A Fast Newton's Algorithm for Entropy Maximization in Phase Determination*

Zhijun Wu[†], George Phillips[‡], Richard Tapia[§], and Yin Zhang[¶]

Abstract. A long-standing issue in the Bayesian statistical approach to the phase problem in X-ray crystallography is to solve an entropy maximization subproblem efficiently in every iteration of phase estimation. The entropy maximization problem is a semi-infinite convex program and can be solved in a finite dual space by using a standard Newton's method. However, the Newton's method is too expensive for this application since it requires $O(n^3)$ floating point operations per iteration, where n corresponds to the number of the phases to be estimated. Other less expensive methods have been used but they cannot guarantee fast convergence. In this paper, we present a fast Newton's algorithm for the entropy maximization problem, which uses Sherman-Morrison-Woodbury Formula and FFT to compute the Newton step and requires only $O(n \log n)$ floating point operations per iteration, yet has the same convergence rate as the standard Newton. We describe the algorithm and discuss related computational issues. Numerical results on simple test cases will also be presented to demonstrate the behavior of the algorithm.

Key Words. X-ray crystallography, protein structure determination, entropy maximization, Newton's method, Fourier transform and convolution

1 Introduction

X-ray crystallography has been one of the most productive approach to protein structure determination. X-ray crystallography determines a protein structure by the following procedure. A protein first is crystallized and an X-ray beam is applied to the crystal. The crystal scatters the X-rays and produces so-called X-ray diffractions or reflections. Different protein structures make different diffraction patterns. The X-ray crystallographers then use the recorded diffraction patterns to determine or distinguish different protein structures [13, 20, 27].

Mathematically, a protein crystal structure can be described by an electron density distribution function $\rho(r)$ where r is an arbitrary point in the crystal. An X-ray diffraction can be represented by a complex number F_H called the structure factor, where H is a three-dimensional integer vector serving as the index of the structure factor.

There is a direct mathematical relationship between the electron density distribution of the crystal and its structure factors: The electron density distribution function ρ can be expanded as a Fourier series with the structure factors F_H as the coefficients, namely,

$$\begin{aligned} \rho(r) &= \sum_H F_H \exp(-2\pi i H^T r), \\ F_H &= \int_{\mathbf{V}} \rho(r) \exp(2\pi i H^T r) dr, \end{aligned}$$

where $\mathbf{V}$ is an unit space of the crystal.

---

*Revised in March 2001.

†Department of Mathematics, Iowa State University, Ames, Iowa, 50011. Work supported in part by the Keck Center for Computational Biology and DOE/LANL Contract 03891-99-23.

‡Department of Biochemistry, University of Wisconsin-Madison, Madison, Wisconsin, 53706. Work supported in part by Welch Foundation Grant C-1142 and NSF RTG Grant BIR-94-13229.

§Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 77005. Work supported in part by DOE DEFG05-86ER25017 and DOE/LANL Contract 03891-99-23.

¶Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 77005. Work supported in part by DOE Grant DE-FG03-97ER25331 and DOE/LANL Contract 03891-99-23.

The structure factors are complex numbers, each having a magnitude and a phase. The magnitudes can be obtained by measuring the intensities of the diffraction spots since the squares of the magnitudes are proportional to the intensities of the corresponding diffraction spots. However, the diffraction image does not contain any information about the phases. In order to fully determine the electron density distribution of the crystal, the phases need to be found. Here then arises the well-known phase problem in X-ray crystallography,

> *Given the magnitudes of the structure factors, find the correct phases that define the electron density distribution function of the crystal system.*

Many efforts have been made to solve the phase problem directly. Karle and Hauptman [21] developed a method based on a nonlinear least-squares formulation of the problem. The method has been successfully applied to small molecules with less than 100 atoms (Karle and Hauptman received the Nobel prize in chemistry in 1985 for their work on the phase problem). However, the method by Karle and Hauptman has not been so successful for large molecules such as proteins. This is because that the model they used is not accurate enough for large molecules and the least-squares problem also becomes too big to solve. For these reasons, Bricogne and several others proposed a Bayesian statistical approach to the phase problem and developed methods more suitable for large molecules. For details about the Bayesian statistical approach to the phase problem, readers are referred to the original papers by Bricogne et al [2, 3, 4, 5, 6, 7, 10, 11, 12, 17, 18, 19, 28, 32] and a recent review by the authors of this paper [31].

One of the major computational components of the Bayesian statistical approach is to compute the joint probability of the structure factors and use them to determine the correct structure factors or equivalently the phases given the fact that the magnitudes of the structure factors are already known. More specifically, let $F = \{F_{H_j},\ j = 1, \ldots, n\}$ be a set of structure factors. Consider the structure factors as a set of random variables. We then want to compute the joint probability of the structure factors F_{H_j} when they are assigned to some given values $F^*_{H_j}$. The probability needs to be evaluated many times for many of the possible values assigned to the structure factors.

Based on the principles of statistical mechanics and information theory [22, 23], the joint probability of the structure factors is proportional to the maximum entropy of the crystal system given structure factors F_{H_j} in F. It follows that it suffices to find the maximum entropy of the crystal system with given structure factors.

Let ρ be a normalized electron density distribution function. Let the structures F_{H_j} in F be given as $F^*_{H_j}$. Then, the joint probability of the structure factors can be obtained by solving the following entropy maximization problem,

$$\max_\rho \quad \mathcal{S}_{\bar{m}}(\rho) \tag{1}$$

$$s.t. \quad \int_{\mathbf{V}} \rho(r) \exp(2\pi i H_j^T r) dr = F^*_{H_j}, \quad j = 1, \ldots, n \tag{2}$$

$$\int_{\mathbf{V}} \rho(r) dr = 1, \tag{3}$$

where $\mathcal{S}_{\bar{m}}$ is an entropy function, and for any ρ,

$$\mathcal{S}_{\bar{m}}(\rho) = -\int_{\mathbf{V}} \rho(r) \log[\rho(r)/\bar{m}(r)] dr, \tag{4}$$

where $\bar{m}$ is the uniform distribution function. More accurately, $\mathcal{S}_{\bar{m}}(\rho)$ is called the entropy of ρ relative to the uniform distribution $\bar{m}$, and more generally, $\bar{m}$ can be replaced by a distribution function containing more prior information about the crystal structure.

Note that the objective function for the entropy maximization problem is a concave function and the constraints are linear. Therefore, the problem must be a concave program. In fact, the objective function is even strictly concave and so the solution to the problem must also be unique.

Since it is a concave program, the entropy maximization problem can be transformed to and solved in a dual form. By following a simple procedure, we can obtain a dual problem,

$$\min_{\lambda_1, \ldots, \lambda_n} \mathcal{D}(\lambda_1, \ldots, \lambda_n), \tag{5}$$

where $\mathcal{D}$ is a convex function of $\lambda_1, \ldots, \lambda_n$. The dual problem is simpler than the original entropy maximization problem. It is an unconstrained optimization problem, and more importantly, a finite dimensional optimization problem, which can be handled much more easily than the original infinite dimensional primal problem.

Actually, the dual objective function $\mathcal{D}$ is a strictly convex function. It follows that it can be minimized by using a standard Newton's method. In general, the Newton's method requires $\mathcal{O}(n^3)$ floating point operations in every iteration, where n is the number of variables. This is too expensive given the fact that in phase estimation, the entropy maximization problem may have tens of thousands of variables and needs to be solved many times.

In this paper, we present a fast Newton's algorithm that we have developed recently for solving problem (5) or equivalently, the entropy maximization problem (1). The algorithm is equivalent to the standard Newton's method in the sense that it generates the same iterates and hence has the same convergence rate as the standard Newton. On the other hand, by using the Sherman-Morrison-Woodbury Formula and FFT, the cost per iteration is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n \log n)$ floating point operations. More specifically, the Sherman-Morrison-Woodbury Formula is used in the algorithm to compute the inverse of the Hessian of the objective function $\mathcal{D}$ in terms of the inverse of a special matrix called the Karle-Hauptman matrix in X-ray crystallography. The inverse of the Karle-Hauptman matrix can be computed by using the fast Fourier transform, which requires only $\mathcal{O}(n \log n)$ floating point operations. Then, the total cost in computing a Newton step is reduced to $\mathcal{O}(n \log n)$ floating point operations.

