CS520 Final Project: Exploration and Study of the Thomson Problem

Yifan Wu, Xiuqi Han
Purdue University
{wu2388, han870}@purdue.edu

Abstract

We study the classical Thomson problem of distributing n identical point charges on the unit sphere in $\mathbb{R}^k$ so as to minimize their total energy. In the low-dimensional regime, we re-derive three baseline optimizers—spherical-coordinate descent, Euclidean gradient-projection, and Riemannian steepest descent—and compare their convergence behavior and wall-clock performance. In the high-dimensional setting, we prove that the Riemannian gradient norm decays while showing experimentally that optimization paradoxically accelerates as k increases; we attribute this to the concentration of random initializations near a near-optimal regular simplex. Finally, we explore advanced manifold optimization algorithms—Riemannian Conjugate Gradient and Riemannian Trust-Region—and demonstrate their advantages and limitations by experiments.

1 Introduction

1.1 Background and Motivation

At the turn of the twentieth century J. J. Thomson asked how n identical electrons would arrange themselves on a sphere in order to minimise their mutual Coulomb repulsion [12, 7]. The resulting *Thomson problem* continues to surface across mathematics, physics, and computer science—from phase transitions of two-dimensional electron gases [8], through the construction of error-correcting codes [4], to blue-noise sampling in photorealistic rendering [13]. While early work focused almost exclusively on the three-dimensional sphere, modern applications such as kernel quadrature [2] and representation learning [3] require a firm grasp of the problem's *high-dimensional* regime.

1.2 Contributions

This paper provides an accessible introduction *and* a scaling study that extends far beyond the classical low-dimensional setting. Our contributions are:

1. **Low-dimensional baselines.** We re-derive three classical optimisers—spherical-coordinate descent, Euclidean gradient–projection, and Riemannian steepest descent.
2. **High-dimensional Analysis** We prove that the Riemannian gradient norm decays as $\mathcal{O}(1/\sqrt{k})$ for fixed n, and we visualize the near-orthogonality of random initializations even when $k \gg n$. We also experimentally observe a paradoxical phenomenon: while the gradient norm decays, convergence accelerates in the regime $k \gg n$. We provide a detailed theoretical explanation for this behavior.
3. **Exploration of manifold optimization methods** We explore manifold optimization techniques—including Riemannian Steepest Descent, Riemannian Conjugate Gradient (RCG), and Riemannian Trust-Region (RTR)—providing a detailed exposition of their theoretical foundations and demonstrating their practical advantages. on of their theoretical foundations and advantages.

1.3 Project Organisation

The manuscript follows:

Section 1 Introduction of Thomson Problem and our contributions.

Section 2 Formalises the problem and reviews the geometry of the sphere.

Section 3 Develops three classical optimisers in low-dimensional cases with visual experiments.

Section 4 Unveils high-dimensional phenomena and its consequences.

Section 5 Introduces Riemannian Conjugate Gradient (RCG) and Riemannian Trust-Region (RTR).

Section 6 Conclusions and limitations

2 Problem Formulation and Geometric Preliminaries

2.1 Notation

Let $k \in \mathbb{N}$ be the ambient Euclidean dimension and n the number of identical point charges. We place the charges at positions

$$X := \{x_1, \dots, x_n\} \subset \mathbb{R}^k, \qquad \|x_i\|_2 = 1 \ \text{ for } i = 1, \dots, n.$$

The unit sphere is denoted by $S^{k-1} = \{x \in \mathbb{R}^k : \|x\|_2 = 1\}$. Throughout the paper indices i, j always range over $\{1, \dots, n\}$ with $i \neq j$ understood when used inside a sum.

Table 1: Frequently used symbols.

Symbol	Meaning
n	number of points (charges)
k	ambient dimension
S^{k-1}	unit sphere in $\mathbb{R}^k$
x_i	position of the i^{th} point
$E(X)$	total Coulomb energy ((**??**))
$\nabla f(x)$	Euclidean gradient of f at x
$\operatorname{grad} f(x)$	Riemannian gradient on S^{k-1}
$T_x S^{k-1}$	tangent space at x
$R_x(\xi)$	retraction of $\xi \in T_x S^{k-1}$ to the sphere

2.2 The optimization problem

Let $\mathcal{X} = \{x_1, \dots, x_n\} \subset \mathbb{R}^k$ with the *unit-sphere constraint*

$$\|x_i\|_2 = 1, \qquad i = 1, \dots, n.$$

The Coulombic (Thomson) energy is

$$E(x_1, \dots, x_n) =$$

2.3 Geometry of the sphere

For any $x \in S^{k-1}$ the tangent space is

$$T_x S^{k-1} = \{\xi \in \mathbb{R}^k : x^\top \xi = 0\}.$$

Given a smooth $f : \mathbb{R}^k \to \mathbb{R}$ the Riemannian gradient is the orthogonal projection of the Euclidean gradient onto $T_x S^{k-1}$:

$$\operatorname{grad} f(x) = \nabla f(x) - (x^\top \nabla f(x))\, x.$$

A popular *retraction* (first–order approximation of the exponential map) is

$$R_x(\xi) = \frac{x + \xi}{\|x + \xi\|_2}, \qquad \xi \in T_x S^{k-1}.$$

It satisfies the standard retraction axioms $R_x(0) = x$ and $\mathrm{D}R_x(0) = \mathrm{id}_{T_x S^{k-1}}$.

2.4 Scope of this section

The notation and geometric facts collected above will be used verbatim in all subsequent sections. Low–dimensional algorithms are developed in Section 3; high–dimensional pathologies and their remedies begin in Section 4. Readers familiar with Riemannian optimization may skim the present section and proceed directly.

3 Low-Dimensional Baselines

We present three classical optimization strategies and reproduce the original visualisations from the student report.

3.1 Baseline A: Spherical-Coordinate optimization

We recall a classic method of polar transformation. Firstly we define

$$\theta_i \sim \mathcal{U}(0, 2\pi) \qquad \text{(Azimuthal angle)}$$
$$\phi_i \sim \mathcal{U}(0, \pi) \qquad \text{(Polar angle)}$$

then the three dimensions of $\mathbf{x}_i$ can be substitute with the angles:

$$\mathbf{x}(\theta_i, \phi_i) = \begin{bmatrix} \cos\theta_i \sin\phi_i \\ \sin\theta_i \sin\phi_i \\ \cos\phi_i \end{bmatrix} \quad \theta_i \in [0, 2\pi), \ \ \phi_i \in [0, \pi]$$

The objective function becomes a function of (θ_i, ϕ_i):

$$U(\theta_i, \phi_i) = \sum_{i<j} \frac{1}{\|\mathbf{x}_i(\theta_i, \phi_i) - \mathbf{x}_j(\theta_j, \phi_j)\|^2}$$

Next we compute the gradients $\frac{\partial U}{\partial \theta_i}$ and $\frac{\partial U}{\partial \phi_i}$. Partial derivatives of $\mathbf{x}_i$ are:

$$\frac{\partial \mathbf{x}_i}{\partial \theta_i} = \begin{bmatrix} -\sin\theta_i \sin\phi_i \\ \cos\theta_i \sin\phi_i \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{x}_i}{\partial \phi_i} = \begin{bmatrix} \cos\theta_i \cos\phi_i \\ \sin\theta_i \cos\phi_i \\ -\sin\phi_i \end{bmatrix}$$

For each $j \neq i$,

$$\frac{\partial}{\partial \mathbf{x}_i}\left(\frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}\right) = -\frac{2(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i - \mathbf{x}_j\|^4}$$

use chain rule, we have

$$\frac{\partial U}{\partial \theta_i} = \sum_{j \neq i} -\frac{2(\mathbf{x}_i - \mathbf{x}_j)^T}{\|\mathbf{x}_i - \mathbf{x}_j\|^4} \cdot \begin{bmatrix} -\sin\theta_i \sin\phi_i \\ \cos\theta_i \sin\phi_i \\ 0 \end{bmatrix},$$

$$\frac{\partial U}{\partial \phi_i} = \sum_{j \neq i} -\frac{2(\mathbf{x}_i - \mathbf{x}_j)^T}{\|\mathbf{x}_i - \mathbf{x}_j\|^4} \cdot \begin{bmatrix} \cos\theta_i \cos\phi_i \\ \sin\theta_i \cos\phi_i \\ -\sin\phi_i \end{bmatrix}$$

then in the procedure of optimization, we update the gradient of the two angles:

$$\theta_i \leftarrow \theta_i - \alpha \frac{\partial U}{\partial \theta_i}, \quad \phi_i \leftarrow \phi_i - \alpha \frac{\partial U}{\partial \phi_i}$$

