
Limited-Memory BFGS for Large-Scale Unconstrained Optimization

Sam Dawley

Department of Chemistry
Princeton University
sd2402@princeton.edu

A Github repository containing the source code to reproduce the results presented herein is located at github.com/APC523-S2024/sd2402-Final-Project.

1 Introduction

Exact and iterative solutions to nonlinear systems of equations are of great theoretical and numerical interest due to their prolific appearance across many of the economic, financial, and scientific fields which drive modern academic and industrial research [1, 2]. Whether it be linear programming, solving systems of differential equations, or statistical learning, the forefront of scientific research can often be framed from the perspective of an *optimization problem*, generally written

$$\begin{aligned} & \min f(x) \\ & \text{such that } x \in \mathbf{D} \end{aligned} \tag{1}$$

where we seek the value of some variable scalar or vector x within a predefined domain \mathbf{D} which minimizes the *objective function* $f(x)$. In case the domain \mathbf{D} defines some set of equalities or inequalities on x the problem is said to be *constrained*; in this instance there are a plethora of solution techniques spanning fields in calculus and linear algebra [3, 4]. In contrast, if \mathbf{D} does not define any constraints on the variables x our problem is *unconstrained* and the set of techniques for finding solutions is dramatically reduced.

The simplest approach to the general unconstrained optimization of a multivariate objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is Newton's method which becomes computationally expensive very quickly when the landscape defined by f is complicated and the Jacobian matrix of f is not easily discovered. To circumvent these issues a class of *quasi-Newton methods* are desired which seek to approximate the Jacobian matrix at each iteration rather than compute it outright. The iterative step of a quasi-Newton method is generically written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbb{B}_k^{-1} f(\mathbf{x}_k) \tag{2}$$

with \mathbb{B}_k a transformation which approximates the Jacobian of f at a position \mathbf{x}_k in phase space, or, in terms of surfaces, the point $f(\mathbf{x}_k)$. Efficient methods of developing and updating \mathbb{B}_k for the purposes of optimization are an active area of research which spawned from major accomplishments during the latter half of the twentieth century [3–6]. Most notably, the so-called “BFGS update” has provided a practical means of tackling unconstrained optimization problems over a highly dimensional parameter space.

The BFGS update, named for Broyden [7], Fletcher [8], Goldfarb [9], and Shanno [10], approximates the inverse of the Jacobian matrix for a quasi-Newton update in the form of (2) as

$$\bar{\mathbb{B}}^{-1} = \bar{\mathbb{H}} = \mathbb{H} + \frac{\mathbf{s}\mathbf{s}^T}{\mathbf{y}^T\mathbf{s}} \left[\frac{\mathbf{y}^T\mathbb{H}\mathbf{y}}{\mathbf{y}^T\mathbf{s}} + 1 \right] - \frac{\mathbf{s}\mathbf{y}^T\mathbb{H} + \mathbb{H}\mathbf{y}\mathbf{s}^T}{\mathbf{y}^T\mathbf{s}} \tag{3}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$ are the differences in position and gradient of the objective function. The terms $\bar{\mathbb{H}}$ and \mathbb{H} should be interpreted as \mathbb{H}_{k+1} and \mathbb{H}_k , respectively. Given

an initial positive definite matrix \mathbb{H}_0 the BFGS update considers successive updates of the form

$$\begin{aligned}\mathbb{H}_1 &= \mathbb{H}_0 + U(\mathbf{s}_0, \mathbf{y}_0, \mathbb{H}_0) \\ \mathbb{H}_2 &= \mathbb{H}_0 + U(\mathbf{s}_1, \mathbf{y}_1, \mathbb{H}_1) + U(\mathbf{s}_0, \mathbf{y}_0, \mathbb{H}_0), \quad \text{etc.}\end{aligned}$$

which requires that the algorithm keep a memory of all previous Hessian matrices until termination. This update formula has proven to be incredibly accurate and blazingly fast, in part because there is no matrix inversion step, unlike the plethora of unconstrained optimization routines.

While the BFGS update formula is excellent in practice, for sufficiently large systems this formula requires an extensive amount of memory which may not always be available in practice. It is for this reason that a more common implementation of the BFGS update is the *limited memory BFGS* formula, or more briefly, the *LBFGS* update.

1.1 Limited-Memory BFGS

Where memory is constrained, Nocedal [11] introduces the alternative representation of (3)

$$\bar{\mathbb{H}} = \mathbb{V}^T \mathbb{H} \mathbb{V} + \rho \mathbf{s} \mathbf{s}^T \quad (4)$$

with $\mathbb{V} = \mathbb{I} - \rho \mathbf{sy}^T$ and $\rho = 1/\mathbf{y}^T \mathbf{s}$. Equation (4) illustrates that termination—in the sense $\bar{\mathbb{H}} = \mathbb{H}$ is unchanged—occurs if $\mathbb{V} = \mathbb{I}$ the identity matrix and $\rho = \mathbf{0}$. The extended representation at each iteration then looks like

$$\begin{aligned}\mathbb{H}_1 &= \mathbb{V}_0^T \mathbb{H}_0 \mathbb{V}_0 + \rho_0 \mathbf{s}_0 \mathbf{s}_0^T \\ \mathbb{H}_2 &= \mathbb{V}_1^T \mathbb{V}_0^T \mathbb{H}_0 \mathbb{V}_0 \mathbb{V}_1 + \mathbb{V}_1^T \rho_0 \mathbf{s}_0 \mathbf{s}_0^T \mathbb{V}_1 + \rho_1 \mathbf{s}_1 \mathbf{s}_1^T \\ &= \mathbb{V}_1^T \mathbb{H}_0 \mathbb{V}_1 + \rho_1 \mathbf{s}_1 \mathbf{s}_1^T, \quad \text{etc.}\end{aligned}$$

with the final line a result of dropping “old information” where $\mathbb{V}_0 = \mathbb{I}$ and $\rho_0 \mathbf{s}_0 \mathbf{s}_0^T = \mathbf{0}$. In general we may choose to discard old information beyond the previous m iterations, giving us a scheme for updating $\bar{\mathbb{H}}$ which relies only the previous m realizations of \mathbb{H} , in contrast to the update formula (3) which *requires* knowledge of all previous iterations. For this reason Nocedal’s method is referred to as the *limited-memory* BFGS update. In certain places the Eqs. (3) and (4) are also referred to as the *product-* and *sum-forms* of the BFGS update, respectively.