In Section 2, we describe the entropy maximization problem in greater detail. We discuss previous approaches in Section 3, and present our algorithm and related convergence and complexity results in Section 4. We present preliminary computational results in Section 5, and conclude the paper in Section 6.

2 Entropy Maximization

In this section, we derive the dual formulation of the entropy maximization problem. We also discuss the properties related to the dual problem. The results are not completely new, but we present them more formally and also include more expository contents for general readers.

Jaynes [22, 23] and Bricogne [2] derived a set of so-called entropy equations for solving the entropy maximization problem. This approach is based on the necessary and sufficient condition of the solution to the problem. Alhassid [1] first discussed the dual formulation of the problem. More general studies followed later in Fang [14] and Decarreau [8]. As we will show below, the dual formulation of the problem is closely related to the entropy equations.

For convenience, we write the entropy maximization problem (1) in the following more general form,

$$\begin{aligned} \max_\rho \quad & \mathcal{S}_{\bar{m}}(\rho) & (6)\\ s.t. \quad & \mathcal{F}_j(\rho) = f_j, \quad j = 1, \dots, n & (7)\\ & \mathcal{F}_0(\rho) = f_0 = 1, & (8)\end{aligned}$$

where f_j, $j = 1, \dots, n$, are complex numbers,

$$\mathcal{F}_j(\rho) = \int_{\mathbf{V}} \rho(r) C_j(r) dr, \quad j = 0, \dots, n$$

and

$$\begin{aligned} C_j(r) &= \exp(2\pi i H_j^T r), \quad j = 1, \dots, n\\ C_0(r) &= 1.\end{aligned}$$

Let $\lambda_0, \lambda_1, \dots, \lambda_n$ be a set of Lagrange multipliers, where λ_j, $j = 1, \dots, n$, are complex. Then, we can form the Lagrangian function for the problem,

$$\mathcal{L}(\rho, \lambda_0, \dots, \lambda_n) = \mathcal{S}_{\bar{m}}(\rho) + \sum_{j=0}^{n} \lambda_j \cdot [\mathcal{F}_j(\rho) - f_j], \tag{9}$$

where $\cdot$ is the inner-product of complex numbers, i.e., $(a_1 + ib_1) \cdot (a_2 + ib_2) = a_1 a_2 + b_1 b_2$.

Based on general optimization theory, a necessary condition for ρ to be a maximizer for the entropy maximization problem is that ρ must satisfy all the constraints of the problem and the partial derivative of the Lagrangian function with respect to ρ must also equal zero. Since the objective function of the problem is concave, the condition is also sufficient. We state the condition formally in the following proposition.

Proposition 2.1 *A necessary and sufficient condition for ρ to be a maximizer of problem (6) is that it must satisfy constraints (7) and (8), and there must also exist a set of parameters, $\lambda_0, \dots, \lambda_n$, such that*

$$\mathcal{L}'_\rho(\rho, \lambda_0, \dots, \lambda_n)(\delta\rho) = 0 \tag{10}$$

for all $\delta\rho \neq 0$.

It is easy to verify that the objective function for problem (6) is strictly concave. Therefore, the maximizer must also be unique.

Note that

$$\mathcal{L}'_\rho(\rho, \lambda_0, \ldots, \lambda_n) = \mathcal{S}'_{\bar{m}}(\rho) + \sum_{j=0}^{n} \lambda_j \cdot \mathcal{F}'_j(\rho).$$

Therefore, from (10), we have

$$\mathcal{S}'_{\bar{m}}(\rho)(\delta\rho) + \sum_{j=0}^{n} \lambda_j \cdot \mathcal{F}'_j(\rho)(\delta\rho) = 0.$$

Since

$$\begin{aligned}
\mathcal{S}'_{\bar{m}}(\rho)(\delta\rho) &= -\int_{\mathbf{V}} [\rho(r)\log(\rho(r)/\bar{m}(r))]'(\delta\rho(r))dr \\
&= -\int_{\mathbf{V}} [\log(\rho(r)/\bar{m}(r)) + 1](\delta\rho(r))dr
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{F}'_j(\rho)(\delta\rho) &= \int_{\mathbf{V}} [\rho(r)C_j(r)]'(\delta\rho(r))dr \\
&= \int_{\mathbf{V}} C_j(r)(\delta\rho(r))dr,
\end{aligned}$$

we then have

$$-\int_{\mathbf{V}} [\log(\rho(r)/\bar{m}(r)) + 1 - \sum_{j=0}^{n} \lambda_j \cdot C_j(r)](\delta\rho(r))dr = 0.$$

Since the integral is equal to zero for all $\delta\rho \neq 0$, the whole integrand must be equal to zero. Therefore,

$$\log[\rho(r)/\bar{m}(r)] + 1 - \sum_{j=0}^{n} \lambda_j \cdot C_j(r) = 0.$$

Solve the equation for ρ to obtain

$$\rho(r) = \bar{m}(r)\exp(\lambda_0 - 1)\exp[\sum_{j=1}^{n} \lambda_j \cdot C_j(r)].$$

Let $\lambda_0 = 1 - \log Z$ or equivalently, $Z = \exp(1 - \lambda_0)$. Then,

$$\rho(r) = \frac{\bar{m}(r)}{Z} \exp[\sum_{j=1}^{n} \lambda_j \cdot C_j(r)]. \tag{11}$$

Since ρ must also satisfy all the constraints, by (8),

$$\int_{\mathbf{V}} \rho(r) dr = \int_{\mathbf{V}} \frac{\bar{m}(r)}{Z} \exp[\sum_{j=1}^{n} \lambda_j \cdot C_j(r)] = 1.$$

We then obtain Z as a function of $\lambda_1, \ldots, \lambda_n$:

$$Z(\lambda_1, \ldots, \lambda_n) = \int_{\mathbf{V}} \bar{m}(r) \exp[\sum_{j=1}^{n} \lambda_j \cdot C_j(r)] dr. \tag{12}$$

By applying constraints (7) to ρ, we obtain the equations

$$\int_{\mathbf{V}} \frac{\bar{m}(r)}{Z(\lambda_1, \ldots, \lambda_n)} \exp[\sum_{l=1}^{n} \lambda_l \cdot C_l(r)] C_j(r) dr = f_j, \qquad j = 1, \ldots, n,$$

or more compactly,

$$\partial_j (\log Z)(\lambda_1, \ldots, \lambda_n) = f_j, \qquad j = 1, \ldots, n, \tag{13}$$

where ∂_j represents $\partial / \partial \lambda_j$. These equations are called the entropy equations and can be used to determine all the parameters $\lambda_1, \ldots, \lambda_n$. Once $\lambda_1, \ldots, \lambda_n$ are determined, ρ can be obtained from (11), and the entropy maximization problem is solved.

We now discuss the dual formulation of problem (6) and show that solving the entropy equations (13) are equivalent to solving the dual problem of the entropy maximization problem.

