PCFFT IMPLEMENTATION NOTES

A PREPRINT

December 19, 2025

1 Computing the spreading grid

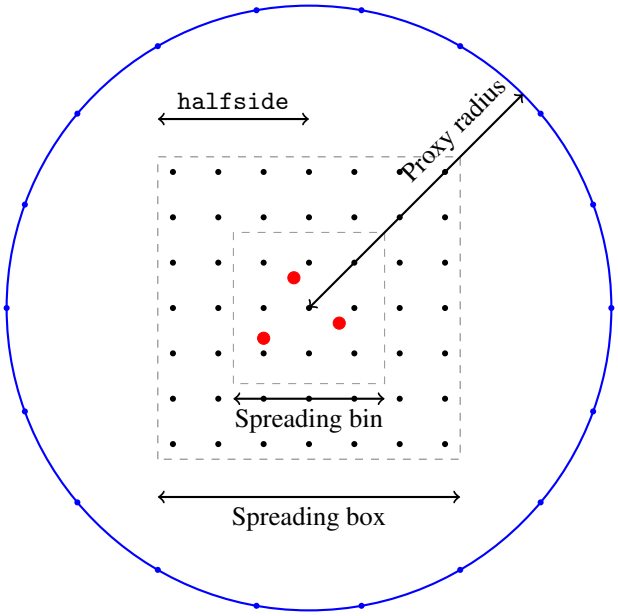

Figure 1: Schematic of the spreading geometry in 2D: sources (red), the regular discretization (black dots), and a proxy ring (blue).

1.1 Computing the spreading box size

The spreading box size is computed by `spread_halfside()`. This is meant to approximately control the number of source points in a particular spreading box.

1.2 Computing the regular grid spacing

This is performed in `dx_nproxy()`. We want to find parameters `dx` and `nproxy`. `dx` is the grid spacing of the regular discretization of the spreading box, which starts at $-\texttt{halfside} + \frac{\texttt{dx}}{2}$ and ends at $\texttt{halfside} - \frac{\texttt{dx}}{2}$. `nproxy` is the number of proxy points placed on a proxy ring (or sphere) outside the spreading box.

Notes on the geometry used:

- We put source points in a bin with sidelength `c_bwidth` $\times$ `halfside`.

(a) nbinpts $= 3$, nspread $= 7$ **(b)** nbinpts $= 4$, nspread $= 8$

Figure 2: Schematic of the padding of the regular grid around the source points. The black lines form the bottom left corner of the bounding rectangle, the blue dashed line is the first spreading bin, and the orange dashed line is its associated spreading box. The regular grid points (black dots) start slightly below the bounding rectangle. There are pad $= \lceil (\texttt{nspread} - \texttt{nbinpts})/2 \rceil$ extra grid points hanging below and to the left of the bounding rectangle to ensure proper padding for spreading.

- We place a proxy ring of radius $\sqrt{\texttt{d}} \times \texttt{halfside} \times \texttt{crad}$ where d is the dimension (2 or 3) and crad is a constant (default 2.0).
- If we consider breaking the spreading box into nspread cells, the grid points are placed at the center of each cell, so $\texttt{dx} = \frac{2 \times \texttt{halfside}}{\texttt{nspread}}$.

Notes on the algorithm used to compute dx and nproxy:

- Generate random sources in the spreading bin with side length halfside. Generate target points on a ring/sphere of radius $1.1\times$ the proxy radius.
- Increase nspread and nproxy until the error tolerance is met.
- Decrease nspread until the error tolerance is no longer met.
- Spreading bin is $\texttt{dx} \times \texttt{nbinpts} = \texttt{dx} \times \lfloor \texttt{nspread}/2 \rfloor$.

1.3 Constructing the regular grid

This is performed in `get_grid()`, which calls `spread_halfside()` and `dx_nproxy()`. Here are the basic steps:

- Compute halfside using `spread_halfside()`.
- Compute dx, nspread, and nproxy using `dx_nproxy()`.
- Construct a bounding rectangle around the source points, which, along with the side length of the spreading bin, determines ngrid, the number of regular grid points in each dimension.
- Add a padding of grid points on each side so that the spreading bins fit properly into the corners of the bounding rectangle. pad $= \lceil (\texttt{nspread} - \texttt{nbinpts})/2 \rceil$.
- Start the regular grid points a bit below the bottom corner of the bounding rectangle. See Figure 2 for an illustration of this padding.