A more rigorous discussion of the BFGS update formula and its use when memory is limited is discussed below in Section 3. Immediately below in Section 2 is a discussion regarding why this particular optimization algorithm best suits our interests. Results of the algorithm are illustrated for a variety of initial conditions a discussion is provided in Section 5.

2 Motivation

The spatial arrangement of N -body systems exhibit a wide diversity of properties and behavior, whether they be consequences of many-body interactions, dynamical phenomena, or external stimuli [12–17]. Among the myriad tools available for studying collective behaviors of particles interacting in space, oftentimes the most natural description is found by using *collective density variables*, conventionally defined by

$$\tilde{\rho}(\mathbf{k}) = \sum_{j=1}^N \exp(i\mathbf{k} \cdot \mathbf{r}_j) \quad (5)$$

where \mathbf{r}_j are the positions of all points $j = 1, \dots, N$ contained in some domain Ω and \mathbf{k} is a wave vector in the appropriately chosen reciprocal space. The non-linearity of the transformation in Eq. (5) entails myriad exotic mathematical properties which have been continually studied for their applications beyond sole curiosity [18–23].

For the present purposes we will be interested in studying $\tilde{\rho}(\mathbf{k})$ as it relates to the *static structure factor*, $S(\mathbf{k})$, directly proportional to the total scattering intensity of a material incident a beam of

light:

$$S(\mathbf{k}) = \frac{|\tilde{\rho}(\mathbf{k})|^2}{N} = 1 + \frac{2}{N}C(\mathbf{k}), \quad (6)$$

$$C(\mathbf{k}) = \sum_{j < k} \cos [\mathbf{k} \cdot (\mathbf{r}_k - \mathbf{r}_j)] \quad \text{with} \quad -\frac{N}{2} \leq C(\mathbf{k}) \leq \frac{N(N-1)}{2}$$

The real quantity $C(\mathbf{k})$ is defined in light of the fact that for many applications the phase angles of each $\tilde{\rho}(\mathbf{k})$ are irrelevant.

2.1 Hyperuniform and Stealthy Hyperuniform Materials

Only recently has the *hyperuniformity* concept appeared in the literature [24] and positioned itself as an integral descriptor of both equilibrium and nonequilibrium systems. Specifically, a hyperuniform material is one in which the normalized infinite-wavelength (long-ranged) density fluctuations are suppressed, implying

$$S(\mathbf{k}) = 0 \quad \text{as} \quad |\mathbf{k}| \rightarrow 0 \quad (7)$$

A *stealthy hyperuniform* system is one in which infinite wavelength density fluctuations are suppressed for a *range* of wave vectors, that is,

$$S(\mathbf{k}) = 0 \quad \text{for} \quad \mathbf{k} \in \mathbf{Q} \quad (8)$$

so that a system is completely ignorant to single scattering events within the prescribed set of wave vectors \mathbf{Q} . In the ensuing discussion we allow \mathbf{Q} to be an exclusion sphere about the origin $\mathbf{k} = \mathbf{0}$ (so that hyperuniformity property is satisfied), denoted simply by the set of wave vectors which satisfy $0 \leq |\mathbf{k}| \leq K$ for a prescribed radius K .

2.2 Generating Stealthy Hyperuniform Materials

The generation of stealthy hyperuniform materials is highly nontrivial and amounts to the *collective coordinate optimization* procedure which seeks a target $S_0(\mathbf{k})$ beginning from an arbitrary initial structure factor. Particularly, in light of (6) we define

$$\Phi(\mathbf{r}^N) = \sum_{\mathbf{k} \in \mathbf{Q}} V(\mathbf{k}) [C(\mathbf{k}) - C_0(\mathbf{k})]^2, \quad V(\mathbf{k}) = \begin{cases} V_0 > 0 & 0 \leq |\mathbf{k}| \leq K \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

with $\mathbf{r}^N = \mathbf{r}_1, \dots, \mathbf{r}_N$ and $V(\mathbf{k})$ is the square-integrable reciprocal-space potential function which we choose to be a step function [16, 20]. The nonnegative objective function (9) is minimized only when the quantity $C(\mathbf{k})$ reaches its target value and, by choosing $C_0(\mathbf{k}) = -N/2$, ensures that $S(\mathbf{k}) = 0$ as desired. Such systems are *classical ground states* of particles interacting via the potential $V(\mathbf{k})$ in (9). Possibly counterintuitively, the plethora (an infinite number, in fact) of these ground state are *disordered* rather than ordered, an observation in direct contrast to the usual classical ground states of interacting particles which usually appear as crystal structures at zero temperature [24]. This unusual property of stealthy materials lends itself to a host of desirable and exotic properties otherwise unrealized by common crystals and liquids [25–27].

Minimizing Φ of Eq. (9) for a d -dimensional system of N particles is tantamount to the unconstrained optimization of dN variables, the dimensionality of which is gargantuan for even the smallest of physically realizable systems. It is for this reason we employ the LBFGS update step in our optimization algorithm in generating stealthy hyperuniform materials beginning from arbitrary initial conditions.

