AMATH 515: COMPUTATIONAL REPORT TEMPLATE

SID MEKA

Applied Mathematics Department, University of Washington, Seattle, WA
gameka2k@uw.edu

1. Introduction

Optimization plays a fundamental role in various scientific and engineering disciplines, particularly in recommendation systems, machine learning, and industrial engineering. The goal of optimization is to define a solution to minimize or maximize an objective function while satisfying certain constraints. Here, we will be minimizing with our constraints by iteratively refining a solution. In this report, we implement and compare and contrast four optimization methods: Steepest Descent, Newton's Method, DFP, and BFGS, each employing a line search strategy based on the weak Wolfe conditions, and further consider the Ackley Function with respect to dimensionality, height, rate of exponential decay, and frequency of cosine oscillations.

We use the Rosenbrock function to evaluate the performance of optimization algorithms. We also do logistic regression on the MNIST datset, which is widely used in classification problems. The Rosenbrock function provides challenges that we overcome as we utilize an effective search direction and further utilize adaptive step sizes for efficient convergence. Logistic regression, on the other hand, involves optimizing a loss function over high dimensional data. After all this, we consider the Ackley Function and edit some of the parameters.

This report provides an analysis of each method's convergence behavior, computational efficiency, and suitability for different problem types. We present comparative results in the form of function value progression, gradient norm reduction, and computational cost over iterations. Finally, we discuss the implications of our findings and how different optimization methods may be best suited for particular problem classes.

2. Methods

For the methods, we outline the optimization algorithms implemented, their respective update rules, and the line search strategy used for step size selection. We focus on: Steepest Descent, Newton's Method, DFP, and BFGS. Each method relies on a line search strategy using the weak Wolfe conditions to determine step sizes efficiently.

(1) **Line Search and Wolfe Conditions**:

For each iteration, we perform a bisection based line search to determine an appropriate step size α^k that satisfies the weak Wolfe conditions:

(W1 Condition) $$f(x^k + \alpha^k d^k) \leq f(x^k) + c_1 \alpha^k (d^k)^T \nabla f(x^k),$$

(W2 Condition) $$c_2 (d^k)^T \nabla f(x^k) \leq (d^k)^T \nabla f(x^k + \alpha^k d^k).$$

Date: February 14, 2025.

where d^k denotes the descent direction, α^k is the step size at the current step, and $0 < c_1 < c_2 < 1$. We have that the

(a) W1 Condition: $W_1(t) = \frac{f(x^k+td^k)-f(x^k)}{t(d^k)^T\nabla f(x^k)}$. This is the Sufficient Decrease Condition, which prevents the step from being too large.

(b) W2 Condition: $W_2(t) = \frac{(d^k)^T\nabla f(x^k+td^k)}{(d^k)^T\nabla f(x^k)}$. This is the Curvature Condition that ensures that the step is not too small and that progress is being made towards convergence.

To satisfy both of these conditions, we employ a bisection strategy that iteratively refines α^k, reducing or increasing it based on whether the conditions hold.

(2) **Optimization Methods Implemented**:
The optimization algorithms iteratively update the solution according to

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)}d^{(k)}\,,\ \text{where } d^{(k)} = -B(x^{(k)})\nabla f(x^{(k)})$$

where the choice of $B^{(k)}$ determines the specific method applied. Specifically:

- Steepest Descent: $B(x^{(k)}) = I$
- Newton Method: $B(x^{(k)}) = \nabla^2 f(x^{(k)})^{-1}$
- DFP: $B(x^{(k+1)}) = B(x^{(k)}) - \frac{\left(B(x^{(k)})y^{(k)}\right)\left(B(x^{(k)})y^{(k)}\right)^T}{(y^{(k)})^T B(x^{(k)})y^{(k)}} + \frac{s^{(k)}(s^{(k)})^T}{(y^{(k)})^T s^{(k)}}$
- BFGS: $B(x^{(k+1)}) = \left(I - \frac{s^{(k)}(y^{(k)})^T}{(y^{(k)})^T s^{(k)}}\right) B(x^{(k)}) \left(I - \frac{y^{(k)}(s^{(k)})^T}{(y^{(k)})^T s^{(k)}}\right) + \frac{s^{(k)}(s^{(k)})^T}{(y^{(k)})^T s^{(k)}}$

Note that all methods employ the `bisection_linesearch` in order to determine the appropriate step size at each iteration.

(3) Ackley Function

(a) We have our function `get_ackley` that defines the function, the gradient of the function, and the Hessian of the function. We test this on $d \in [2,5]\big|_{d\in\mathbb{Z}}$, where d is our number of dimensions. The Ackley function is used widely in optimization as it features multiple local minima and a single global minimum at the origin. It is defined by

$$f(\vec{x}) = -a\,\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)\,,\ \text{where}$$

- a represents height with a default value of 20
- b represents the rate of exponential decay with a default value of 0.2
- c represents the frequency of cosine oscillations with a default value of 2π
- d represents dimensionality

(b) Ackley for Plotting:
The function `ackley_for_plotting` is a modified version of the Ackley function designed specifically for visualization in two dimensions. It takes two inputs, x and y, along with the number of dimensions d, and parameters a, b, and c that control the function's shape. We use contour plots to first evaluate the function value when d is changed. We then change d accordingly to see how the Three Dimensional Surface Plot and Contour Plot change in shape and intensities, respectively. It is worth noting that within our implementation, we experimented with non integer dimensionality as that allows us to visualize what happens when we have small $0 < d < 1$; while the Ackley function is traditionally defined for integer dimensions, the reason we do this is to visualize what can happen for non traditional values for d, which we wouldn't have been able to do otherwise. We then experiment around by seeing what happens when we change a, b, and c while keeping $d = 2$.

3. Results

We see that here the Rosenbrock function has best convergence for the Newton and BFGS Methods depending on n and the worst convergence for Steepest Descent.We see that here the Rosenbrock function has best convergence for the Newton and BFGS Methods depending on n and the worst convergence for Steepest Descent. Performance of optimization methods for logistic regression on the MNIST dataset, with different regularization parameters: $\lambda = 0.001$, $\lambda = 0.01$, and $\lambda = 0.1$. The function value is plotted against iterations, showing the effect of different regularization strengths on convergence speed and accuracy. Lower λ values have worse convergence, whereas higher λ values employ better convergence due to their stronger regularization. We see best convergence to worst convergence for Newton's Method, Steepest Descent, BFGS, and DFP in that order.Performance of optimization methods for logistic regression on the MNIST dataset, with different regularization parameters: $\lambda = 0.001$, $\lambda = 0.01$, and $\lambda = 0.1$. The function value is plotted against iterations, showing the effect of different regularization strengths on convergence speed and accuracy. Lower λ values have worse convergence, whereas higher λ values employ better convergence due to their stronger regularization. We see best convergence to worst convergence for Newton's Method, Steepest Descent, BFGS, and DFP in that order. Overall, Newton's Method seems to be the best consistently across the Rosenbrock function and Logistic Regression.

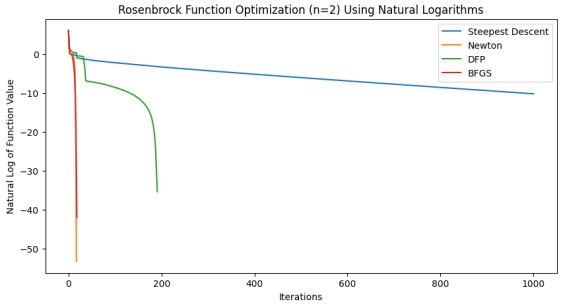
We also experiment with the Ackley function. The Ackley function's behavior varies significantly with dimensionality. At lower dimensions, such as $d < 1$, the function exhibits pronounced peaks and valleys, making the landscape highly oscillatory. As dimensionality increases: $1 \leq d \leq 10$, the function smooths out, with local minima becoming less distinct. In high dimensions, $d > 50$, the function flattens, reducing the effect of cosine oscillations and making the optimization landscape less rugged. For extremely high dimensions like $d \geq 1000$, the function approaches a nearly constant surface, resembling a quadratic bowl. These results suggest that in lower dimensions, optimization algorithms may struggle with local minima, while in very high dimensions, they may benefit from a smoother landscape but lose meaningful structural variations.