3.2 Baseline B: Euclidean Gradient + Projection

This idea is based on *Erber, T. & Hockney, G. M. (1991). Equilibrium configurations of N equal charges on a sphere* [6]. Firstly, compute the gradient of the objective function U with respect to x_i:

$$\nabla_{x_i} f = -2 \sum_{j \neq i} \frac{x_i - x_j}{\|x_i - x_j\|^4}$$

Then we "ignore" the constraint on a sphere for a while, directly perform regular gradient descent.

$$x_{i_{k+1}} \leftarrow x_{i_k} - \alpha g_{i_k}$$

After each particle update, project it back onto the unit sphere:

$$x_{i_{k+1}} \leftarrow \frac{x_{i_{k+1}}}{\|x_{i_{k+1}}\|}$$

We might try to prove it convergence in the final report. But intuitively, this works. We visualize the process in a very simple case, when $n = 2$, $k = 2$.

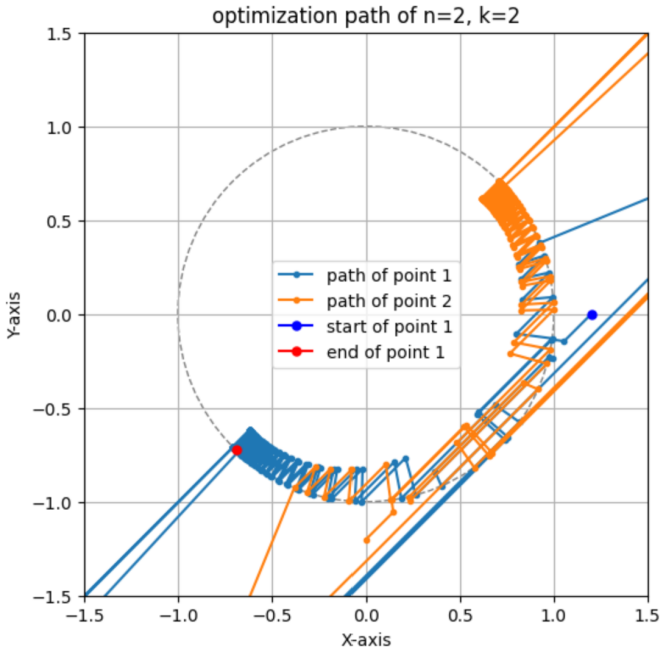


Figure 1: simple case with n = 2, k = 2

It can be seen that starting from a initial point (e.g. start of point 1 in the figure), each time the points could leave the sphere due to the best descent direction but then forced back to sphere. Generally, points are heading for global optimized outcome with the constraint. If we put the hint given in our class website page,

$$X^T e = 0$$

which means the center of all points should be kept to $(0, 0)$.So after each iteration, adjust the centroid back to the origin through:

$$x_{i_{k+1}} \leftarrow x_{i_{k+1}} - \frac{1}{n}\sum_{t=1}^{n} x_{t_{k+1}}.$$

Then we find the optimization process is even more efficient:

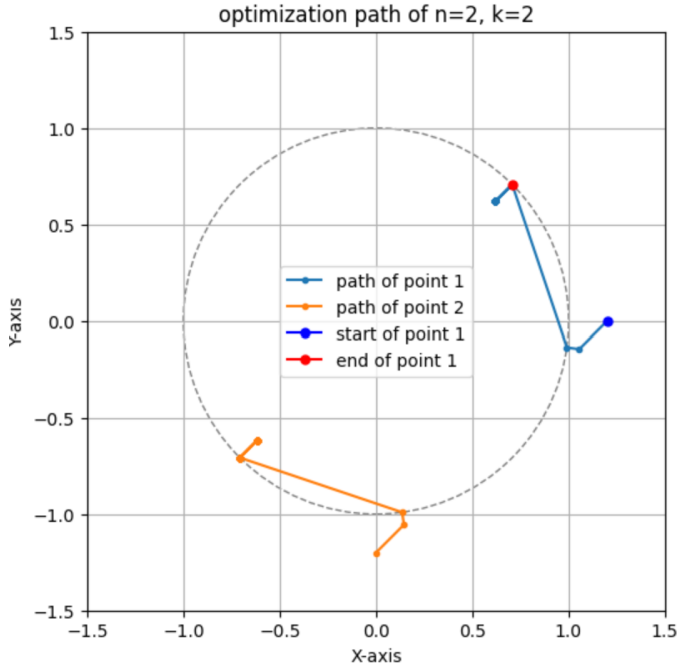


Figure 2: simple case: add constraint

3.3 Baseline C: Riemannian Steepest Descent

Before we start to explain how does Gradient Descent work on Manifolds, we introduce some basic concepts about manifolds as preparation. A standard reference is [1].

Introduction to Manifold
The $(n-1)$-dimensional sphere is defined as:

$$S^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$$

Usually, a general manifold is denoted as $\mathcal{M}$. For convenience, we directly explain things on sphere S^{n-1}. But actually the following notions could have been generally defined on $\mathcal{M}$ instead of S^{n-1}. The tangent space at $x \in S^{n-1}$ is:

$$T_x S^{n-1} = \{\xi \in \mathbb{R}^n : x^\top \xi = 0\}$$

For a function $f : \mathbb{R}^n \to \mathbb{R}$, the Riemannian gradient is obtained by orthogonal projection:

$$\xi := \text{grad } f(x) = \nabla f(x) - (x^\top \nabla f(x))x$$

The retraction on S^{n-1} is given by:

$$R_x(\xi) = \frac{x+\xi}{\|x+\xi\|}, \quad \xi \in T_x S^{n-1}$$

This satisfies the retraction axioms:

- $R_x(0_x) = x$
- $DR_x(0_x) = \text{id}_{T_x S^{n-1}}$ (local rigidity condition)

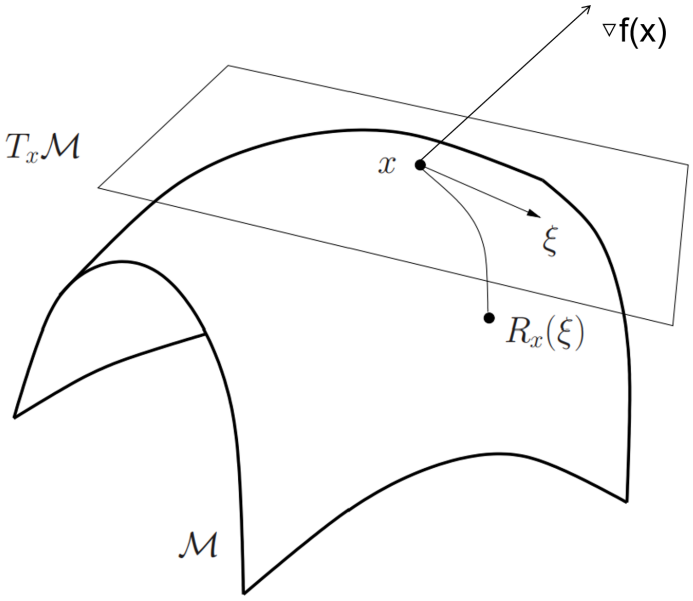


Figure 3: This graph explains the relation among the above notions. In conclusion, firstly compute $\nabla f(x)$, then project it onto $T_x S^{n-1}$ as ξ, finally retract on sphere as $R_x(\xi)$

