Referee report:

Fast Algorithms for the Multi-dimensional Jacobi Polynomial Transform,

by James Bremer, Qiyuan Pang and Haizhao Yang,
submitted to *Applied and Computational Harmonic Analysis*
(ACHA-2019-11).

This article developed a fast algorithm for computing multi-dimensional Jacobi polynomial transforms. The paper's chief contribution is that it extended the basic idea of the method in [19] from one dimension to multi-dimensions and from $(a, b) \subset (-1/2, 1/2)$ to $(a, b) \subset (-1, 1)$.

The paper is interesting and well written. The mathematical results are correct and the numerical results appear to be plausible and consistent. This paper deserves to be published in *Applied and Computational Harmonic Analysis.*

However, the authors should address and clarify the following comments and questions before the paper can be published.

1. **Theoretical complexity.** For the Chebyshev-Jacobi and even Jacobi-Jacobi transform, several algorithms with provable quasi-linear complexity are known as follows.

   [1] Alex Townsend, Marcus Webb and Sheehan Olver, Fast polynomial transforms based on Toeplitz and Hankel matrices, *Mathematics of Computation*, 87 (2018), pp. 1913-1934.

   [2] Richard Mikaël Slevinsky, On the use of Hahn's asymptotic formula and stabilized recurrence for a fast, simple and stable Chebyshev-Jacobi transform, *IMA Journal of Numerical Analysis*, 38(2018), pp. 102-124.

   [3] Jie Shen, Yingwei Wang, Jianlin Xia, Fast structured Jacobi-Jacobi transforms, *Mathematics of Computation*, 88(2019), pp. 1743-1772.

   While the authors demonstrate good scaling performance in this article, say quasi-linear complexity, by numerical examples, they do not have proofs of this behavior in the general setting (only observe it).

Basically, the algorithmic complexity of the proposed algorithm heavily depends on the (numerical) rank r in Eq.(43) or Eq.(44). It is important to have the asymptotic analysis for r as $n \to \infty$. At least a conjecture like $r = O(\log n)$ or sth else should be given.

Besides, it seems that the fast algorithms in both [19] and this paper are based on the low-rank property of the matrices. Why the algorithm in [19] only works for the cases with $(a, b) \subset (-1/2, 1/2)$ while the one in this paper works for a lager range $(a, b) \subset (-1, 1)$? This point is a great contribution of this paper and deserves better explanation.

2. **Numerical results.**

   (a) Numerical ranks. It is shown in the bottom left of Figure 1 that for given tolerance, the RS method provides a more compact matrix compression competitive to that of the CHEB method. However, in both of the two cases, the numerical ranks grow similarly like $r = O(\log(N))$. How about other values of (a, b)? Try more numerical tests and observe the different behaviors of numerical ranks for various pairs of (a, b), even for the cases with (a, b) outside $(-1, 1)$.

   (b) Numerical stability. No discussion is presented of the stability of any of these fast transformations, which presumably may be quite poorly conditioned. It would be nice to check if the fast transforms are actually accurate numerical inverses. Something like

   $$\|\text{Backward transform}\,(\text{Forward transform}(\boldsymbol{v})) - \boldsymbol{v}\|_2/\|\boldsymbol{v}\|_2, \qquad (1)$$

   for some random vector $\boldsymbol{v}$ will do.

3. **Applications.** The paper is weakly motivated. A reader needs an application in mind, where the authors advocate using the proposed algorithm.

4. **Minor issues**.

   (a) The notation of points $\{x_i\}_{i=1,\ldots,n} \subset (-1, 1)$ appears four times in the first paragraph, which is redundant. The reviewer suggests that it should be denoted as I for short.

   (b) When you introduce the Eqs.(3) and (4), it would be better to give the range for the variables t and ν clearly. For example, $t \in [0, \pi]$ and ν is a positive integer.

   (c) What are the boundary/initial conditions for the third order ODE (22)?

   (d) The reference [22] should have a year = 2018.