3 Methodology

Unconstrained optimization of the objective function (9) will be done using the LBFGS update step (4) as outlined by Nocedal [11]. We will be most interested in systems containing a number of points useful for modelling simple to complex physical systems in the thermodynamic limit for general two- and three-dimensional spaces. In practice this restriction is met by implementing periodic

boundary conditions in a sufficiently large simulation box of side lengths far greater than a particle radius or other characteristic length scale. Values for K which define the exclusion sphere of (8) vary throughout the literature, though, since it will not significantly change the optimization procedure varying the set of constrained wave vectors in \mathbf{Q} will not be tested.

The above are physical constraints. The most important aspect of our implementation will be ensuring that the optimization performs well for a host of arbitrary initial conditions and as such we will test the algorithm using various unique point patterns as initial conditions. In addition to this error analysis of the algorithm will be performed in regard to how the rate of convergence is changed with increasing particle number and dimensionality.

Our unconstrained optimization scheme will be done in C++ and succeeding error analysis in Python. The former implementation will make use of the Basic Linear Algebra Subprograms (BLAS) available in the standard library for C++ as well as some function implementations for manipulating vectors and matrices provided by the GNU Scientific Library (GSL).

3.1 A Special BFGS Update Formula

Implementation of the LBFGS update step follows that of Nocedal [11] closely. We discuss salient points of the implementation here. Beginning from the same problem statement as in (1), consider the objective function $f : \mathbb{R}^N \rightarrow \mathbb{R}$ with a gradient g and hessian matrix h . Define the quantities

$$\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k \quad \text{and} \quad \mathbf{y}_k = g_{k+1} - g_k \quad (10)$$

which are the differences in the position and gradient between successive iterations. The quasi-Newton update step becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbb{B}_k^{-1} f(\mathbf{x}_k) = \mathbf{x}_{k+1} - \mathbb{H}_k f(\mathbf{x}_k) \quad (11)$$

where \mathbb{B}_k is a suitable approximation to the Jacobian matrix of $f(\mathbf{x}_k)$. We introduce the matrix \mathbb{H}_k , an approximation to the *inverse* of the Jacobian matrix, as a means of avoiding the computationally expensive matrix inverse operation. In particular, the *special BFGS matrix* \mathbb{H}_{new} may be written

$$\mathbb{H}_{\text{new}} = [\mathbb{I} - \rho \mathbf{s} \mathbf{y}^T] \mathbb{H}_{\text{old}} [\mathbb{I} - \rho \mathbf{y} \mathbf{s}^T] + \rho \mathbf{s} \mathbf{s}^T = \mathbb{V}^T \mathbb{H}_{\text{old}} \mathbb{V} + \rho \mathbf{s} \mathbf{s}^T \quad (12)$$

where $\rho = 1/\mathbf{y}^T \mathbf{s}$. Successive “corrections” to our approximation to the inverse Jacobian are immediately clear: When $\mathbb{V} = \mathbb{I}$ and $\rho \mathbf{s} \mathbf{s}^T = 0$ the new matrix $\mathbb{H}_{k+1} = \mathbb{H}_k$ and no new information is gathered from this update. As an illustrative example, consider application of the modified BFGS update (12) including only two previous corrections. We write

$$\rho_i = 1/\mathbf{y}_i^T \mathbf{s}_i \quad \text{and} \quad \mathbb{V}_i = (\mathbb{I} - \rho_i \mathbf{y}_i \mathbf{s}_i^T)$$

\mathbb{H}_0 a given positive definite matrix

$$\mathbb{H}_1 = \mathbb{V}_0^T \mathbb{H}_0 \mathbb{V}_0 + \rho_0 \mathbf{s}_0 \mathbf{s}_0^T$$

$$\mathbb{H}_2 = \mathbb{V}_1^T \mathbb{V}_0^T \mathbb{H}_0 \mathbb{V}_0 \mathbb{V}_1 + \mathbb{V}_1^T \rho_0 \mathbf{s}_0 \mathbf{s}_0^T \mathbb{V}_1 + \rho_1 \mathbf{s}_1 \mathbf{s}_1^T$$

Discarding old information, that is, allowing $\mathbb{V}_0 = \mathbb{I}$ and $\rho_0 \mathbf{s}_0 \mathbf{s}_0^T = \mathbf{0}$, leaves

$$\mathbb{H}_2 = \mathbb{V}_1^T \mathbb{H}_0 \mathbb{V}_1 + \rho_1 \mathbf{s}_1 \mathbf{s}_1^T \quad (13)$$

Updating this new matrix and those going forward in the same fashion,

$$\mathbb{H}_3 = \mathbb{V}_2^T \mathbb{V}_1^T \mathbb{H}_0 \mathbb{V}_1 \mathbb{V}_2 + \mathbb{V}_2^T \rho_1 \mathbf{s}_1 \mathbf{s}_1^T \mathbb{V}_2 + \rho_2 \mathbf{s}_2 \mathbf{s}_2^T$$

$$\mathbb{H}_4 = \mathbb{V}_3^T \mathbb{V}_2^T \mathbb{H}_0 \mathbb{V}_2 \mathbb{V}_3 + \mathbb{V}_3^T \rho_2 \mathbf{s}_2 \mathbf{s}_2^T \mathbb{V}_3 + \rho_3 \mathbf{s}_3 \mathbf{s}_3^T$$

⋮

Now, choose m to be the number of previous corrections included to the modified BFGS update in deriving a formula for the matrix \mathbb{H}_{k+1} . In general, for $k+1 \leq m$ the usual BFGS update is

$$\begin{aligned} \mathbb{H}_{k+1} &= \mathbb{V}_k^T \mathbb{V}_{k-1}^T \cdots \mathbb{V}_0^T \mathbb{H}_0 \mathbb{V}_0 \cdots \mathbb{V}_{k-1} \mathbb{V}_k \\ &\quad + \mathbb{V}_k^T \cdots \mathbb{V}_1^T \rho_0 \mathbf{s}_0 \mathbf{s}_0^T \mathbb{V}_1 \cdots \mathbb{V}_k \\ &\quad \vdots \\ &\quad + \mathbb{V}_k^T \mathbb{V}_{k-1}^T \rho_{k-2} \mathbf{s}_{k-2} \mathbf{s}_{k-2}^T \mathbb{V}_{k-1} \mathbb{V}_k \\ &\quad + \mathbb{V}_k^T \rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T \mathbb{V}_k \\ &\quad + \rho_k \mathbf{s}_k \mathbf{s}_k^T \end{aligned} \quad (14)$$

