大規模分散並列環境におけるコレスキーQR 型アルゴリズムによる縦長行列の列ピボット付き QR 分解の性能評価（続）

深谷 猛

北海道大学 情報基盤センター

１．はじめに

行列分解は科学技術計算の基盤技術の一つであり、高性能な数値計算技術が求められている。著者らは、基本的な行列分解の一種である QR 分解について、これまでに様々な研究開発を行ってきた。前回の記事[1]では、縦長行列に対する列ピボット付き QR 分解（通称：QRCP：QR factorization with column pivoting）[2]に対して、著者らが現在開発している CholeskyQR 型アルゴリズムを、2023 年 1 月実施の大規模 HPC チャレンジで評価した結果の速報を報告した。その後、再度、2023 年 8 月に「Wisteria/BDEC-01（Odyssey）」および「Oakbridge-CX」の両システムにおける大規模 HPC チャレンジの機会を頂き、初回の性能評価の後に改良を加えたアルゴリズムについて、より詳細な性能評価を実施することができた。そこで、本稿では、2 回目の大規模 HPC チャレンジにおける性能評価を通して得られた最新の結果のハイライトを紹介する。

２．前回の記事の概要

本節では、前回の記事[1]の概要を述べる。詳細については、前回の記事を参照されたい。

２．１．問題設定および代表的なアルゴリズム

与えられた行列$A \in \mathbb{R}^{m\times n}$ $(m \geq n)$に対して

$$AP = QR$$

の形で行列を分解する計算を QRCP と呼ぶ[2]。ここで、$P \in \mathbb{R}^{n\times n}$は置換行列、$Q \in \mathbb{R}^{m\times n}$は列直交行列、$R \in \mathbb{R}^{n\times n}$は上三角行列である。QRCP は Rank Revealing QR 分解の一種であり、行列の低ランク近似等の応用を持つ[3, 4, 5]。なお、本研究では行列Aが縦長（tall-skinny）の場合、つまり、$m \gg n$の場合を想定する。

QRCP に対して、列ピボット付きハウスホルダーQR 分解（HQR-CP：Householder QR with Column Pivoting）が代表的な数値計算アルゴリズムとして知られている[3, 6]。HQR-CP では、各ステップで貪欲法的にピボット列を選択（Pの構成と本質的に等価）しながら、ハウスホルダー変換による行列の上三角化を行う。これらの詳細は、例えば、文献[7]等を参照されたい。

２．２．QRCP に対するコレスキーQR 型アルゴリズムの開発

縦長行列に対する通常（列ピボットなし）の QR 分解において、コレスキーQR 型アルゴリズムが有効である[8, 9, 10, 11, 12]。この事実を踏まえて、コレスキーQR 型アルゴリズムを QRCP に拡張し、その有効性を示すことが我々の研究目的である。数学的には、通常のコレスキーQR アルゴリズムに対して、コレスキー分解を完全ピボット（Complete Pivoting）付きコレスキー分解：$P^{\mathsf{T}}WP \rightarrow R^{\mathsf{T}}R$ [13]に置き換えることで、QRCP への拡張が可能となる。しかし、数値計算の

場合、丸め誤差の影響を受けるため、一般的に、正しい（HQR-CP と等しい）ピボットの選択に失敗する。

そこで、我々は、完全ピボット付きコレスキー分解の計算過程で現れる値をチェックして、ある条件を満たした場合に計算を打ち切り（それ以降の結果の信頼性が低いと判断）、それまでに得られた部分的なピボット選択の情報のみを採用するアプローチを考案した。そして、反復的に（部分的な）完全ピボット付きコレスキー分解を繰り返す形のアルゴリズムを構築し、Ite-CholQR-CP（Iterative CholeskyQR with Column Pivoting）と名付けた。Ite-CholQR-CP の基本的な構造は、従来のコレスキーQR 型アルゴリズムと同じであり、計算の大部分が行列積（GEMM）に代表される Level-3 BLAS で実行可能であり、分散並列計算における集団通信の回数が$O(1)$である（CA: Communication-Avoiding）、といった高性能計算に適した特徴を維持している。なお、Ite-CholQR-CP の詳細については、本研究の成果をまとめた論文[14]が国際会議に採択されたので、そちらに委ねる形としたい。