Algorithm

Algorithm 1 Steepest Descent on S^{n-1}

Input: Initial point $x_0 \in S^{n-1}$, parameters $\bar{\alpha} > 0$, $\beta, \sigma \in (0, 1)$
for $k = 0, 1, 2, \ldots$ **do**
    Compute Euclidean gradient $\nabla f(x_k)$
    Project to tangent space: $\eta_k(\text{previous } \xi) = -(\nabla f(x_k) - (x_k^\top \nabla f(x_k))x_k)$
    Find Armijo step size $t_k = \beta^m \bar{\alpha}$:
    while $f(x_k) - f(R_{x_k}(t_k\eta_k)) < -\sigma t_k \langle \text{grad } f(x_k), \eta_k \rangle$ **do**
        $m \leftarrow m + 1$
    end while
    Update: $x_{k+1} = R_{x_k}(t_k\eta_k) = \frac{x_k + t_k\eta_k}{\|x_k + t_k\eta_k\|}$
end for

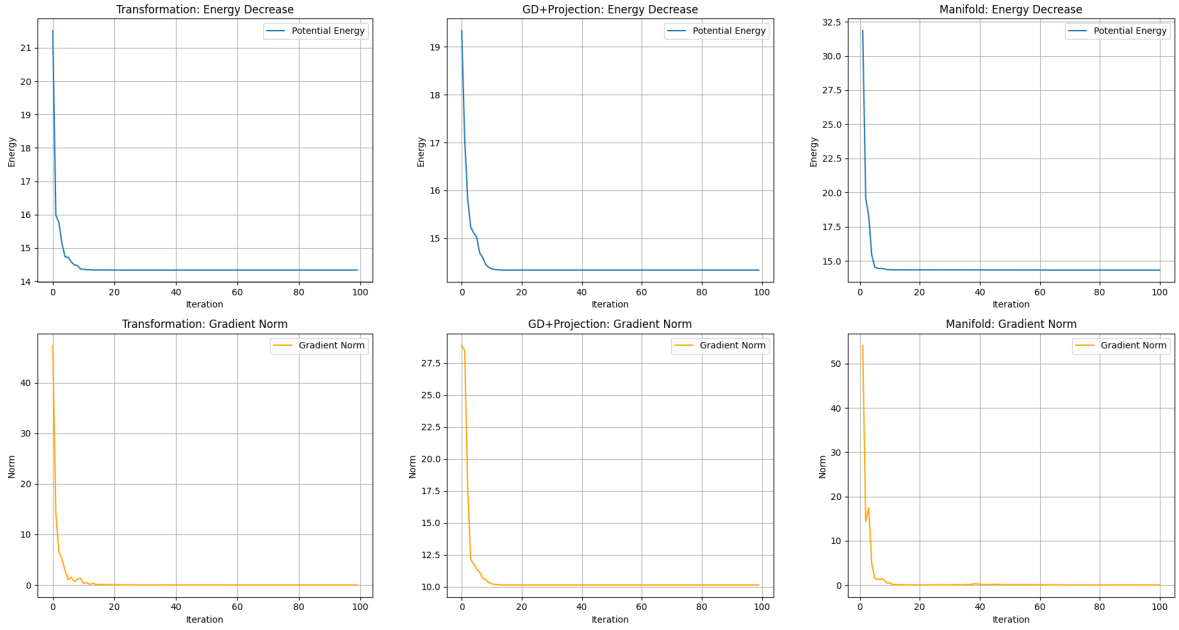
We wrote a small demo to verify the whole descent procedure when working on the sphere. We started by choosing a fixed point, computing the gradient by the formula provided before, and projected.

```
x_k: [[ 0.3052382  -0.35548152 -0.39851008]
 [-0.25951884 -0.45224471 -0.07559032]
 [-0.03542096  0.26236184  1.21039427]
 [-0.01029839  0.54536439 -0.73629387]]
Energy_k: 4.990417328963714
g(x_k)(Elucian gradient):
[[-6.7078047   0.80982692  3.14622346]
 [ 6.3505664   2.21961947 -3.51892929]
 [-0.01650278 -0.38333853 -1.12212044]
 [ 0.37374108 -2.64610786  1.49482626]]
\xi:=grad(f)(gradient on tangent space):
[[-5.61225643e+00 -4.66052596e-01  1.71590761e+00]
 [ 5.73137889e+00  1.14060619e+00 -3.69928067e+00]
 [-6.81535222e-02 -7.63275688e-04  6.42872869e-01]
 [ 3.47505167e-01 -1.25675209e+00 -3.80936364e-01]]
\xi norm: 9.206332562452575
Armijo step: 1.0
x_k+1(after retraction):
[[ 0.9415444   0.01759318 -0.33642923]
 [-0.83433216 -0.22183098  0.50465916]
 [ 0.05225439  0.420054    0.90599344]
 [-0.19120144  0.96300683 -0.18989433]]
Energy_k+1: 2.9242998709474985
```

The change of energy exactly coincide the optimization history when using a package. Thus software differentiation is correct.

3.4 Experiments in $k = 3$

With $n = 8$ charges we run all three baselines. Energy traces are shown in Figure 4; final energies and wall-times appear in Table 2.



(a) Coordinate transform (b) Proj. gradient (c) Riem. GD

Figure 4: optimization history ($n = 8$, $k = 3$).

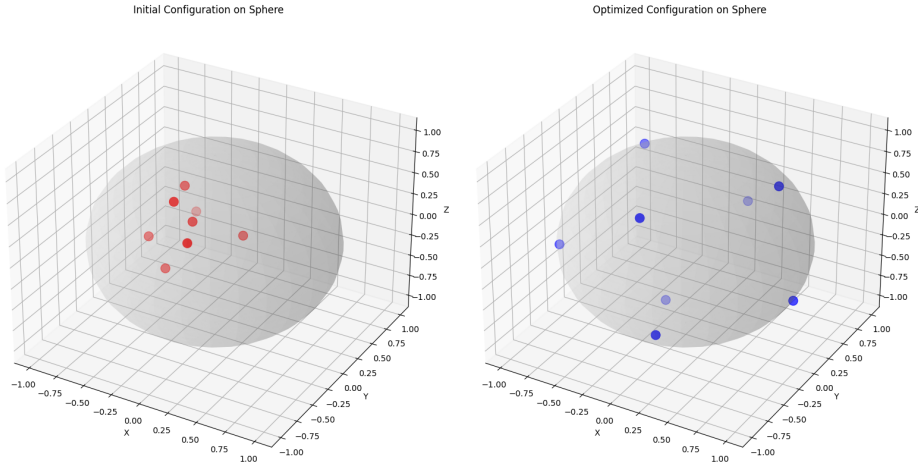


Figure 5: Generally, no matter which one we use, the algorithms transform the distribution of particles from left(initialized) to right(optimized).

Plus we compare the final outcome:

Table 2: Optimized objective value (lowest energy) and running time

-	Coordinate Trans	Classic GD	Manifolds GD
Energy	14.33679108	14.338188	14.33679127
Time/s	0.47	1.04	1.16

Observation. All three methods reach essentially the same minimum; spherical coordinates converge fastest in wall-clock, while Riemannian GD gives the tightest constraint satisfaction $\|x_i\|_2 \equiv 1$.

3.5 Bridge to High Dimensions

In the previous section, we compared three methods for solving the Thomson problem in low-dimensional settings. However, as the ambient dimension k increases, a critical phenomenon emerges:

The norm of the Riemannian gradient scales as $\mathcal{O}(1/\sqrt{k})$, causing optimization signals to weaken significantly. Intuitively, this phenomenon should lead to the failure of gradient-based methods. Yet, through our experiments and theoretical analysis, we found a surprising result: when $k \gg n$, increasing k not only does not hinder optimization but actually makes the problem much easier to solve. The deeper reason behind this lies in the near-orthogonality of random initialization points in high-dimensional spaces.