In the special case that $k + 1 > m$, the BFGS update becomes

$$\begin{aligned}\mathbb{H}_{k+1} = & \mathbb{V}_k^T \mathbb{V}_{k-1}^T \dots \mathbb{V}_{k-m+1}^T \mathbb{H}_0 \mathbb{V}_{k-m+1} \dots \mathbb{V}_{k-1} \mathbb{V}_k \\ & + \mathbb{V}_k^T \dots \mathbb{V}_{k-m+2}^T \rho_{k-m+1} \mathbf{s}_{k-m+1} \mathbf{s}_{k-m+1}^T \mathbb{V}_{k-m+2} \dots \mathbb{V}_k \\ & \vdots \\ & + \mathbb{V}_k^T \rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T \mathbb{V}_k \\ & + \rho_k \mathbf{s}_k \mathbf{s}_k^T\end{aligned}\tag{15}$$

Eqs. (14) and (15) are our special BFGS update matrices. Importantly, if \mathbb{H}_0 is a positive definite matrix—most easily realized by allowing our initial guess to be the identity matrix—it is straightforward to verify that all subsequent matrices derived from (14) and (15) are positive definite also, provided $\mathbf{y}_i^T \mathbf{s}_i > 0$ for all iterations i .

3.2 Gradient-Related Methods

In the simplest setting the special update formulas (14) and (15) may be used for quasi-Newton updates of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \quad \mathbf{d}_k = -\mathbb{H}_k g_k\tag{16}$$

where each iteration uses a limited memory update to determine the search direction \mathbf{d}_k . Alternatively, a preconditioned conjugate gradient iteration of the form

$$\begin{aligned}\mathbf{d}_0 &= -\mathbb{H}_0 g_0 \\ \mathbf{d}_k &= -\mathbb{H}_k g_k + \beta_k \mathbf{d}_{k-1}, \quad \beta_k = \mathbf{y}_{k-1}^T g_k / \mathbf{y}_{k-1}^T \mathbf{d}_{k-1} \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{d}_k\end{aligned}$$

may be used instead. In any case, the special BFGS update formula applied to either of these two schemes yields an algorithm with quadratic termination. For details regarding this proof the reader is referred to Ref. [11].

3.3 A Recursive Formula to Compute the Descent Direction

When using the special matrices (14) and (15) in quasi-Newton steps we require the product $\mathbf{d} = -\mathbb{H}g$. Nocedal [11] introduces a recursive algorithm for performing this product efficiently; similar to above, m is the number of corrections stored at each step and I is the iteration number. Pseudo-code for this algorithm is presented below.

3.4 Implementation

The algorithm used herein takes advantage of quasi-Newton updates in an effort to minimize the objective function (9). In particular, Eq. (16) will update the positions of particles $\mathbf{x}_1, \dots, \mathbf{x}_N$ using the update formula (14) for the matrix \mathbb{H} . The line search employed for getting α_k from (16) uses the safe-guarded cubic interpolation [28]

$$f(\mathbf{x} + \alpha \mathbf{d}) < f(\mathbf{x}) + 0.0001 \alpha \mathbf{d}^T g(\mathbf{x})\tag{17}$$

and

$$\left| \mathbf{d}^T g(\mathbf{x} + \alpha \mathbf{d}) / \mathbf{d}^T g(\mathbf{x}) \right| < 0.9\tag{18}$$

with g the gradient to the objective function as above. The recursive algorithm above can be used in each instance where the descent direction \mathbf{d} must be found.

3.5 Analyzing Results

To test the efficacy of our algorithm we rely on some methods from materials science. From Eq. (8) of Sec. 2.1 we see that ground states of our algorithm—the outputs—are determined by a structure factor which vanishes for a range of wave vectors about the origin of reciprocal space. For this reason

Data: The special BFGS matrix \mathbb{H} and gradient of the objective function g .
Result: The descent direction $\mathbf{d} \propto \mathbb{H}g$ used in quasi-Newton iterations above.

```

begin
  if  $I \leq m$  then
    | idx = 0           /* Offset for determining if (14) and (15) is used */
    | B = I             /* Bound on indexing variable */
  else
    | idx =  $I - m$ 
    | B =  $m$ 
  end
   $q_B \leftarrow g_{\text{idx}}$ 
  for  $i = (B - 1), \dots, 0$       /* Iterate backward through update variables */
    do
      |  $j \leftarrow i + \text{idx}$ 
      |  $\alpha_i \leftarrow \rho_j \mathbf{s}_j^T q_{i+1}$                                 /* Store  $\alpha_i$  */
      |  $q_i \leftarrow q_{i+1} - \alpha_i \mathbf{y}_j$ 
    end
     $r_0 \leftarrow \mathbb{H}_0 g_0$ 
    for  $i = 0, 1, \dots, (B - 1)$       /* Update elements of matrix individually */
      do
        |  $j \leftarrow i + \text{idx}$ 
        |  $\beta_j \leftarrow \rho_j \mathbf{y}_j^T r_i$ 
        |  $r_{i+1} \leftarrow r_i + \mathbf{s}_j (\alpha_i - \beta_i)$ 
      end
  end

```

we will analyze plots of $S(\mathbf{k})$ measured from each configuration using Eq. (6) to determine how well our algorithm reached the minimum value of the objective function.

