

An Equivariant Fast Fourier Transform Algorithm

L. Auslander
Department of Mathematics
The City University of New York
New York, NY 10036

J. R. Johnson
Department of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104

R. W. Johnson
Department of Computer Science
St. Cloud State University
St. Cloud, MN 56301

Technical Report DU-MCS-96-02
Department of Mathematica and Computer Science
Drexel University
Philadelphia, PA 19104

Abstract

This paper presents a generalization of the Cooley-Tukey fast Fourier transform algorithm that respects group symmetries. The algorithm, when applied to a function invariant under a group of symmetries, fully exploits these symmetries to reduce both the number of arithmetic operations and the amount of memory used. The symmetries accommodated by the algorithm include all of the crystallographic groups. These groups arise in crystallographic structure analysis, which was the motivating application for the algorithm presented in this paper.

0.1 Introduction

In this paper, it is shown that a generalization of the Cooley-Tukey Fast Fourier Transform (FFT), presented by the authors in [1] can be modified to take advantage of a wide class of symmetries in the data, including crystallographic group symmetries, to produce a fast ($N \log N$) algorithm that fully exploits these symmetries to reduce the computational cost. The algorithm is derived by a divide and conquer procedure in the same spirit as the Cooley-Tukey Fast Fourier Transform (FFT) [2]. The symmetries are viewed as the action of a group on a set of functions and the divide part of the “divide and conquer” is shown to respect this action. The resulting algorithm therefore respects the action of the group, and hence, is equivariant.

The motivating application for our work is crystallographic structure analysis, where the Fourier transform is used to derive the structure of a crystal from its x-ray diffraction pattern. The electron density function $\rho(r)$ in a crystal determines its diffraction pattern and conversely. The function $\rho(r)$ is a triply periodic function of the position vector r , and consequently can be expanded in a Fourier series. The coefficients of the Fourier series are the structure factors and their magnitudes are determined from the x-ray diffraction pattern. Since the structure factors are in general complex, the determination of the structure of a crystal is equivalent to finding the phases of the structure factors: the phase problem of x-ray crystallography. Hauptman [3], discusses the phase problem and shows that, due to the atomicity of crystal structures and the redundancy of observed magnitudes, the problem is, in principle, solvable. In practice, this means that for small structures the phase problem is directly solvable. However, for larger structures indirect methods are used. Indirect methods involve computing the Fourier transform and its inverse many times to calculate the structure factors from their magnitudes.

The computational cost of these calculations is enormous and many efforts have been made to reduce this cost. Since crystals exhibit symmetry leading to redundancy in the data from the diffraction pattern, it is natural to try to exploit this redundancy to reduce the cost of computing the Fourier transform. The symmetry of crystals are described by the crystallographic groups. In three dimensions there are 230 possible crystallographic groups. The introduction in the paper by Auslander and Shenefelt [4] contains a discussion of crystallographic group.

In his pioneering work on crystallographic fast Fourier transforms Ten Eyck [5] showed how symmetries from some of the crystallographic groups could be combined with the FFT to reduce the computational effort. Ten Eyck stated that his methods would not work for general crystallographic groups. Additional techniques and groups were investigated by Agarwal [6]. The papers by Bantz and Zwick [7] and Brünger [8] show how to use the symmetries of all of the crystallographic groups to reduce the amount of required memory.

In an effort to save both arithmetic operations and memory use for general crystallographic groups, Auslander and Shenfelt [4] suggested using the multiplicative algorithms of Rader [9] and Winograd [10]. This idea was further investigated by Auslander, Johnson and Vulis [11], Bricogne and Tolimieri [12], An, Cooley and Tolimieri [13] and An, Lu, Prince, and Tolimieri [14].

In this paper it is shown that the symmetries of all of the crystallographic groups can be handled with a Cooley-Tukey type FFT. The key to this result is the interpretation of the domain of the electron density function, after finite sampling, as a finite Abelian group. As a function of a finite Abelian group, the Cooley-Tukey FFT generalized to Abelian groups, can be applied. Furthermore, the crystallographic symmetries of the electron density function become affine symmetries of the finite Abelian group.

In Section 0.2 the Fourier transform of an Abelian group with symmetries is defined. In Section 0.3 a generalization of the Cooley-Tukey FFT is presented, and in Section 0.4 this generalization is modified to work with group symmetries.