It follows from general optimization theory that the dual problem for problem (6) can be derived via minimizing the Lagrangian function $\mathcal{L}$ subject to the necessary condition that the partial derivative of $\mathcal{L}$ with respect to ρ is equal to zero in any feasible direction $\delta\rho$, that is,

$$\begin{aligned} &\min_{\rho, \lambda_0, \ldots, \lambda_n} \quad \mathcal{L}(\rho, \lambda_0, \ldots, \lambda_n) & (14) \\ &s.t. \quad \mathcal{L}'_\rho(\rho, \lambda_0, \ldots, \lambda_n)(\delta\rho) = 0. & (15) \end{aligned}$$

From condition (15), we obtain a function ρ in terms of $\lambda_1, \ldots, \lambda_n$ as in (11). Substitute this ρ into the objective function $\mathcal{L}$ and require ρ to be normalized to one, i.e., $\int_{\mathbf{V}} \rho(r)dr = 1$. Then, $\mathcal{L}$ becomes depending on $\lambda_1, \ldots, \lambda_n$ only:

$$\begin{aligned}
& \mathcal{L}(\rho, \lambda_0, \ldots, \lambda_n) \\
= \; & \mathcal{S}_{\bar{m}}(\rho) + \sum_{j=0}^{n} \lambda_j \cdot [\mathcal{F}_j(\rho) - f_j] \\
= \; & \mathcal{S}_{\bar{m}}(\rho) + \sum_{j=1}^{n} \lambda_j \cdot [\mathcal{F}_j(\rho) - f_j] \\
= \; & -\int_{\mathbf{V}} \rho(r) \log[\rho(r)/\bar{m}(r)]dr + \sum_{j=1}^{n} \lambda_j \cdot \int_{\mathbf{V}} \rho(r) C_j(r) dr - \sum_{j=1}^{n} \lambda_j \cdot f_j \\
= \; & \log Z - \int_{\mathbf{V}} \rho(r) \log \frac{Z\rho(r)}{\bar{m}(r)} dr + \sum_{j=1}^{n} \lambda_j \cdot \int_{\mathbf{V}} \rho(r) C_j(r) dr - \sum_{j=1}^{n} \lambda_j \cdot f_j \\
= \; & \log Z - \int_{\mathbf{V}} \rho(r) \sum_{j=1}^{n} \lambda_j \cdot C_j(r) dr + \sum_{j=1}^{n} \lambda_j \cdot \int_{\mathbf{V}} \rho(r) C_j(r) dr - \sum_{j=1}^{n} \lambda_j \cdot f_j \\
= \; & \log Z - \sum_{j=1}^{n} \lambda_j \cdot f_j.
\end{aligned}$$

It follows that the dual problem (14) can be simplified to an unconstrained minimization problem,

$$\min_{\lambda_1, \ldots, \lambda_n} \mathcal{D}(\lambda_1, \ldots, \lambda_n) = \log Z(\lambda_1, \ldots, \lambda_n) - \sum_{j=1}^{n} f_j \cdot \lambda_j. \tag{16}$$

Let G be a function and $\langle G \rangle$ the average value of G by a probability distribution ρ,

$$\langle G \rangle = \int_{\mathbf{V}} \rho(r) G(r) dr.$$

where $\int_{\mathbf{V}} \rho(r)dr = 1$. Then, it is easy to verify that

$$\begin{aligned}
\partial_j(\log Z) &= \langle C_j \rangle, \\
\partial^2_{jk}(\log Z) &= \langle C_j \overline{C}_k \rangle - \langle C_j \rangle \langle \overline{C}_k \rangle = \langle (C_j - \langle C_j \rangle) \overline{(C_k - \langle C_k \rangle)} \rangle,
\end{aligned}$$

where $\overline{C}_k$ is the complex conjugate of C_k, and $\overline{(C_k - \langle C_k \rangle)}$ the complex conjugate of $(C_k - \langle C_k \rangle)$. This implies that the Hessian of $\log Z$ is a covariance matrix of the deviation of C_j's from their averaged values.

Proposition 2.2 *The Hessian of* $\log Z$ *is the covariance matrix of the deviation of* C_j*'s from their averaged values by the probability distribution* ρ*, and*

$$\nabla^2(\log Z) = \langle (C - \langle C \rangle)(C - \langle C \rangle)^H \rangle, \tag{17}$$

where $C = (C_1, \ldots, C_m)^T$, $(C - \langle C \rangle)^H$ *is the complex conjugate of* $(C - \langle C \rangle)$*, and* $\langle\rangle$ *is taken component-wise.*

Proof. By the definition of Z in (12),

$$\begin{aligned}
\partial_j(\log Z) &= \frac{1}{Z}\partial_j Z \\
&= \int_{\mathbf{V}} \frac{\bar{m}(r)}{Z(\lambda_1 \ldots, \lambda_n)} \exp[\sum_{l=1}^{n} \lambda_l \cdot C_l(r)] C_j(r) dr \\
&= \int_{\mathbf{V}} \rho(r) C_j(r) dr \\
&= \langle C_j \rangle.
\end{aligned}$$

It follows that

$$\begin{aligned}
\partial^2_{jk}(\log Z) &= \frac{1}{Z}\partial^2_{jk} Z - \frac{1}{Z^2}\partial_j Z \partial_k Z \\
&= \langle C_j \overline{C}_k \rangle - \langle C_j \rangle \langle \overline{C}_k \rangle \\
&= \langle (C_j - \langle C_j \rangle)\overline{(C_k - \langle C_k \rangle)} \rangle.
\end{aligned}$$

The Hessian of $\log Z$ is then obtained in the form of (17). $\Box$

Proposition 2.3 *The Hessian of* $\log Z$ *is positive definite.*

Proof. Let $x = (x_1, \ldots, x_n)^T$ be a nonzero vector and x^H the complex conjugate of x.

$$\begin{aligned}
x^H \nabla^2(\log Z) x &= x^H \langle (C - \langle C \rangle)(C - \langle C \rangle)^H \rangle x \\
&= \langle x^H (C - \langle C \rangle)(C - \langle C \rangle)^H x \rangle \\
&= \langle |x^H (C - \langle C \rangle)|^2 \rangle \\
&\geq 0.
\end{aligned}$$

Assume that the equality holds for some x,

$$x^H \nabla^2 (\log Z) x = \langle |x^H (C - \langle C \rangle)|^2 \rangle = 0.$$

We then have

$$x^H (C - \langle C \rangle) = 0.$$

Given the fact that $C_j \neq \langle C_j \rangle$ and $C_j - \langle C_j \rangle$ are linearly independent of each other, x must be equal to zero, contradicting to the assumption that x be a nonzero vector. Therefore,

$$x^H \nabla^2 (\log Z) x = \langle |x^H (C - \langle C \rangle)|^2 \rangle > 0,$$

and $\nabla^2 (\log Z)$ is positive definite. $\square$

Corollary 2.1 *The Hessian of the function $\mathcal{D}$ is positive definite and therefore, $\mathcal{D}$ is a strictly convex function.*

Proof. Note that $\nabla^2 \mathcal{D} = \nabla^2 (\log Z)$. The corollary then follows immediately from Proposition 2.2 and 2.3. $\square$

Corollary 2.2 *The entropy equations (13) are equivalent to the necessary and sufficient conditions for the solution to the dual problem (16). Therefore, solving the entropy equations (13) is equivalent to solving the dual problem (16).*

Proof. Since $\mathcal{D}$ is a strictly convex function, a necessary and sufficient condition for some $\lambda_1, \ldots, \lambda_n$ to be the minimizer of the function is that

$$\partial_j \mathcal{D}(\lambda_1, \ldots, \lambda_n) = 0, \qquad j = 1, \ldots, n.$$

The corollary then follows immediately from the fact that

$$\partial_j \mathcal{D}(\lambda_1 \ldots, \lambda_n) = \partial_j (\log Z)(\lambda_1, \ldots, \lambda_n) - f_j, \qquad j = 1, \ldots, n.$$

$\square$

Finally we show an important relationship between the entropy maximization problem (6) and its dual problem (16).

Proposition 2.4 *Let ρ be a feasible solution to the entropy maximization problem (6). Then, for any $\lambda = (\lambda_1, \ldots, \lambda_n)^T$,*

$$\mathcal{S}_{\bar{m}}(\rho) \leq \mathcal{D}(\lambda). \tag{18}$$

Moreover, if ρ is the maximizer of problem (6) and λ the minimizer of problem (16), the equality will hold, and vice versa.