２．３．性能評価結果の概要：HQR-CP との比較

Wisteria/BDEC-01（Odyssey）（以下、BDEC-O）と Oakbridge-CX（以下、OBCX）の両システム上で、MPI（と 1 次元データ分散）を用いて分散並列実装した Ite-CholQR-CP を、代表的な既存手法である HQR-CP（の簡略版）と比較した。BDEC-O（4096 ノード）では、nが比較的小さい（$n = 16, 32, 64$）の場合に、Ite-CholQR-CP は HQR-CP よりも高速（最大で 4 倍弱）であった。一方、OBCX（1024 ノード）の場合に、テストした全てのn（最大で$n = 1024$）で HQR-CP よりも高速（最大で 25 倍強）であった。これらの結果より、我々が開発した Ite-CholQR-CP は BDEC-O や OBCX のような一般的な大規模分散並列環境（所謂、スパコン）において、代表的既存手法である HQR-CP よりも有効な手法となり得る可能性が高いことが示された。なお、共有メモリ環境（マルチコア CPU 環境）における性能評価結果も上記の国際会議論文[14]で提示しているので、興味がある場合は参照されたい。

３．今回の性能評価の概要と主結果

本節では、今回の大規模 HPC チャレンジの実施内容の概要と主な結果を報告する。

３．１．実施内容の概要

前回の性能評価では、代表的な既存手法の一つである HQR-CP を比較対象とした。一方、計算対象の行列が縦長の場合、以下の計算手法も有力となる。まず、行列Aに対して、通常の QR 分解

$$A = Q_1 R_1$$

を行い、次に、得られた上三角行列R_1に対して、HQR-CP（あるいは他の手法）により

$$R_1 P = Q_2 R$$

と QRCP を行う。これにより

$$AP = Q_1 Q_2 R = QR,\ \ Q = Q_1 Q_2$$

と行列Aの QRCP が得られる[15]。前半のAに対する通常の QR 分解の計算量は$O(mn^2)$であり、一方、後半のR_1に対する QRCP の計算量は$O(n^3)$であるため、行列Aが縦長（$m \gg n$）の場合には、前半の計算が支配的となる。したがって、（列ピボットなしの）コレスキーQR 型アルゴリズムなど、縦長行列に対する高性能な QR 分解の計算手法を活用することで、高速に QRCP を計算するこ

とが可能となる。今回、我々が開発した Ite-CholQR-CP の有効性を議論する上で、この計算手法との比較は不可欠であり、今回の性能評価では、これを主眼に置く。

また、Ite-CholQR-CP の実装方法に関して、いくつか改良できる可能性があることが分かった。具体的な実装の内容については、現在論文を執筆中であり、それに委ねる形としたい。方針としては、アルゴリズムに対する数学的な考察に基づいた、一部の演算（例：数学的にゼロになることが分かっている部分の演算）の省略や、ループの順番を入れ替えることによる、反復を途中で打ち切る場合に無駄になる可能性の高い演算の削減、という改良の導入を試みた。今回の実装方法の改変により、理論的な演算量の削減は期待できるが、一方で、各処理の演算効率（FLOPS）も変わるため、計算時間の意味での有効性は断言できない。そこで、今回の性能評価では、Ite-CholQR-CP の新旧の実装方法の比較も行う。

最後に、問題設定として、前回の大規模 HPC チャレンジの際の設定（Q,R,Pを全て求める）に加えて、事前に指定された一部の結果（例えば、必要なランクが指定されていて、それに対応する結果）のみを求める場合も考える。このような問題設定に対して、Ite-CholQR-CP は途中で計算を打ち切ることが可能であるが、一方で、上述した縦長行列に対する通常の QR 分解に基づく手法は、前半の QR 分解は完全に行うことが求められる（ランクの情報は後半で得られる）。そのため、Ite-CholQR-CP が有効となる可能の高い問題設定であり、検証する価値があると考える。

性能評価における設定や条件は前回と同様であり、前回の記事[1]を参照されたい。今回、新たに比較対象とした、通常の QR 分解に基づく手法に関しては、コレスキーQR 型アルゴリズムと TSQR アルゴリズムをそれぞれ用いた手法を実装した。前者については、シフト付きコレスキーQR（Shifted CholeskyQR）アルゴリズムを前処理に用いた手法であり、前処理の回数を動的に決定する形を採用した。これらのアルゴリズムの詳細については、文献[10, 11, 12]などを参照されたい。なお、上三角行列の QRCP は HQR-CP（LAPACK の関数）を用いた。