Explicit implementation of the LBFGS update in our quasi-Newton scheme also allows us to capture intermediate states of the system and therefore calculate particle trajectories during optimization. This information is useful for applications which require a method of finding the closest ground state given an arbitrary initial condition [29] and also more broadly within the context of transport phenomena [30], probability theory [31], and fluid mechanics [32], to name a few applications. Although this aspect of our algorithm won't be explored in much detail some results in Sec. 5 will illustrate its usefulness to this end.

In addition to this we can consider the value of the objective function as a function of the number of iterations to gather a sense of the convergence rate of our algorithm. The rate of convergence for the BFGS update step in the quasi-Newton scheme is well documented and therefore can be compared to the literature. This will be another test of efficacy for our algorithm.

3.5.1 Initial Conditions

Similar to how optimization algorithms are tested against various benchmark functions which challenge a global optimization scheme, here we will test our algorithm against various initial conditions which span from completely random to strongly locally correlated. A list of initial conditions considered is found in Table 1. For a detailed discussion on how to generate the URL and Saturated RSA packings the reader is referred to Refs. [33] and [34], respectively.

Initial Conditions					
Packing	Final χ	ρ	ϕ	N	M
URL		1.0	0.08	1024	50
Poisson	0.02	1.0	0.54	1024	50
Saturated RSA ¹		1.0	0.54	1005	44

Table 1: Initial conditions considered to for the unconstrained optimization of Eq. (9) using the LBFGS update formula. Here ρ is the number density of particles, ϕ is the dimensionless density, also known as the volume fraction, N is the number of points considered, M is the number of members within each ensemble, and χ is a parameter which controls the exclusion region K chosen to enforce $K = 1$ in the ground state.

4 Usage

In this section we assume the user has cloned the associated repository and is in the root directory. The directory structure looks something like

```
/sd2402-Final-Project
  analysis
    cco
      CCOptimization
        main.cpp
        sam-cco.slurm
    EXC
      manager.out
      soft_core_stealthy2.out
    EXC_formats
    CONFIGURATIONS
      POISSON
      RSA
      URL
    README.md
    results
      poisson
      rsa
      url
```

The CONFIGURATIONS and results folders contain the initial and final configurational data associated with each initial condition and may be perused at one's leisure. The most important aspects of the program are those in red, the SLURM script `sam-cco.slurm` and the executable `soft_core_stealthy2.out`. The former submits a batch job to the cluster and contains all necessary command SLURM commands for submitting a job on the cluster. The latter is the executable made by the program after compiling the MakeFile.

4.1 Compiling the MakeFile

Assuming we are in the root directory, navigate to the folder `EXC_formats` and open `makefile`. Before making the executables the variable `EXEdir` must be changed to correspond to wherever the user wants to store the executables on their machine. Then, running `make` from within this folder will create the executable `manager.out` for reading and writing to output files.

Now we compile the executable `soft_core_stealthy2.out`. From the root directory, navigate to `CCOptimization` and alter the makefile identically to above. Running `make` from this directory will create the desired executable in wherever `EXEdir` was specified.

If the default `./EXC` is changed for `EXEdir` the corresponding change must be made in the SLURM submission script `sam-cco.slurm` at the bottom where the executable is called. Outputs of submitting `sam-cco.slurm` to Adroit will be created in the working directory.

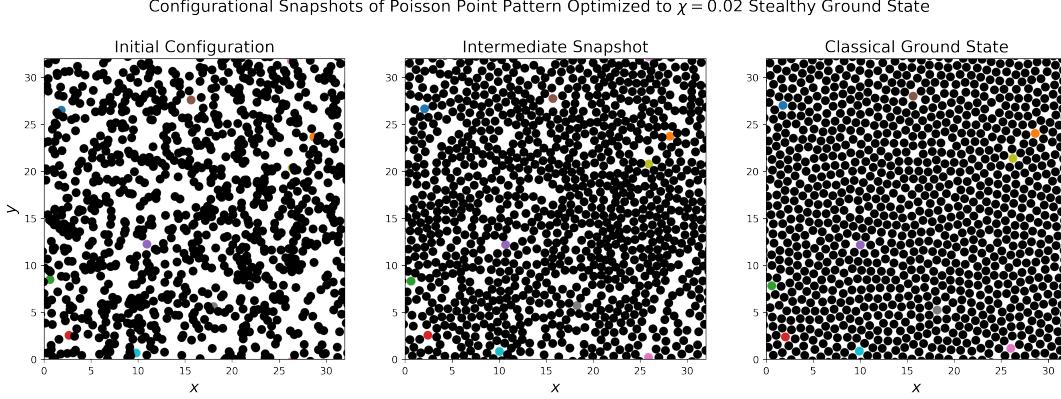


Figure 1: Temporal evolution of Poisson point pattern from initial condition to stealthy ground state with $\chi = 0.02$ following collective coordinate optimization.

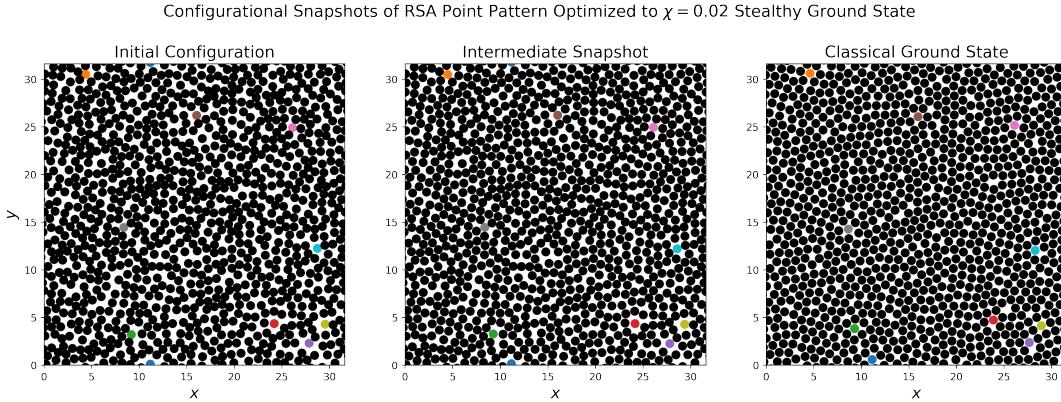


Figure 2: Temporal evolution of saturated RSA point pattern from initial condition to stealthy ground state with $\chi = 0.02$ following collective coordinate optimization.