Proof. Given $\lambda = (\lambda_1, \ldots, \lambda_n)^T$, let $\lambda_0 = 1 - \log Z(\lambda_1, \ldots, \lambda_n)$ and

$$\rho_\lambda(r) = \frac{\bar{m}(r)}{Z(\lambda_1, \ldots, \lambda_n)} \exp[\sum_{j=1}^{n} \lambda_j \cdot C_j(r)].$$

Let ρ be a feasible solution for problem (6). Since $\mathcal{S}_{\bar{m}}$ is concave,

$$\mathcal{S}_{\bar{m}}(\rho) \quad \leq \quad \mathcal{S}_{\bar{m}}(\rho_\lambda) + \mathcal{S}'_{\bar{m}}(\rho_\lambda)(\rho - \rho_\lambda).$$

By the fact that $\mathcal{L}'_\rho(\rho_\lambda, \lambda_0, \ldots, \lambda_n)(\rho - \rho_\lambda) = 0$,

$$\mathcal{S}'_{\bar{m}}(\rho_\lambda)(\rho - \rho_\lambda) = -\sum_{j=0}^{n} \lambda_j \cdot \mathcal{F}'_j(\rho_\lambda)(\rho - \rho_\lambda).$$

Since $\mathcal{F}_j$ are linear and ρ is feasible,

$$\mathcal{F}'_j(\rho_\lambda)(\rho - \rho_\lambda) = \mathcal{F}_j(\rho) - \mathcal{F}_j(\rho_\lambda) = f_j - \mathcal{F}_j(\rho_\lambda).$$

By the definition of ρ_λ,

$$\mathcal{S}_{\bar{m}}(\rho_\lambda) + \sum_{j=0}^{n} \lambda_j \cdot [\mathcal{F}_j(\rho_\lambda) - f_j] = \log Z(\lambda_1, \ldots, \lambda_n) - \sum_{j=1}^{n} f_j \cdot \lambda_j.$$

It then follows that

$$\begin{aligned}
\mathcal{S}_{\bar{m}}(\rho) &\leq \mathcal{S}_{\bar{m}}(\rho_\lambda) + \mathcal{S}'_{\bar{m}}(\rho_\lambda)(\rho - \rho_\lambda) \\
&= \mathcal{S}_{\bar{m}}(\rho_\lambda) - \sum_{j=0}^{n} \lambda_j \cdot \mathcal{F}'_j(\rho_\lambda)(\rho - \rho_\lambda) \\
&= \mathcal{S}_{\bar{m}}(\rho_\lambda) + \sum_{j=0}^{n} \lambda_j \cdot [\mathcal{F}_j(\rho_\lambda) - \mathcal{F}_j(\rho)] \\
&= \mathcal{S}_{\bar{m}}(\rho_\lambda) + \sum_{j=0}^{n} \lambda_j \cdot [\mathcal{F}_j(\rho_\lambda) - f_j] \\
&= \log Z(\lambda_1, \ldots, \lambda_n) - \sum_{j=1}^{n} f_j \cdot \lambda_j \\
&= D(\lambda).
\end{aligned}$$

The inequality (18) is proved.

Let ρ^* be the maximizer of problem (6) and $\lambda^* = (\lambda_1^*, \ldots, \lambda_n^*)^T$ the minimizer of problem (16). We now show that $\mathcal{S}_{\bar{m}}(\rho^*) = \mathcal{D}(\lambda^*)$.

Since ρ^* is a maximizer of problem (6), there must be a set of Lagrange multipliers, $\lambda_0, \lambda_1, \ldots, \lambda_n$, such that

$$\rho^*(r) = \frac{\bar{m}(r)}{Z(\lambda_1, \ldots, \lambda_n)} \exp[\sum_{j=1}^{n} \lambda_j \cdot C_j(r)].$$

It then follows that

$$\begin{aligned}
\mathcal{S}_{\bar{m}}(\rho^*) &= -\int_{\mathbf{V}} \rho^*(r) \log[\rho^*(r)/\bar{m}(r)] dr \\
&= \log Z(\lambda_1, \ldots, \lambda_n) - \int_{\mathbf{V}} \rho^*(r) \sum_{j=1}^{n} \lambda_j \cdot C_j(r) dr \\
&= \log Z(\lambda_1, \ldots, \lambda_n) - \sum_{j=1}^{n} f_j \cdot \lambda_j \\
&= \mathcal{D}(\lambda).
\end{aligned}$$

Since $\mathcal{D}(\lambda) \geq \mathcal{D}(\lambda^*)$ for all λ, $\mathcal{S}_{\bar{m}}(\rho^*) \geq \mathcal{D}(\lambda^*)$. However, by (18), $\mathcal{D}(\lambda^*) \geq \mathcal{S}_{\bar{m}}(\rho^*)$. Therefore, $\mathcal{S}_{\bar{m}}(\rho^*) = \mathcal{D}(\lambda^*)$.

On the other hand, suppose that the equality of (18) holds for some ρ^* and λ^*. We show that ρ^* must be the maximizer of problem (6) and λ^* the minimizer of problem (16).

Since $\mathcal{S}_{\bar{m}}(\rho) \leq \mathcal{D}(\lambda)$ for any feasible ρ and λ, there cannot be $\rho \neq \rho^*$ or $\lambda \neq \lambda^*$ such that

$$\mathcal{S}_{\bar{m}}(\rho) > \mathcal{S}_{\bar{m}}(\rho^*) = \mathcal{D}(\lambda^*) > \mathcal{D}(\lambda).$$

Therefore, ρ^* must be optimal for (6) and λ^* for (16). $\square$

3 Previous Approaches

Since the objective function of (16) is strictly convex, the problem can be solved by using a standard Newton's method. The method consists of an iterative procedure. In each iteration, a Newton's step is computed to update the current iterate to obtain a new iterate. More specifically, if

$\lambda^{(l)} = (\lambda_1^{(l)}, \ldots, \lambda_n^{(l)})^T$ is the current iterate, the next iterate $\lambda^{(l+1)}$ is computed by an iterative procedure,

$$\lambda^{(l+1)} = \lambda^{(l)} - [\nabla^2 \mathcal{D}(\lambda^{(l)})]^{-1} \nabla \mathcal{D}(\lambda^{(l)}). \tag{19}$$

If we repeat the procedure, we will obtain a sequence of iterates. Since the objective function is strictly convex, as a standard result, the sequence will converge to the minimizer of the function, and also converge quadratically when the iterate is sufficiently close to the minimizer. However, the Newton's method is costly in general in the sense that in each iteration, it requires $\mathcal{O}(n^3)$ floating point operations, in order to form the Newton step, i.e., the inverse of the Hessian times the negative gradient of the objective function.

The Newton's method (19) converges to the solution to problem (16) quadratically, but in general, requires $\mathcal{O}(n^3)$ floating point operations in each iteration.

In the applications in X-ray crystallography, the entropy maximization problem often is large in the sense that when transformed to problem (16), n often is in the order of tens of thousands. A straightforward implementation of the Newton's method will not be practical, especially when the problem needs to be solved many times, as required by the Bayesian statistical approach to structural determination.

In order to reduce the cost of the Newton's method, Bricogne [2] suggested to approximate the Hessian of $\mathcal{D}$ so that the inverse of the approximated Hessian can be computed in a cheaper way. More specifically, as we will show later, the Hessian of $\mathcal{D}$ is actually equal to a special matrix K minus a matrix FF^H for some vector F. This matrix K is known as the Karle-Hauptman matrix in X-ray crystallography [24]. Bricogne [2] suggested to use this K matrix as an approximation to the Hessian of $\mathcal{D}$ since the inverse of K can be computed by using fast Fourier transform, which costs only $\mathcal{O}(n \log n)$ floating point operations. However, the fast convergence property of the original Newton's method is no longer guaranteed by this approximation. As a trade-off between fast convergence and low computational cost, an ad-hoc approach is taken in practice: For example, in BUSTER [5], the Newton's method is used only for small problems, and is switched to the approximation method when a large problem is considered.

