Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., referee comment RC1
https://doi.org/10.5194/gmd-2022-116-RC1, 2022

Comment on gmd-2022-116

Anonymous Referee #1

Referee comment on "Fast approximate Barnes interpolation: illustrated by Python-Numba implementation fast-barnes-py v1.0" by Bruno K. Zürcher, Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2022-116-RC1, 2022

It is a good paper. The idea of approximating Guassian weights using the central limit theorem (CLT) is very good. The Approximation 3 is a happy result mainly because of cancellation of the common factors in the numerator and the denominator. The algorithms for implementation are also nice and thorough. It is surprising to read how the computing time is reduced from a naive Barnes interpolation. Nevertheless, I have some comments and questions, basically for a better presentation.

1. It uses a special kind of the uniform random variable on an interval with a threshold. Hope the author describe why this threshold (sqrt(3/n) sigma) was chosen.

2. page 3, line 54: hope the author provide the definition of the n-fold convolution of p(x) with itself.

3. p5 line 92: hope the author provide more concretely what *^nx and *^yn are, because some readers may know it after reading the Algorithms 2, 3, and 4.

4. In Section 5, the author applied 4-fold convolution. It is known that a summation of uniform random variables converge slowly to the CLT compared to other uni-modal random variables, even though the convergence in distribution and the convergence in pdf are different. Thus I think only 4-fold is too small. I would recommend to apply at least 10-fold convolutions in real applications to ensure a good approximation.

5. The result in this paper is only a nice approximation to Barnes interpolation. I think the title can be changed to 'Fast approximation to Barnes interpolation'.

6. It uses a special kind of the uniform random variable and a numerical integration for convolution in Algorithm 2. It is still good, but I guess one may consider other random variables which may accelerate the convergene to the CLT. Then numerical integration may be a little more complex than that in this paper. This may increase computing time in Algorithm 2, but may be alright with a smaller n.

7. In Algorithms 2 and 3, I was a bit confused with the notations g, r_T, and F[i,j]. Hope the author kindly give short descriptions on these notations to improve reader's understanding.

8. Finally, I think that the idea of approximating Guassian weights using the CLT is very good and so it can be applied to the other area of numerical computations which use Gaussian weights. Hope the author try to search the literature if any body has already published or applied this idea.