Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., author comment AC1
https://doi.org/10.5194/gmd-2022-116-AC1, 2022

Reply on RC1

Bruno Zürcher

Author 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-AC1, 2022

Dear Referee,

Thank you very much for reading the manuscript carefully and for your valuable comments. Below I give a first response to these comments, point after point.

1. page 3, line 65: The calculation, why the interval has the length (sqrt(3/n) sigma) is actually not obvious. It is basically chosen in this way, such that the variance of the resulting uniform distribution u_n(x) is (sigma^2/n).
I will add in the manuscript a short comment about that and a calculation that verifies this.

2. page 3, line 54: I propose to provide the definition of the n-fold convolution of p(x) with itself in an appendix to the manuscript.

3. page 5, line 92: I agree, the explanation of the operators *^nx and *^yn in the text is minimalistic. I propose to give a thorough derivation of them in the appendix.

4. Section 5, rate of convergence: Maybe one has to differentiate from apllication to application. When Barnes interpolation is used to visualize data by means of isolines - as done in the manuscript - I noticed that the resulting isolines found more or less their stable form already after applying a 3-fold convolution. For other applications, which require a higher precision, a 10-fold convolution or even more might make sense.
I extend the corresponding remark on p. 18 line 295 accordingly.

5. I can change the title to something like "Fast Approximative Barnes Interpolation". I did not use the term "approximative" in the title, because other _feasible_ methods to compute Barnes interpolation are approximations as well, but do not emphasize this in an obvious way.

6. Yes, there are many other PDFs that have a faster CLT-convergence to a Gaussian than the uniform PDF - in the extreme case one could even take a normal distribution itself for which we would have n = 1.
But when using a non-trivial PDF, it is clear that putting and moving the weight window (as described on p. 9 line 161) over the data to be convolved requires in general 2T+1 multiplications and 2T additions per element. Thus, the simplicity of algorithm 2 is lost to a far part and the algorithmic complexity of it grows to O(L*T) instead of O(L). Overall we

then could not claim a complexity of O(N+W*H) anymore, this would be rather O(N+W*H*T).
Your comment leads me to consider to add a remark on p. 6 line 128 that Approximation 3 is valid in a much more general context, i.e. also for other (non-uniform) PDFs. In the case of a normal distribution even equality holds.

7. I try to write this more clear.

8. This is a good question. I know that the CLT is used by some applications to quickly generate the weights of a normal distribution (as above, of course only in an approximative way). Further I assume that some image processing programs like Photoshop employ this technique in order to apply a Gaussian blur filter on an image. This process can be regarded as a special case of the method presented in this manuscript, where the observation points in question are given by the image matrix. In this case the points are located on a regular grid and the denominator of equation (8) simplifies to 1. I try to find references for such applications and will mention them in the text of the conclusions section.

Best regards and thank you once more
Bruno