Other approaches have also been taken to avoid the high cost of the Newton's method such as using the BFGS method [26]. The BFGS method does not compute the inverse of the Hessian, which actually is required in practice, however. Therefore, it is still an issue how to compute the inverse of the Hessian with less cost than $\mathcal{O}(n^3)$.

4 A Fast Newton's Algorithm

Here we present a fast Newton's algorithm for solving the entropy maximization problem. The algorithm requires only $\mathcal{O}(n \log n)$ floating point operations for each of its iterates, yet has the same convergence rate as the Newton's method.

First, consider the Newton's iteration (19). Note that the major computation in the iteration is the Newton step,

$$-(\nabla^2 \mathcal{D})^{-1} \nabla \mathcal{D}. \tag{20}$$

Based on the previous discussion, the gradient and the Hessian of $\mathcal{D}$ can be computed in the following form.

$$\begin{aligned} \nabla_j \mathcal{D} &= \nabla_j (\log Z) - f_j = \langle C_j \rangle - f_j, \\ \nabla^2_{jk} \mathcal{D} &= \nabla^2_{jk} (\log Z) = \langle C_j \overline{C_k} \rangle - \langle C_j \rangle \langle \overline{C_k} \rangle. \end{aligned}$$

By the definition, $\langle C_j \rangle$ is a Fourier transform of ρ and corresponds to a structure factor F_{H_j}:

$$\langle C_j \rangle = \mathcal{F}(\rho) = \int_{\mathbf{V}} \rho(r) C_j(r) dr = F_{H_j}.$$

Let f_j be equal to given structure factors $F^*_{H_j}$. We can write the gradient and the Hessian in terms of the structure factors,

$$\begin{aligned} \nabla_j \mathcal{D} &= F_{H_j} - F^*_{H_j}, \\ \nabla^2_{jk} \mathcal{D} &= F_{H_j - H_k} - F_{H_j} F_{-H_k}. \end{aligned}$$

In a more compact form,

$$\begin{aligned} \nabla \mathcal{D} &= F - F^*, \\ \nabla^2 \mathcal{D} &= K - F F^H, \end{aligned}$$

where $F = (F_{H_1}, \ldots, F_{H_n})^T$, $F^* = (F^*_{H_1}, \ldots, F^*_{H_n})^T$, and K is a matrix such that $K_{jk} = F_{H_j - H_k}$. In practice, $H_1, \ldots, H_n$ can be selected such that $F_{H_j - H_k}$ changes periodically, and $K_{\cdot k}$ and $K_{\cdot l}$ are simply permutations of each other for different k and l. A matrix K with these characteristics is called a Karle-Hauptman matrix [24].

Because of the special structure of the matrix K, its inverse can be computed by using fast Fourier transform with less computational cost than conventional matrix factorization methods. On the other hand, since $\nabla^2 \mathcal{D}$ is equal to $K - FF^H$, by using the Sherman-Morrison-Woodbury Formula, the inverse of $\nabla^2 \mathcal{D}$ can be computed in terms of the inverse of K. In other words, in order to reduce the cost for computing the Newton step, we can use the Sherman-Morrison-Woodbury Formula to reformulate the Newton step in terms of the inverse of K. We will then be able to obtain a faster algorithm for solving the entropy maximization problem.

The Sherman-Morrison-Woodbury Formula usually applies to nonsingular matrices (see [9, 15, 29]). For our purpose, we give a more specific version of the formula for positive definite matrices.

Theorem 4.1 *Let T and S be two Hermite matrices, U a vector, and*

$$T = S - UU^H. \tag{21}$$

Let S be a nonsingular matrix and $\sigma = U^H S^{-1} U$. Then T is positive definite if and only if S is positive definite and $\sigma < 1$. Furthermore,

$$T^{-1} = S^{-1} + \frac{S^{-1} U U^H S^{-1}}{1 - \sigma}. \tag{22}$$

Proof. We first show that if T is positive definite, S is positive definite and $\sigma < 1$, and the inverse of T can be computed by (22).

If T is positive definite, $S = T + UU^H$ must be positive definite. Then, T^{-1} and S^{-1} exist and are also positive definite. Multiply (21) by S^{-1} from left and by T^{-1} from right to obtain

$$S^{-1} = T^{-1} - S^{-1} U U^H T^{-1}. \tag{23}$$

It follows that

$$U^H S^{-1} U = (1 - U^H S^{-1} U) U^H T^{-1} U > 0,$$

and $U^H S^{-1} U = \sigma$ must be less than 1.

From (23), we have

$$T^{-1} = S^{-1} + S^{-1} U U^H T^{-1}. \tag{24}$$

Note that in the second part of the formula, if we substitute T^{-1} recursively,

$$\begin{aligned} UU^H T^{-1} &= UU^H (S^{-1} + S^{-1} UU^H T^{-1}) \\ &= UU^H S^{-1} + UU^H S^{-1} UU^H T^{-1} \\ &= UU^H S^{-1} + U^H S^{-1} UUU^H T^{-1} \\ &= UU^H S^{-1} + \sigma UU^H T^{-1}. \end{aligned}$$

Therefore,

$$UU^H T^{-1} = \frac{UU^H S^{-1}}{1 - \sigma}. \tag{25}$$

Substitute (25) into (24) to obtain

$$T^{-1} = S^{-1} + \frac{S^{-1} UU^H S^{-1}}{1 - \sigma}.$$

We now show that if S is positive definite and $\sigma < 1$, T must be positive definite, and the same formula (22) for the inverse of T follows.

If S is positive definite and $\sigma < 1$, we can construct a matrix,

$$T' = S^{-1} + \frac{S^{-1} UU^H S^{-1}}{1 - \sigma}.$$

It is easy to verify that T' is positive definite and

$$T'T = T'(S - UU^H) = TT' = (S - UU^H)T' = I.$$

Therefore, T' is the inverse of T and T must be positive definite. $\square$

By applying the Sherman-Morrison-Woodbury Formula (22) to $\nabla^2 \mathcal{D}$, we obtain the following results.

Proposition 4.1 *Let $\nabla^2 \mathcal{D} = K - FF^H$, where K is a Karle-Hauptman matrix. Then,*

$$0 < F^H K^{-1} F < 1. \tag{26}$$

Proof. The result follows directly from Theorem 4.1 and the fact that $\nabla^2\mathcal{D}$ is positive definite. $\Box$

Proposition 4.2 *Let $\nabla^2\mathcal{D} = K - FF^H$, where K be a Karle-Hauptman matrix. Then, the inverse of $\nabla^2\mathcal{D}$ can be computed by the following formula.*

$$(\nabla^2\mathcal{D})^{-1} = K^{-1} + \frac{K^{-1}FF^HK^{-1}}{1-\sigma}, \tag{27}$$

where $\sigma = F^HK^{-1}F$.

Proof. Since $\nabla^2\mathcal{D}$ is positive definite, K is also positive definite and $\sigma < 1$ by Theorem 4.1. The formula can then be derived by using (22) for $\nabla^2\mathcal{D}$. $\Box$

By using formula (27), we can write the Newton step (20) in the following form.

$$-(\nabla^2\mathcal{D})^{-1}\nabla\mathcal{D} = -(K - FF^H)^{-1}(F - F^*) = V + \frac{F^HV}{1 - F^HU}U \tag{28}$$

where

$$V = K^{-1}(F^* - F), \qquad U = K^{-1}F. \tag{29}$$

In this formula, the computation for the Newton step is reduced to computing the inverse of K times a vector. The inverse of K and its product with a vector can all be computed by using fast Fourier transform, which we verify later in the section.

