実対称固有値問題に対する多分割の分割統治法の並列化

坪 谷 怜
埼玉大学大学院理工学研究科

1. はじめに

固有値問題は，量子力学，量子化学，構造解析，経済など様々な分野で現れる問題で，大規模な問題を扱う分野も多く，この問題の数値解法には高速性が求められている。本稿では特に応用の広い実対称固有値問題を対象とする。実対称固有値問題に対する数値解法として，相似変換を行い実対称三重対角行列の固有値問題に帰着して解く方法が一般に用いられている。実対称三重対角行列の固有値問題に対する最新の数値解法として，二分割の分割統治法 (以降 DC-2 と呼ぶ)[1] や MRRR 法 (以降 MR^3 と呼ぶ)[2] が広く知られている。また近年，桑島らにより DC-2 は，多分割の分割統治法 (以降 DC-k と呼ぶ)[3, 4] に拡張された。DC-k の特徴は分割数 k を 3 以上に大きく出来ることで，分割数の増加は DC-2 の主要演算である行列行列積の演算量削減を可能とし，逐次計算において DC-k が高速に計算出来ることが示されている。また，DC-k は DC-2 同様の高い並列性が期待され，並列計算における研究が必要とされている。

本研究の目的は DC-k を共有メモリ型並列計算機に並列実装し高速化すること，また，並列計算機上におけるアルゴリズムの有用性を確認することである。研究内容の骨子に関しては既出のスーパーコンピューティングニュース [5] に記載した。本稿では，その後のプログラムの改善点やこれに付随して改善した計測結果について報告する。

以降の流れとして，次の節では DC-k のアルゴリズムを簡単に紹介し，第 3 節ではプログラムの改善点について説明する。第 4 節では実験の内容を説明し，第 5 節では実験の結果を説明する。最後に第 6 節でまとめを述べる。

2. DC-k のアルゴリズム

DC-k のアルゴリズムは n 次実対称三重対角行列 T と分割数 k を入力として，$T = Q\Lambda Q^\top$ となる直交行列 Q と実対角行列 Λ を出力とする。この時，Q の第 j 列ベクトル $\boldsymbol{q}_j$ と Λ の j 行 j 列要素 λ_j は T の固有対である。全体の工程は大きく分けて次の 3 つのステップからなる。

Step 1. T を実対角行列と低階数の摂動の和 $(D + UU^\top)$ に同値変形する ($T = Q_p(D+UU^\top)Q_p^\top$)。この時，$D$ は実対角行列，U は $n \times (k-1)$ の摂動行列，Q_p は k 個の小さな直交行列ブロックが対角に並ぶ直交行列である。

Step 2. $D + UU^\top$ の固有値問題を解く ($D + UU^\top = Q'\Lambda Q'^\top$)。

1. 全固有値 $\lambda_1, \cdots, \lambda_n$ を計算する。$\Lambda := \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ とする。
2. 全固有ベクトル $Q' := (\boldsymbol{q}'_1, \cdots, \boldsymbol{q}'_n)$ を計算する。
3. 直交性の悪い固有ベクトルに対して再直交化する。

Step 3. $Q := Q_p \times Q'$ を計算する。

このアルゴリズムでは分割数 k の増加に伴い，Q_p の対角ブロック数が増え零要素が増える。この構造を利用すると Step 3. の演算量を削減することが出来，$O(n^3)$ である Step 3. の高速化に大きく貢献する。逆に分割数 k が増加すると，U の列数が増える。これによって，Step 2. の演算量が増加してしまい，およそ $O(n^2)$ の Step 2. の速度を徐々に劣化させる。このため Step 2., 3. はトレードオフの関係にあり，最も高速に計算出来る分割数は問題とする行列の性質やサイズによって変化する。本稿では，ある行列に対して最も高速に計算出来ると思われる分割数をその行列の最適分割数と呼ぶこととする。