３．２．評価結果

図 1 に、BDEC-O（4096 ノード）および OBCX（1024 ノード）における、各手法（の各実装）の HQR-CP に対する速度向上を示す。なお、「Ite-CholQR-CP」に関して、1 が以前の実装、2 が新しい実装である。また、「CholQR-QRCP」は通常のコレスキーQR 型アルゴリズムに基づいた手法、「TSQR-QRCP」は TSQR アルゴリズムに基づいた手法であり、それぞれ複数の実装を試した（BDEC-O では一部の LAPACK 関数が利用できないため、評価した TSQR-QRCP の実装が OBCX よりも少ない）。なお、mを固定（$m = 16777216$）して、$n = 16, \ldots, 512$と変えて性能を測定した。

図 1 のグラフから、縦長行列に対する通常の QR 分解を用いた手法も十分に有効であり、条件によっては Ite-CholQR-CP よりも高性能となっていることが確認できる。また、通常の QR 分解に対するコレスキーQR 型アルゴリズムや TSQR アルゴリズムの性能は文献[11, 12]などでも報告されており、nが大きくなった場合に TSQR の性能が低下する点や BDEC-O と OBCX で性能の挙動が異なる点など、これまでの報告と整合性が確認できる結果であることも分かる。

また、図 1 より、Ite-CholQR-CP について、今回の実装方法の変更（グラフ中の 2）が計算時間の意味でも概ね有効であったことも確認できる。特に、OBCX においてnが大きい場合に効果が高くなっていることが観察できる。

図 1 の結果から、Ite-CholQR-CP と通常の QR 分解に基づく手法は、どちらも縦長行列の QRCP を計算する際に有望であることが確認できる。また、Ite-CholQR-CP および通常のコレスキーQR

型アルゴリズムに基づく手法は、どちらも反復型の計算手法であり、計算対象の行列の条件（や手法内部のパラメータの設定）に応じて反復回数が変化する。そのため、限られた実験結果のみから、両者の優劣を断言すべきではないことにも注意が必要である。以上の内容を踏まえると、我々が開発した Ite-CholQR-CP について、現時点では、縦長行列の QRCP の計算において、既存の有力な手法と同程度の性能を持った（有効な選択肢の一つになり得る）手法、と評価するのが妥当であると考える。

次に、部分的な QR 分解を計算する場合を考える。具体的には、Qの一部（前半部分）と対応するRの部分行列のみを必要とする場合を想定する。このような設定において、Ite-CholQR-CP では反復の途中で計算をストップすることが可能であるが、最初に通常の QR 分解を行う手法では、QR 分解全体を計算する必要がある。そのため、Ite-CholQR-CP の有効性が期待できる問題設定である。図 2 に、BDEC-O（4096 ノード）および OBCX（1024 ノード）において、Qの最初の 1/4 の列ベクトル（$n/4$本の列ベクトル）のみを計算する場合の、各手法（の各実装）の HQR-CP を基準にした速度向上を示す。