Figure 1 contains an outline of the fast Newton's algorithm constructed by using the above formulas. In the first step, an initial guess $\lambda^{(0)}$ is given and l is set to zero. Then the step 2 is repeated. First, in step 2a, given $\lambda_1^{(l)}, \ldots, \lambda_n^{(l)}$, the corresponding $\rho^{(l)}$ and $Z(\lambda^{(l)})$ are computed; they can be computed together through an inverse Fourier transform, which requires only $\mathcal{O}(n \log n)$ floating point operations. In step 2b, another Fourier transform is applied to $\rho^{(l)}$ to obtain the structure factors $F_j^{(l)}$, and the cost is again in the order of $n \log n$. In step 2c, two Fourier transforms are required for two matrix-vector products, $[K^{(l)}]^{-1}\Delta^{(l)}$ and $[K^{(l)}]^{-1}F^{(l)}$. Finally, in step 2d, the Newton step is formed with the Sherman-Morrison-Woodbury Formula, and a new iterate $\lambda^{(l+1)}$ is obtained. The whole step 2d requires only vector-vector operations and costs $\mathcal{O}(n)$ floating point operations. In the end of the iteration, if the new iterate is optimal, the algorithm stops; otherwise, the iteration continues until it converges. In any case, each iteration requires only $\mathcal{O}(n \log n)$ floating point operations.

> *The fast Newton's algorithm converges to the solution to problem (16) quadratically, and in each iteration, requires only $\mathcal{O}(n \log n)$ floating point operations.*

To complete the section, we verify the facts that the inverse of K as well as the matrix-vector products $K^{-1}(F^* - F)$ and $K^{-1}F$ can all be obtained through certain forms of Fourier transforms.

Proposition 4.3 *Let K be a Karle-Hauptman matrix, $K_{jk} = F_{H_j - H_k}$, and for all k,*

$$\begin{aligned} F_{H_j-H_k} &= \int_{\mathbf{V}} \rho(r) \exp[2\pi i (H_j - H_k)^T r] dr, \ j = 1, \ldots, n, && (30) \\ \rho(r) &= \sum_{j=1}^{n} F_{H_j-H_k} exp[-2\pi i (H_j - H_k)^T r]. && (31) \end{aligned}$$

Then K^{-1} can be obtained with $[K^{-1}]_{jk} = E_{H_j-H_k}$, and for all k

$$\begin{aligned} E_{H_j-H_k} &= \int_{\mathbf{V}} \rho^{-1}(r) \exp[2\pi i (H_j - H_k)^T r] dr, \ j = 1, \ldots, n, && (32) \\ \rho^{-1}(r) &= \sum_{j=1}^{n} E_{H_j-H_k} exp[-2\pi i (H_j - H_k)^T r]. && (33) \end{aligned}$$

Proof. Let $L = KK^{-1}$. We show that L is an identity matrix. By the definitions of K and K^{-1}, we have

$$\begin{aligned} L_{jk} &= \sum_{l=1}^{n} F_{H_j-H_l} E_{H_l-H_k} \\ &= \sum_{l=1}^{n} E_{H_l-H_k} \int_{\mathbf{V}} \rho(r) \exp[2\pi i (H_j - H_l)^T r] dr \\ &= \int_{\mathbf{V}} \rho(r) \exp[2\pi i (H_j - H_k)^T r] \\ & \quad \sum_{l=1}^{n} E_{H_l-H_k} \exp[-2\pi i (H_l - H_k)^T r] dr \end{aligned}$$

The Fast Newton's Algorithm

1. Input initial $\lambda^{(0)}$. Set $l = 0$.
2. Repeat
   (a) Compute
   $$\begin{aligned} Z(\lambda^{(l)}) &= \int_{\mathbf{V}} \bar{m}(r) \exp[\sum_{j=1}^{n} \lambda_j^{(l)} C_j(r)] dr \\ \rho^{(l)}(r) &= \frac{\bar{m}(r)}{Z(\lambda^{(l)})} \exp[\sum_{j=1}^{n} \lambda_j^{(l)} C_j(r)] \end{aligned}$$
   (b) Compute, for $j = 1, \ldots, n$,
   $$F_j^{(l)} = \int_{\mathbf{V}} \rho^{(l)}(r) \exp(2\pi i H_j^T r) dr$$
   (c) Set
   $$\begin{aligned} \Delta^{(l)} &= F^* - F^{(l)} \\ V^{(l)} &= [K^{(l)}]^{-1} \Delta^{(l)} \\ U^{(l)} &= [K^{(l)}]^{-1} F^{(l)} \end{aligned}$$
   (d) Compute
   $$\begin{aligned} \Delta\lambda^{(l)} &= V^{(l)} + \frac{[F^{(l)}]^H V^{(l)}}{1 - [F^{(l)}]^H U^{(l)}} U^{(l)} \\ \lambda^{(l+1)} &= \lambda^{(l)} + \Delta\lambda^{(l)} \\ l &= l + 1 \end{aligned}$$
   (e) If the optimality condition is satisfied, go to 3.
3. Set $\lambda^* = \lambda^{(l)}$, $\rho^* = \rho^{(l)}$. Stop.

Figure 1: Outline of the fast Newton's algorithm

$$
\begin{aligned}
&= \int_{\mathbf{V}} \rho(r) \exp[2\pi i (H_j - H_k)^T r] \rho^{-1}(r) dr \\
&= \int_{\mathbf{V}} \exp[2\pi i (H_j - H_k)^T r] dr.
\end{aligned}
$$

It is easy to see that $L_{jk} = 0$ if $j \neq k$ and $L_{jk} = 1$ if $j = k$. L is indeed an identity matrix. The same result can be obtained for $L = K^{-1}K$. $\Box$

Proposition 4.4 *Let K be a Karle-Hauptman matrix, $K_{jk} = F_{H_j - H_k}$, and for all k,*

$$
\begin{aligned}
F_{H_j - H_k} &= \int_{\mathbf{V}} \rho(r) \exp[2\pi i (H_j - H_k)^T r] dr, \; j = 1, \ldots, n, \qquad (34) \\
\rho(r) &= \sum_{j=1}^{n} F_{H_j - H_k} exp[-2\pi i (H_j - H_k)^T r]. \qquad (35)
\end{aligned}
$$

Let $U = K^{-1}F$. Then

$$
U_l = \int_{\mathbf{V}} \rho^{-1}(r) \bar{\rho}(r) \exp[2\pi i H_l^T r] dr, \qquad (36)
$$

where

$$
\bar{\rho}(r) = \sum_{j=1}^{n} F_{H_j} \exp(-2\pi i H_j^T r). \qquad (37)
$$

Proof. We show that $U_l = \sum_{j=1}^{n} [K^{-1}]_{lj} F_{H_j}$. By Proposition 4.3 and the definition of $\bar{\rho}(r)$,

$$
\begin{aligned}
U_l &= \int_{\mathbf{V}} \rho^{-1}(r) \bar{\rho}(r) \exp[2\pi i H_l^T r] dr \\
&= \int_{\mathbf{V}} \rho^{-1}(r) \sum_{j=1}^{n} F_{H_j} \exp(-2\pi i H_j^T r) \exp[2\pi i H_l^T r] dr \\
&= \sum_{j=1}^{n} F_{H_j} \int_{\mathbf{V}} \rho^{-1}(r) \exp[2\pi i (H_l - H_j)^T r] dr \\
&= \sum_{j=1}^{n} E_{H_l - H_j} F_{H_j} \\
&= \sum_{j=1}^{n} [K^{-1}]_{lj} F_{H_j}.
\end{aligned}
$$

$\Box$

Proposition 4.5 *Let K be a Karle-Hauptman matrix, $K_{jk} = F_{H_j - H_k}$, and for all k,*

$$F_{H_j-H_k} = \int_{\mathbf{V}} \rho(r) \exp[2\pi i (H_j - H_k)^T r] dr, \; j = 1, \ldots, n, \tag{38}$$

$$\rho(r) = \sum_{j=1}^{n} F_{H_j-H_k} exp[-2\pi i (H_j - H_k)^T r]. \tag{39}$$

Let $V = K^{-1}(F^* - F)$. Then

$$V_l = \int_{\mathbf{V}} \rho^{-1}(r) \bar{\rho}(r) \exp[2\pi i H_l^T r] dr, \tag{40}$$

where

$$\bar{\rho}(r) = \sum_{j=1}^{n} (F_{H_j}^* - F_{H_j}) \exp(-2\pi i H_j^T r) dr. \tag{41}$$