5 Results and Conclusions

The initial and final configurations provided by our algorithm are illustrated by Figs. 1 through 3 for each of the initial conditions listed in Table 1. Some arbitrary points are colored so we get an idea of how much each particle was displaced from its original position. The structure factors for the same initial conditions and final ground states are shown in Figs. 4 through 6. A dotted red line running vertical along these graphs depicts the extent of the exclusion region in terms of the dimensionless wave number kD .

In addition to these analytic measures of determining how successful our algorithm was we also consider the total energy of the system, in this case a proxy for the error, as a function of the number of iterations in Fig. 7. The rate of convergence of the error in this plot indicates strong agreement with quadratic convergence as expected from the BFGS update step. Overhead from separate parts of the algorithm and other stochastic factors are responsible for the variation in iterations required for termination.

6 Outlook

Although the implemented optimization algorithm taking advantage of a limited-memory BFGS update step provides excellent accuracy and efficiency for our purposes the final algorithm is far from

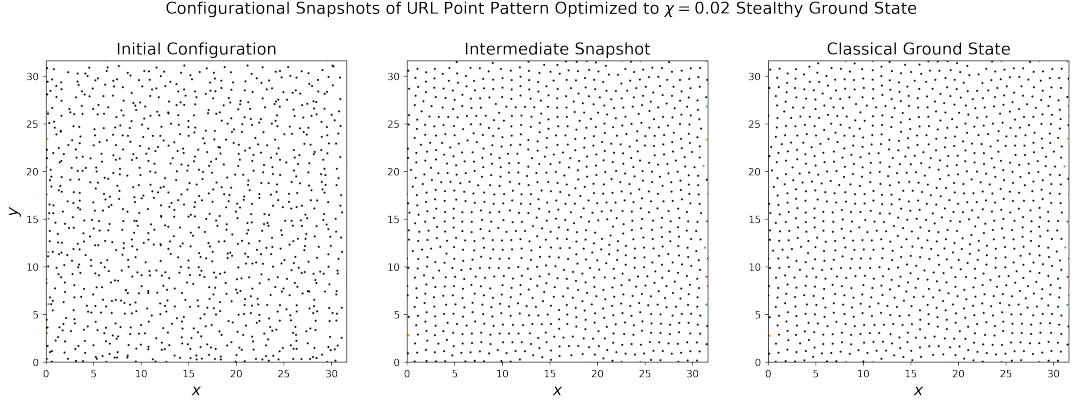


Figure 3: Temporal evolution of URL point pattern from initial condition to stealthy ground state with $\chi = 0.02$ following collective coordinate optimization.

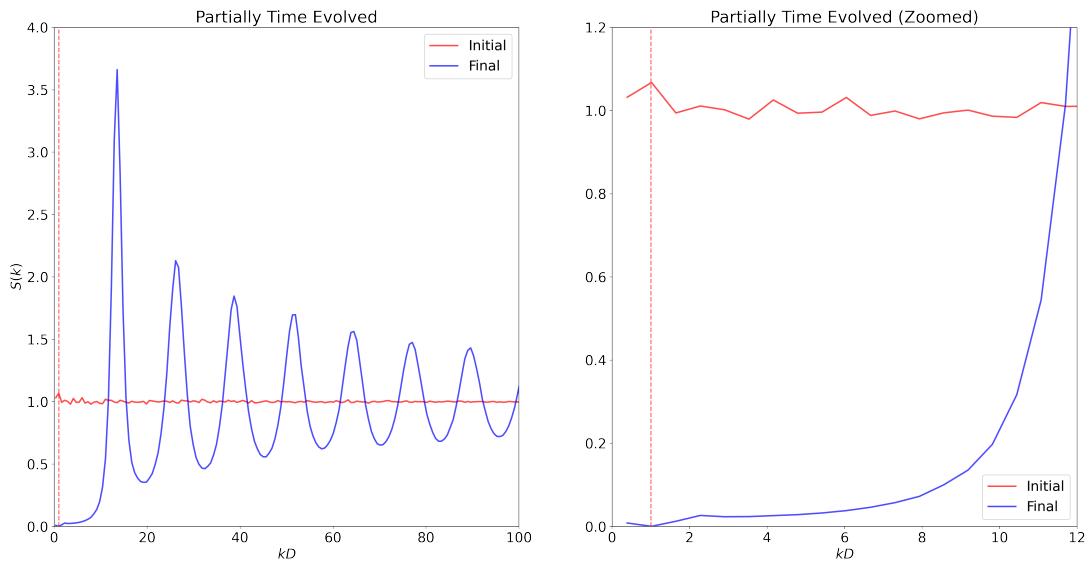


Figure 4: Time evolution of $S(k)$ from initial Poisson point pattern to stealthy ground state with $\chi = 0.02$.

