャート

【図12】同実施の形態による特異値分解装置の動作を示すフローチャート

【図13】同実施の形態における上2重対角行列Bの特異値の算出について説明するための図

【図14】同実施の形態におけるコレスキー分解について説明するための図

10

20

30

40

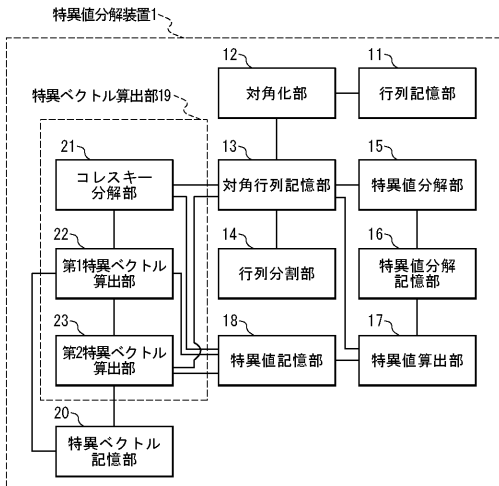
50

- 【図15】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図16】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図17】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図18】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図19】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図20】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図21A】同実施の形態におけるq d型ツイスト分解法について説明するための図
- 【図21B】同実施の形態におけるL V型ツイスト分解法について説明するための図
- 【図22】同実施の形態による特異値分解装置の動作を示すフローチャート
- 【図23】同実施の形態による特異値分解装置の構成の他の一例を示すブロック図
- 【図24】同実施の形態による特異値分解装置の構成の他の一例を示すブロック図
- 【図25】同実施の形態における画像処理の一例を示すフローチャート
- 【図26】同実施の形態における文書検索処理の一例を示すフローチャート
- 【図27】同実施の形態による特異値分解と従来の特異値分解との比較について説明するための図
- 【図28】同実施の形態による特異値分解と従来の特異値分解との比較について説明するための図
- 【図29】同実施の形態による特異値分解と従来の特異値分解との比較について説明するための図
- 【図30】同実施の形態におけるコンピュータシステムの外観一例を示す模式図
- 【図31】同実施の形態におけるコンピュータシステムの構成の一例を示す図