Step 2. では U の第 j 行ベクトルが零である時，D の j 行 j 列要素 d_j を固有値とし，単位行列の第 j 列ベクトル $\boldsymbol{e}_j$ を固有ベクトルとする自明解が存在する。固有値問題から自明解を取り除くことが出来れば問題の規模を縮小することが出来，その後の演算量を削減することが出来る。このため，Step 2. の初めに自明解を取り除く操作が行われ，この操作を Deflation と呼ぶ。Deflation の効果は行列の性質によって異なり，自明解が多く Deflation の効果が大きい行列もあれば，自明解が少なく効果が小さい行列も存在する。

3. プログラムの改善点

アルゴリズム Step 2. の固有ベクトルの計算に関わる項目 2. でプログラムの改善を行った。実対角行列と低階数の摂動の和の固有ベクトル計算には，対応する固有値 λ_j に依存する行列 $\tilde{F}$ の固有対 $(0, \boldsymbol{x})$ を利用して固有ベクトルを計算することが出来る [4]。行列 $\tilde{F}$ は，

$$\tilde{F} = \begin{pmatrix} A & B^\top \\ B & C \end{pmatrix}, \qquad \begin{pmatrix} A : s \text{ 次実対称行列，} B : t \times s \text{ 行列} \\ C : t \text{ 次実対角行列，} s = k-1,\ 0 \le t \le n,\ k : \text{分割数} \end{pmatrix}$$

の形の構造をしており，$\boldsymbol{x}$ の計算には逆反復法を用いる。

文献 [5] の逆反復法内部では，$m(= s + t)$ 次実対称行列 $\tilde{F}$ の LU 分解を計算し，1 つの固有ベクトルにつき $O(m^3)$ の演算量を必要とする実装であった。一般に t の大きさは $\boldsymbol{x}$ の精度確保の難易度と関係しており，平均的には $t \simeq k$ であるが，$\boldsymbol{x}$ の精度確保が難しい固有値 λ_j の場合には t を k に比べ十分大きくとる必要がある。例えば，$n = 20000$ の問題に対して $k = 30$ の時 $m = 700$ 程度となることもある。C が実対角行列であることを活用し，零要素をうまく利用することで $\tilde{F}$ の三角行列への分解の演算量を $O(km^2)$ 程度に削減することを行った。

4. 実験概要

数値実験では東京大学情報基盤センター HITACHI SR 11000 を用いて実対称三重対角行列の固有値問題を解くために必要な実行時間を計測した。並列計算は，1 ノード 16 スレッドを用いた共有メモリ型並列計算を行う。また，出力として得られた直交行列 Q と実対角行列 Λ の精

度を評価するために,

$$\text{相対残差 } \epsilon_r = \max_{1\leq i\leq n} \frac{||T\boldsymbol{q}_i - \lambda_i \boldsymbol{q}_i||_2}{||T||_2}, \qquad \text{直交誤差 } \epsilon_o = \max_{1\leq i\leq j\leq n} \left|\boldsymbol{q}_i^T \boldsymbol{q}_j - \delta_{ij}\right|$$

を用いる。ただし，T は入力とする n 次実対称三重対角行列，λ_j は T の第 j 固有値，$\boldsymbol{q}_j$ は T の第 j 固有ベクトル，δ_{ij} はクロネッカのデルタである。

入力として試した行列は以下の 3 種類である。

- 行列 DS：j 番目の主対角要素に $j \times 10^{-6}$，副対角要素に 1 を持つ実対称三重対角行列。Deflation の効果が小さい。
- 行列 DL：主対角要素に $(2, 4]$ の，副対角要素に $(1, 2]$ の一様乱数を持つ実対称三重対角行列。Deflation の効果が大きい。
- 行列 QC：正規乱数を要素に持つ実対称行列を，相似変換した実対称三重対角行列 [6]。量子カオス系のモデルである [7]。

比較対象として，逐次，並列共に SR 11000 に備え付けの LAPACK [8] モジュールから DC-2 および MR^3 のソルバを利用する。MR^3 はアルゴリズム上固有ベクトル計算に並列性がある。しかし，MR^3 の並列ソルバは並列化による高速化が乏しかったため，ソルバが十分に並列化され，固有ベクトル計算部分の並列実行時間が無視出来るほどに短縮したと仮定し，MR^3 による全固有値計算のみの並列実行時間を全固有値固有ベクトル計算の並列実行時間として扱う。よって以降で紹介する MR^3 による並列実行時間は全固有値のみの並列実行時間である。