0.2 Finite Fourier Transform and Group Symmetries

This section defines the Fourier transform of a finite Abelian group with symmetries. Let A be a finite Abelian group, $|A|$ denote the order of A , and $L^2(A)$ be the space of complex functions on A with inner product $f \cdot g = 1/|A| \sum_{a \in A} f(a)\overline{g(a)}$. A character of A is a homomorphism from A into \mathbf{T} , the complex numbers of absolute value one. The set of characters, \hat{A} , form a group called the dual group, with group operation $(\hat{a}_1 + \hat{a}_2)(a) = \hat{a}_1(a)\hat{a}_2(a)$. The dual group of \hat{A} is naturally isomorphic to A , and since A is finite, \hat{A} is also isomorphic to A .

Let $\langle , \rangle : A \times \hat{A} \rightarrow \mathbf{T}$ be the bilinear pairing of A with its dual, defined by $\langle a, \hat{a} \rangle = \hat{a}(a)$. The Fourier transform of A is the linear mapping, $F(A)$, from $L^2(A)$ to $L^2(\hat{A})$, which maps f to $\hat{f} = F(A)f$, where

$$\hat{f}(\hat{a}) = \sum_{a \in A} f(a) \langle a, \hat{a} \rangle, \text{ for } \hat{a} \in \hat{A}.$$

In crystallographic structure analysis, the Fourier transform is applied to functions symmetric under affine operations of A , namely functions $f(x)$ such that $f(\theta(x) + a) = f(x)$, where $\theta \in \text{Aut}(A)$, the automorphism group of A , and $a \in A$. However, the Fourier transform a function with affine symmetries introduces symmetries that may not be affine.

Let $S(a)f(x) = f(x+a)$, be the regular representation of A in $L^2(A)$ and let M be the unitary representation of \hat{A} in $L^2(A)$ defined by $M(\hat{a})f(x) = \langle x, \hat{a} \rangle f(x)$. Similarly S and M define representations of \hat{A} and A in $L^2(\hat{A})$. A fundamental property of the Fourier transform is

$$\begin{aligned} F(A)S(a) &= M(-a)F(A), \\ F(A)M(\hat{a}) &= S(\hat{a})F(A). \end{aligned}$$

This implies that the Fourier transform maps translational symmetries to multiplicative symmetries. Indeed, if $f(x+a) = f(x)$ for $x \in A$, then $\langle -a, \hat{x} \rangle \hat{f}(\hat{x}) = \hat{f}(\hat{x})$ for $\hat{x} \in \hat{A}$, where $\hat{f} = F(A)f$, and if $\langle x, \hat{a} \rangle f(x) = f(x)$ for $x \in A$ then $\hat{f}(\hat{x} + \hat{a}) = \hat{f}(\hat{x})$ for $\hat{x} \in \hat{A}$.

The operators S and M do not commute; in fact, $S(a)M(\hat{a}) = \langle a, \hat{a} \rangle M(\hat{a})S(a)$. The group generated by the operators S and M is isomorphic to the Heisenberg group of A , $N(A) = A \times \hat{A} \times T$ with multiplication given by

$$(a_1, \hat{a}_1, t_1)(a_2, \hat{a}_2, t_2) = (a_1 + a_2, \hat{a}_1 + \hat{a}_2, t_1 t_2 \langle a_1, \hat{a}_2 \rangle).$$

The Dirac representation of N on $L^2(A)$ defined by

$$D(a, \hat{a}, t)f(x) = t \langle x, \hat{a} \rangle f(x + a),$$

is equal to the group of operators generated by S and M .

To incorporate crystallographic symmetries, the Heisenberg group must be combined with $\text{Aut}(A)$. The automorphism group of \hat{A} , $\text{Aut}(\hat{A})$, is isomorphic to $\text{Aut}(A)$ by the following identification $\langle \theta(a), \hat{a} \rangle = \langle a, \hat{\theta}(\hat{a}) \rangle$. The

action of $\text{Aut}(A)$ can be extended to N by $\theta(a, \hat{a}, t) = (\theta(a), \hat{\theta}^{-1}(\hat{a}), t)$, and the semi-direct product

$$H(A) = \text{Aut}(A) \ltimes N,$$

called the *symmetry group* of A , contains all of the necessary symmetries. The Dirac representation, D , extends to a unitary representation of $H(A)$ on $L^2(A)$, by defining $D(\theta)f(x) = f(\theta^{-1}(x))$ and then

$$D(\theta)^{-1}D(a, \hat{a}, t)D(\theta) = D(\theta(a), \hat{\theta}^{-1}(\hat{a}), t).$$

The isomorphism $H(A)$ onto $H(\hat{A})$ defined by $g \mapsto \hat{g}$ where $g = (\theta, a, \hat{a}, t)$ and $\hat{g} = (\hat{\theta}^{-1}, \hat{a}, -a, t)$ is compatible with the Fourier transform $F(A)$, namely,

$$F(A)D(g) = D(\hat{g})F(A).$$

Let G be a subgroup of $H(A)$ and \hat{G} the corresponding isomorphic subgroup of $H(\hat{A})$. Then the Fourier transform of A maps G -invariant functions onto \hat{G} -invariant functions. If $L_G^2(A)$ denotes the subspace of G -invariant functions and $L_{\hat{G}}^2(\hat{A})$ denotes the subspace of \hat{G} -invariant functions, then the restriction of $F(A)$ to G -invariant functions, $F_G(A)$, maps $L_G^2(A)$ onto $L_{\hat{G}}^2(\hat{A})$.