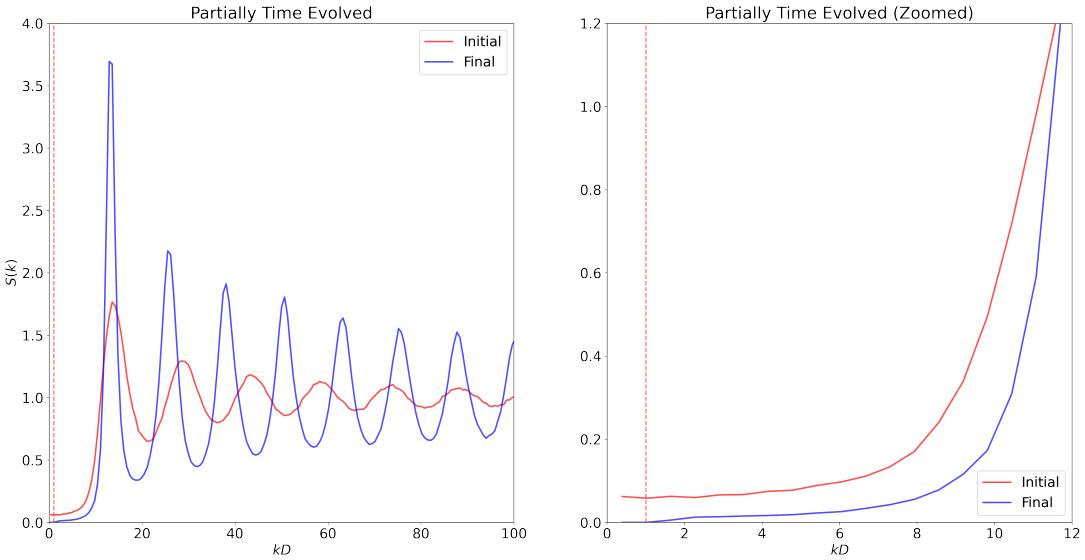


Figure 5: Time evolution of $S(k)$ from initially saturated RSA sphere packing to stealthy ground state with $\chi = 0.02$.

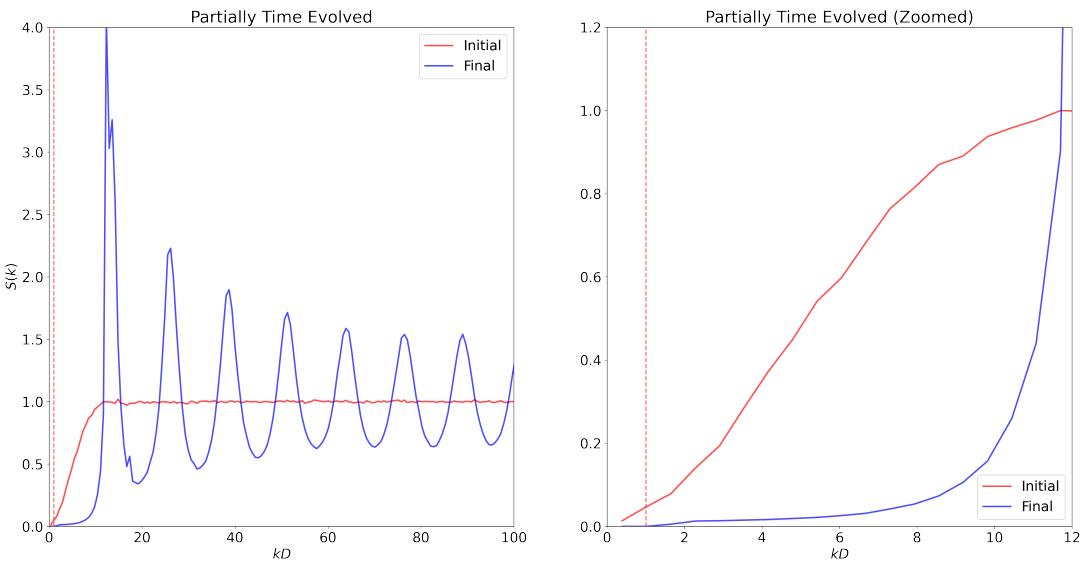


Figure 6: Time evolution of $S(k)$ from initial URL point pattern to stealthy ground state with $\chi = 0.02$.

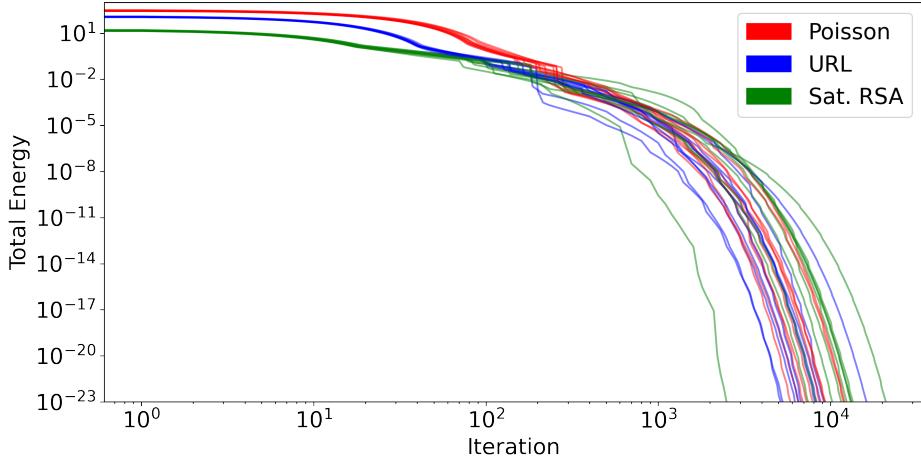


Figure 7: Energy as a function of number of iterations required by collective coordinate optimization using the LBFGS update step.

optimized. Particularly, an analysis of the optimal number m of previous iterations to store could not be performed and likely contributes to suboptimal performance of the optimizer.