5. 実験結果

表 1 は分割数別に DC-k の並列実行時間を計測した結果であり，20000 次元の行列 DS に対して，2 分割から 50 分割まで計測を行った。図 1 は表 1 のデータに各ステップでの実行時間のデータを加えてグラフ化したものである。第 2 節で述べた通り，分割数の増加に伴って Step 3. の実行時間は短縮し，Step 2. の実行時間は増加している。行列 DS では，全体時間の最小値周辺での変化は緩やかで最小値はおよそ 30 の時である。このため，20000 次元の行列 DS の最適分割数を 30 とする。また，同様の実験から，行列 DL の最適分割数を 2，行列 QC の最適分割数を 30 とした。以降ではこの最適分割数を選んで計算を行うものとする。

表 2 は DC-k の逐次実行時間と並列実行時間を計測した結果である。行列 DS に対して，2500 次元から 20000 次元まで計測を行った。“効率” の欄は並列化効率を表している。並列化効率とは，逐次実行時間を並列実行時間とスレッド数の積で割った値を百分率で表したものである。20000 次元の問題では 16 スレッドを用いて 8 倍程度高速化することが出来，並列化効率は 50 % 程度であった。表 3 はアルゴリズムの各ステップでの実行時間を計測した結果である。Step 1. はサイズの小さい行列を複数個扱う処理で構成され，個々の処理に対して並列化を行っている現状では高い並列化は望めない。Step 2., 3. は問題サイズが十分大きく，うまい並列化を行えば高い並列化効率が得られるはずである。仮に Step 2., 3. の並列実行時間を逐次実行時間

表 1: 分割数に対する実行時間

分割数	時間 [sec]	分割数	時間 [sec]
2	137.16	28	36.01
4	74.51	30	35.33
6	59.19	32	36.61
8	48.68	34	40.58
10	47.07	36	41.00
12	42.36	38	41.22
14	38.59	40	48.26
16	37.80	42	52.56
18	42.84	44	50.69
20	38.58	46	58.11
22	37.04	48	61.61
24	36.31	50	72.74
26	36.63		

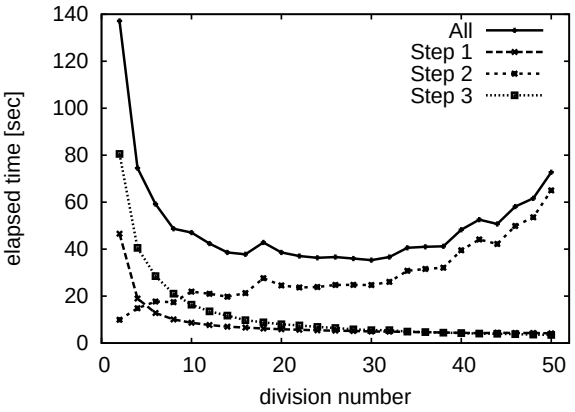

図 1: 分割数に対する実行時間

の 16 分の 1 に短縮することが出来れば，全体の並列実行時間は 20000 次元の問題の時 22.88 秒に短縮出来，並列化効率を 81.2% まで向上出来る。現在の実装では，全体の並列化効率は 50% 程度であり，プログラムにはまだ高速化の余地が残されていると考えられる。

表 2: 並列化に伴う高速化

サイズ	2500	5000	7500	10000
逐次 [sec]	2.17	9.66	24.17	49.71
並列 [sec]	1.19	3.30	5.93	10.39
効率 [%]	11.3	18.3	25.5	29.9
サイズ	12500	15000	17500	20000
逐次 [sec]	88.37	145.12	211.26	297.15
並列 [sec]	14.00	19.85	26.16	34.83
効率 [%]	39.4	45.7	50.5	53.3