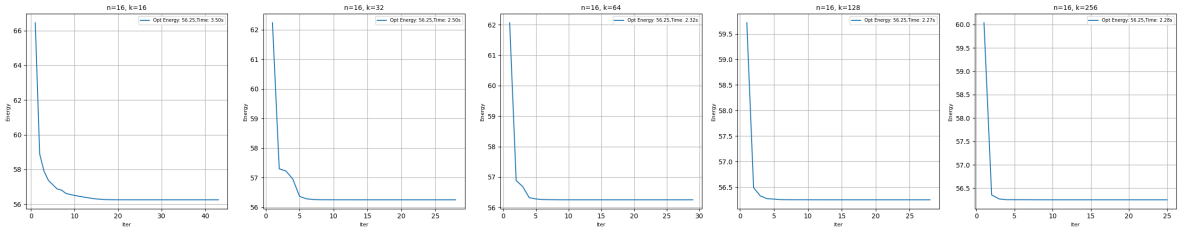
4 High-Dimensional Phenomena

We now revisit the so-called "high-dimensional paradox" using results from a Riemannian Gradient Descent (RGD) optimizer. In this experiment, we fix the number of points $n \in \{8, 16\}$, and progressively increase the ambient dimension $k \in \{8, 16, 32, 64, 128, 256\}$. For each (n, k) pair, we initialize points randomly on the sphere and optimize the objective using RGD until convergence.

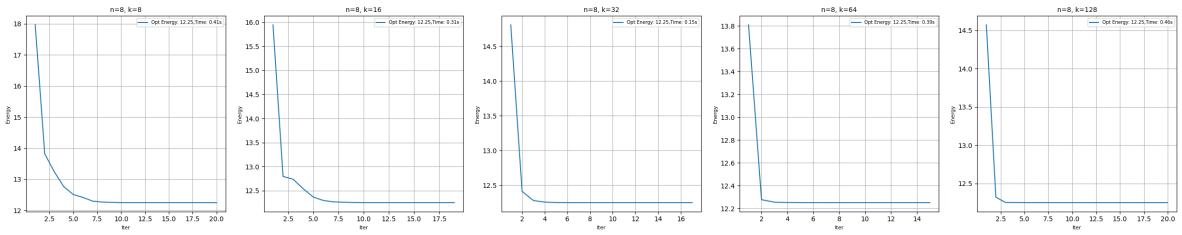
Figure 6 summarizes the results. Each subplot plots the energy value over RGD iterations for a fixed n, comparing multiple values of k. As the ambient dimension k increases, the convergence becomes significantly *faster*, both in terms of iteration count and wall-clock time—even though the Riemannian gradient norm formally shrinks as $\mathcal{O}(k^{-1/2})$.

This seemingly paradoxical behavior is particularly evident in the case $n = 8$ (bottom row), where increasing k from 8 to 128 reduces the number of required iterations from nearly 20 to fewer than 10. Similarly, for $n = 16$ (top row), the number of iterations decreases steadily as k increases from 16 to 256.

Notably, the final optimized energy remains effectively unchanged across different values of k, suggesting that high-dimensional initialization lies closer to the global minimizer in energy space. This empirical observation supports the theoretical prediction that random points on high-dimensional spheres are nearly orthogonal, thus yielding near-optimal configurations from the outset.



(a) Energy descent curves for $n = 8$ with varying k.



(b) Energy descent curves for $n = 16$ with varying k.

Figure 6: Comparison of optimization trajectories for different (n, k) settings.

This empirical observation motivates a resolution of the paradox: although gradients indeed become smaller as $k \to \infty$, the random initialization is *already* energetically close to a near-optimal configuration. In fact, the expected energy gap between a uniformly random point on S^{k-1} and the optimal configuration is of order $\tilde{\mathcal{O}}(k^{-1/2})$. Consequently, a *constant* number of optimization iterations is sufficient to reduce the energy by this small margin, once $k \gg n$.

This behavior aligns with the theoretical analysis presented in the next subsection, which shows that high-dimensional concentration effects cause random initializations to lie near-optimal in energy—despite being far in distance from the true minimizers.

4.0.1 Random Initialization and Uniform Measure on S^{k-1}

A classical presentation of concentration of measure on the sphere is in [9]. The fact that the projection of a Gaussian onto the sphere is uniform appears in [5].

Let $Z \sim \mathcal{N}(0, I_k)$ be a k-dimensional standard normal vector. Define

$$X = \frac{Z}{\|Z\|_2}, \quad \|Z\|_2 = \sqrt{\sum_{i=1}^{k} Z_i^2}\,.$$

Theorem 1 (Uniform Distribution on S^{k-1})**.** *For any measurable $A \subset S^{k-1}$,*

$$\Pr(X \in A) = \frac{\mathrm{Vol}_{k-1}(A)}{\mathrm{Vol}_{k-1}(S^{k-1})}\,,$$

i.e. X is uniformly distributed on the unit sphere.

Proof sketch: The joint density of Z is rotationally invariant, $f_Z(z) \propto e^{-\|z\|^2/2}$, and under the change to spherical coordinates,

$$dz = r^{k-1}\, dr\, d\Omega(\theta),$$

the angular part $d\Omega(\theta)$ is uniform on S^{k-1}. Conditioning on $\|Z\| = r$ and then setting $r = 1$ by normalization yields the result.

Asymptotic regime. In the remainder of this subsection we *fix the number of points n* and let the ambient dimension $k \to \infty$. All $O(\cdot)$ and $o(\cdot)$ statements are understood in this regime.

4.0.2 Energy Function and Its Gradient Vanising

Define for $i \neq j$,

$$f_{ij}(x_i, x_j) = \frac{1}{\|x_i - x_j\|^2}.$$

Then the total energy is

$$E(x_1, \ldots, x_n) = \sum_{1 \le i < j \le n} f_{ij}(x_i, x_j).$$

For a given x_i, differentiating in the Euclidean space yields

$$\nabla_{x_i} f_{ij}(x_i, x_j) = -\frac{\nabla_{x_i} \|x_i - x_j\|^2}{\|x_i - x_j\|^4}.$$

Since

$$\|x_i - x_j\|^2 = \langle x_i - x_j, x_i - x_j \rangle \quad \text{and} \quad \nabla_{x_i} \|x_i - x_j\|^2 = 2(x_i - x_j),$$

we obtain

$$\nabla_{x_i} f_{ij}(x_i, x_j) = -\frac{2(x_i - x_j)}{\|x_i - x_j\|^4}.$$

Thus, the gradient with respect to x_i is

$$\nabla_{x_i} E = -2 \sum_{j \neq i} \frac{x_i - x_j}{\|x_i - x_j\|^4}.$$

Since the search must be constrained on the sphere S^{k-1}, we project the gradient onto the tangent space $T_{x_i} S^{k-1}$. The projection operator is given by

$$P_{x_i}(v) = v - \langle v, x_i \rangle\, x_i,$$

and the effective gradient is

$$g_i = P_{x_i}\big(\nabla_{x_i} E\big).$$

Using the relation

$$\|x_i - x_j\|^2 = 2\big(1 - u_{ij}\big), \quad \text{where} \quad u_{ij} = \langle x_i, x_j \rangle,$$

we have

$$\frac{1}{\|x_i - x_j\|^4} = \frac{1}{[2(1 - u_{ij})]^2} = \frac{1}{4}\, \frac{1}{(1 - u_{ij})^2}.$$

For $|u_{ij}| < 1$, expand using the (exact) Taylor series

$$\frac{1}{(1-u_{ij})^2} = \sum_{m=0}^{\infty} (m+1)\, u_{ij}^m,$$

so

$$\frac{1}{\|x_i - x_j\|^4} = \frac{1}{4} \sum_{m=0}^{\infty} (m+1)\, u_{ij}^m.$$

Substituting back,

$$\nabla_{x_i} E = -2 \sum_{j \neq i} (x_i - x_j) \cdot \frac{1}{4} \sum_{m=0}^{\infty} (m+1)\, u_{ij}^m = -\frac{1}{2} \sum_{j \neq i} (x_i - x_j) \Big\{ 1 + 2u_{ij} + 3u_{ij}^2 + \cdots \Big\}.$$