Note: Figure 1, Figure 2, Text output for the Ackley Function, and Ackley Function Images are in the Appendix

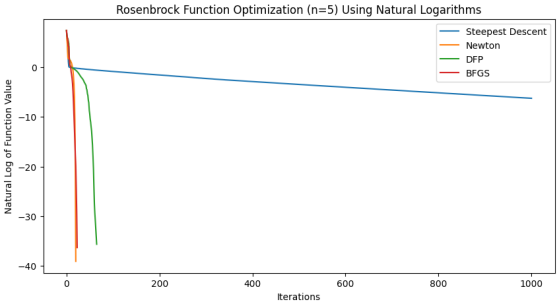
4. Summary and Conclusions

We have computationally analyzed certain convergence methods, which proved useful for their practicality, efficiency, and learning in solving optimization problems. Our results indicate that Newton's Method consistently demonstrates superior convergence, while quasi Newton methods like BFGS provide a good balance between efficiency and computational cost. Steepest Descent, though simple, converges more slowly, particularly for complex landscapes like the Rosenbrock function. Additionally, our exploration of the Ackley function highlights how dimensionality affects the optimization landscape, with higher dimensions leading to a smoother, less oscillatory function. These findings reinforce the importance of selecting appropriate optimization techniques based on the problem structure and computational constraints.

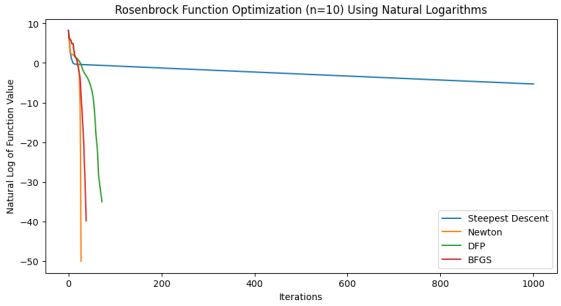
5. Appendix



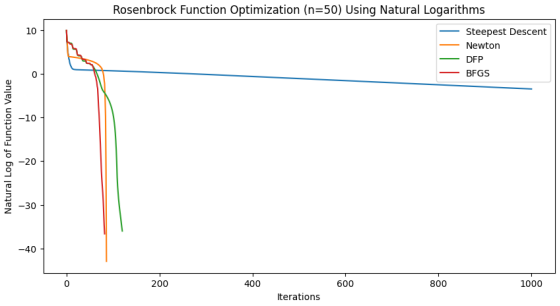
(A) Rosenbrock function with $n = 2$



(B) Rosenbrock function with $n = 5$

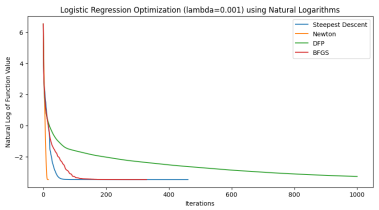


(C) Rosenbrock function with $n = 10$

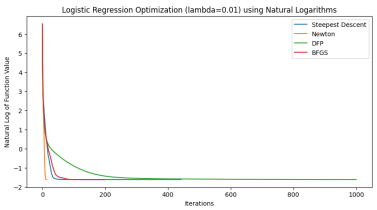


(D) Rosenbrock function with $n = 50$

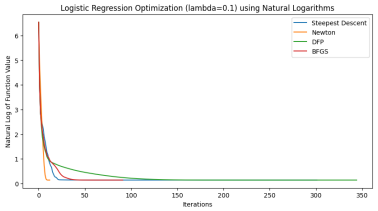
Figure 1. Rosenbrock function with different n