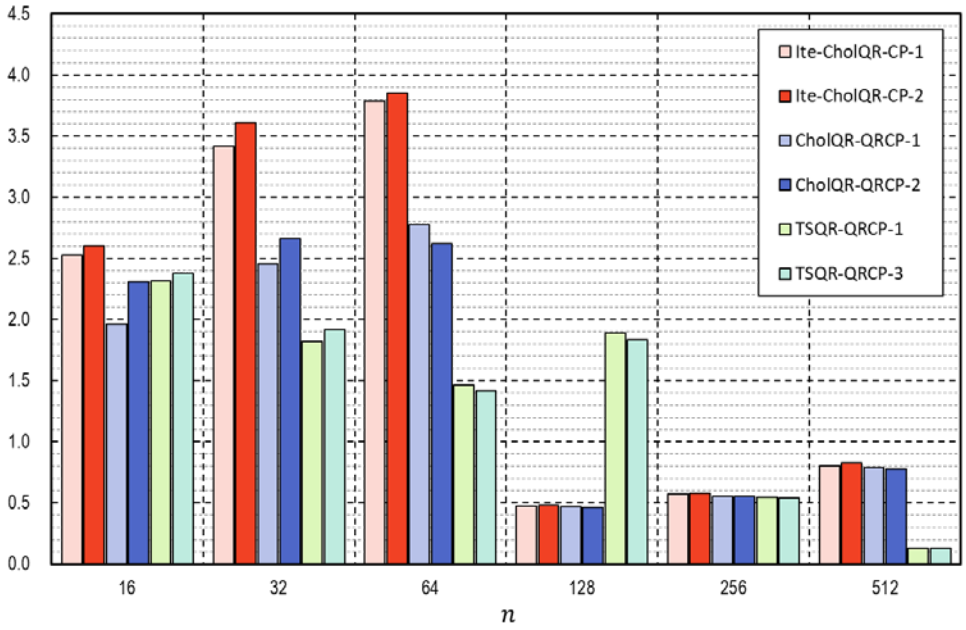
図 2 より、部分的な QRCP を計算する設定では、Ite-CholQR-CP が、通常の QR 分解を用いる手法よりも有効となるケースが多く、また、両者の差も図 1 の結果より大きいことが確認できる。特に、nが大きくなるほど、両者の差が拡大する傾向にあることが読み取れる。ただし、HQR-CP も途中で計算をストップすることができるため、HQR-CP に対する速度向上はnに応じて大きくなるわけではなく、注意が必要である。

４．おわりに

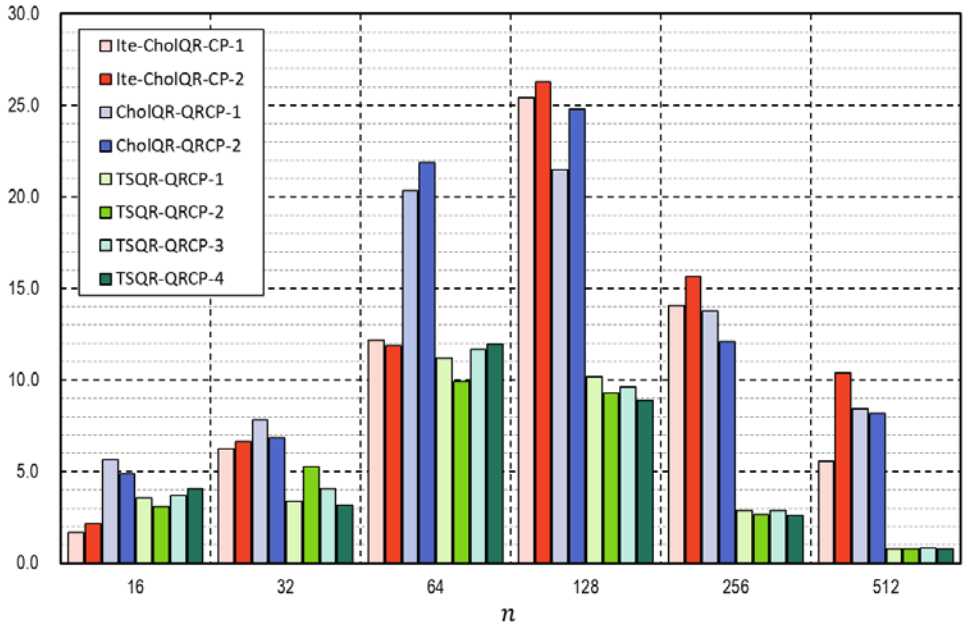
本稿では、前回の記事に引き続き、我々が開発しているコレスキーQR 型の QRCP アルゴリズム（Ite-CholQR-CP）に関する性能評価の結果を中心に報告した。特に、縦長行列の QRCP を、通常の QR 分解を利用して計算する手法を比較対象とした性能評価の結果を示した。これまでに得られた性能評価の結果より、Ite-CholQR-CP は、通常の QR 分解を利用する手法と同程度の性能が期待され、状況に応じて両者の優劣は変化することが確認できた。乱択アルゴリズム[16]などのその他の最新のアルゴリズムを含めて、より詳細な性能評価を行うことが今後の課題である。また、部分的な QRCP を計算する設定では、Ite-CholQR-CP の有効性が十分に期待できることが分かった。各アプリケーションでのニーズに応じて、適切な計算手法を使い分けることが重要であり、そのための判断材料になる性能データの収集や分析を今後行うことが必要である。