0.3 FFT Algorithm for Abelian Groups

This section presents a generalization of the Cooley-Tukey FFT algorithm that applies to functions of finite Abelian groups. Let $B < A$ be a subgroup of A and $C = A/B$. The set of characters in \hat{A} which vanish on B , $\mathcal{A}(B) = \{\hat{a} \in \hat{A} \mid \langle b, \hat{a} \rangle = 1, \text{ for } b \in B\}$, form a subgroup of \hat{A} called the annihilator of B . The subgroup $\mathcal{A}(B)$ is isomorphic to \hat{C} and $\hat{A}/\mathcal{A}(B)$ is isomorphic to \hat{B} .

Let $\xi : C \rightarrow A$ and $\hat{\eta} : \hat{B} \rightarrow \hat{A}$ be a choices of coset representatives for A/B and $\hat{A}/\mathcal{A}(B)$. A function $f \in L^2(A)$ can be partitioned into $|C|$ *section functions*, $f_{\xi(c)} \in L^2(B)$, where $f_{\xi(c)}(b) = f(\xi(c) + b)$. The following theorem, which is a generalization of the construction in the paper of Cooley and Tukey [2], shows how to compute $F(A)$ using $|C|$ copies of $F(B)$, $|B|$ copies of $F(C)$ and $|A|$ complex multiplications. The complex multiplications depend on the choices of coset representatives ξ and $\hat{\eta}$.

Theorem 0.3.1 *Let $f \in L^2(A)$ and $\hat{f} = F(A)f$. If $B < A$, $C = A/B$, $\hat{C} = \mathcal{A}(B)$, $\hat{B} = \hat{A}/\hat{C}$, $\xi : C \rightarrow A$ be a choice of coset representatives for A/B , and $\hat{\eta} : \hat{B} \rightarrow \hat{A}$ be a choice of coset representatives for \hat{A}/\hat{C} , then*

$$\hat{f}(\hat{a}) = \sum_{c \in C} \langle c, \hat{c} \rangle \left(\langle \xi(c), \hat{\eta}(\hat{b}) \rangle \sum_{b \in B} f_{\xi(c)}(b) \langle b, \hat{b} \rangle \right), \text{ for } \hat{a} \in \hat{A},$$

where $f_{\xi(c)}(b) = f(b + \xi(c))$ and $\hat{a} = \hat{\eta}(\hat{b}) + \hat{c}$, with $\hat{b} \in \hat{B}$ and $\hat{c} \in \hat{C}$.

This theorem is the basis for a divide and conquer algorithm for computing $F(A)$, which is described in the meta-algorithm Generalized FFT. Parameters, such as A , B , C , \hat{B} , \hat{C} , ξ , and $\hat{\eta}$ that are intended to be supplied, or computed, at compile-time are called meta-variables. These variables are distinguished from variables such as f and \hat{f} that are supplied or computed at run-time. An instance of the algorithm Generalized FFT is obtained when the meta-variables are instantiated with particular groups, subgroups, quotient groups, and coset representatives. The resulting algorithm can then be used to compute \hat{f} from the input parameter f .

Algorithm Generalized FFT

Compute $F(A)$ using the generalized Cooley-Tukey FFT algorithm.

Inputs: $f \in L^2(A)$.

Output: $\hat{f} \in L^2(\hat{A})$, $\hat{f} = F(A)f$.

Meta-Variables: A is a finite Abelian group. $B < A$, $C = A/B$ and $\xi : C \rightarrow A$ is a set of coset representatives. $\hat{C} = \mathcal{A}(B)$, $\hat{B} = \hat{A}/\hat{C}$, and $\hat{\eta} : \hat{B} \rightarrow \hat{A}$ is a set of coset representatives.