(A) $\lambda = 0.001$



(B) $\lambda = 0.01$

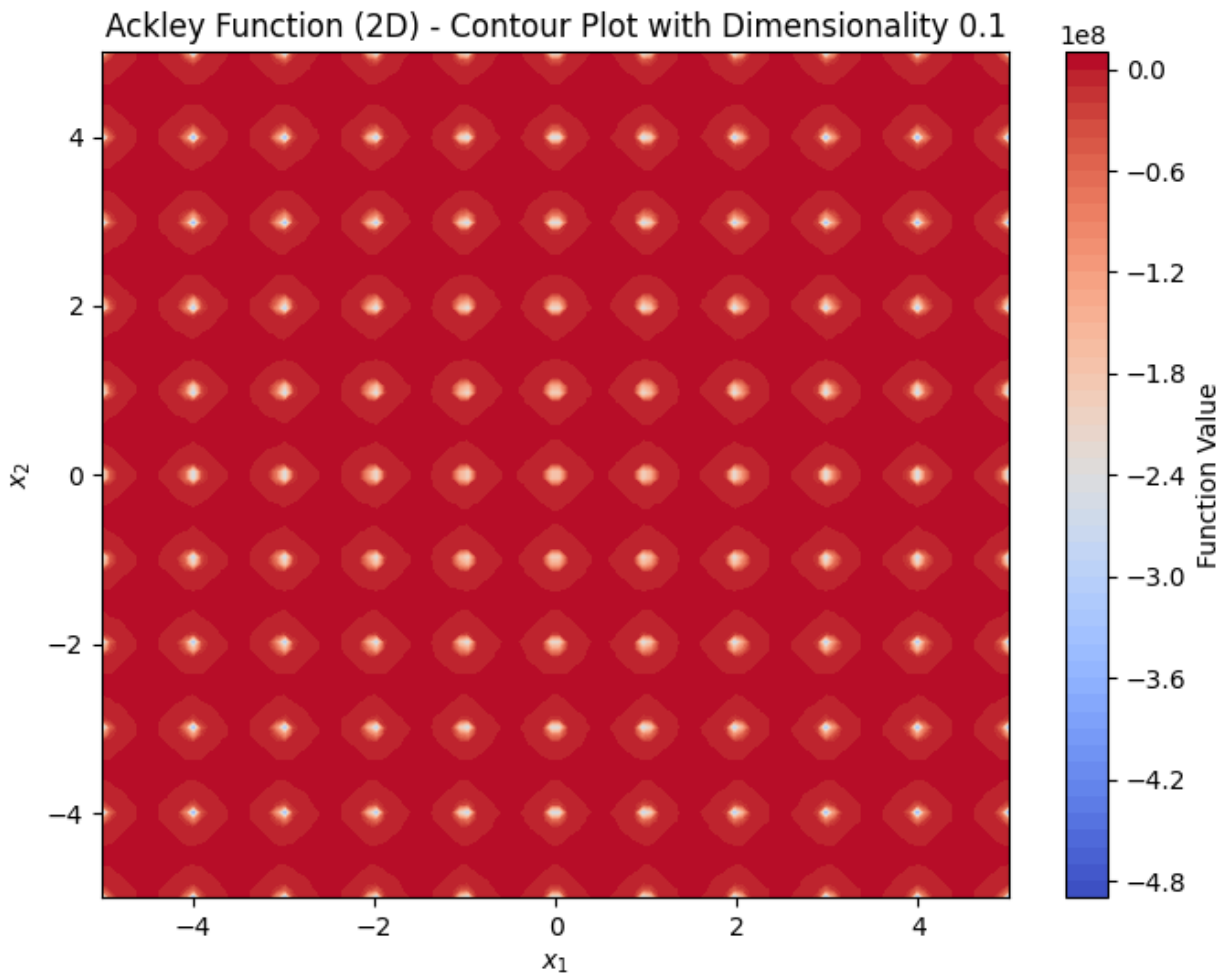


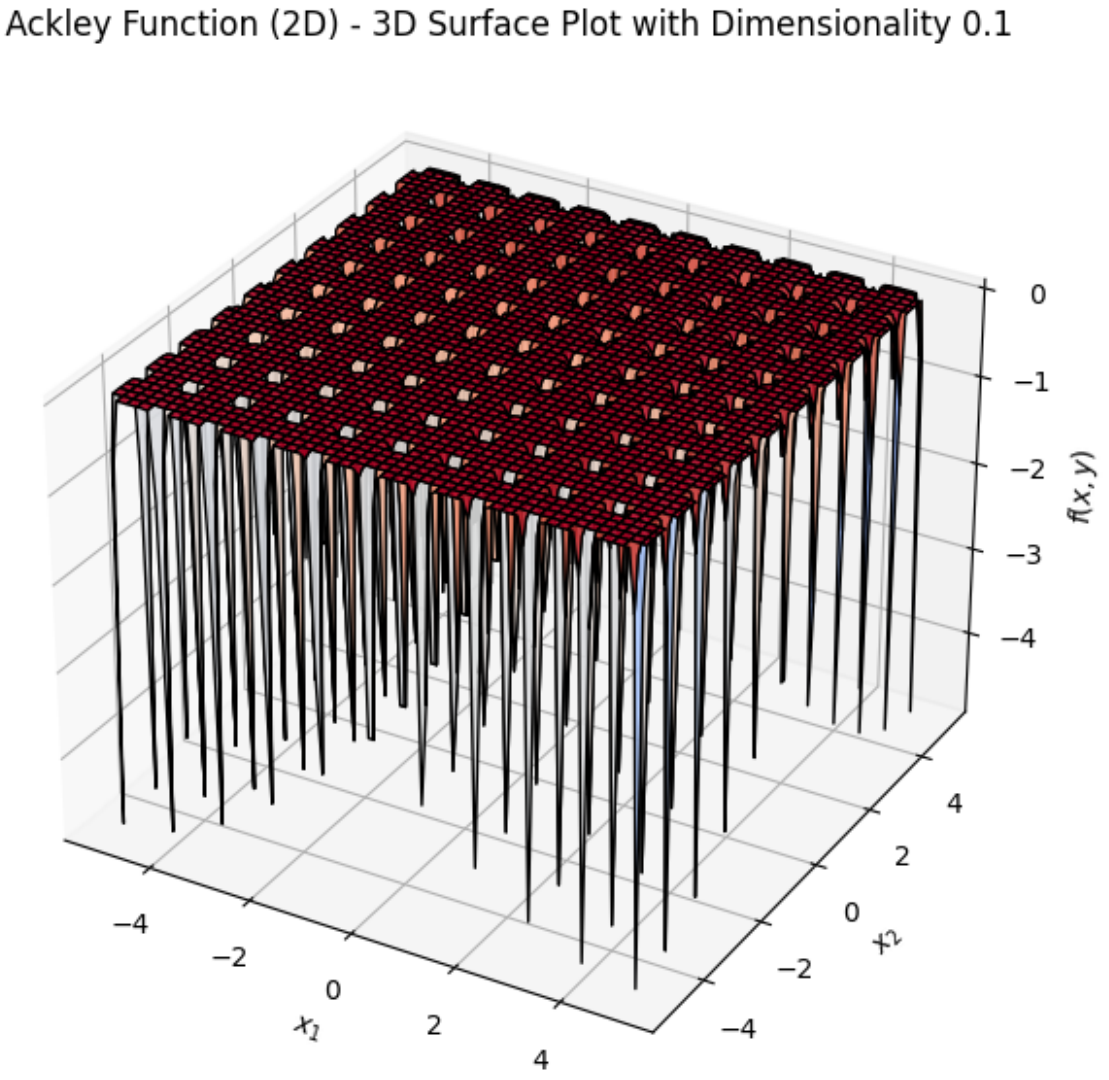
(C) $\lambda = 0.1$

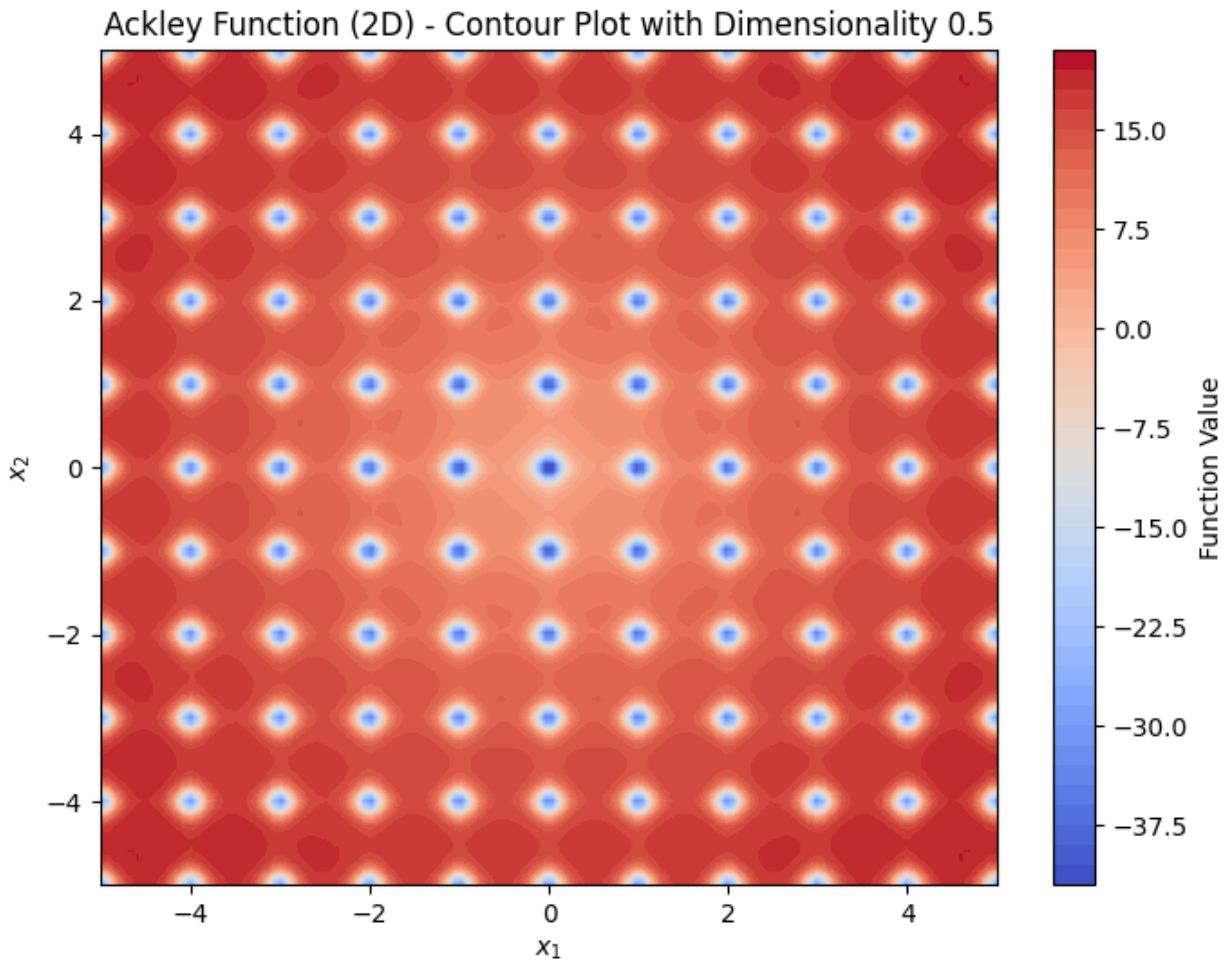
Figure 2. Logistic Regression with different λ