Separate the $m = 0$ term:

$$T_0 = -\frac{1}{2} \sum_{j \neq i} (x_i - x_j) = -\frac{1}{2} \Big[(n-1)x_i - \sum_{j \neq i} x_j \Big].$$

When projected onto the tangent space,

$$P_{x_i}(x_i) = 0, \quad \text{so} \quad P_{x_i}(T_0) = \frac{1}{2} P_{x_i}\big(\sum_{j \neq i} x_j\big).$$

Denote the remaining series (for $m \geq 1$) by

$$\Delta_i = -\frac{1}{2} \sum_{j \neq i} (x_i - x_j) \sum_{m \geq 1} (m+1)\, u_{ij}^m.$$

Hence,

$$\nabla_{x_i} E = T_0 + \Delta_i \quad \text{and} \quad g_i = P_{x_i}\big(\nabla_{x_i} E\big) = \frac{1}{2}\, P_{x_i}\big(\sum_{j \neq i} x_j\big) + P_{x_i}(\Delta_i).$$

Because for uniform sampling on S^{k-1},

$$\mathbb{E}[u_{ij}] = 0 \quad \text{and} \quad \mathrm{Var}(u_{ij}) = \frac{1}{k},$$

with high probability

$$|u_{ij}| \leq \frac{C}{\sqrt{k}}.$$

As a consequence, each series term for $m \geq 1$ is $O(1/\sqrt{k})$, implying

$$\|P_{x_i}\big(\sum_{j \neq i} x_j\big)\| = O\Big(\sqrt{\tfrac{n-1}{k}}\Big), \quad \|P_{x_i}(\Delta_i)\| = O\big((n-1)\tfrac{1}{\sqrt{k}}\big).$$

Therefore,

$$\|g_i\| = O\Big(\tfrac{1}{\sqrt{k}}\Big), \qquad (\text{with } n \text{ fixed and } k \to \infty).$$

Hence, under a fixed-n asymptotic, the effective gradient indeed vanishes as the dimension grows, confirming the "high-dimensional flatness" intuition.

4.0.3 Optimal Configuration as a Regular Simplex

We consider n unit vectors $x_1, \ldots, x_n \in \mathbb{S}^{k-1} \subset \mathbb{R}^k$ minimizing the pairwise repulsion energy

$$E(X) \;=\; \sum_{i \neq j} \frac{1}{\|x_i - x_j\|^2}\,, \qquad \|x_i - x_j\|^2 \;=\; 2\big(1 - \cos\theta_{ij}\big)\,,$$

where $\theta_{ij} = \arccos(x_i^\top x_j)$. Minimizing E is equivalent to maximizing every θ_{ij}.

A *regular* $(n-1)$*-simplex* on the unit sphere is a configuration of n points satisfying

$$x_i^\top x_j \;=\; -\frac{1}{n-1} \quad (i \neq j).$$

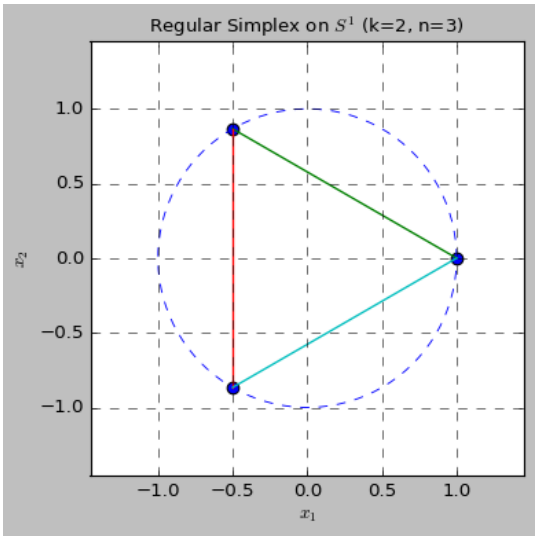
Such a configuration achieves maximal pairwise angle and hence global minimal energy. Linear-algebraically, if we assemble $X = [x_1\, x_2\, \cdots\, x_n] \in \mathbb{R}^{k\times n}$, the Gram matrix

$$G \;=\; X^\top X \;=\; \begin{pmatrix} 1 & -\frac{1}{n-1} & \cdots \\ -\frac{1}{n-1} & 1 & \\ \vdots & & \ddots \end{pmatrix}$$

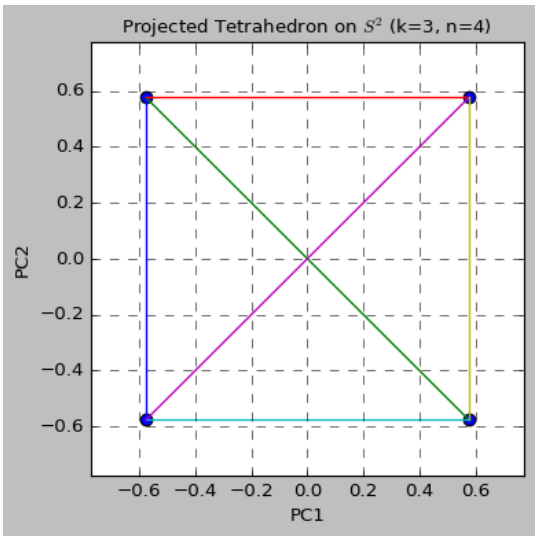
has rank n. Since $\operatorname{rank}(X^\top X) \le \operatorname{rank}(X) \le k$, this is only possible when

$$n \;\le\; k+1\,.$$

Thus whenever $n \le k+1$, the regular simplex exists and is the unique (up to rotation) global minimizer.



(a) Equilateral triangle on S^1 $(k=2, n=3)$.



(b) Projected regular tetrahedron on S^2 $(k=3, n=4)$.

Figure 7: Examples of regular simplices on low-dimensional spheres.

Figure 7a shows the only nontrivial simplex on S^1: an equilateral triangle. Figure 7b depicts the projection of the regular tetrahedron on S^2 into the plane via PCA.

4.1 High-dimensional approximately orthogonal initial points

Let $t = x^\top y = \cos\theta$. Then by symmetry,

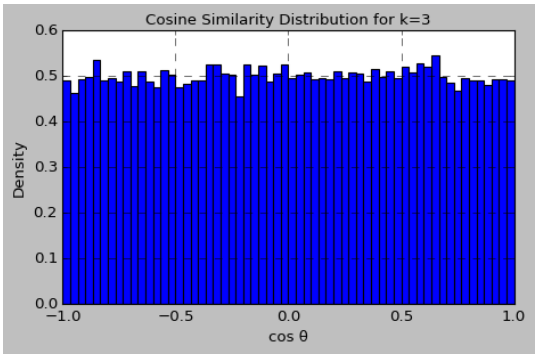
$$\mathbb{E}[t] = \sum_{m=1}^{k} \mathbb{E}[x_m]\mathbb{E}[y_m] = 0,$$

and using $\mathbb{E}[x_i x_j] = \delta_{ij}/k$, one finds

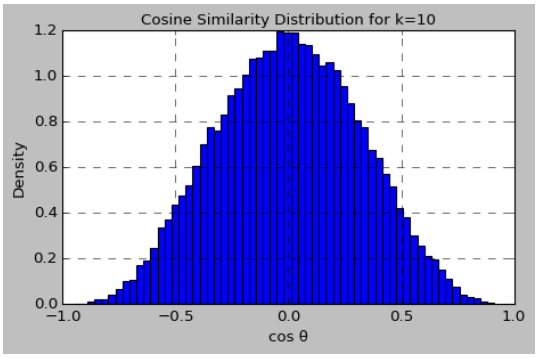
$$\mathbb{E}[t^2] = \sum_{i,j} \mathbb{E}[x_i x_j]\mathbb{E}[y_i y_j] = \frac{1}{k}, \quad Var[t] = \frac{1}{k}.$$

