PCFFT IMPLEMENTATION NOTES

A PREPRINT

December 4, 2025

1 Computing the spreading grid

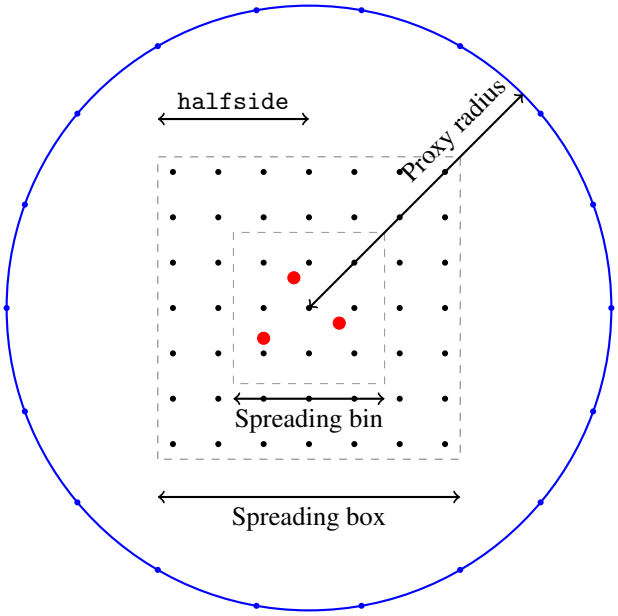

Figure 1: Schematic of the spreading geometry in 2D: sources (red), the regular discretization (black dots), and a proxy ring (blue).

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`.

- 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 `nproxy` until the error tolerance is no longer met.
- Re-generate random source points inside a spreading bin of size $\texttt{dx} \times \lceil \texttt{nspread}/2 \rceil$, which may be slightly larger than the original spreading bin. Increase `nspread` until the error tolerance is met.