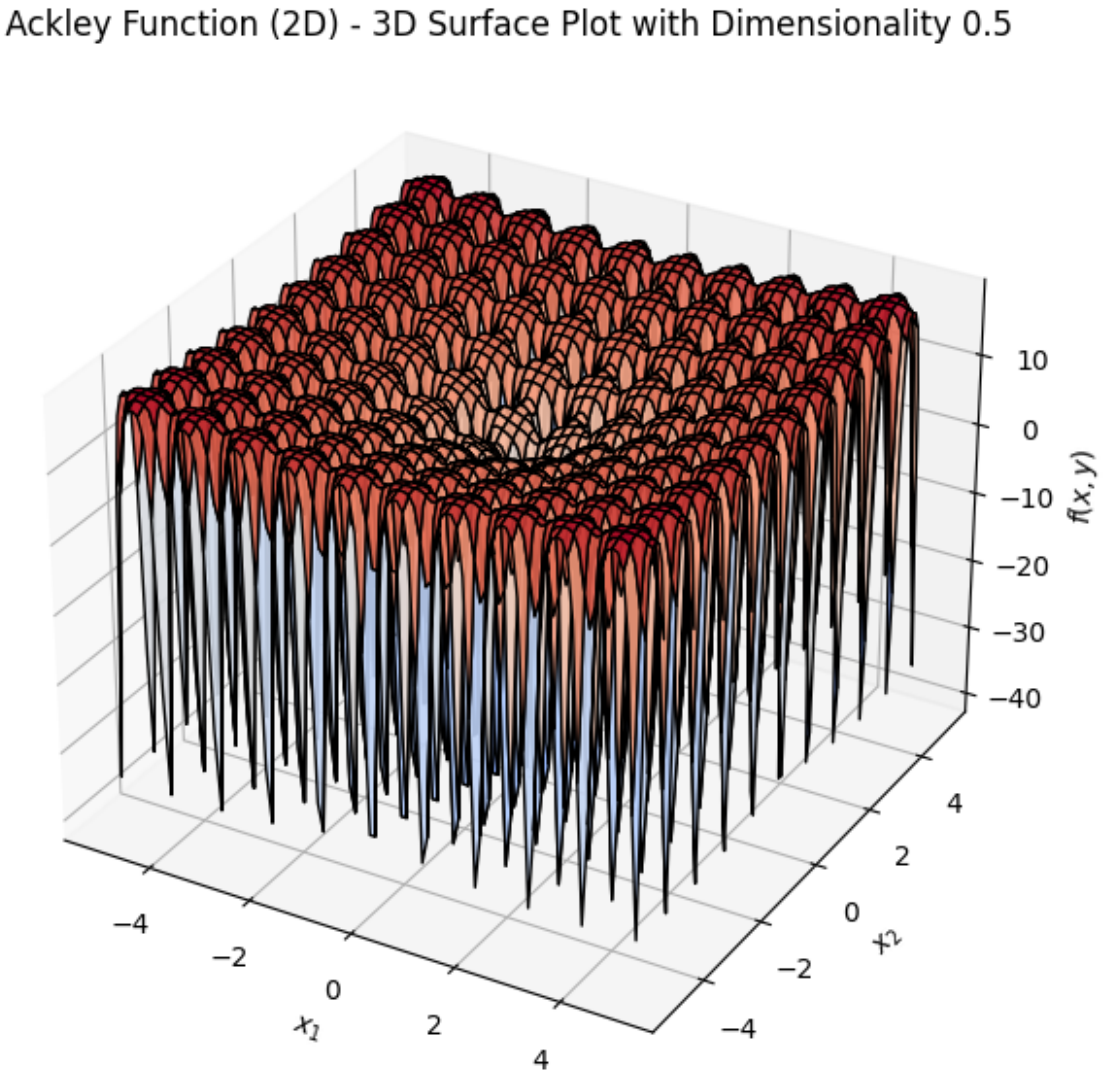
Ackley Function Test:

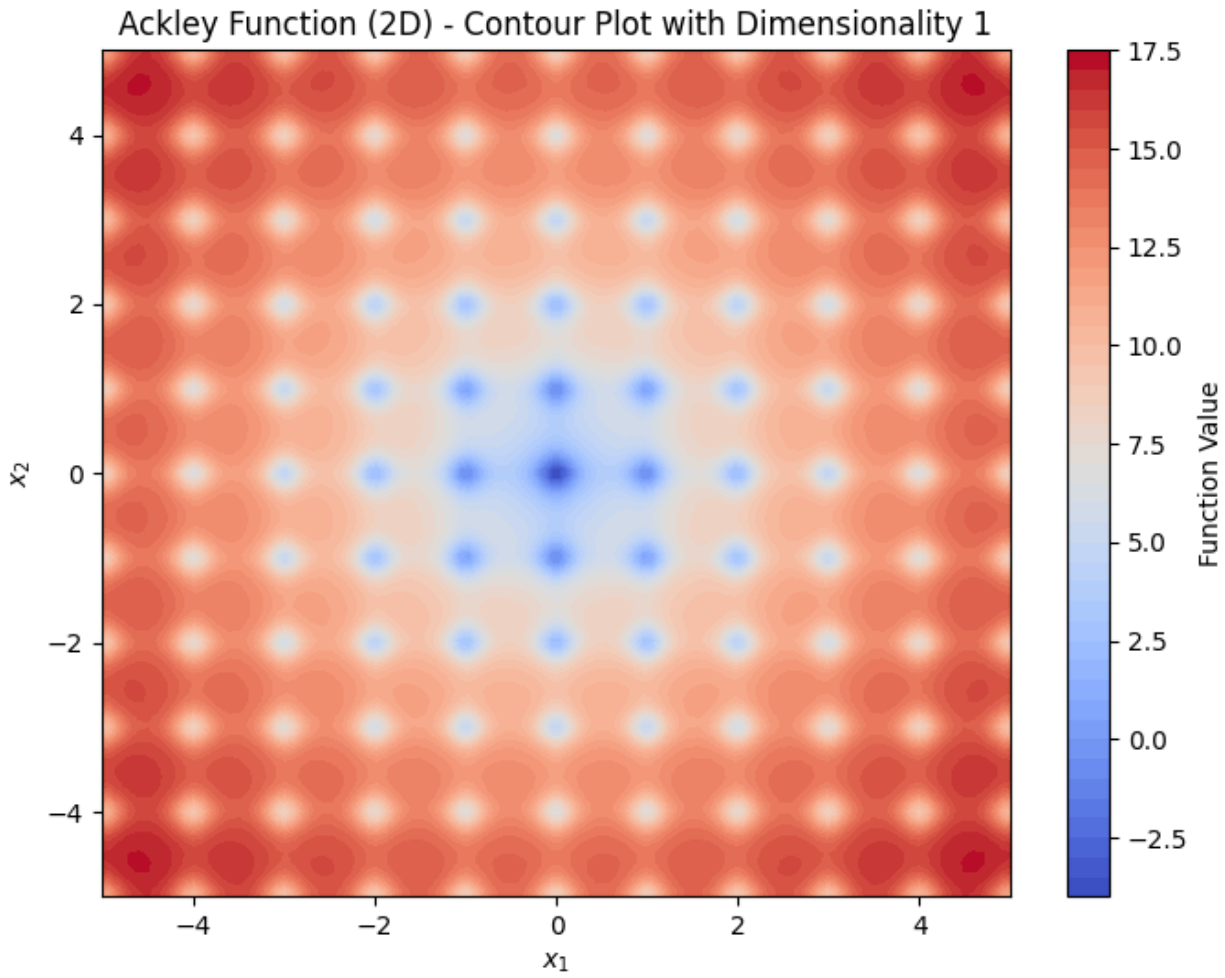
```
Number of Dimensions: 2
Ackley Function Test:
f(x0): 9.684621694099041
Gradient Norm at x0: 2.4696033601135365
Hessian at x0:
 [[-11.38703077   0.        ]
 [  0.           9.83212428]]

Number of Dimensions: 3
Ackley Function Test:
f(x0): 11.05739567440051
Gradient Norm at x0: 3.7559217495220283
Hessian at x0:
 [[16.72816444  0.          0.        ]
 [ 0.         -2.27599577  0.        ]
 [ 0.          0.         16.95623356]]

Number of Dimensions: 4
Ackley Function Test:
f(x0): 11.28769581323352
Gradient Norm at x0: 1.5167838725843785
Hessian at x0:
 [[-6.01728516  0.          0.          0.        ]
 [ 0.         -4.42236218  0.          0.        ]
 [ 0.          0.         -4.34513202  0.        ]
 [ 0.          0.          0.          4.14741686]]

Number of Dimensions: 5
Ackley Function Test:
f(x0): 7.502076421302988
Gradient Norm at x0: 3.1705678967061948
Hessian at x0:
 [[10.46764811  0.          0.          0.          0.        ]
 [ 0.          9.92008302  0.          0.          0.        ]
 [ 0.          0.          8.86849283  0.          0.        ]
 [ 0.          0.          0.          9.70733238  0.        ]
 [ 0.          0.          0.          0.          3.76506618]]
```