Hence

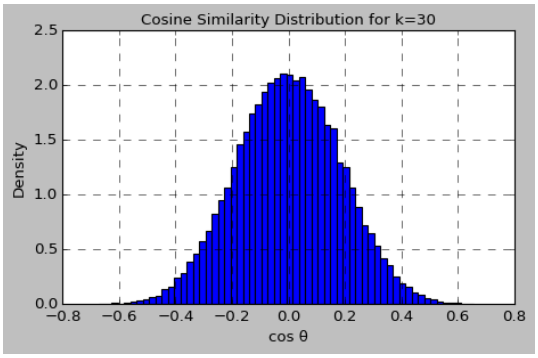
$$\|x-y\|^2 = 2(1-t) \implies Var(\|x-y\|^2) = 4\,Var[t] = \frac{4}{k} \to 0.$$



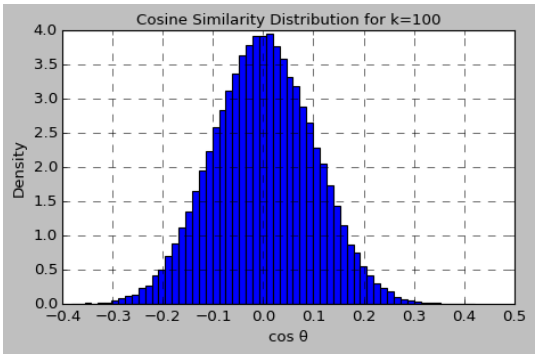
(a) $k = 3$.



(b) $k = 10$.



(c) $k = 30$.



(d) $k = 100$.

Figure 8: Histograms of $\cos\theta = x^\top y$ for random $x, y \in \mathbb{S}^{k-1}$. As k grows, the distribution concentrates sharply near 0.

Figure 8 confirms that for $k \geq 30$, almost all pairwise cosines lie in $[-0.2, 0.2]$. Thus a random initialization of $n \leq k+1$ points is already nearly a regular simplex.

We now show how the above facts imply

$$\underbrace{\|\nabla E\|}_{\text{vanishing}} \quad \text{and} \quad \underbrace{E_{\text{init}} - E^*}_{O(1/k)} \longrightarrow \text{ very few iterations to converge.}$$

The optimal energy for a regular simplex is

$$E^* = \binom{n}{2} \frac{1}{\frac{2n}{n-1}} = \frac{n(n-1)}{4}.$$

A random pair has $\mathbb{E}[\,1/\|x-y\|^2\,] \approx \frac{1}{2}(1 + 1/k)$, so

$$\mathbb{E}[E_{\text{init}}] = \binom{n}{2} \cdot \tfrac{1}{2}\Big(1 + \frac{1}{k}\Big) = E^*\Big(1 + \tfrac{1}{k}\Big) \quad \Longrightarrow \quad \frac{E_{\text{init}} - E^*}{E^*} = O\big(1/k\big).$$

That is why the puzzle happens.

5 Optimization Method Improvement

5.1 Experimental Phenomenon: Slow Convergence when $n > k$

In Part 4, we discussed how the optimization process accelerates with increasing dimension $k \gg n$, particularly when random initial points are close to an optimal configuration. Despite the vanishing gradient norm scaling as $\mathcal{O}(k^{-1/2})$, the optimization process remains efficient, and convergence to a near-optimal solution occurs within a constant number of iterations.

However, when the number of points n becomes larger than the ambient dimension k, this efficiency starts to deteriorate. We observe in our experiments that as n increases, even if k is large, the optimization speed slows significantly. This happens because:

As the number of points n increases, the relationships between points become more intricate, and local minima or saddle points in the energy landscape start to trap the optimizer, resulting in slower convergence.

When $n > k$, the updates become very small as the gradient magnitude diminishes. The optimizer struggles to make substantial progress in each iteration, resulting in what appears to be a plateau in convergence.

To illustrate this, we conducted a series of experiments across three dimensions $k = 3, 20, 100$, and varied $n = 10, 50, 100, 200$. The resulting convergence profiles are shown in Figure 9. We observe:

- For fixed low $k = 3$, the required iteration count increases steeply with n, reaching over 150 iterations for $n = 200$.
- For moderate $k = 20$, the convergence becomes slightly more stable, but still slows significantly for large n.
- For high $k = 100$, the iteration count is much less sensitive to n; even with $n = 200$, convergence occurs within 140 iterations.

Figure 9 visualizes the iteration count as a function of n for multiple fixed values of k. The curves confirm that convergence becomes slower when $n > k$, particularly in low-dimensional settings.

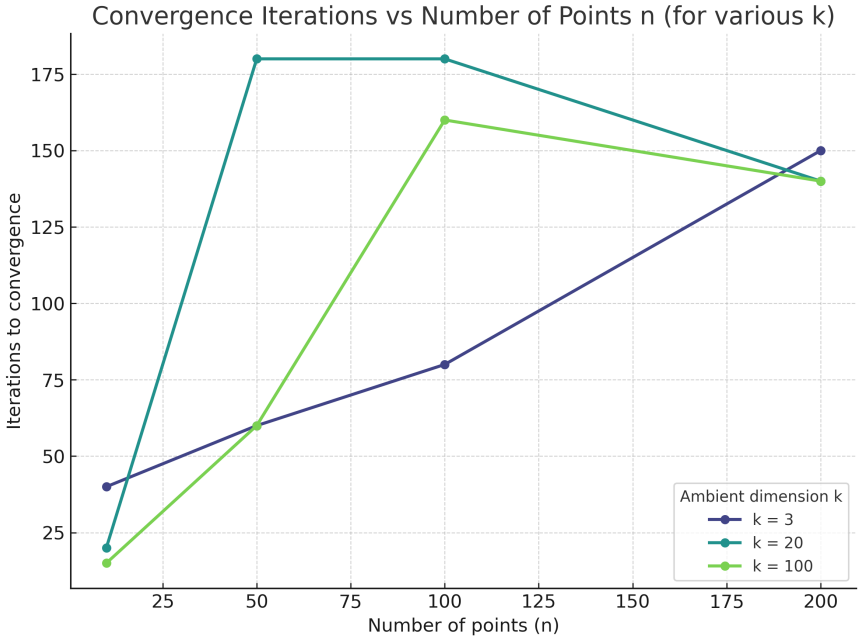


Figure 9: Iterations to reach convergence for various values of n, under different fixed ambient dimensions k. When $n > k$, convergence slows dramatically, especially for low k.

This slow convergence when $n > k$ highlights a key challenge in high-dimensional optimization and motivates the need for more sophisticated optimization methods. In the next section, we will discuss advanced algorithms such as Conjugate Gradient (CG), which can overcome the stagnation caused by large n.

5.2 Introduction of Improved Method

5.2.1 Manifolds: Conjugate Gradient

Theory

In this part, we firstly briefly review linear CG descent in $\mathbb{R}^n$ (but we skip non-linear case, which requires improved choice of β). Then in comparison, we extend to manifolds problem.

Classic Conjugate Gradient

We recall that classic CG method[10] solves the quadratic minimization problem in $\mathbb{R}^n$:

$$\min_x \phi(x) = \frac{1}{2}x^T Ax - x^T b$$

where A is symmetric positive definite. Initialize residual $r_k = b - Ax_0$, then in each iteration,

1. Compute step size and update solution:

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}, \quad x_{k+1} = x_k + \alpha_k p_k$$

2. Then update following items:

$$\text{Residuals } r_{k+1} = r_k - \alpha_k A p_k$$

$$\text{Conjugation coefficient } \beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \quad \text{(FR, and other options)}$$

$$\text{Search direction } p_{k+1} = r_{k+1} + \beta_{k+1} p_k$$

Conjugate Gradient on Riemannian manifolds

Choose initial point $x_0 \in \mathcal{M}$, and set initial search direction[11, 1] $\eta_0 = -\text{grad } f(x_0) \in T_{x_0}\mathcal{M}$. Then in each iteration,

1. Compute step size $\alpha_k > 0$ via Backtracking line search and Armijo rule.
2. Update position using retraction:

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$$

3. Compute new search direction:

$$\eta_{k+1} = -\text{grad } f(x_{k+1}) + \beta_{k+1}\mathcal{T}_{\alpha_k \eta_k}(\eta_k)$$

where

(a) $\mathcal{T}_{\alpha_k \eta_k} : T_{x_k}\mathcal{M} \to T_{x_{k+1}}\mathcal{M}$ is called vector transport, since η between different iterations are on different tangent space and to get use of η_k for η_{k+1}, this special moving method is needed. Specifically on sphere, to transport vector ξ from point x's tangent space to point y's, the form is

$$\mathcal{T}_{\eta_x}\xi_x = \frac{1}{\|x + \eta_x\|} P_{x+\eta_x}\xi_x$$

and $P_y = I - yy^T/\|y\|^2$ is the projection onto $T_y S^{n-1}$.