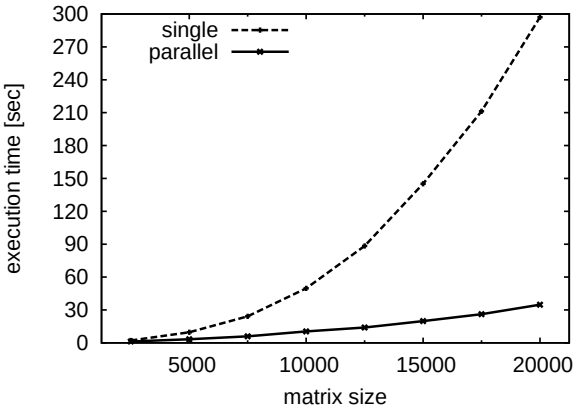

図 2: 並列化に伴う高速化

表 4 は Deflation の効果の小さい行列 DS に対して MR^3，DC-2，DC-k のそれぞれを比較するために，2500 次元から 20000 次元まで並列実行時間を計測した結果である。DC-k のデータは表 2 より再掲である。図 3 は表 4 をグラフ化したものである。逐次計算では MR^3 に速度で劣る DC-k であるが，十分に並列台数があれば，並列計算することで DC-k は MR^3 よりも高速に計算出来ることが分かる。

表 5，6 は行列 DS に対して，2500 次元から 20000 次元まで並列計算を行った時の相対残差および直交誤差を計算した結果であり，図 4 は表 5 を，図 5 は表 6 をグラフ化したものである。行列の分解法を改善した影響で精度が大幅に劣化することはなく，MR^3 に対して相対残差では

表 3: 各ステップでの実行時間の内訳

	全体	Step 1.	Step 2.			Step 3.	その他
			固有値	固有ベクトル	再直交化		
逐次 [sec]	297.15	3.75	42.32	80.72	68.50	101.02	0.85
並列 [sec]	34.83	4.98	4.90	5.94	13.01	5.44	0.56
効率 [%]	53.3	4.7	54.0	85.0	32.9	116.1	9.4

表 4: 行列 DS の実行時間 [sec]

サイズ	2500	5000	7500	10000
MR^3	0.70	2.83	6.52	11.64
DC-2	1.23	4.74	11.41	23.26
DC-k	1.19	3.30	5.93	10.39
サイズ	12500	15000	17500	20000
MR^3	18.21	25.15	35.19	46.17
DC-2	40.92	68.91	97.54	143.93
DC-k	14.00	19.85	26.16	34.83

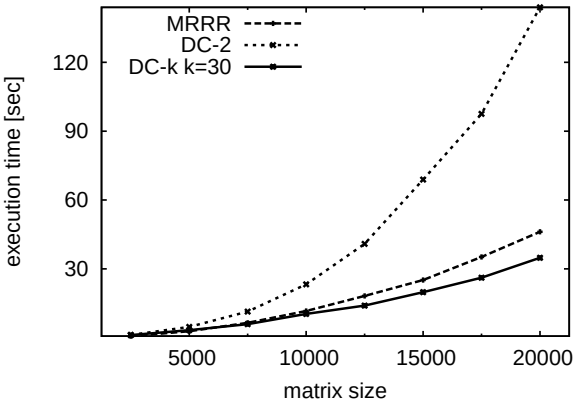

図 3: 行列 DS の実行時間

1 桁程度，直交誤差では 2 桁程度精度良く求まっている。また直交誤差は，固有ベクトルの再直交化により DC-2 よりも高い直交性を保つことが出来た。行列 DS と同じく行列 QC，DL に対しても同様の結果を得ることが出来，精度の高いことが確認出来た。

表 7 は Deflation 効果の大きい行列 DL に対して MR^3，DC-2，DC-k のそれぞれを比較するために，2500 次元から 20000 次元まで並列実行時間を計測した結果である。また，図 6 は表 7 をグラフ化したものである。行列 DL では Deflation の効果によって固有値問題の規模を縮小することが出来，DC-2，DC-k 共に MR^3 よりも非常に高速に計算出来た。

表 8 は量子カオス系のモデル行列 QC に対して MR^3，DC-2，DC-k のそれぞれを比較するために，2500 次元から 20000 次元まで並列実行時間を計測した結果である。図 7 は表 8 をグラフ化したものである。行列 DS と同じように DC-k，MR^3，DC-2 の順に高速で DC-k は MR^3 よりも 2 倍ほど高速に計算出来た。また，行列 QC は行列 DS のような作為的でめったに現れない行列ではなく，一般的によく用いられる行列であるため，行列 QC の固有値問題が高速に解けたことは DC-k が実用的なアルゴリズムであることを示している。