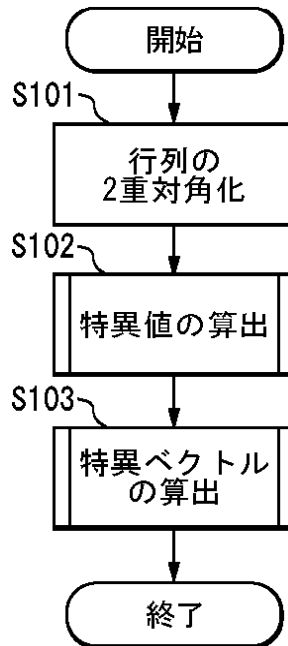
10

20

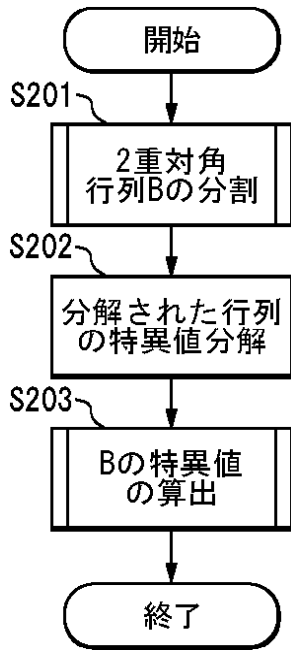
【図1】



【図2】



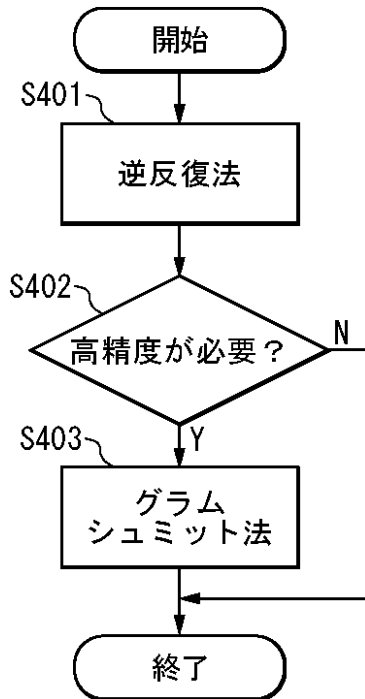
【図3】



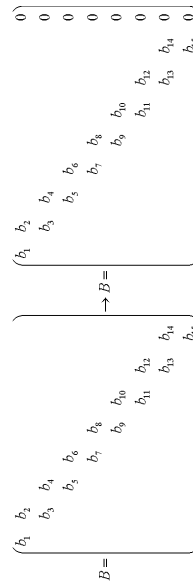
【図4】



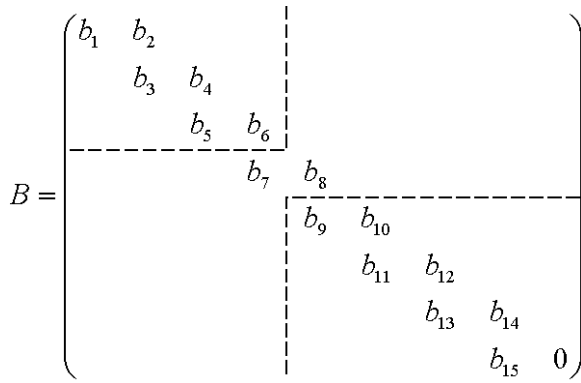
【図5】



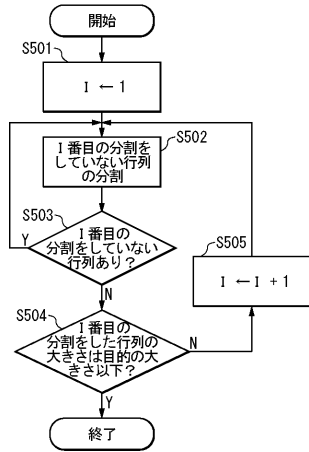
【図6】



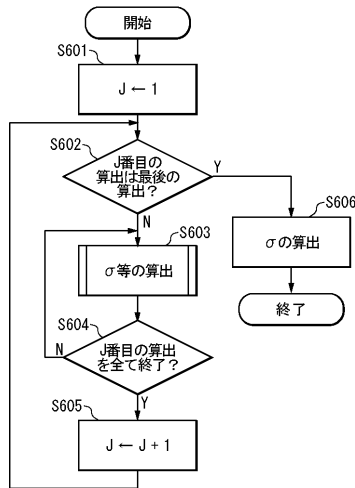
【図7】



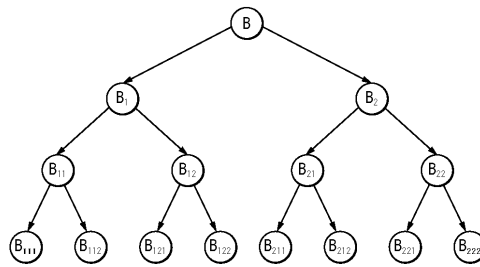
【図8】



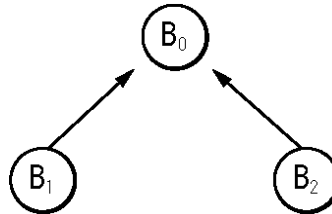
【図11】