(b) The conjugate gradient parameter β has a few options, for example, Fletcher-Reeves's:

$$\beta_{k+1}^{FR} = \frac{g(\text{grad } f(x_{k+1}), \text{grad } f(x_{k+1}))}{g(\text{grad } f(x_k), \text{grad } f(x_k))}$$

5.2.2 Manifolds: Trust Region

Theory
Again, we firstly briefly review Trust-region methods in $\mathbb{R}^n$. Then correspondingly compared, step by step, we explain how it is extended to manifolds problem.
Classic Trust-region methods
Quadratic Model

$$f(x_k + p) = f_k + g_k^T p + \tfrac{1}{2} p^T \nabla^2 f(x_k + tp) p,$$

Subproblem: Dogleg and Cauchy point
At the k-th iteration, our subproblem is:

$$\min_p m_k(p) = f(x_k) + g^T p + \frac{1}{2} p^T B_k p \quad \text{subject to } \|p\| \le \Delta_k$$

The Dogleg method constructs the descent direction by combining two classical directions:

$$\text{Steepest descent} \quad p_u = -\frac{g^T g}{g^T B g} g$$

$$\text{Newton} \quad p_b = -B^{-1} g.$$

The final Dogleg point is decided in different case as:

- If the Newton step p_b is within the trust region ($\|p_b\| \le \Delta_k$), directly take $p_{\text{dog}} = p_b$.
- If p_u exceeds the trust region ($\|p_u\| \ge \Delta_k$), scale p_u to the boundary: $p_{\text{dog}} = \Delta_k \cdot \frac{p_u}{\|p_u\|}$, i.e.recess to Cauchy point.
- Or find the point along the Dogleg path that exactly reaches the trust-region boundary. Interpolate between p_u and p_b, find $\tau \in (0, 1)$ such that:

$$\|p(\tau)\| = \|p_u + \tau(p_b - p_u)\| = \Delta_k$$

Quotient Assessment
As long as p_k obtained, compare

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$$

with $\frac{1}{4}$, $\frac{3}{4}$.

Trust-Region Methods on Riemannian Manifolds

Quadratic Model

The second-order model at $x_k \in \mathcal{M}$ is defined on the tangent space $T_{x_k}\mathcal{M}$ as:

$$m_k(\eta) = f(x_k) + \langle \operatorname{grad} f(x_k), \eta \rangle + \frac{1}{2} \langle H_k[\eta], \eta \rangle,$$

where $\eta \in T_{x_k}\mathcal{M}$, $\operatorname{grad} f(x_k)$ is the Riemannian gradient, and H_k is a symmetric operator (typically the Riemannian Hessian $\operatorname{Hess} f(x_k)$).

Trust-Region Subproblem and tCG

At each iteration, we solve:

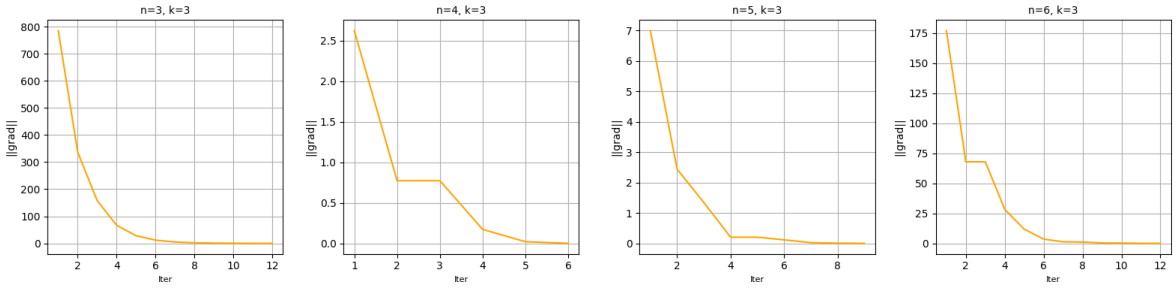
$$\min_{\eta \in T_{x_k}\mathcal{M}} m_k(\eta) \quad \text{subject to} \quad \|\eta\| \le \Delta_k,$$

where $\Delta_k > 0$ is the trust-region radius. The subproblem is approximately solved using a truncated Conjugate Gradient (tCG) method, which:

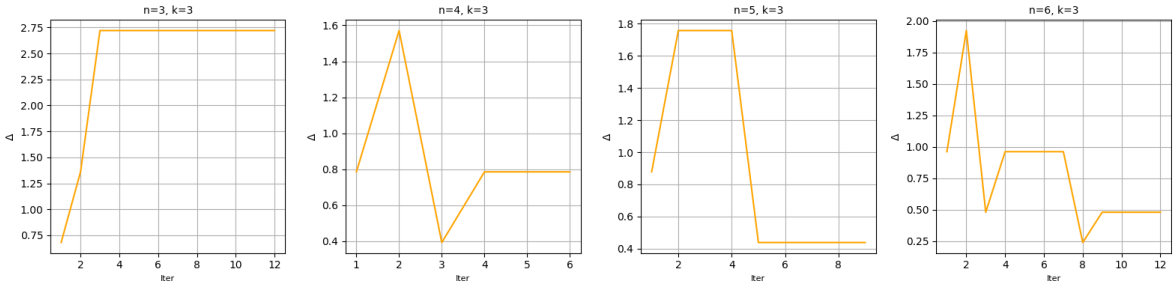
- Maintains feasibility: $\|\eta^j\| \le \Delta_k$

- Ensures decrease of $m_k(\eta^j)$
- Terminates when one of the following holds:
  - The boundary is reached: $\|\eta^{j+1}\| \geq \Delta_k$
  - Negative curvature is encountered: $\langle \delta_j, H_k \delta_j \rangle \leq 0$
  - The residual norm $\|r_j\|$ falls below a given threshold

Notice that on Riemannian manifolds, all gradient and Hessian computations respect the manifold's metric structure.



(a) Tracking grad norm



(b) Tracking Δ_k

Figure 10: To better understand Trust-Region method, we observe the change of its specific parameters.

The tCG inner iteration may be considered to fail or require termination in the following cases:

- When the Hessian operator H_k exhibits negative curvature along the search direction δ_j:

$$\langle \delta_j, H_k[\delta_j] \rangle_{x_k} \leq 0$$

  which indicates the model is non-convex in this direction, and tCG cannot continue safely.