References

- ¹Z. Chen, Y. Liu, and H. Sun, “Physics-informed learning of governing equations from scarce data”, *Nature Communications* **12**, 6136 (2021).
- ²B. Oksendal, *Stochastic differential equations: an introduction with applications* (Springer Science & Business Media, 2013).
- ³J. E. Dennis Jr and J. J. Moré, “Quasi-newton methods, motivation and theory”, *SIAM review* **19**, 46–89 (1977).
- ⁴R. Fletcher, *Practical methods of optimization* (Wiley, 2000).
- ⁵A. Buckley and A. Lenir, “Qn-like variable storage conjugate gradients”, *Mathematical Programming* **27**, 155–175 (1983).
- ⁶R. H. Byrd, J. Nocedal, and R. B. Schnabel, “Representations of quasi-newton matrices and their use in limited memory methods”, *Mathematical Programming* **63**, 129–156 (1994).
- ⁷C. G. Broyden, “A class of methods for solving nonlinear simultaneous equations”, *Mathematics of computation* **19**, 577–593 (1965).
- ⁸R. Fletcher, “A new approach to variable metric algorithms”, *The Computer Journal* **13**, 317–322 (1970).
- ⁹D. Goldfarb, “A family of variable-metric methods derived by variational means”, *Mathematics of computation* **24**, 23–26 (1970).
- ¹⁰D. F. Shanno, “Conditioning of quasi-newton methods for function minimization”, *Mathematics of computation* **24**, 647–656 (1970).
- ¹¹J. Nocedal, “Updating quasi-newton matrices with limited storage”, *Mathematics of computation* **35**, 773–782 (1980).
- ¹²S. Ghosh and J. L. Lebowitz, “Generalized stealthy hyperuniform processes: maximal rigidity and the bounded holes conjecture”, *Communications in Mathematical Physics* **363**, 97–110 (2018).
- ¹³E. Leutheusser, “Dynamical model of the liquid-glass transition”, *Phys. Rev. A* **29**, 2765–2773 (1984).
- ¹⁴R. D. Mountain and D. Thirumalai, “Relationship between the fluctuation metric and the non-ergodicity parameter: incoherent scattering function”, *Physica A: Statistical Mechanics and its Applications* **192**, 543–549 (1993).

- ¹⁵D. Pines, “Electron interaction in metals”, in *Solid state physics*, Vol. 1 (Elsevier, 1955), pp. 367–450.
- ¹⁶O. U. Uche, F. H. Stillinger, and S. Torquato, “Constraints on collective density variables: two dimensions”, *Physical Review E* **70**, 046122 (2004).
- ¹⁷G. Zhang, F. H. Stillinger, and S. Torquato, “Transport, geometrical, and topological properties of stealthy disordered hyperuniform two-phase systems”, *The Journal of Chemical Physics* **145**, 10.1063/1.4972862 (2016).
- ¹⁸J. K. Percus and G. J. Yevick, “Analysis of classical statistical mechanics by means of collective coordinates”, *Physical Review* **110**, 1 (1958).
- ¹⁹A. Sütő, “Crystalline ground states for classical particles”, *Physical review letters* **95**, 265501 (2005).
- ²⁰O. U. Uche, S. Torquato, and F. H. Stillinger, “Collective coordinate control of density distributions”, *Physical Review E* **74**, 031104 (2006).
- ²¹S. Torquato, G. Zhang, and F. H. Stillinger, “Ensemble theory for stealthy hyperuniform disordered ground states”, *Physical Review X* **5**, 021020 (2015).
- ²²G. Zhang, F. H. Stillinger, and S. Torquato, “Ground states of stealthy hyperuniform potentials: i. entropically favored configurations”, *Physical Review E* **92**, 022119 (2015).
- ²³G. Zhang, F. H. Stillinger, and S. Torquato, “Ground states of stealthy hyperuniform potentials. ii. stacked-slider phases”, *Physical Review E* **92**, 022120 (2015).
- ²⁴S. Torquato, “Hyperuniform states of matter”, *Physics Reports* **745**, 1–95 (2018).
- ²⁵W. Man, M. Florescu, K. Matsuyama, P. Yadak, G. Nahal, S. Hashemizad, E. Williamson, P. Steinhardt, S. Torquato, and P. Chaikin, “Photonic band gap in isotropic hyperuniform disordered solids with low dielectric contrast”, *Optics Express* **21**, 19972–19981 (2013).
- ²⁶J. Haberko, L. S. Froufe-Pérez, and F. Scheffold, “Transition from light diffusion to localization in three-dimensional amorphous dielectric networks near the band edge”, *Nature Communications* **11**, 4867 (2020).
- ²⁷L. Siedentop, G. Lui, G. Maret, P. M. Chaikin, P. J. Steinhardt, S. Torquato, P. Keim, and M. Florescu, *Stealthy and hyperuniform isotropic photonic bandgap structure in 3d*, 2024.
- ²⁸D. F. Shanno and K. H. Phua, “Algorithm 500: minimization of unconstrained multivariate functions”, *ACM Transactions on Mathematical Software (TOMS)* **2**, 87–94 (1976).
- ²⁹R. D. Batten, F. H. Stillinger, and S. Torquato, “Inherent structures for soft long-range interactions in two-dimensional many-particle systems”, *The Journal of chemical physics* **135** (2011).
- ³⁰J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, “Iterative bregman projections for regularized transportation problems”, *SIAM Journal on Scientific Computing* **37**, A1111–A1138 (2015).
- ³¹P. Cattiaux and C. Léonard, “Minimization of the kullback information of diffusion processes”, in *Annales de l’ihp probabilités et statistiques*, Vol. 30 (), pp. 83–132.
- ³²J.-D. Benamou, G. Carlier, and L. Nenna, “Generalized incompressible flows, multi-marginal transport and sinkhorn algorithm”, *Numerische Mathematik* **142**, 33–54 (2019).
- ³³M. A. Klatt, J. Kim, and S. Torquato, “Cloaking the underlying long-range order of randomly perturbed lattices”, *Physical Review E* **101**, 032118 (2020).
- ³⁴G. Zhang and S. Torquato, “Precise algorithm to generate random sequential addition of hard hyperspheres at saturation”, *Physical Review E* **88**, PRE, 053312 (2013).