表 9，10，11 は LU 分解の改善前と改善後の並列実行時間をそれぞれの行列で計測した結果であり，2500 次元から 20000 次元まで計測を行った。比較のために MR^3 の計測結果も再掲する。図 8 は表 9 を，図 9 は表 10 を，図 10 は表 11 をグラフ化したものである。$\tilde{F}$ の疎行列性を利用することで行列 DS に対して高速化することが出来，特に 20000 次元の問題を扱った際には 2 倍以上の速度向上を達成した。また，改善前は並列化しても DC-k は MR^3 より遅かったが，$\tilde{F}$ の分解法の改善によって MR^3 よりも高速に計算出来た。ただし，行列 DL，QC に関しては m の大きさが極端に大きくならないため大きな変化はなかった。

表 5: 相対残差 [$\times 10^{-15}$]

サイズ	2500	5000	7500	10000
MR^3	25.9	39.3	52.8	61.1
DC-2	3.08	4.35	5.21	6.20
DC-k	7.07	9.00	14.1	11.8
サイズ	12500	15000	17500	20000
MR^3	75.4	79.4	94.9	90.8
DC-2	6.75	7.24	10.6	8.96
DC-k	15.3	16.3	18.5	17.1

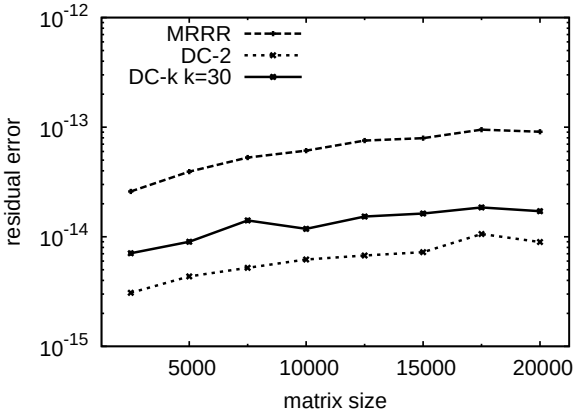

図 4: 相対残差

表 6: 直交誤差 [$\times 10^{-14}$]

サイズ	2500	5000	7500	10000
MR^3	77.7	171.	287.	394.
DC-2	2.13	3.46	5.73	7.35
DC-k	1.11	1.15	1.22	1.32
サイズ	12500	15000	17500	20000
MR^3	549.	608.	893.	1130.
DC-2	10.3	10.1	10.6	14.5
DC-k	1.36	1.41	1.63	1.51

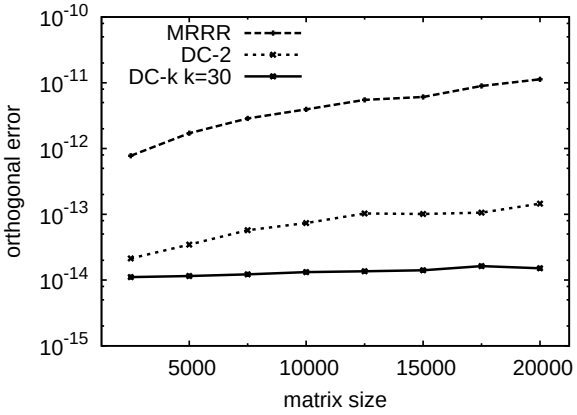

図 5: 直交誤差

表 7: 行列 DL の実行時間 [sec]

サイズ	2500	5000	7500	10000
MR^3	1.21	4.23	9.20	16.06
DC-2	0.74	1.58	2.45	3.35
DC-k	0.72	1.56	2.57	3.58
サイズ	12500	15000	17500	20000
MR^3	24.49	35.08	47.41	61.24
DC-2	4.37	5.24	5.89	7.15
DC-k	4.54	5.46	6.29	7.64