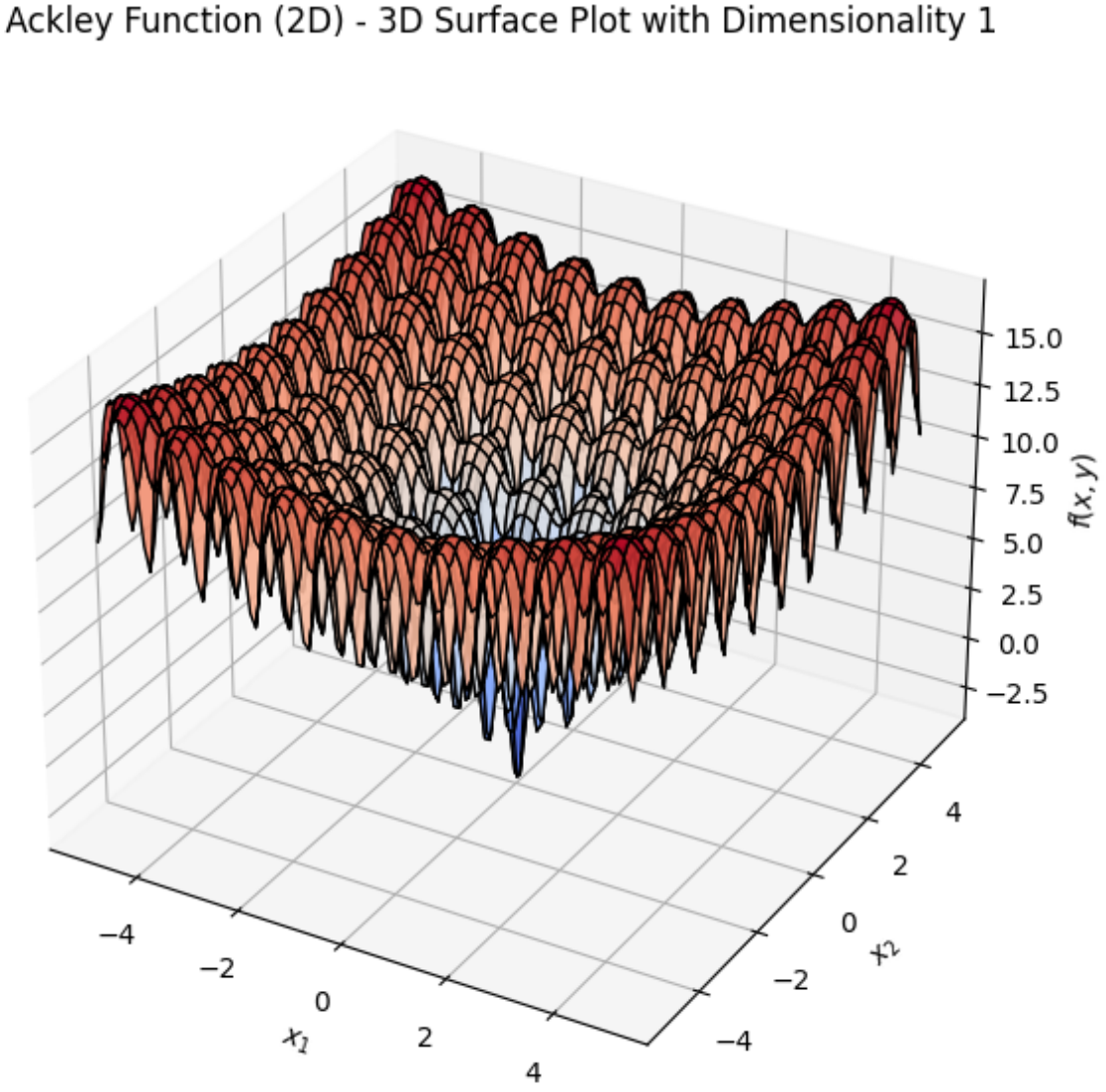

Ackley Function (2D) - Contour Plot with Dimensionality 0.1
1e8
0.0
−0.6
−1.2
−1.8
−2.4
−3.0
−3.6
−4.2
−4.8
Function Value
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1

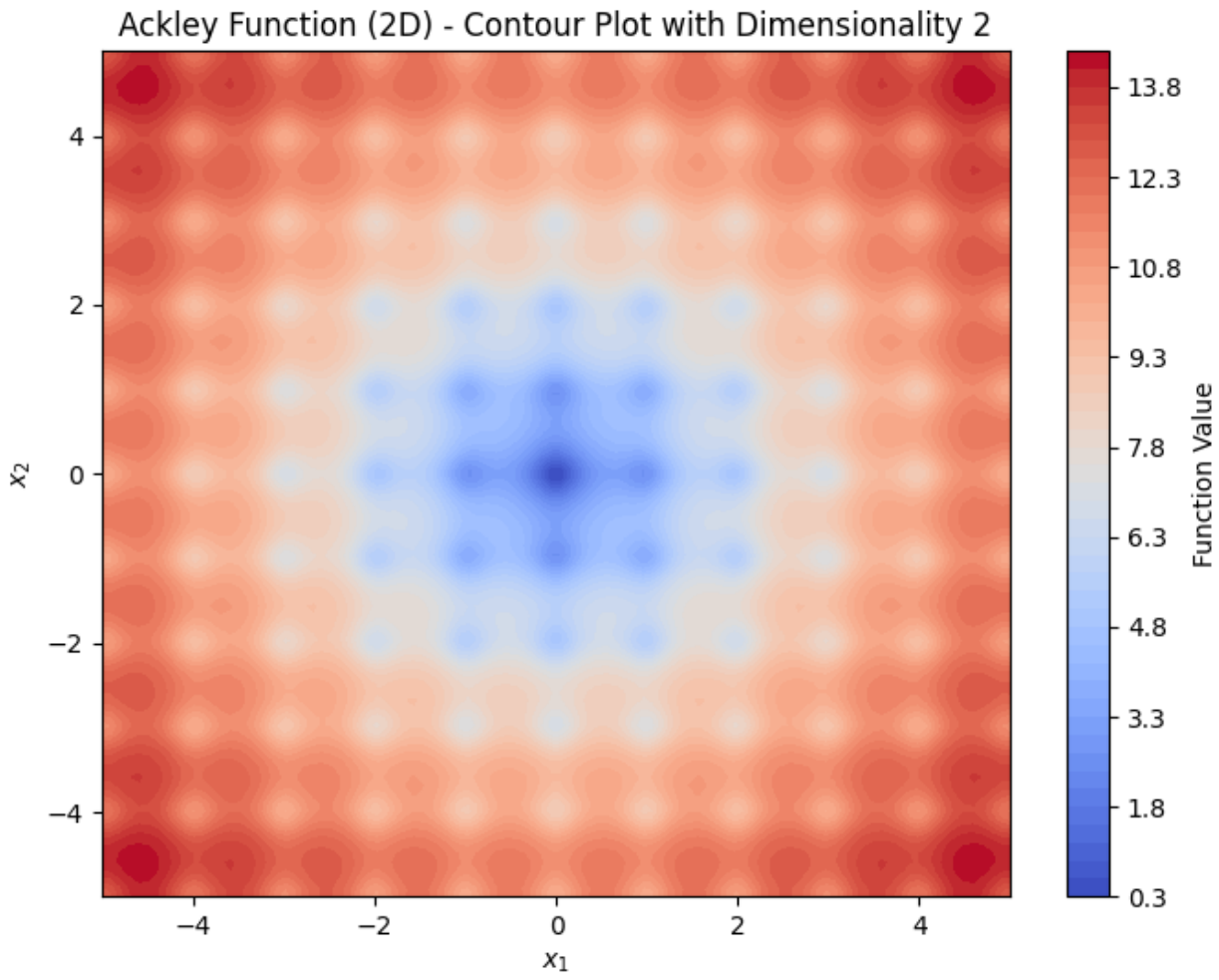

Ackley Function (2D) - 3D Surface Plot with Dimensionality 0.1
0
−1
−2
−3
−4
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