- When the proposed step η^{j+1} would exceed the trust-region radius:

$$\|\eta^{j+1}\|_{x_k} \geq \Delta_k$$

- When the residual norm becomes sufficiently small, e.g.:

$$\|r_{j+1}\| \leq \|r_0\| \min(\|r_0\|^{\theta}, \kappa).$$

In cases (1) and (2), the algorithm falls back to computing a step along the boundary that guarantees at least the Cauchy decrease, which is computed as follows. Since the Cauchy point lies along the negative gradient direction in $T_{x_k}\mathcal{M}$:

$$\eta_k^C = -\tau \cdot \text{grad} f(x_k),$$

the step size τ is chosen to minimize the model $\hat{m}_{x_k}$ while respecting the trust-region constraint:

$$\tau^* = \underset{\tau>0}{\operatorname{argmin}}\, \hat{m}_{x_k}(-\tau \cdot \operatorname{grad} f(x_k)) \quad \text{s.t.} \quad \|\tau \cdot \operatorname{grad} f(x_k)\| \leq \Delta_k$$

Then the solution can be computed explicitly as:

$$\tau^* = \min\left(\frac{\|\operatorname{grad} f(x_k)\|^2}{\langle \operatorname{grad} f(x_k), H_k[\operatorname{grad} f(x_k)]\rangle}, \frac{\Delta_k}{\|\operatorname{grad} f(x_k)\|}\right)$$

where the first term is the unconstrained minimizer and the second term enforces the trust-region constraint, and resulting Cauchy point is:

$$\eta_k^C = -\tau^* \cdot \operatorname{grad} f(x_k)$$

This point guarantees the *Cauchy decrease*:

$$\hat{m}_{x_k}(0) - \hat{m}_{x_k}(\eta_k^C) \geq \frac{1}{2}\|\operatorname{grad} f(x_k)\| \min\left(\Delta_k, \frac{\|\operatorname{grad} f(x_k)\|}{\|H_k\|}\right)$$

Quotient Assessment
Finally, evaluate the acceptance ratio:

$$\rho_k = \frac{f(x_k) - f(R_{x_k}(\eta_k))}{\hat{m}_{x_k}(0) - \hat{m}_{x_k}(\eta_k)}$$

Update trust-region radius Δ_k based on ρ_k. If $\rho_k > \rho'$, accept $x_{k+1} = R_{x_k}(\eta_k)$; else reject.

5.2.3 Experimental Results

We now present experimental results comparing the effectiveness of three optimization strategies in high-dimensional settings: standard Gradient Descent (GD), Riemannian Conjugate Gradient (CG), and Riemannian Trust-Region (TR) methods. All experiments are conducted on the unit sphere manifold, where the goal is to minimize Coulomb energy for increasing values of n, with ambient dimension fixed at $k = 16$.

Figures 11 and 12 show the convergence behavior of these methods across four problem sizes: $n = 32, 64, 128, 256$. The vertical axis reports total energy; the horizontal axis reports iteration count.

In Figure 11, we compare CG and GD. The CG method consistently converges faster in both iteration count and wall-clock time, with nearly identical final energy. For example, at $n = 64$, CG converges in 419.76 seconds compared to GD's 421.45 seconds; at $n = 256$, CG is slightly faster and reaches slightly lower energy (17256.47 vs 17263.13).

Figure 12 displays convergence of the Trust-Region method. While TR yields energy values close to CG and GD, its iteration count is significantly lower. For instance, at $n = 128$, TR converges in only 12 iterations, compared to over 300 for GD/CG. However, TR incurs higher per-iteration cost, and the wall-clock time becomes competitive only for smaller n.

Figures 11 and 12 show the convergence behavior of the different methods for various values of n and k.

Table 3 summarizes the final energy values and total optimization time for each method. We observe:

- All three methods achieve nearly identical final energy values across all problem sizes.
- CG is the most time-efficient method overall.
- TR uses the fewest iterations but has higher per-iteration overhead, especially noticeable for large n.

These results confirm that while standard GD can be slow, Riemannian CG and TR methods offer practical improvements in both speed and stability for high-dimensional Coulomb optimization.

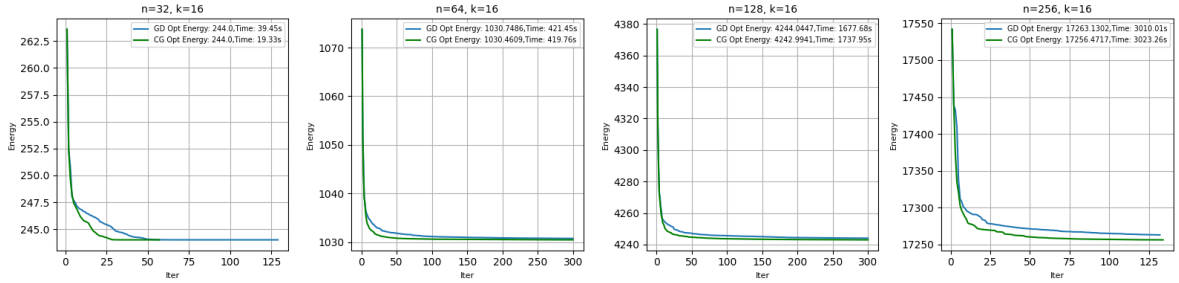


Figure 11: Convergence of Conjugate Gradient Method compared with Gradient Descent in high-dimensional optimization problems.

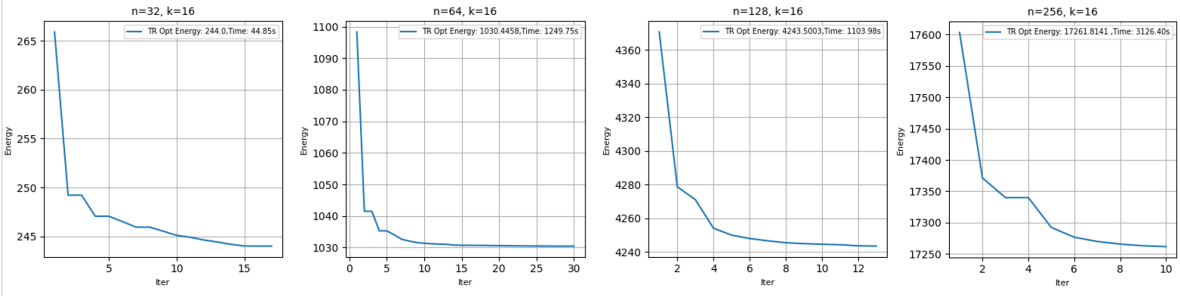


Figure 12: Convergence of Trust-Region Method in high-dimensional optimization problems.

Table 3: Optimized Energy In Summary

-	$n = 32, k = 16$	$n = 64, k = 16$	$n = 128, k = 16$	$n = 256, k = 16$
GD Energy	244.0	1030.7486	4244.0447	17263.1302
GD Time/s	39.45	421.45	1677.68	3010.01
CG Energy	244.0	1030.4609	4242.9941	17256.4717
CG Time/s	19.33	419.76	1737.95	3023.26
TR Energy	244.0	1030.4458	4243.5003	17261.8141
TR Time/s	44.85	1249.75	1103.98	3126.40

6 Conclusion

Our study presents a detailed investigation of the Thomson optimization problem, including an in-depth exploration of manifold optimization methods and a rigorous theoretical analysis of high-dimensional phenomena in the regime $k \gg n$. In addition, we compare multiple optimization strategies for the $n > k$ setting. The extensive set of visual experiments provides a solid foundation for researchers interested in both theoretical and practical aspects of this problem.

Limitations. Despite these contributions, more complex scenarios—such as those where both n and k are very large and comparable in magnitude—remain insufficiently explored due to time and computational constraints.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] F. Bach. On the equivalence between quadrature rules and random features. *Journal of Machine Learning Research*, 18(21):1–38, 2017.

[3] J. Chung and J. Glass. Learning word embeddings on the sphere. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4140–4150. Association for Computational Linguistics, 2019.

[4] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer, 3rd edition, 1998.

[5] P. Diaconis and D. Freedman. *A dozen de Finetti–style results in search of a theory*. IMS Lecture Notes, 1987.

[6] T. Erber and M. Hockney, G.˙Equilibrium configurations of n equal charges on a sphere. *J. Physics A*, 24:L1369–L1377, 1991.

[7] P. Hardin, D. and B. Saff, E.˙Discretizing manifolds via minimum energy points. *Notices of the AMS*, 51(10):1186–1194, 2004.

[8] B. Laughlin, R.˙Anomalous quantum hall effect: An incompressible quantum fluid with fractionally charged excitations. *Physical Review Letters*, 50(18):1395–1398, 1983.

[9] M. Ledoux. *The Concentration of Measure Phenomenon*. AMS, 2001.

[10] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis*, 3(R1):35–43, 1969.

[11] T. Smith, S.˙Optimization techniques on riemannian manifolds. *Fields Institute Communications*, 3:113–146, 1994.

[12] J. Thomson, J.˙On the structure of the atom. *Philosophical Magazine*, 7:237–265, 1904.

[13] R. Ulichney. *Digital Halftoning*. MIT Press, 1987.