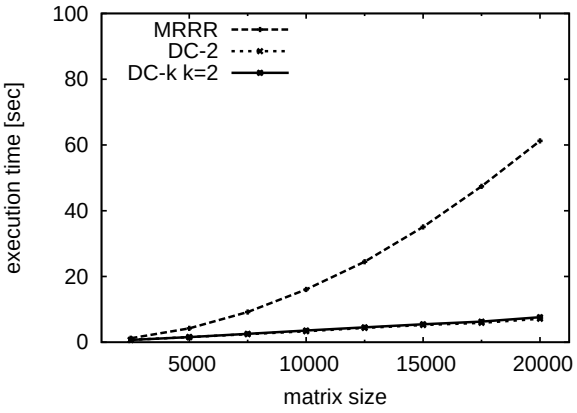

図 6: 行列 DL の実行時間

表 8: 行列 QC の実行時間 [sec]

サイズ	2500	5000	7500	10000
MR^3	1.03	3.72	8.32	14.68
DC-2	1.20	4.18	9.72	19.28
DC-k	1.05	2.60	4.75	7.12
サイズ	12500	15000	17500	20000
MR^3	22.90	32.88	44.55	58.10
DC-2	33.59	51.82	77.44	115.46
DC-k	10.34	14.12	18.86	25.06

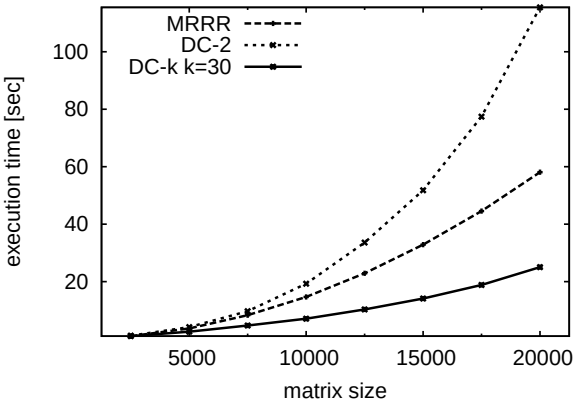

図 7: 行列 QC の実行時間

表 9: 行列 DS: 分解法の違い [sec]

サイズ	2500	5000	7500	10000
MR^3	0.70	2.83	6.52	11.64
改善前	1.15	7.60	7.14	14.96
改善後	1.19	3.30	5.93	10.39
サイズ	12500	15000	17500	20000
MR^3	18.21	25.15	35.19	46.17
改善前	22.76	38.44	56.09	83.66
改善後	14.00	19.85	26.16	34.83

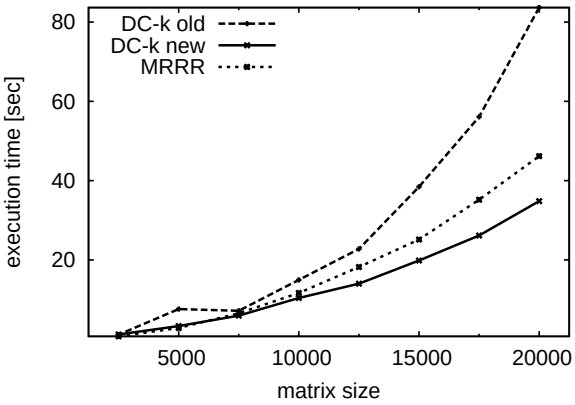

図 8: 行列 DS: 分解法の違い

表 10: 行列 DL: 分解法の違い [sec]

サイズ	2500	5000	7500	10000
MR^3	1.21	4.23	9.20	16.06
改善前	0.73	1.60	2.49	3.48
改善後	0.72	1.56	2.57	3.58
サイズ	12500	15000	17500	20000
MR^3	24.49	35.08	47.41	61.24
改善前	4.55	5.32	6.21	7.47
改善後	4.54	5.46	6.29	7.64