Proof. Similar to the previous proposition, we show that

$$V_l = \sum_{j=1}^{n} [K^{-1}]_{lj} (F_{H_j}^* - F_{H_j}).$$

By Proposition 4.3 and the definition of $\bar{\rho}(r)$,

$$\begin{aligned}
V_l &= \int_{\mathbf{V}} \rho^{-1}(r) \bar{\rho}(r) \exp[2\pi i H_l^T r] dr \\
&= \int_{\mathbf{V}} \rho^{-1}(r) \sum_{j=1}^{n} (F_{H_j}^* - F_{H_j}) \exp(-2\pi i H_j^T r) \exp[2\pi i H_l^T r] dr \\
&= \sum_{j=1}^{n} (F_{H_j}^* - F_{H_j}) \int_{\mathbf{V}} \rho^{-1}(r) \exp[2\pi i (H_l - H_j)^T r] dr \\
&= \sum_{j=1}^{n} E_{H_l - H_j} (F_{H_j}^* - F_{H_j}) \\
&= \sum_{j=1}^{n} [K^{-1}]_{lj} (F_{H_j}^* - F_{H_j}).
\end{aligned}$$

□

Finally, from a matrix computation point of view, we note that the Karle-Hauptman matrix K and K^{-1} are basically circulant matrices, and the results in Proposition 4.4 and 4.5 can therefore be obtained in terms of the properties of circulant matrices as well.

Let $A = (A_1, \ldots, A_n)^T$ be a vector. Then, a circulant matrix $C(A)$ is a matrix with its first column equal to $(A_1, A_2 \ldots, A_n)^T$, the second $(A_n, A_1, \ldots, A_{n-1})^T$, the third $(A_{n-1}, A_n, \ldots, A_{n-2})^T$, ...

Let $A = (A_1, \ldots, A_n)^T$ and $B = (B_1, \ldots, B_n)^T$ be two vectors. A discrete convolution of A and B, denoted by $*$, is defined as

$$A * B = C(A)B, \tag{42}$$

where $C(A)$ is a circulant matrix generated by A.

It follows that the product of a circulant matrix and a vector is a discrete convolution. Based on discrete convolution theory, a discrete convolution of two vectors can be computed via discrete Fourier transform. Let x be a vector and $FT(x)$ the discrete Fourier transform of x.

Theorem 4.2 *Let $A = (A_1, \ldots, A_n)^T$, $B = (B_1, \ldots, B_n)^T$, $a = (a_1, \ldots, a_n)^T$, $b = (b_1, \ldots, b_n)^T$, and $A = FT(a)$ and $B = FT(b)$. Then,*

$$A * B = FT(a) * FT(b) = FT(a \cdot b), \tag{43}$$

where $a \cdot b = (a_1 b_1, \ldots, a_n b_n)^T$ is the component-wised product of a and b.

Proof: See standard text books on discrete Fourier transform and convolution, for example, [25] and [30]. □

By applying this theorem to matrix K^{-1}, we obtain the formula for computing the product of K^{-1} and a vector X: Let $X = FT(\delta)$ for some δ. Since $K^{-1} = C(E)$, where $E = FT(\rho^{-1})$. Then,

$$K^{-1}X = E * X = FT(\rho^{-1}) * FT(\delta) = FT(\rho^{-1} \cdot \delta). \tag{44}$$

It follows that the product of K^{-1} and X can be obtained with fast Fourier transform, which requires only $\mathcal{O}(n \log n)$ floating point operations.

5 Preliminary Computational Results

We have implemented the fast Newton's algorithm in Matlab and compared it with several other algorithms including a gradient algorithm, a standard

Newton's algorithm, and an algorithm with Hessian approximation. By Hessian approximation we mean that we use K instead of $K - FF^H$ as an approximation to the Hessian of the function $\mathcal{D}$.

We have tested the algorithms with a set of model problems generated in the following procedure. We first constructed a one-dimensional density distribution function. We then generated five sets of structure factors from the function. The first set has 8 structure factors, the second 16, the third 32, the fourth 64, and the fifth 128. From each set of structure factors we define an entropy maximization problem with the corresponding structure factors as the constraints. We then obtain five entropy maximization problems.

We applied the algorithms to the model problems and recorded the number of iterations required for the algorithm to converge to the solution to the problem. We also recorded the total number of floating point operations for each run.

Note that we used the Matlab routine for the fast Fourier transform required in the algorithms. We also used the Matlab routine for the linear system solve in the standard Newton's algorithm. In each iteration, the gradient algorithm needs to compute the function and the gradient. Given $\lambda_1, \ldots, \lambda_n$, the function and the gradient of $\mathcal{D}$ can both be obtained by doing some fast Fourier transforms. Therefore, the cost of each iteration in the gradient algorithm is $\mathcal{O}(n \log n)$. However, the gradient algorithm converges only linearly and may take too many iterations to reach a solution.

When the Hessian is approximated by K, the inverse of the Hessian can be computed in $\mathcal{O}(n \log n)$. So the approximation algorithm should take the same order of floating point operations as the fast Newton's algorithm. However, because of the approximation, the algorithm is no longer a Newton's algorithm, and therefore, the fast convergence rate of the Newton's algorithm will be lost.

Table 1, 2, and 3 showed the performance of the algorithms in terms of the number of iterations or the total number of floating point operations. Table 1 showed the numbers of iterations required for the gradient algorithm. Clearly, it took many more iterations to converge than the fast Newton's algorithm, which for all the test problems, took only 6 iterations.

Table 2 showed the number of iterations required by the approximation algorithm. For all the problem instances, the algorithm took 2 to 3 times more iterations than the fast Newton's algorithm. Since the test problems we have constructed are relatively simple and easy, we expect the difference between the two algorithms to be bigger in practice when the problem is larger and more complicated. In any case, there is no theory to guarantee the approximation algorithm to converge fast.

Table 3 showed the total numbers of floating point operations required by the standard and fast Newton's algorithms. As we can see from the table, the standard Newton's algorithm required much more floating point operations than the fast Newton's algorithm. For example, for the test problem with 128 structure factors, the fast Newton's algorithm required about 200 thousand floating point operations, while the standard Newton's algorithm did more than 13 million floating point operations. The fast Newton's algorithm was more than 60 times faster. When the problem size becomes larger, we expect the difference to be even bigger.

Table 1: Comparison with a Gradient Algorithm (# Iterations)

# Factors:	8	16	32	64	128
Gradient:	65	68	68	67	65
Fast Newton:	6	6	6	6	6

Table 2: Comparison with an Approximation Algorithm (# Iterations)

# Factors:	8	16	32	64	128
Approximation:	18	16	16	14	14
Fast Newton:	6	6	6	6	6

6 Concluding Remarks

In this paper, we studied the entropy maximization problem in the Bayesian statistical approach to the phase problem in protein X-ray crystallography. Since the solution to the problem is required in every step of the Bayesian method, an efficient algorithm for solving the problem is important especially

Table 3: Comparison with a Standard Newton's Algorithm (# Flops)

# Factors:	8	16	32	64	128
Standard:	12838	57794	308614	1911746	13191742
Fast Newton:	8561	22272	46152	103896	223768

for large-scale applications. Previous approaches used standard Newton's or approximation methods. They were either costly, requiring $\mathcal{O}(n^3)$ computation time, or not able to guarantee the fast convergence, where n is the number of structure factors of interest. We derived a formula to compute the inverse of the Hessian in $\mathcal{O}(n \log n)$ computation time, thereby reducing the time complexity of the Newton's method. As a result, we should now be able to apply the Newton's method to large-scale problems with both low computational cost and fast convergence rate.

