Multi-Dimensional Generalized Fast Fourier Transforms

June 14, 2026

ER Technical Report

Excelsior Research LLC
https://excelsiorresearch.org

Matt Fulkerson

1.0 Generalized Fast Fourier Transforms

The Generalized Fast Fourier Transform (GFFT) algorithms defined herein are based on problem solving methods developed by the author. The author has developed a 1-D GFFT algorithm called the Non-Uniform Frequency-Dependent Out-of-Order Fast Fourier Transform (NUFDOOFFT), and under ER optimized it in double precision. NUFDOOFFT provides a rapid solution to

$$F(u) = \sum_m f_m(u) e^{-i2\pi u x_m} , \tag{1}$$

where the inputs $f_m(u)$ can depend on u, and x_m can be unequally spaced and arrive in any order (i.e. not necessarily monotonic).

Figure 1 shows a diagram of the 1-D algorithm implemented by ER.

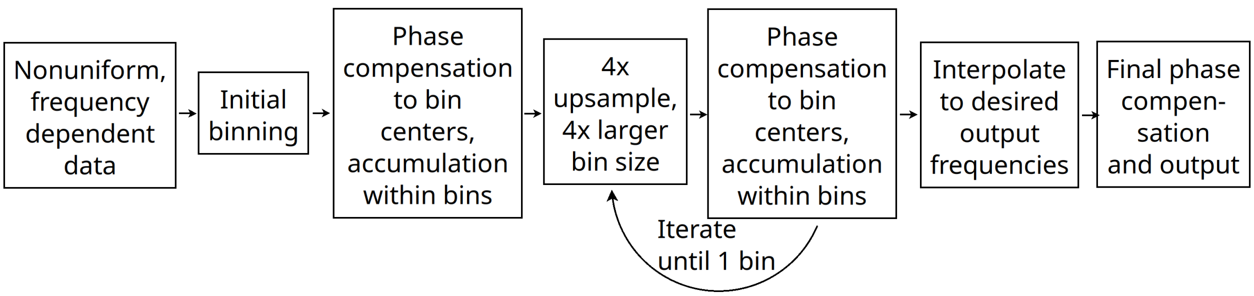

Figure 1: A diagram of 1-D GFFT.

In D dimensions, the math is very similar and more general. The equation to be solved for D dimensions is

$$F(\vec{u}) = \sum_m f_m(\vec{u}) e^{-i2\pi \vec{u}\cdot\vec{x}_m} . \tag{2}$$

Figure 2 shows a diagram of the multi-dimensional algorithm, to be implemented by ER.