1. Decomposition Phase

1a. Partition f into $|C|$ section functions $f_{\xi(c)}$ of B .

for $c \in C$

for $b \in B$

$$f_{\xi(c)}(b) = f(\xi(c) + b);$$

1b. Compute $|C|$ $F(B)$'s, $\hat{f}_{\xi(c)}$.

for $c \in C$

$$\hat{f}_{\xi(c)} = F(B)f_{\xi(c)};$$

2. Twiddle Factor Phase

Construct $|\hat{B}|$ **Intermediate Functions** $h_{\hat{\eta}(\hat{b})}$ **of** C .

for $\hat{b} \in \hat{B}$
for $c \in C$
 $h_{\hat{\eta}(\hat{b})}(c) = \langle \xi(c), \hat{\eta}(\hat{b}) \rangle \hat{f}_{\xi(c)}(\hat{b});$

3. Synthesis Phase

3a. Compute $|\hat{B}|$ $F(C)$'s, $\hat{h}_{\hat{\eta}(\hat{b})}$.

for $\hat{b} \in \hat{B}$
 $\hat{h}_{\hat{\eta}(\hat{b})} = F(C)h_{\hat{\eta}(\hat{b})};$

3b. Recover \hat{f} **from the** $|\hat{B}|$ **functions** $\hat{h}_{\hat{\eta}(\hat{b})}$ **of** \hat{C} .

for $\hat{b} \in \hat{B}$
for $\hat{c} \in \hat{C}$
 $\hat{f}(\hat{\eta}(\hat{b}) + \hat{c}) = \hat{h}_{\hat{\eta}(\hat{b})}(\hat{c});$

The straight forward computation of $F(A)$ as a matrix-vector product requires $O(|A|^2)$ operations. Assume the Generalized FFT algorithm uses the straight forward algorithm for all recursive calls. Then $O(|C||B|^2)$ operations are needed in the first phase, $O(|\hat{B}||C|)$ operations are needed in the second phase, and $O(|\hat{B}||C|^2)$ operations are needed in the third phase. Therefore, using the fact that $|\hat{B}| = |B|$ and $|A| = |B||C|$, the total number of operations is $O(|A|(|B| + |C|))$.

0.4 An Equivariant FFT Algorithm

Let f be a function of a finite Abelian group A that is invariant under the action of a subgroup G of $H(A)$, the symmetries of A . The results of Section 0.2 showed that the Fourier transform $\hat{f} = F(A)f$ is invariant under the action of \hat{G} , the isomorphic image of G in $H(\hat{A})$. In this section the symmetries in G and \hat{G} are used to reduce the amount of computation required

by the FFT algorithm in Section 0.3. This is achieved by using the actions of G and \hat{G} to obtain many of the intermediate results, which ordinarily would have been computed using recursive computation of smaller Fourier transforms. At the end of this section the complexity of this equivariant algorithm is determined and compared with direct computation of the Fourier transform taking advantage of symmetry and the FFT not taking advantage of symmetry.

Several technical definitions are required to describe the algorithm in this section. Let $g = (\theta, r, \chi, t) \in G$. The linear part of g , $\lambda(g) = (\theta, 0, 0, 1)$ and the affine part of g , $\alpha(g) = (\theta, r, 0, 1)$. The linear part can be thought of as an automorphism of A and the affine part can be thought of as an affine transformation of A . The linear part of G , $\lambda(G)$, is the set of the linear parts of the elements of G . Similarly the affine part of G , $\alpha(G)$, is the set of the affine parts of the elements of G . The linear and affine parts of G are subgroups of $H(A)$. The linear parts of G and \hat{G} are always isomorphic, but the affine parts need not be.

In order to use the group action of G on the intermediate computations in the FFT algorithm, the subgroup B must be invariant under the linear part of G . This implies that its annihilator, $\mathcal{A}(B)$, is invariant under \hat{G} . Then $\alpha(G)$ acts on $C = A/B$; for $c = a + B$, $\alpha(g)(c) = \alpha(g)(a) + B$. Similarly $\alpha(\hat{G})$ acts on $\hat{B} = \hat{A}/\mathcal{A}(B)$.

Recall that the algorithm in Section 0.3 has three stages: (1) for a set of coset representatives of A/B partition the function f into the $|C|$ section functions $f_{\xi(c)}$ of B and compute the Fourier transforms $\hat{f}_{\xi(c)} = F(B)f_{\xi(c)}$, (2) form the $|\hat{B}|$ functions $h_{\hat{\eta}(\hat{b})}$ of C by fixing \hat{b} and multiplying by the “twiddle factors”, and (3) compute the $|\hat{B}|$ Fourier transforms $\hat{h}_{\hat{\eta}(\hat{b})} = F(C)h_{\hat{\eta}(\hat{b})}$.