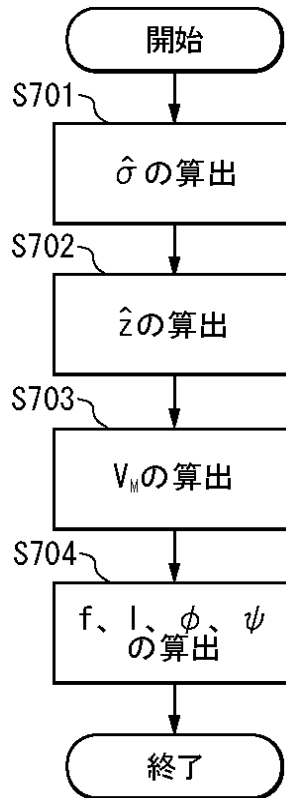
【図9】



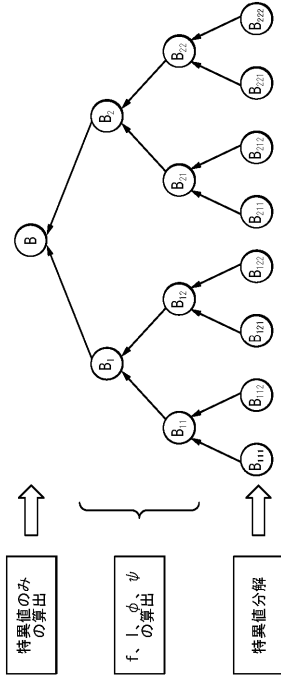
【図10】



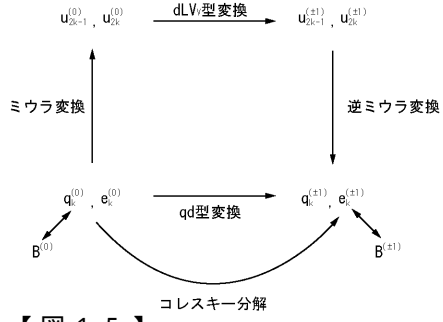
【図12】



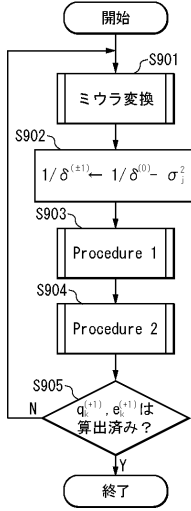
【図13】



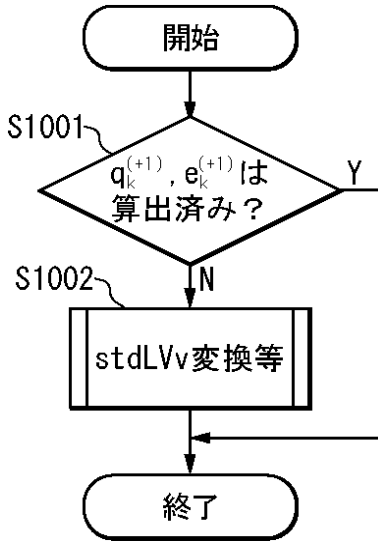
【図14】



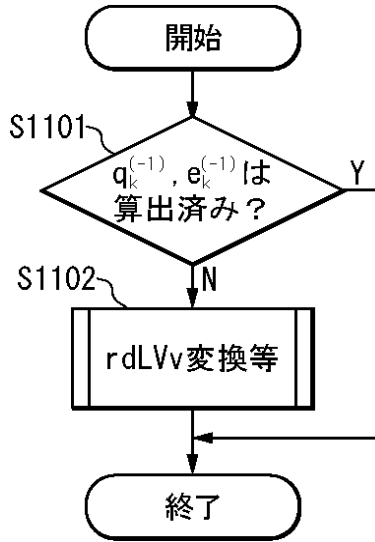
【図15】



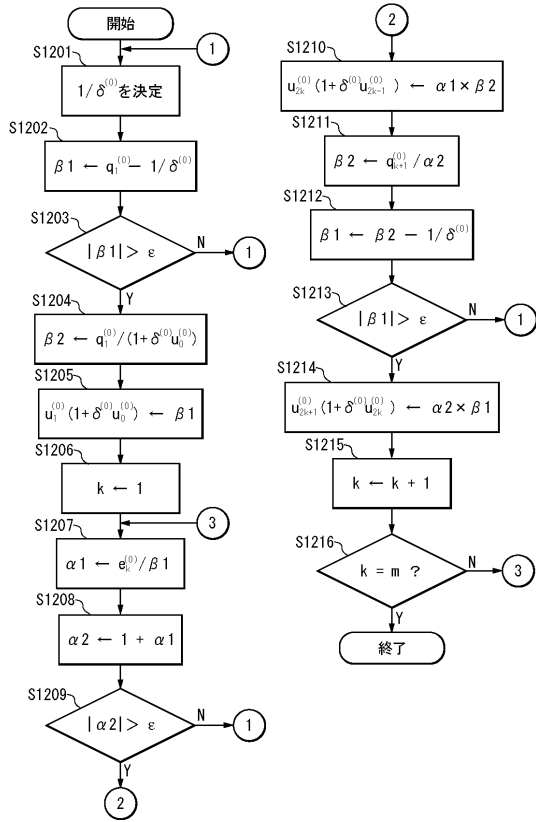
【図16】



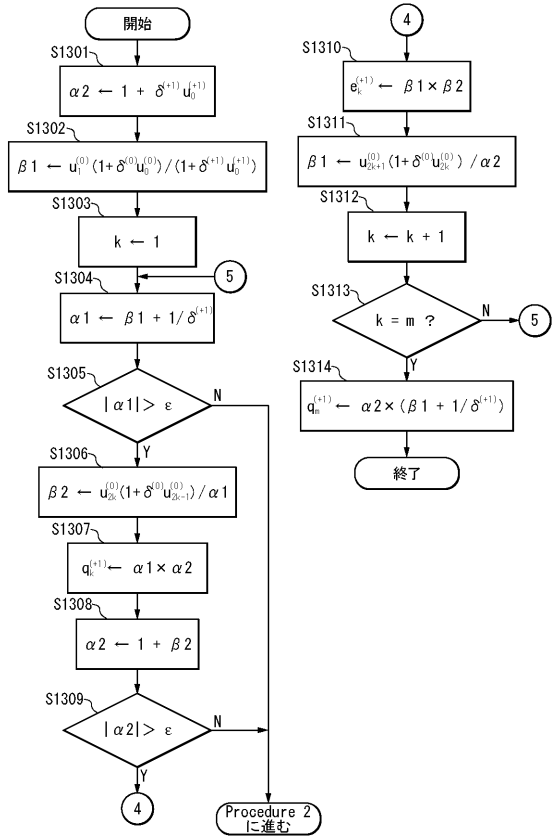
【図17】



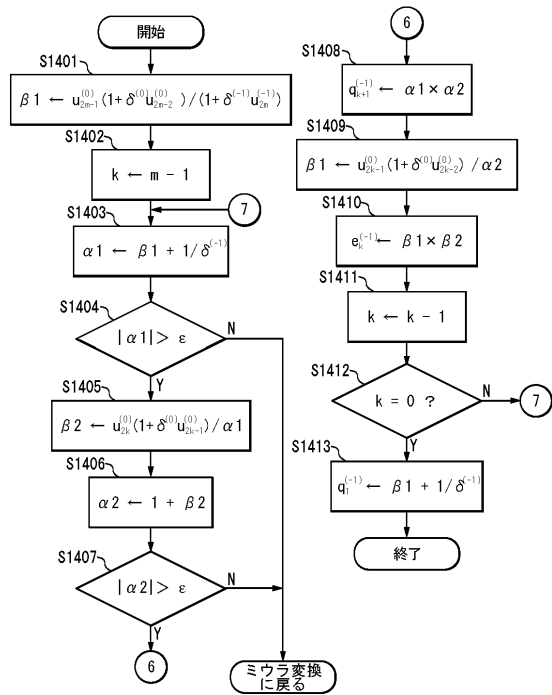
【図18】