We described the entropy maximization problem and reviewed previous approaches to the problem. Some of the previous results were given only informally in literature. We gave more formal descriptions and provided accurate proofs for key mathematical facts. In particular, we re-derived the entropy equations for solving the entropy maximization problems and the dual formulation of the problem, and showed the close relationship between the two approaches.

We focused on the dual problem of the entropy maximization problem, and derived the Sherman-Morrison-Woodbury Formula for the computation of the inverse of the Hessian of the objective function. We presented the fast Newton's algorithm based on this formula, and verified the time complexity for each step of the computation.

We also described our computational experiments with the fast Newton's algorithm. We presented the results from using the algorithm for a set of simple test problems, and compared them with some other methods. The results showed that the fast Newton's algorithm converged in fewer iterations than a typical gradient algorithm and an algorithm with Hessian approximation, although they all required the same order of floating point operations in each iteration. On the other hand, the fast Newton's algorithm required much less computation than the standard Newton's algorithm, although they both converged in the same rate. The results implied that the fast Newton's algorithm can be used to reduce the high cost of the standard Newton's algorithm, while converging as fast as the standard Newton and certainly faster than the gradient and Hessian-approximation methods. This makes it possible for solving large-scale entropy maximization problems in practice and for developing more efficient and reliable phase estimation procedures for structure determination.

References

[1] Y. Alhassid, N. Agmon, R. D. Levin, *An Upper Bound for the Entropy and Its Applications to the Maximal Entropy Problem*, Chem. Phys. Lett. Vol. 53, 1978, pp. 22–26.

[2] G. Bricogne, *Maximum Entropy and the Foundations of Direct Methods*, Acta Cryst. A40, 1984, pp. 410–445.

[3] G. Bricogne, *A Bayesian Statistical Theory of the Phase Problem. I. A Multichannel Maximum-Entropy Formalism for Constructing Generalized Joint Probability Distributions of Structure Factors*, Acta Cryst. A44, 1988, pp. 517–545.

[4] G. Bricogne, *A Multisolution Method of Phase Determination by Combined Maximization of Entropy and Likelihood. III. Extension to Powder Diffraction Data*, Acta Cryst. A47, 1991, pp. 803–829.

[5] G. Bricogne, *Direct Phase Determination by Entropy Maximization and Likelihood Ranking: Status Report and Perspectives*, Acta Cryst. D49, 1993, pp. 37–60.

[6] G. Bricogne, *Bayesian Statistical Viewpoint on Structure Determination: Basic Concepts and Examples*, in Methods in Enzymology, Vol. 276, 1997, Academic Press, pp. 361 – 423.

[7] G. Bricogne and C. J. Gilmore, *A Multisolution Method of Phase Determination by Combined Maximization of Entropy and Likelihood. I. Theory, Algorithms and Strategy*, Acta Cryst. A46, 1990, pp. 284–297.

[8] A. Decarreau, D. Hilhorst, C. Lemaréchal, and J. Navaza, *Dual Methods in Entropy Maximization: Applications to Some Problems in Crystallography*, SIAM J. on Opt., Vol 2, No. 2, 1992, pp. 173–197.

[9] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[10] W. Dong, T. Baird, J. R. Fryer, C. J. Gilmore, D. D. MacNicol, G. Bricogne, D. J. Smith, M. A. O'Keefe, and S. Hövmoller, *Electron Microscopy at 1-ÅResolution by Entropy Maximization and Likelihood Ranking*, Nature Vol. 355, 1992, pp. 605–609.

[11] S. Doublie, S. Xiang, C. J. Gilmore, G. Bricogne, and C. W. Carter Jr., *Overcoming Non-Isomorphism by Phase Permutation and Likelihood Scoring: Solution of the TrpRS Crystal Structure*, Acta Cryst. A50, 1994, pp. 164–182.

[12] S. Doublie, G. Bricogne, C. J. Gilmore, and C. W. Carter Jr., *Tryptophanyl-tRNA Synthetase Crystal Structure Reveals an Unexpected Homology to Tyrosyl-tRNA Synthetase*, Structure 15, 1995, 3:17–31.

[13] J. Drenth, *Principles of Protein X-ray Crystallography*, Springer-Verlag, New York, 1994.

[14] S. C. Fang, J. R. Rajasekera, and H. S. J. Tsao, *Entropy Optimization and Mathematical Programming*, Kluwer Academic Publishers, Boston, Massachusetts, 1997.

[15] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, 1987.

[16] E.L. Fortelle and G. Bricogne, *Maximum-Likelihood Heavy-Atom Parameter Refinement for Multiple Isomorphous Replacement and Multiwavelength Anomalous Diffraction Methods*, in Methods in Enzymology, Vol. 276, 1997, pp. 472 – 494.

[17] C. J. Gilmore, G. Bricogne, and C. Bannister, *A Multisolution Method of Phase Determination by Combined Maximization of Entropy*

and Likelihood. II. Applications to Small Molecules, Acta Cryst. A46, 1990, pp. 297–308.

[18] C. J. Gilmore, K. Henderson, and G. Bricogne, *A Multisolution Method of Phase Determination by Combined Maximization of Entropy and Likelihood. IV. The Ab Initio Solution of Crystal Structures from Their X-ray Powder Data*, Acta Cryst. A47, 1991, pp. 830–841.

[19] C. J. Gilmore, K. Henderson, and G. Bricogne, *A Multisolution Method of Phase Determination by Combined Maximization of Entropy and Likelihood. V. The Use of Likelihood as a Discriminator of Phase Sets Produced by the SAYTAN Program for a Small Protein*, Acta Cryst. A47, 1991, pp. 842–846.

[20] J. P. Glusker and K. N. Trueblood, *Crystal Structure Analysis*, Oxford University Press, New York, 1985.

[21] H. Hauptman and J. Karle, *The Solution of the Phase Problem I. The Centrosymmetric Crystal*, ACA Monograph No. 3. Polycrystal Book Service, New York, 1953.

[22] E. T. Jaynes, *Information Theory and Statistical Mechanics*, Phys. Rev. Vol. 106, pp. 620–630.

[23] E. T. Jaynes, *Prior Probabilities*, IEEE Trans. SSC-4, pp. 227–241.

[24] J. Karle and H. Hauptman, *The Phases and Magnitudes of the Structure Factors*, Acta Cryst. 3, 1952, pp. 181–187.

[25] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.

[26] E. Prince, *The Maximum Entropy Distribution Consistent with Observed Structure Amplitudes*, Acta Cryst. A45, 1989, pp. 200-203.

[27] D. E. Sands, *Introduction to Crystallography*, Dover Publications, Inc., New York, 1975.

[28] K. Shankland, C. J. Gilmore, G. Bricogne, and Hashizume, *A Multisolution Method of Phase Determination by Combined Maximization of Entropy and Likelihood. VI. Automatic Likelihood Analysis via*

the Student t Test, with an Application to the Powder Structure of Magnesium Boron Nitride, Mg_3BN_3, Acta Cryst. A49, 1993, pp. 493–501.

[29] A. H. Sherman, *On Newton Iterative Methods for the Solution of Systems of Nonlinear Equations*, SIAM J. Num. Anal., Vol. 15, 1978, pp. 755–771.

[30] R. Tolimieri, Myoung An, and Chao Lu, *Algorithms for Discrete Fourier Transform and Convolution*, Springer, New York, 1997.

[31] Zhijun Wu, George Phillips, Richard Tapia, and Yin Zhang, *The Bayesian Statistical Approach to the Phase Problem in Protein X-ray Crystallography*, Technical Report, TR99-13, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1999.

[32] S. Xiang, C. W. Carter Jr., G. Bricogne, and C. J. Gilmore, *Entropy Maximization Constrained by Solvent Flatness: a New Method for Macromolecular Phase Extension and Map Improvement*, Acta Cryst. D49, 1993, pp. 193–212.