The equivariant algorithm is based on two key observations:

1. If γ is the affine part of $g \in G$, then $\hat{f}_{\xi(\gamma(c))}$ can be obtained from $\hat{f}_{\xi(c)}$ using a permutation and complex multiplications determined by g . Therefore $\hat{f}_{\xi(c)}$ is computed only for $c \in C/\alpha(G)$, where $\alpha(G)$ is an asymmetric unit of C under the action of $\alpha(G)$.
2. The function $f_{\xi(c)}$ is invariant under the action of G_c , a subgroup of $H(B)$ obtained from G and depending on c . Therefore, a recursive call of the equivariant algorithm can be used to compute $F_{G_c}(B)f_{\xi(c)}$.

Using an analogous argument it is possible to show that similar savings occur in stage (3) of the algorithm. In this stage, the group $\alpha(\hat{G})$ acts on $\hat{B} = \hat{A}/\mathcal{A}(B)$ to reduce the number of recursive computations of $F(C)$, and each recursive computation uses an equivariant algorithm applied to functions invariant under the action of $G_{\hat{b}}$, a subgroup of $H(C)$ obtained from \hat{G} and depending on \hat{b} .

The following theorems make these ideas more precise. The first theorem shows how to compute $F(B)f_{\gamma(\xi(c))}$ from $F(B)f_{\xi(c)}$.

Theorem 0.4.1 *Assume that G is a subgroup of $H(A)$ and f is a G -invariant function of A . If $g = (\theta, r, \chi, \omega) \in G$ and $\gamma = \alpha(g)$ the affine part of g , then*

$$\hat{f}_{\xi(\gamma(c))}(\hat{b}) = \overline{\omega\langle\theta(\xi(c)), \chi\rangle\langle b_\gamma, \hat{b}\rangle} \hat{f}_{\xi(c)}(\hat{\theta}(\hat{b} - \chi)), \text{ for } \hat{b} \in \hat{B} \text{ and } c \in C.$$

where $b_\gamma = \xi(\gamma(c)) - \gamma(\xi(c))$.

The second theorem shows how to compute the symmetry group of the section function $f_{\xi(c)}$. Let $g = (\theta, r, \chi, \omega) \in G$ and $\gamma = \alpha(g)$. Since $\alpha(G)$ acts on $C = A/B$, there exists $b_\gamma \in B$ such that $\gamma(\xi(c)) = \xi(c) + b_\gamma$. Define $g_c = (\theta, b_\gamma, \chi, \omega\langle\theta(\xi(c)), \chi\rangle)$ to be the induced symmetry of g with respect to the section function $f_{\xi(c)}$. Since $\lambda(G)(B) = B$, $b_\gamma \in B$, and χ restricted to B is a character of B , $g_c \in H(B)$. The set of induced symmetries, $G_c = \{g_c \mid g \in G \text{ and } \alpha(g)(c) = c\}$, is called the *section symmetries* corresponding to the section function $f_{\xi(c)}$.

Theorem 0.4.2 *Assume that G is a subgroup of $H(A)$ and f is a G -invariant function of A . Let G_c be the section symmetries corresponding to the section function $f_{\xi(c)}$. Then for $c \in C$, G_c is a subgroup of $H(B)$ and $f_{\xi(c)}$ is a G_c -invariant function of B .*

The next two theorems are analogs of Theorems 0.4.1 and 0.4.2 which are needed for stage (3) of the algorithm. The first analog shows how to compute $F(C)h_{\hat{\gamma}(\hat{b})}$ from $F(C)h_{\hat{\eta}(\hat{b})}$, where $\hat{\gamma} = \alpha(\hat{g})$ for $\hat{g} \in \hat{G}$. The result follows from Theorem 0.3.1, which implies that $F(C)h_{\hat{\eta}(\hat{b})}(\hat{c}) = F(A)f(\hat{\eta}(\hat{b}) + \hat{c})$ and the fact that $\hat{f} = F(A)f$ is invariant under the action of \hat{G} .

Theorem 0.4.3 *Assume that G is a subgroup of $H(A)$, f is a G -invariant function of A , $\hat{f} = F(A)f$ and \hat{G} is the isomorphic image of G in $H(\hat{A})$. Let*

$\hat{g} = (\theta, \chi, r, \omega) \in \hat{G}$ and $\hat{\gamma} = \alpha(\hat{g})$ be the affine part of \hat{g} . Then

$$\hat{h}_{\hat{\gamma}(\hat{b})}(\hat{c} - \hat{c}_\gamma) = \overline{\omega\langle r, \theta(\hat{\eta}(\hat{b})) \rangle} \langle r, \hat{c} \rangle \hat{h}_{\hat{\eta}(\hat{b})}(\theta^{-1}(\hat{c})), \text{ for } \hat{b} \in \hat{B} \text{ and } \hat{c} \in \hat{C}.$$

where $\hat{c}_\gamma = \hat{\eta}(\hat{\gamma}(\hat{b})) - \hat{\gamma}(\hat{\eta}(\hat{b}))$.