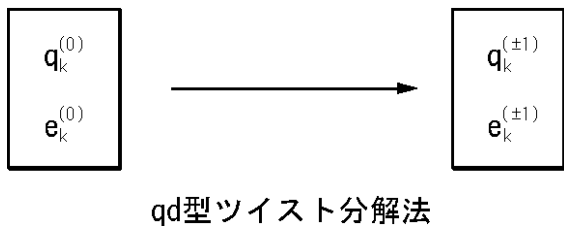
【図19】



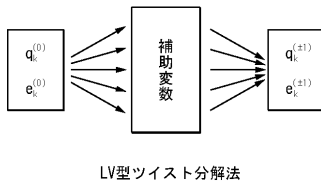
【図20】



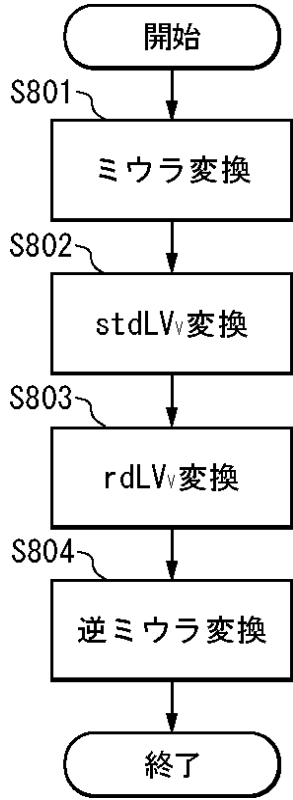
【図21A】



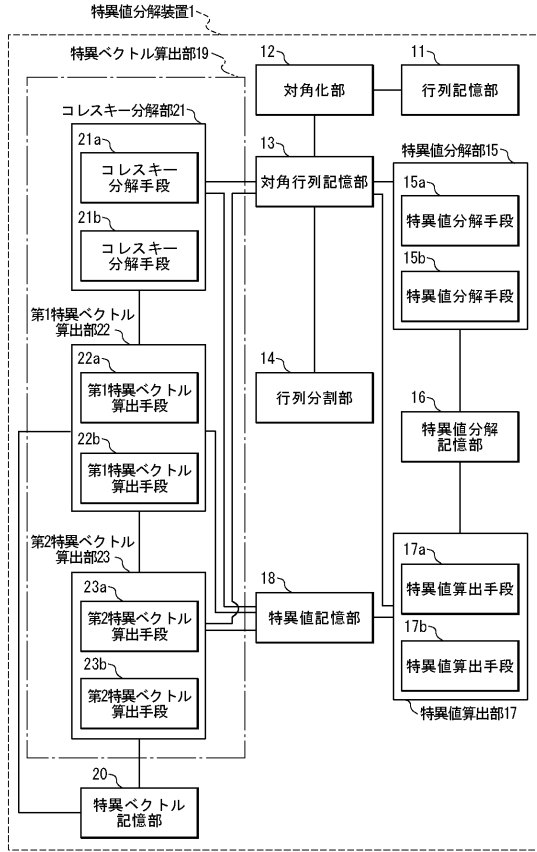
【図21B】



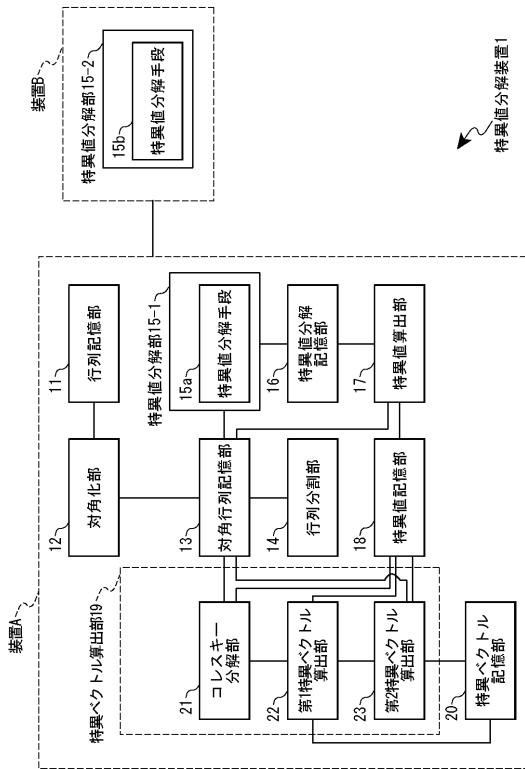
【図22】



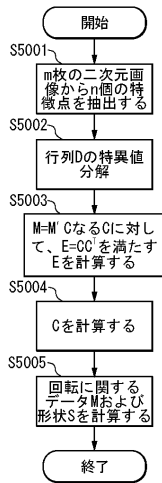
【図23】



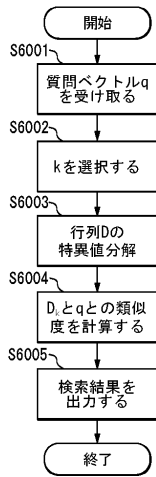
【図24】



【図25】



【図 26】



【図 27】

| | 本発明 | D & C | QR |
|----------------|------|--------|---------|