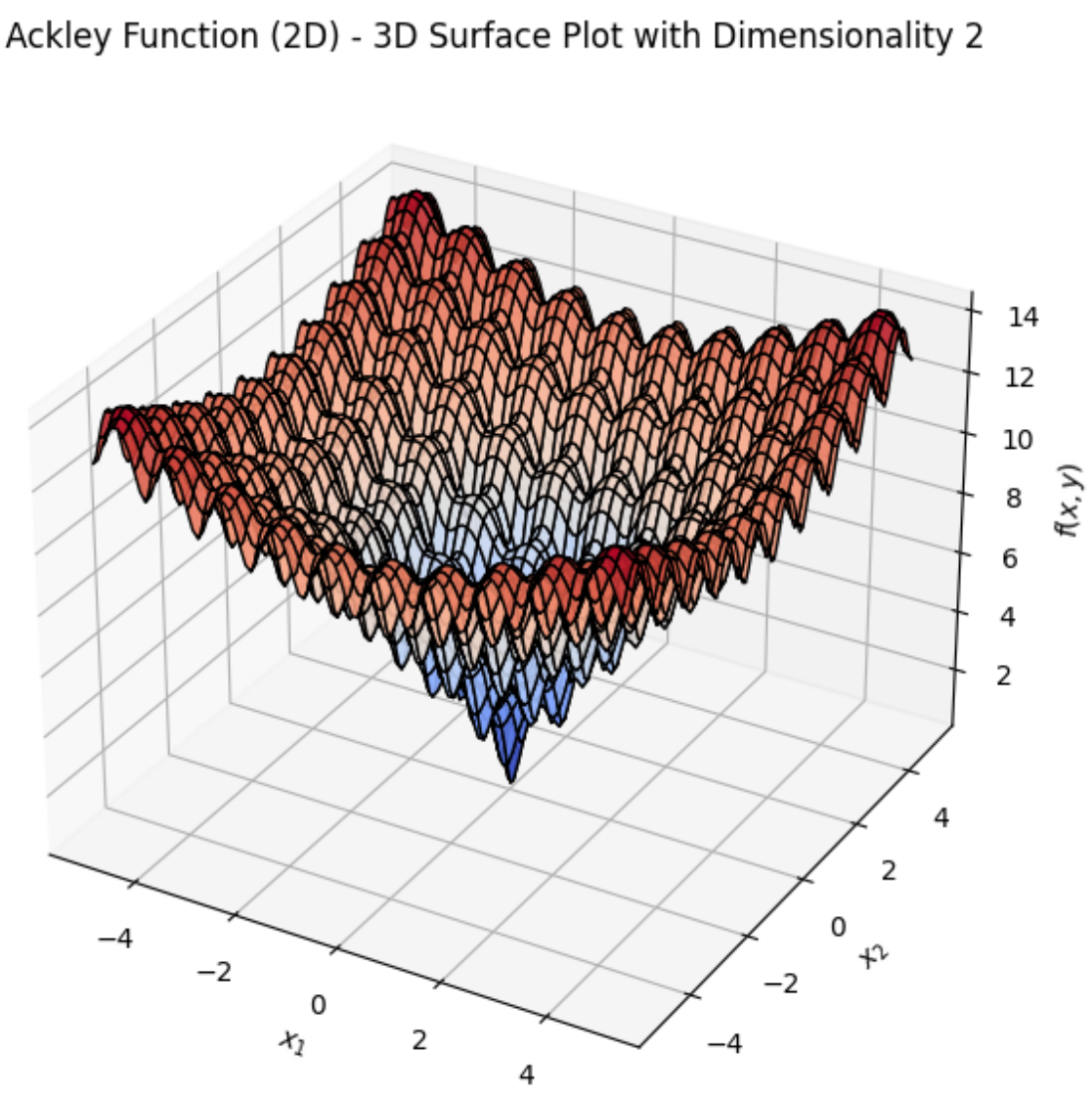

Ackley Function (2D) - Contour Plot with Dimensionality 0.5
x_1
x_2
Function Value
15.0
7.5
0.0
−7.5
−15.0
−22.5
−30.0
−37.5
4
2
0
−2
−4

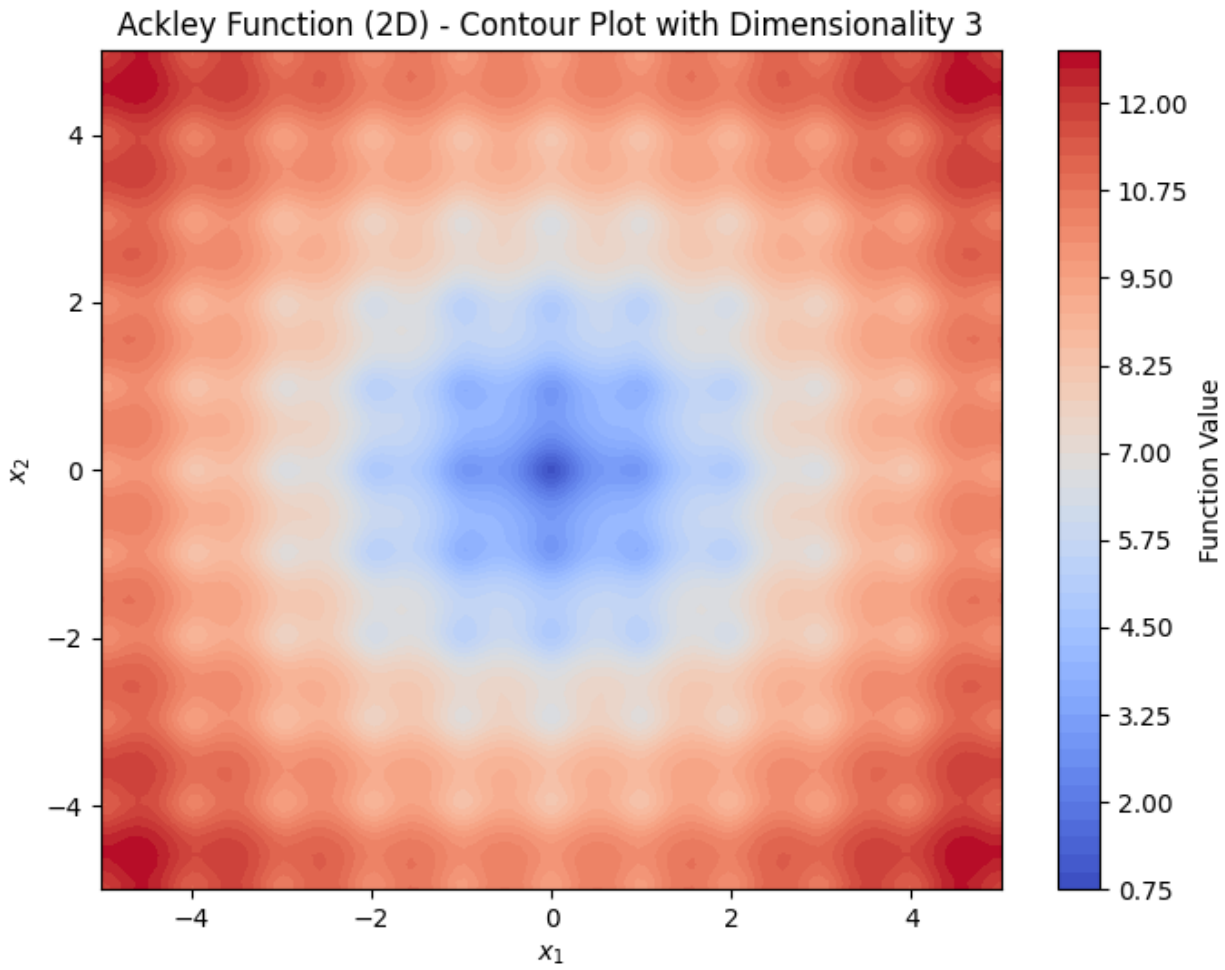

Ackley Function (2D) - 3D Surface Plot with Dimensionality 0.5
10
0
−10
−20
−30
−40
f(x, y)
−4
−2
0
2
4
x1
−4
−2
0
2
4
x2