The analog to Theorem 0.4.2 is proven using duality. The dual situation arises when A is replaced by \hat{A} , G is replaced by \hat{G} , B is replaced by $\mathcal{A}(B) = \hat{C}$, $C = A/B$ is replaced by $\hat{B} = \hat{A}/\hat{C}$, ξ is replaced by $\hat{\eta}$, and f is replaced by $\hat{f} = F(A)f$. The symmetry group, $\hat{G}_{\hat{b}}$ of the section function $\hat{f}_{\hat{\eta}(\hat{b})}$ can be computed using Theorem 0.4.2. Theorem 0.3.1 implies that $F(C)h_{\hat{\eta}(\hat{b})}(\hat{c}) = \hat{f}(\hat{\eta}(\hat{b}) + \hat{c}) = \hat{f}_{\hat{\eta}(\hat{b})}(\hat{c})$, and therefore the symmetry group of $h_{\hat{\eta}(\hat{b})}$ is equal to $G_{\hat{b}}$, where $D(G_{\hat{b}}) = F(C)^{-1}D(\hat{G}_{\hat{b}})F(C)$.

Theorem 0.4.4 *Assume that G is a subgroup of $H(A)$, f is a G -invariant function of A , $\hat{f} = F(A)f$ and \hat{G} is the isomorphic image of G in $H(\hat{A})$. For $\hat{b} \in \hat{B}$, let $\hat{f}_{\hat{\eta}(\hat{b})}$ be the section function of \hat{f} corresponding to $\hat{\eta}(\hat{b})$. Let $\hat{G}_{\hat{b}}$ be the symmetry group of $\hat{f}_{\hat{\eta}(\hat{b})}$ computed by Theorem 0.4.2 and let $h_{\hat{\eta}(\hat{b})}$ be the function of C in algorithm FFT such that $\hat{f}_{\hat{\eta}(\hat{b})}(\hat{c}) = F(C)h_{\hat{\eta}(\hat{b})}(\hat{c})$. Then the symmetry group of $h_{\hat{\eta}(\hat{b})}$ is equal to $G_{\hat{b}}$, where $D(G_{\hat{b}}) = F(C)^{-1}D(\hat{G}_{\hat{b}})F(C)$.*

Theorems 0.4.1–0.4.4 lead to the following equivariant version of the generalized Cooley-Tukey FFT algorithm presented in Section 0.3. Like the algorithm FFT in Section 0.3, the algorithm Equivariant FFT, is a meta-algorithm. Additional meta-variables for the symmetry groups are required. The run-time input to the instantiated algorithm is a function $f \in L_G^2(A)$ and the output is the function $\hat{f} \in L_{\hat{G}}^2(\hat{A})$ is equal to $F_G(A)f$. Instead, of using all $|A|$ values of f to represent the input, the input function is represented by the values $f(a)$ for a in the asymmetric unit $A/\alpha(G)$. Similarly the output function \hat{f} is represented by the values $\hat{f}(\hat{a})$ for \hat{a} in the asymmetric unit $\hat{A}/\alpha(\hat{G})$. The remaining values of the functions f and \hat{f} can be computed using Theorems 0.4.1 and 0.4.3.

Algorithm Equivariant FFT

Compute $F_G(A)$ using an equivariant version of the generalized Cooley-Tukey algorithm.

Inputs: $f \in L_G^2(A)$. Only the values $f(a)$ for a in the asymmetric unit $A/\alpha(G)$ of A under the action of the affine part of G are needed.

Output: $\hat{f} \in L_{\hat{G}}^2(\hat{A})$, where $D(\hat{G}) = F(A)D(G)F(A)^{-1}$. $\hat{f} = F_G(A)f$. Only the values $\hat{f}(\hat{a})$ for \hat{a} in the asymmetric unit $\hat{A}/\alpha(\hat{G})$ of \hat{A} under the action of the affine part of \hat{G} are returned.

Meta-Variables: A is a finite Abelian group. $G < H(A)$. $B < A$ such that $\lambda(G)(B) = B$. $C = A/B$ and $\xi : C \rightarrow A$ is a set of coset representatives. $\hat{C} = \mathcal{A}(B)$, $\hat{B} = \hat{A}/\hat{C}$, and $\hat{\eta} : \hat{B} \rightarrow \hat{A}$ is a set of coset representatives. $\hat{G} < H(\hat{A})$ such that $D(\hat{G}) = F(A)D(G)F(A)^{-1}$. $G_c < H(B)$, $G_{\hat{b}} < H(C)$, $\hat{G}_{\hat{b}} < H(\hat{C})$. $D(\hat{G}_{\hat{b}}) = F(C)D(G_{\hat{b}})F(C)^{-1}$.