謝　辞

大規模 HPC チャレンジの実施に関してお世話になりました、東京大学情報基盤センターの関係者の皆様に深く感謝いたします。本稿は、山本有作 教授（電気通信大学）および中務佑治 准教授（University of Oxford）との共同研究に基づいた内容であり、両氏に感謝いたします。本研究の一部は、JST さきがけ（課題番号：JPMJPR20M8）、JSPS 科研費（課題番号：JP21K11909、JP23H00462）、学際大規模情報基盤共同利用・共同研究拠点（JHPCN）、および、革新的ハイパフォーマンス・コンピューティング・インフラ（HPCI）（課題番号：jh230010）の支援を受けています。

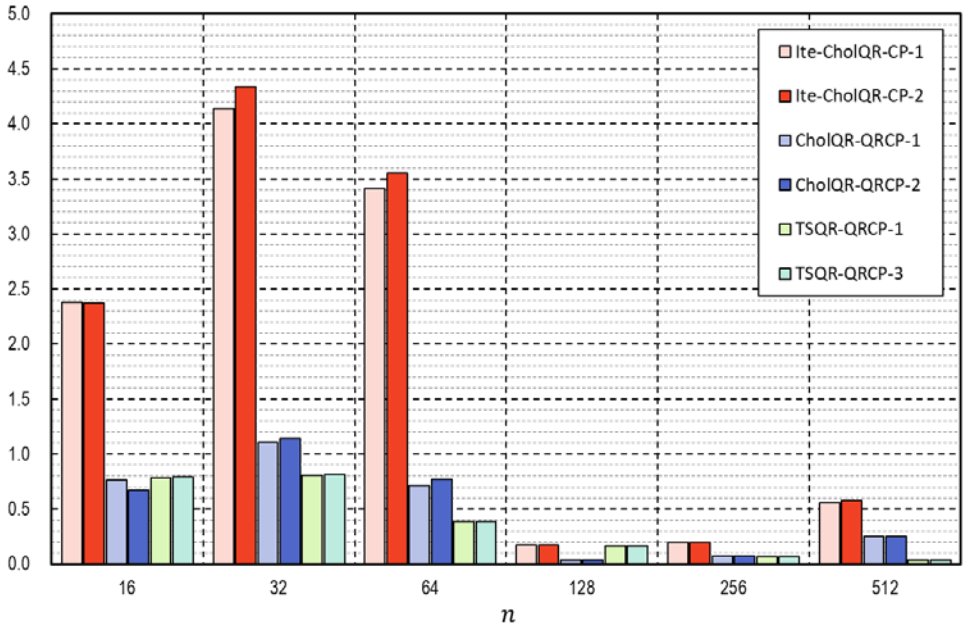


(a) BDEC-O (4096 ノード)