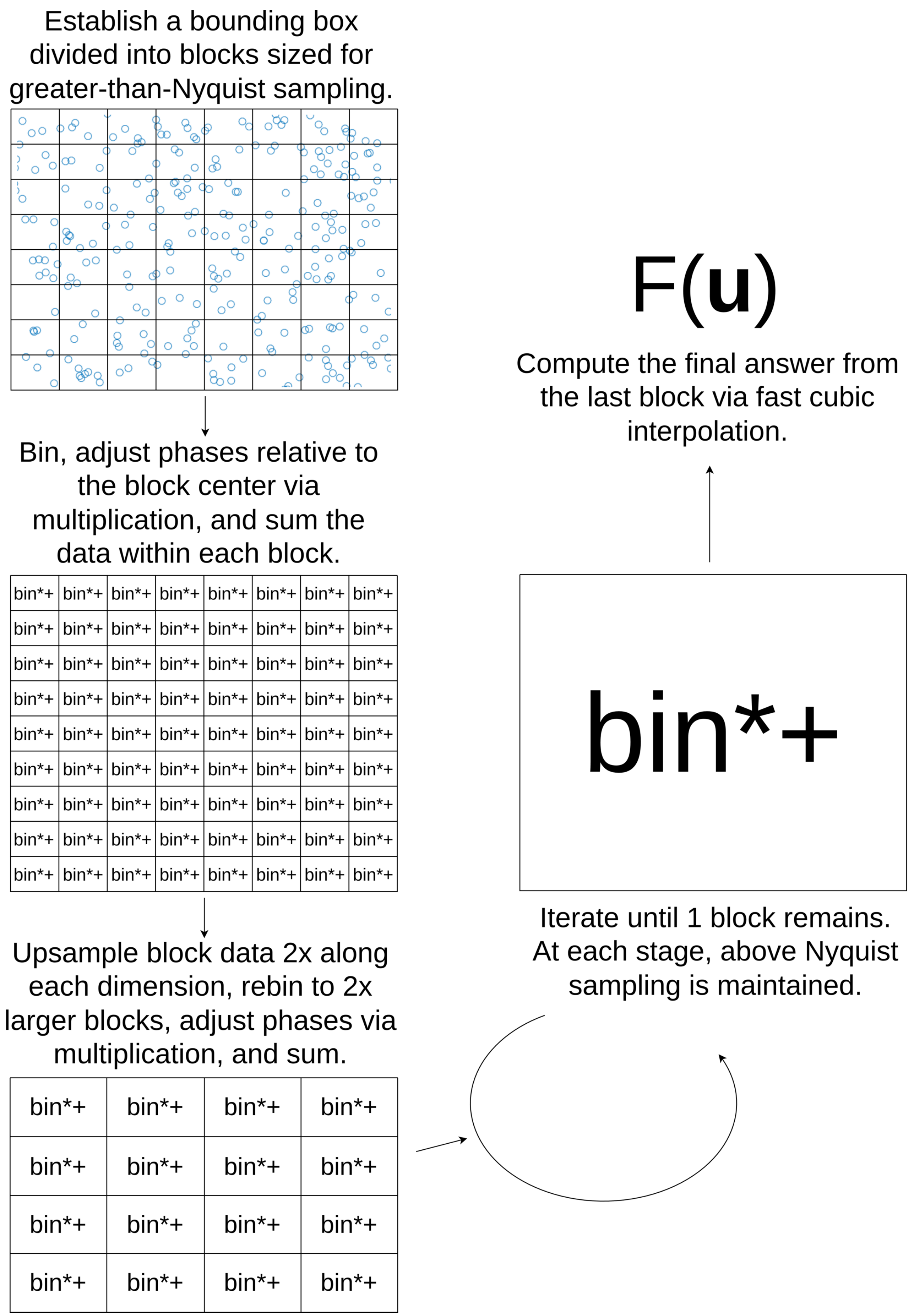

Figure 2: A diagram of multi-dimensional GFFT, illustrated in two dimensions.

2.0 What Are These Complex Exponentials and How Do They Work?

What you need to know is

- i is the imaginary unit $\sqrt{-1}$
- $e^{ix} \equiv \cos(x) + i\sin(x)$
  - When the argument x to a complex exponential is purely real, it is a phase to sine / cosine
- Complex exponentials follow the usual rules for ordinary exponentials
  - When multiplying two exponentials of the same base, you keep the base and add the exponents

3.0 Multi-Dimensional GFFT Mathematics from One Stage to the Next

GFFT involves stages that bin the data into multidimensional-dimensional blocks, multiply by complex sinusoids relative to each block center, and sum the result within each block. Let $B_n(\vec{u})$ be the partial sum of the data within block n, such that we adjust the origin to be relative to the center of block n. We define

$$B_n(\vec{u}) \equiv \sum_{m \in \text{block}\, n} f_m(\vec{u}) e^{-i2\pi\vec{u}\cdot(\vec{x}_m - \vec{x}_n)} \,. \tag{3}$$

Now we show how to recover $F(\vec{u})$ after this operation.