1. Decomposition Phase

1a. Partition f into $|C/\alpha(G)|$ section functions $f_{\xi(c)} \in L_{G_c}^2(B)$.

for $c \in C/\alpha(G)$
for $b \in B/\alpha(G_c)$
 $f_{\xi(c)}(b) = f(\xi(c) + b);$

1b. Compute $|C/\alpha(G)|$ $F_{G_c}(B)$'s, $\hat{f}_{\xi(c)}$.

for $c \in C/\alpha(G)$
 $\hat{f}_{\xi(c)} = F_{G_c}(B)f_{\xi(c)};$

2. Twiddle Factor Phase

Construct $|\hat{B}/\alpha(\hat{G})|$ intermediate functions $h_{\hat{\eta}(\hat{b})} \in L_{G_{\hat{b}}}^2(C)$.

for $\hat{b} \in \hat{B}/\alpha(\hat{G})$
for $c \in C/\alpha(G_{\hat{b}})$
 $h_{\hat{\eta}(\hat{b})}(c) = \langle \xi(c), \hat{\eta}(\hat{b}) \rangle \hat{f}_{\xi(c)}(\hat{b});$

3. Synthesis Phase

3a. Compute $|\hat{B}/\alpha(\hat{G})|$ $F_{G_{\hat{b}}}(C)$'s, $\hat{h}_{\hat{\eta}(\hat{b})}$.

for $\hat{b} \in \hat{B}/\alpha(\hat{G})$
 $\hat{h}_{\hat{\eta}(\hat{b})} = F_{G_{\hat{b}}}(C)h_{\hat{\eta}(\hat{b})};$

3b. Recover \hat{f} from the $|\hat{B}/\alpha(\hat{G})|$ functions $\hat{h}_{\hat{\eta}(\hat{b})} \in L_{\hat{G}_b}^2(\hat{C})$.

for $\hat{b} \in \hat{B}/\alpha(\hat{G})$
for $\hat{c} \in \hat{C}/\alpha(\hat{G}_b)$
 $\hat{f}(\hat{\eta}(\hat{b}) + \hat{c}) = \hat{h}_{\hat{\eta}(\hat{b})}(\hat{c});$

In addition to Theorems 0.4.1–0.4.4 several remarks are required to show that the equivariant algorithm is correct. First note that the computations of $F_{G_c}(B)$ in step (1b) and the computations of $F_{\hat{G}_b}(C)$ in step (3a) can be computed either by definition or a recursive call to the equivariant algorithm. The base case of the recursive algorithm, which has not been specified, must be computed by definition. In step (1a) the function f is accessed at $\xi(c) + b$ for $c \in C/\alpha(G)$ and $b \in B/\alpha(G_c)$ for each c . It is essential to observe that the elements $\xi(c) + b$ that are accessed are the points in $A/\alpha(G)$, and these are exactly the points required in step (1b). Similarly, the points accessed in step (3b) are those in $\hat{A}/\alpha(\hat{G})$, and these are exactly the points produced in step (3a). Finally observe that the intermediate points accessed in step (2) are exactly those needed for step (3a); however, it may be the case that some of the values $\hat{f}_{\xi(c)}(\hat{b})$ needed in step (2) have not been computed in step (1b). The points not computed in step (1b) needed in step (2) can be recovered using Theorem 0.4.1. The extra multiplications implied by Theorem 0.4.1 can be combined with the twiddle factors already present in step (2).

For the complexity analysis of the equivariant algorithm, it is assumed that the group of symmetries, G , of the input function f acts affinely. This means that G is isomorphic to $\alpha(G)$ and that \hat{G} is isomorphic to $\alpha(\hat{G})$. The consequence of this assumption that is used in the complexity analysis is that the subspace of G -invariant functions of A is isomorphic to the functions defined on the asymmetric unit $A/\alpha(G)$, and similarly the subspace of \hat{G} -invariant functions of \hat{A} is isomorphic to the functions defined on the asymmetric unit $\hat{A}/\alpha(\hat{G})$. The group G acts affinely if G has no pure translational, character, or scalar symmetry. For the application to functions with crystallographic symmetry this condition is always satisfied. For general G , a reduction to the case where G acts affinely is possible.

The assertion that the groups act affinely is preserved in the recursive calls in steps (1b) and (3a). That G_c is isomorphic to $\alpha(G_c)$ follows from the construction in Theorem 0.4.2 and the assumption that G acts affinely.

That \hat{G}_c is isomorphic to $\alpha(\hat{G}_c)$ follows from $D(\hat{G}_c) = F(C)D(G_c)F(C)^{-1}$ and the fact that \hat{G} acts affinely.