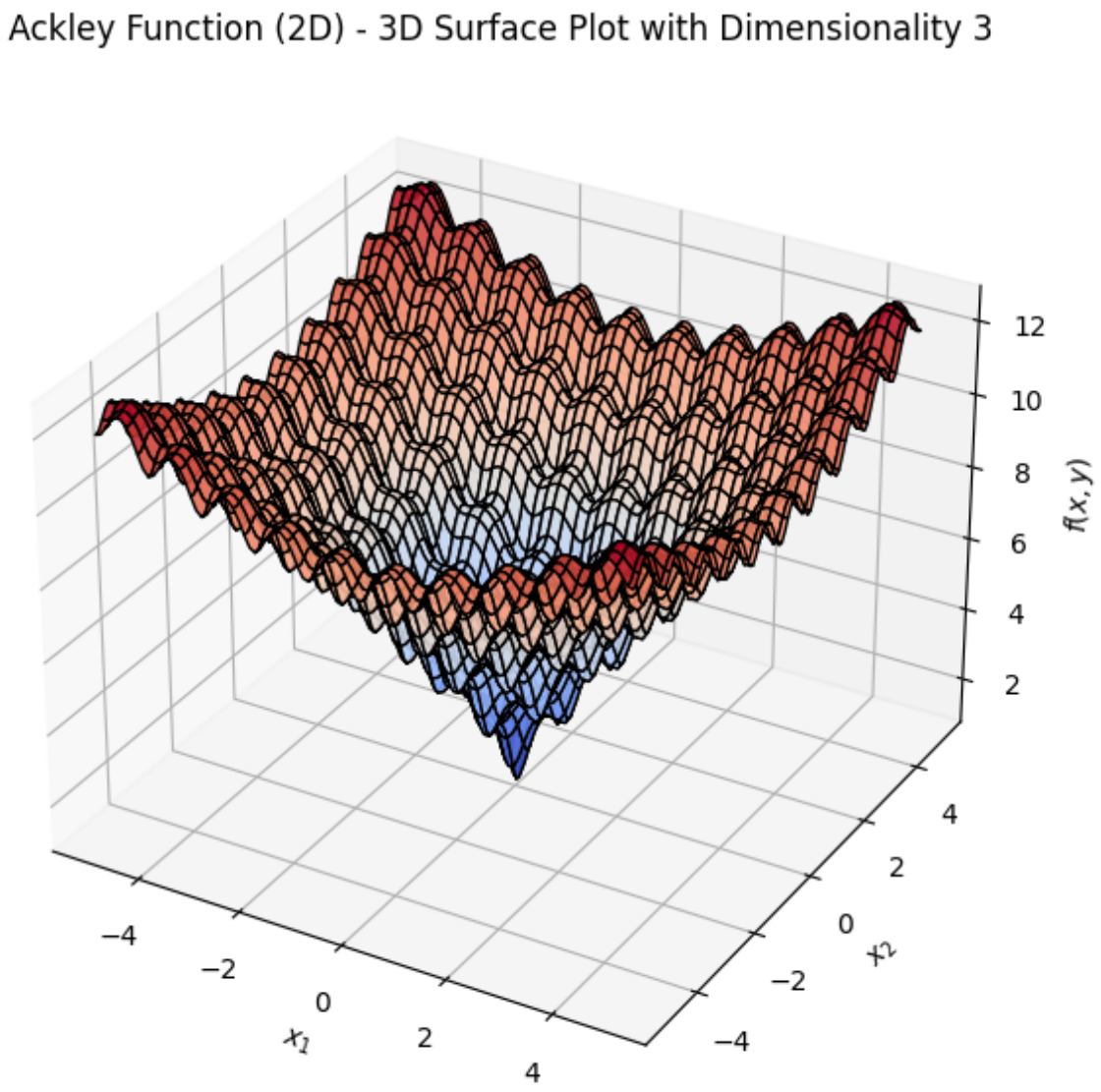

Ackley Function (2D) - Contour Plot with Dimensionality 1
x_1
x_2
Function Value
−4
−2
0
2
4
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5

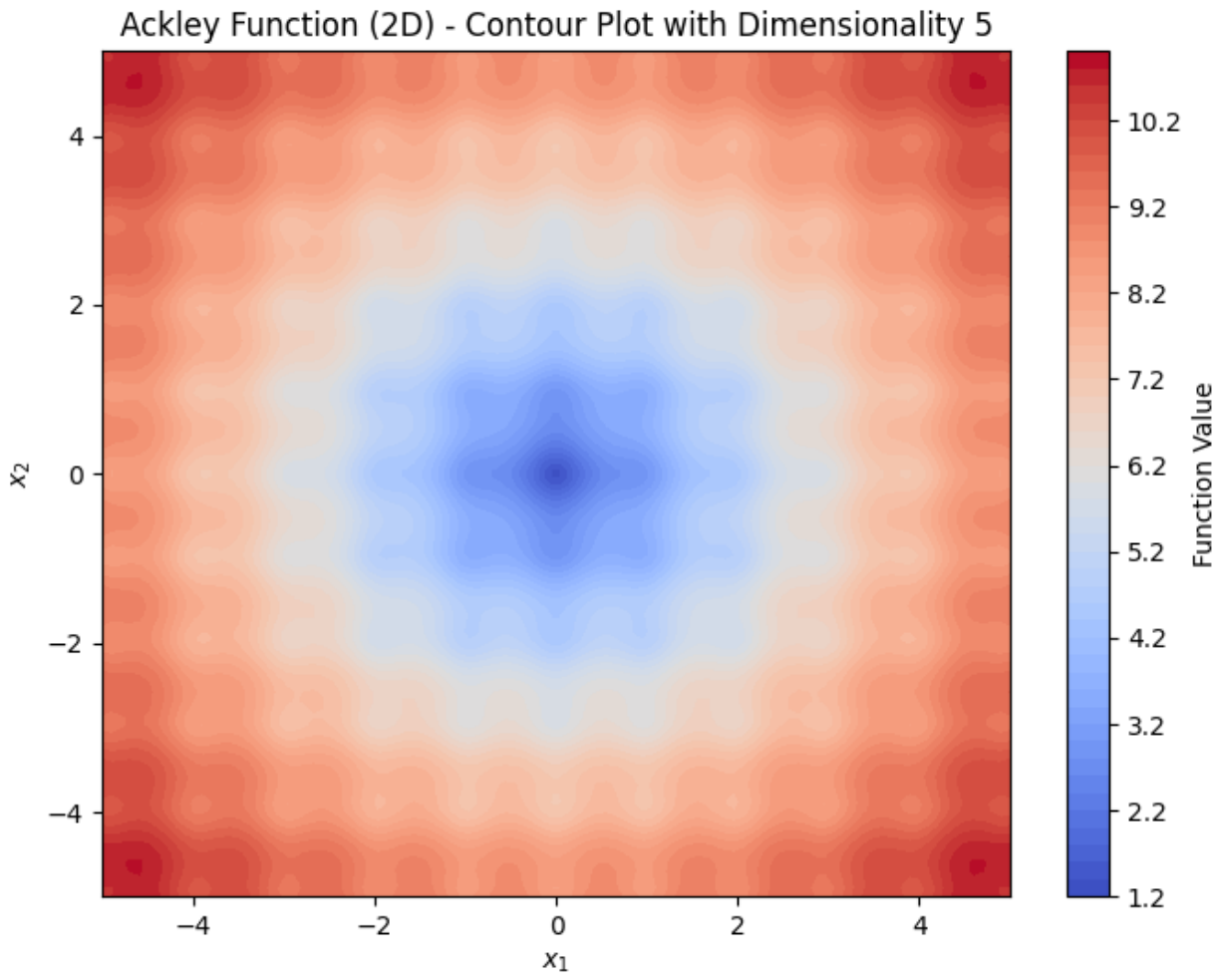

Ackley Function (2D) - 3D Surface Plot with Dimensionality 1
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
f(x, y)
−4
−2
0
2
4
x_1
−4
−2
0
2
4
x_2