$$\begin{aligned}
\sum_n B_n(\vec{u}) e^{-i2\pi\vec{u}\cdot\vec{x}_n} &= \sum_n \left(\sum_{m \in \text{block}\, n} f_m(\vec{u}) e^{-i2\pi\vec{u}\cdot(\vec{x}_m - \vec{x}_n)} \right) e^{-i2\pi\vec{u}\cdot\vec{x}_n} && (4)\\
&= \sum_n \sum_{m \in \text{block}\, n} f_m(\vec{u}) e^{-i2\pi\vec{u}\cdot\vec{x}_m} && (5)\\
&= F(\vec{u}) && (6)
\end{aligned}$$

Thus the binning process has been shown to reproduce the desired GFFT result. The mathematics of further stages is similar, and the result is that any number of GFFT stages is possible, resulting in computing the transform accurately.

To show this by induction, we only have to show that the next stage produces a valid result for $F(\vec{u})$ from the previous stage. And so we have

$$C_p(\vec{u}) \equiv \sum_{n \in \text{block}\, p} B_n(\vec{u}) e^{-i2\pi\vec{u}\cdot(\vec{x}_n - \vec{x}_p)} \tag{7}$$

$$\begin{aligned}
\sum_p C_p(\vec{u}) e^{-i2\pi\vec{u}\cdot\vec{x}_p} &= \sum_p \left(\sum_{n \in \text{block}\, p} B_n(\vec{u}) e^{-i2\pi\vec{u}\cdot(\vec{x}_n - \vec{x}_p)} \right) e^{-i2\pi\vec{u}\cdot\vec{x}_p} && (8)\\
&= \sum_n B_n(\vec{u}) e^{-i2\pi\vec{u}\cdot\vec{x}_n} && (9)\\
&= F(\vec{u}) \,. && (10)
\end{aligned}$$

Therefore, by induction we can apply the mathematics for any number of stages in order to fully achieve the efficiency of the transform.

4.0 Multi-Dimensional GFFT Number of Operations

If a GFFT problem is characterized by M inputs and N resulting blocks into which the data are binned, the order of operations is $\mathcal{O}(M)$ + $\mathcal{O}(N \log N)$. The $\mathcal{O}(M)$ scaling comes from the up-front binning, phase adjustment via multiplication, and summation steps. The $\mathcal{O}(N \log N)$ scaling comes from the fact that for each iterative stage, blocks are increased in size by 2x along each dimension. For example, if the initial binning process results in a perfect D-dimensional hypercube, and there are 1024 initial blocks along each dimension ($N = 1024^D$ total blocks), then there are $\log_2(1024) = 10$ stages. In terms of the total number of blocks in the hypercube, there will be $\log_{2D}(N)$ stages. Since the number of operations from one stage to the next remains constant is $\mathcal{O}(N)$, the total number of operations for all stages is $\mathcal{O}(N \log_{2D} N)$.

If the initial binning does not result in a perfect D-dimensional hypercube, GFFT algorithms will simply stop the iterative process on each dimension once the number of blocks along that dimension is 1.

5.0 Handling of Sparsity

GFFT algorithms keep track of whether or not a block is occupied. Operations on unoccupied blocks will simply be skipped with minimal overhead. Thus GFFT is very efficient for sparse problems. The accumulation process will eventually incorporate blocks that are occupied, and so in subsequent stages the blocks will be fully processed.

6.0 Applications

- Simulation of signal construction for radar, given frequency-dependent scatterers, for any waveform
- Radar cross section calculation, given a CAD model and RF (radio frequency) physics, feeding into 2-D imaging
  - Range-Doppler imaging
  - Synthetic Aperture Radar imaging
- Medical imaging?
  - Stay tuned for the next revision!

7.0 Conclusion

GFFT is a completely new algorithm for computing Fast Fourier Transforms that are more general than an ordinary Fast Fourier Transform or even an ordinary Non-Uniform Fast Fourier Transform. Excelsior Research will further pursue applications of GFFT. If you are interested in conversing email `mattfulkerson@excelsiorreach.org`.