Since $L_G^2(A)$ is isomorphic to $L^2(A/\alpha(G))$ and $L_G^2(\hat{A})$ is isomorphic to $L^2(\hat{A}/\alpha(\hat{G}))$, the dimension of $L_G^2(A)$ is equal to $|A/\alpha(G)|$ and the dimension of $L_G^2(\hat{A})$ is equal to $|\hat{A}/\alpha(\hat{G})|$. Furthermore, since in general $L_G^2(A)$ is isomorphic to $L_G^2(\hat{A})$, $|A/\alpha(G)| = |\hat{A}/\alpha(\hat{G})|$. Therefore, under the assumption that G acts affinely, the algorithm which computes $F_G(A)$ by definition as a linear transformation takes time proportional to $|A/\alpha(G)|^2$. Another consequence is that $|C/\alpha(G_b)| = |\hat{C}/\alpha(\hat{G}_b)|$, and the number of points accessed in step (2) of the algorithm is equal to $|\hat{A}/\alpha(\hat{G})|$ which is the same number as in steps (1) and (3).

Assuming that G acts affinely and the equivariant FFT algorithm uses the straightforward algorithm to compute $F_{G_c}(B)$ and $F_{G_b}(C)$ the following complexity result is obtained.

Theorem 0.4.5 *The number of operations required by the equivariant FFT algorithm is proportional to*

$$\sum_{c \in C/\alpha(G)} |B/\alpha(G_c)|^2 + \sum_{\hat{b} \in \hat{B}/\alpha(\hat{G})} |C/\alpha(\hat{G}_b)|^2$$

and is less than or equal to

$$|A/\alpha(G)| \left(\max_{c \in C/\alpha(G)} (|B/\alpha(G_c)|) + \max_{\hat{b} \in \hat{B}/\alpha(\hat{G})} (|C/\alpha(\hat{G}_b)|) \right).$$

This means that the equivariant FFT reduces the size of the input function to an asymmetric unit of the action of the group of symmetries on the sample space and reduces the number of arithmetic operations to essentially the cost of the FFT applied to a function on the asymmetric unit.

Bibliography

- [1] L. Auslander, J. R. Johnson, and R. W. Johnson. Multidimensional Cooley-Tukey algorithms revisited. *Adv. in Appl. Math.*, 1996. To Appear.
- [2] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19(90):297–301, April 1965.
- [3] H. Hauptman. The phase problem of x-ray crystallography. In F. A. Grünbaum, J. W. Helton, and P. Khargonekar, editors, *Signal Processing Part II: Control Theory and Applications*, volume 23 of *The IMA Volumes in Mathematics and Its Applications*, pages 257 – 273. Springer-Verlag, New York, 1990.
- [4] L. Auslander and M. Shenefelt. Fourier transforms that respect crystallographic symmetries. *IBM J. Res. Develop.*, 31(2):213–223, March 1987.
- [5] L. F. Ten Eyck. Crystallographic Fast Fourier Transforms. *Acta Cryst.*, A29:183–191, 1973.
- [6] R. C. Agarwal. A new least-squares refinement technique based on the fast Fourier transform algorithm. *Acta Cryst.*, A34:791–809, 1978.
- [7] D. A. Bantz and M. Zwick. The use of symmetry with the fast Fourier algorithm. *Acta Cryst.*, A30:257–260, 1974.
- [8] A. T. Brünger. A memory-efficient fast Fourier transform algorithm for crystallographic refinement on supercomputers. *Acta Cryst.*, A45:42–50, 1989.

- [9] C. M. Rader. Discrete Fourier transforms when the number of data samples is prime. *Proc. IEEE.*, 56:1107–1108, 1968.
- [10] S. Winograd. *Arithmetic Complexity of Computations*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1980.
- [11] L. Auslander, R. W. Johnson, and M. Vulis. Evaluating finite Fourier transforms that respect group symmetries. *Acta Cryst.*, A44:467–478, 1988.
- [12] G. Bricogne and R. Tolimieri. Two dimensional FFT algorithms on data admitting 90°-rotational symmetry. In L. Auslander, T. Kailath, and S. Mitter, editors, *Signal Processing Part I: Signal Processing Theory*, volume 22 of *The IMA Volumes in Mathematics and Its Applications*, pages 25 – 35. Springer-Verlag, New York, 1990.
- [13] M. An, J. W. Cooley, and R. Tolimieri. Factorization method for crystallographic Fourier transforms. *Adv. in Appl. Math.*, 11:358–371, 1990.
- [14] M. An, C. Lu, E. Prince, and R. Tolimieri. Fast Fourier transforms for space groups containing rotation axes of order three and higher. *Acta Cryst.*, A48:346–349, 1992.