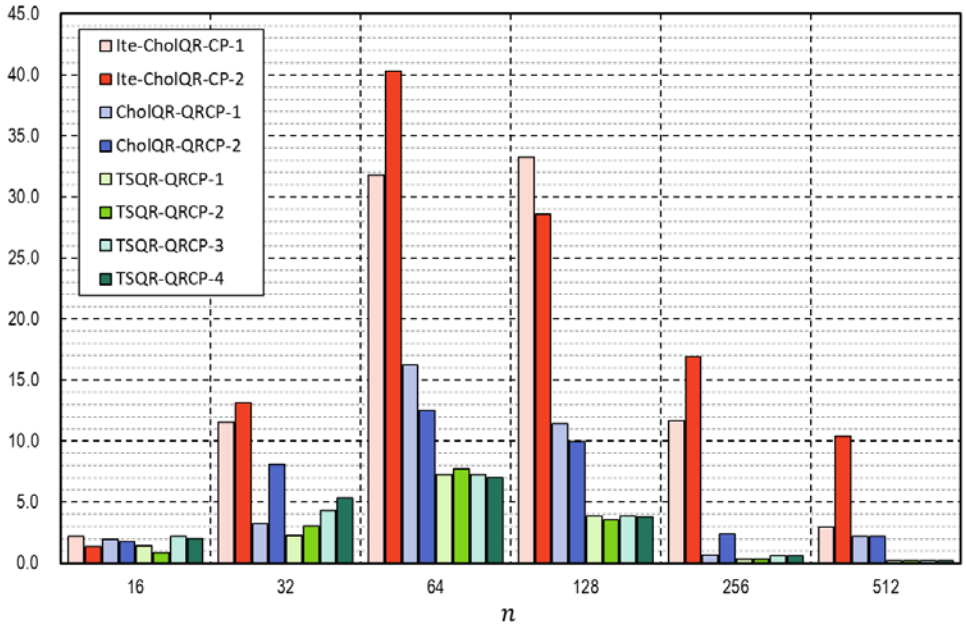


(b) OBCX (1024 ノード)

図 1：HQR-CP に対する速度向上（$m = 16777216$）。



(a) BDEC-O (4096 ノード)



(b) OBCX (1024 ノード)

図 2：部分的な QRCP（Qの先頭から 1/4 の列のみを計算する場合）における
HQR-CP に対する速度向上（$m = 16777216$）。

参 考 文 献

[1] 深谷 猛, 大規模分散並列環境におけるコレスキーQR型アルゴリズムによる縦長行列の列ピボット付きQR分解の性能評価, 東京大学情報基盤センター スーパーコンピューティングニュース, Vol. 25, No. 4 (2023), pp. 20-28.

[2] T. F. Chan, Rank revealing QR factorizations, Linear Algebra Appl., Vol. 88-99 (1987), pp. 67-82.

[3] G. Golub, Numerical methods for solving linear least squares problems, Numer. Math., Vol. 7 (1965), pp. 206-216.

[4] S. Chandrasekaran and I. C. F. Ipsen, On Rank-Revealing Factorizations, SIAM J. Matrix Anal. Appl., Vol. 15 (1994), pp. 592-622.

[5] S. Chandrasekaran and I. C. F. Ipsen, On Rank-Revealing Factorizations, SIAM J. Matrix Anal. Appl., Vol. 15 (1994), pp. 592-622.

[6] P. Businger and G. H. Golub, Linear least squares solutions by Householder transformations, Numer. Math., Vol. 7 (1965), pp. 269-276.

[7] M. Gu and S. C. Eisenstat, Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization, SIAM J. Sci. Comput., Vol. 17 (1996), pp. 848-869.

[8] T. Fukaya, Y. Nakatsukasa, Y. Yanagisawa, and Y. Yamamoto, CholeskyQR2: A Simple and Communication-Avoiding Algorithm for Computing a Tall-Skinny QR Factorization on a Large-Scale Parallel System, ScalA'14, pp. 31-38, 2014.

[9] Y. Yamamoto, Y. Nakatsukasa, Y. Yanagisawa, and T. Fukaya, Roundoff Error Analysis of the CholeskyQR2 Algorithm, ETNA, Vol. 44 (2015), pp. 306-326.

[10] T. Fukaya, R. Kannan, Y. Nakatsukasa, Y. Yamamoto, and Y. Yanagisawa, Shifted Cholesky QR for Computing the QR Factorization of Ill-Conditioned Matrices, SIAM J. Sci. Comput., Vol. 42 (2020), pp. A477-A503.

[11] 深谷 猛, 縦長行列のQR分解に対する各種アルゴリズムの比較:Oakforest-PACS上での性能評価, 東京大学情報基盤センター スーパーコンピューティングニュース, Vol. 22, No. 6 (2020), pp. 28-39.

[12] T. Fukaya, Distributed Parallel Tall-Skinny QR Factorization: Performance Evaluation of Various Algorithms on Various Systems, PDCAT 2022 (LNCS Vol. 13798), pp. 275-287, 2022.

[13] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed., SIAM, 2002

[14] T. Fukaya, Y. Nakatsukasa, and Y. Yamamoto, A Cholesky QR type algorithm for computing tall-skinny QR factorization with column pivoting, IPDPS 2024, 2024. (accepted).

[15] R. Cunha, D. Becker, and J. Patterson, New Parallel (Rank-Revealing) QR Factorization Algorithms, Euro-Par 2002, pp. 677-686, 2002.

[16] J. A. Duersch and M. Gu, Randomized Projection for Rank-Revealing Matrix Factorizations and Low-Rank Approximations, SIAM Rev., Vol. 62 (2020), pp. 661-682.