| 特異値の計算時間(秒) | 1.51 | — | — |
| 特異ベクトルの計算時間(秒) | 4.30 | — | — |
| 全体の計算時間(秒) | 5.81 | 170.36 | 1814.79 |

【図 28】

計算時間(秒)

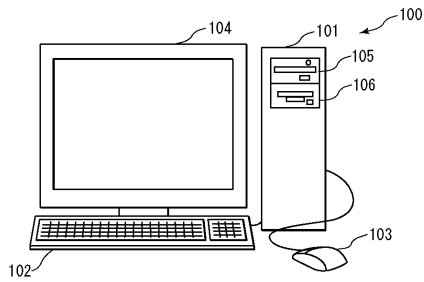
| 行列サイズ | 本発明 | D & C | QR |
|-------|-------|--------|---------|
| 3000 | 5.81 | 170.36 | 1814.79 |
| 5000 | 18.17 | 809.93 | — |
| 7000 | 37.86 | — | — |

【図 29】

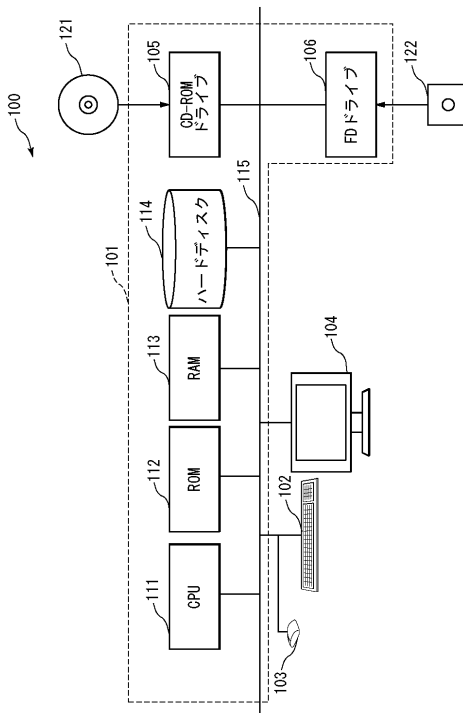
計算精度

| | 本発明 | D & C | QR |
|--------------------|-----------|-----------|-----------|
| Σ | $2.1E-13$ | $2.1E-13$ | $1.3E-12$ |
| $V^T V - I$ | $3.7E-10$ | $5.1E-11$ | $1.3E-10$ |
| $U^T U - I$ | $3.6E-10$ | $5.1E-11$ | $1.3E-10$ |
| $B - U \Sigma V^T$ | $4.2E-09$ | $3.0E-11$ | $1.1E-10$ |

【図 30】



【図 31】



フロントページの続き

(72)発明者 阪野 真也

京都府京都市左京区吉田本町 国立大学法人京都大学大学院情報学研究科内

(72)発明者 高田 雅美

京都府京都市左京区吉田本町 国立大学法人京都大学大学院情報学研究科内

審査官 田中 幸雄

(56)参考文献 国際公開第2005/119507(WO, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 17/16

G06T 1/00