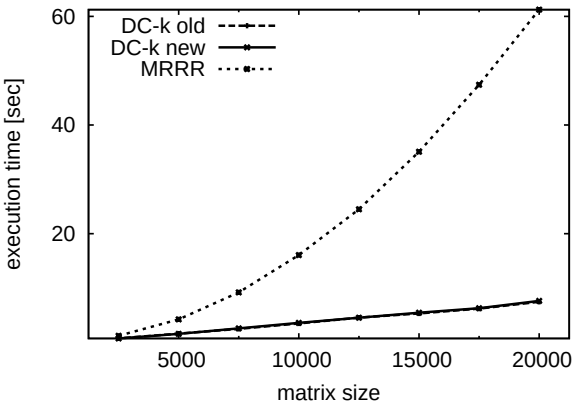

図 9: 行列 DL: 分解法の違い

表 11: 行列 QC: 分解法の違い [sec]

サイズ	2500	5000	7500	10000
MR^3	1.03	3.72	8.32	14.68
改善前	0.94	2.50	4.76	7.34
改善後	1.05	2.60	4.75	7.12
サイズ	12500	15000	17500	20000
MR^3	22.90	32.88	44.55	58.10
改善前	10.54	14.89	20.21	27.25
改善後	10.34	14.12	18.86	25.06

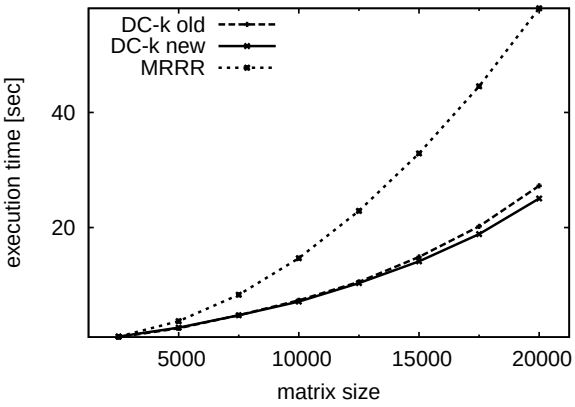

図 10: 行列 QC: 分解法の違い

6. まとめ

行列 $\tilde{F}$ の分解法を改善することによって実対称固有値問題に対する多分割の分割統治法のプログラムを高速化し，Deflation の効果が小さい行列に対し文献 [5] 記載当時と比較して 2 倍以上も高速に計算出来た。行列の分解法を変えたことによる演算精度の劣化は起こっておらず，MRRR 法と比較してより高精度に計算出来た。並列計算を行えば，紹介した行列の多くで，MRRR 法よりも高速に計算出来，特に，実用上現れるであろう量子カオス系のモデル行列に対しても高速に計算出来た。これはアルゴリズムの有用性を示している。よって，実対称固有値問題に対する多分割の分割統治法が並列計算機上において有用なアルゴリズムであることを確認出来た。

最後に本研究は，重原 孝臣，桑島 豊，田村 純一の三氏との共同研究である。

謝辞

本研究は東京大学情報基盤センター若手利用者推薦制度の下で遂行しました。サポートに感謝致します。

参考文献

[1] J. J. M. Cuppen, “A divide and conquer method for the symmetric tridiagonal eigenproblem”, Numerische Mathmatik, Vol. 36, pp. 177–195 (1981).

[2] I. S. Dhillon, “A new $O(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem”, University of California at Berkeley, Berkelery, CA (1998).

[3] 桑島豊, 重原孝臣, 「実対称三重対角固有値問題の分割統治法の拡張」, 日本応用数理学会論文誌, Vol. 15, No. 2, pp. 89–115 (2005).

[4] 桑島豊, 重原孝臣, 「実対称三重対角固有値問題に対する多分割の分割統治法の改良」, 日本応用数理学会論文誌, Vol. 16, No. 4, pp. 453–480 (2006).

[5] 桑島豊, 坪谷怜, 田村純一, 重原孝臣, 「実対称固有値問題に対する多分割の分割統治法の SR11000 への一実装」, スーパーコンピューティングニュース, Vol. 9, No. Special Issue 1, pp. 47–70 (2008).

[6] I. Dumitriu, A. Edelman, "Matrix models for beta ensembles", Journal of Mathematical Physics, Vol. 43, No. 11, pp. 5830–5847 (2002).

[7] M. L. Mehta, "Random matrices", Academic Press (1990).

[8] E. Anderson, et al., "LAPACK users' guide", SIAM, Philadelphia, PA, USA (1999).