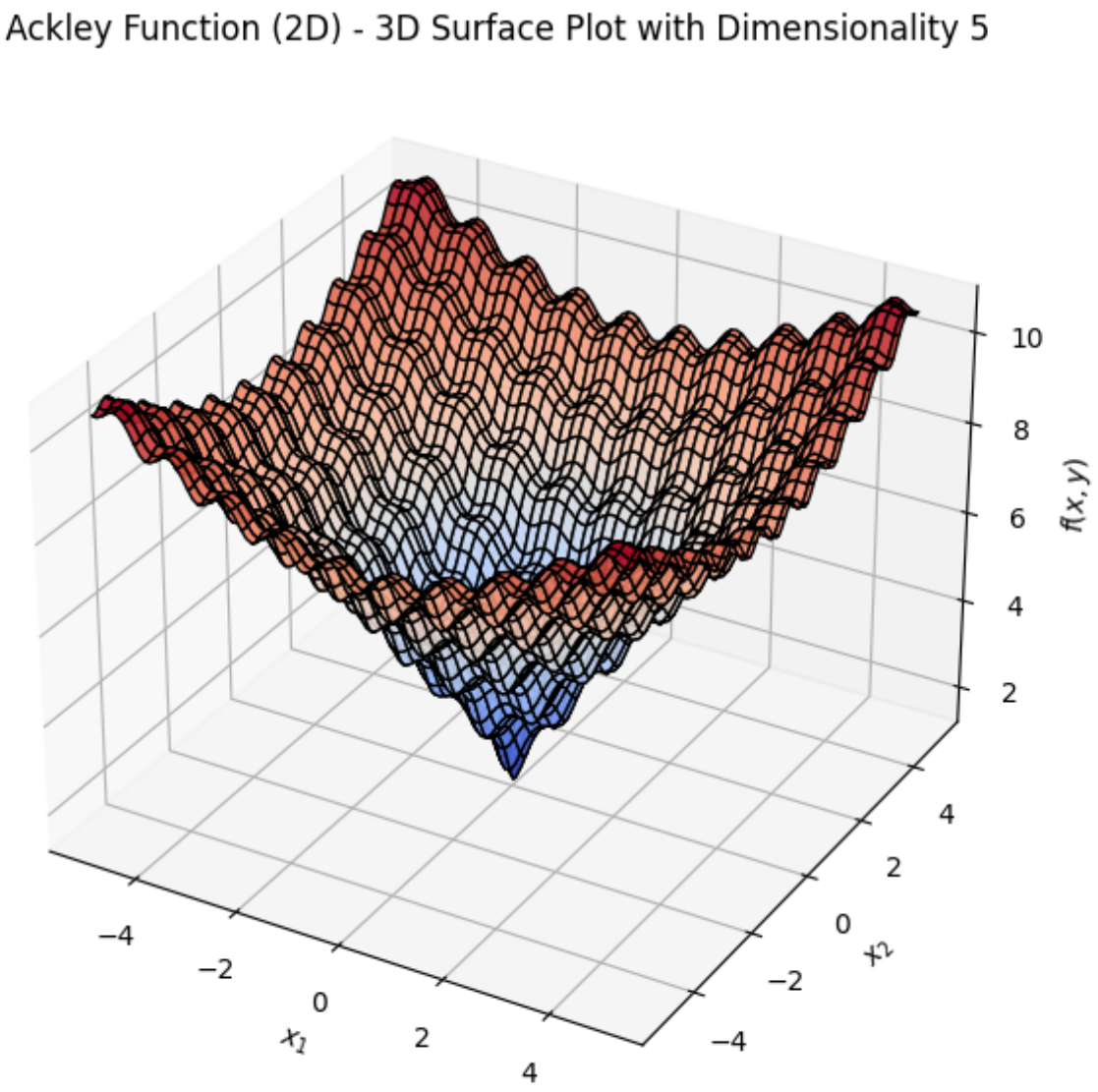

Ackley Function (2D) - Contour Plot with Dimensionality 2
x_1
x_2
Function Value
13.8
12.3
10.8
9.3
7.8
6.3
4.8
3.3
1.8
0.3
−4
−2
0
2
4

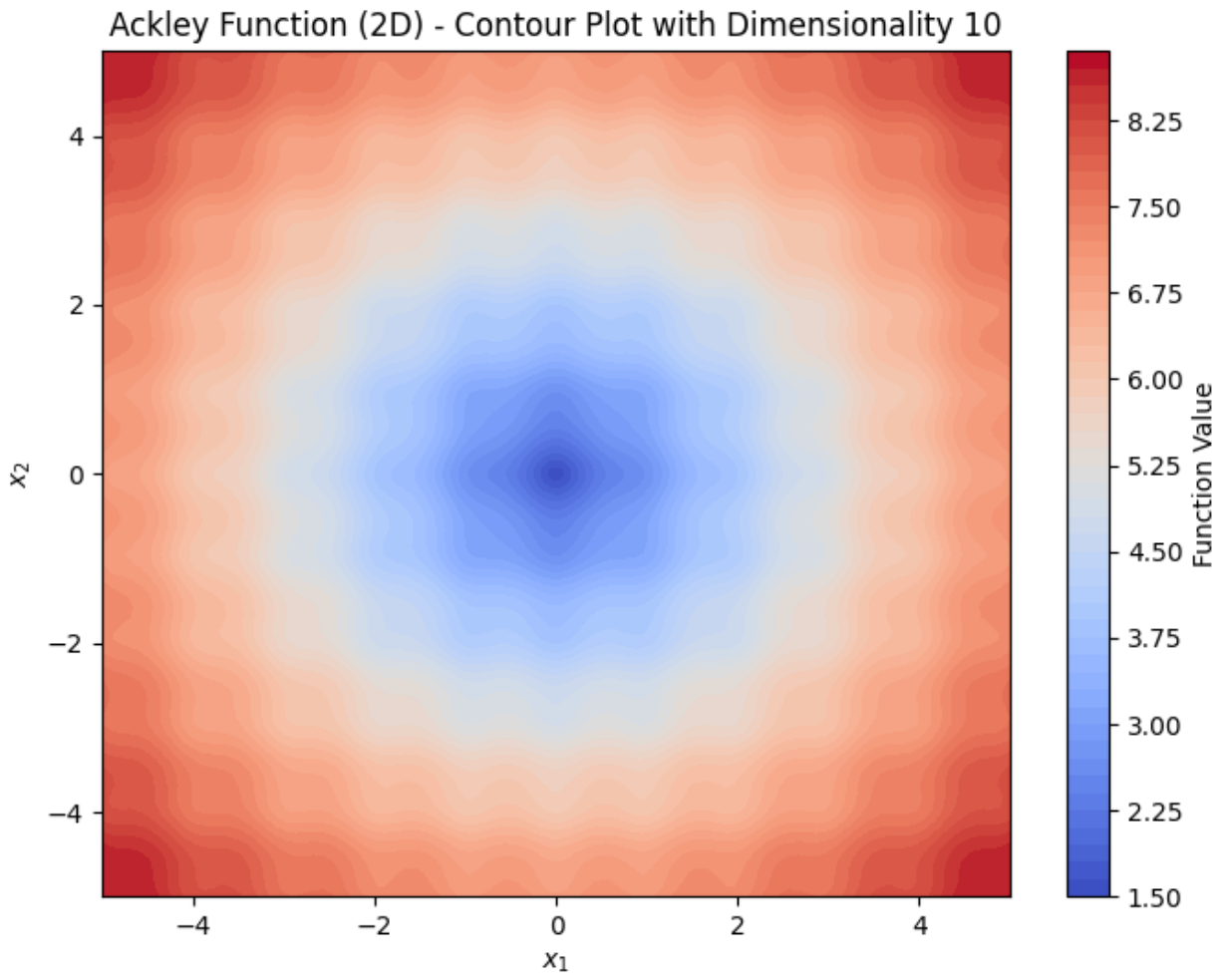

Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
14
12
10
8
6
4
2
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

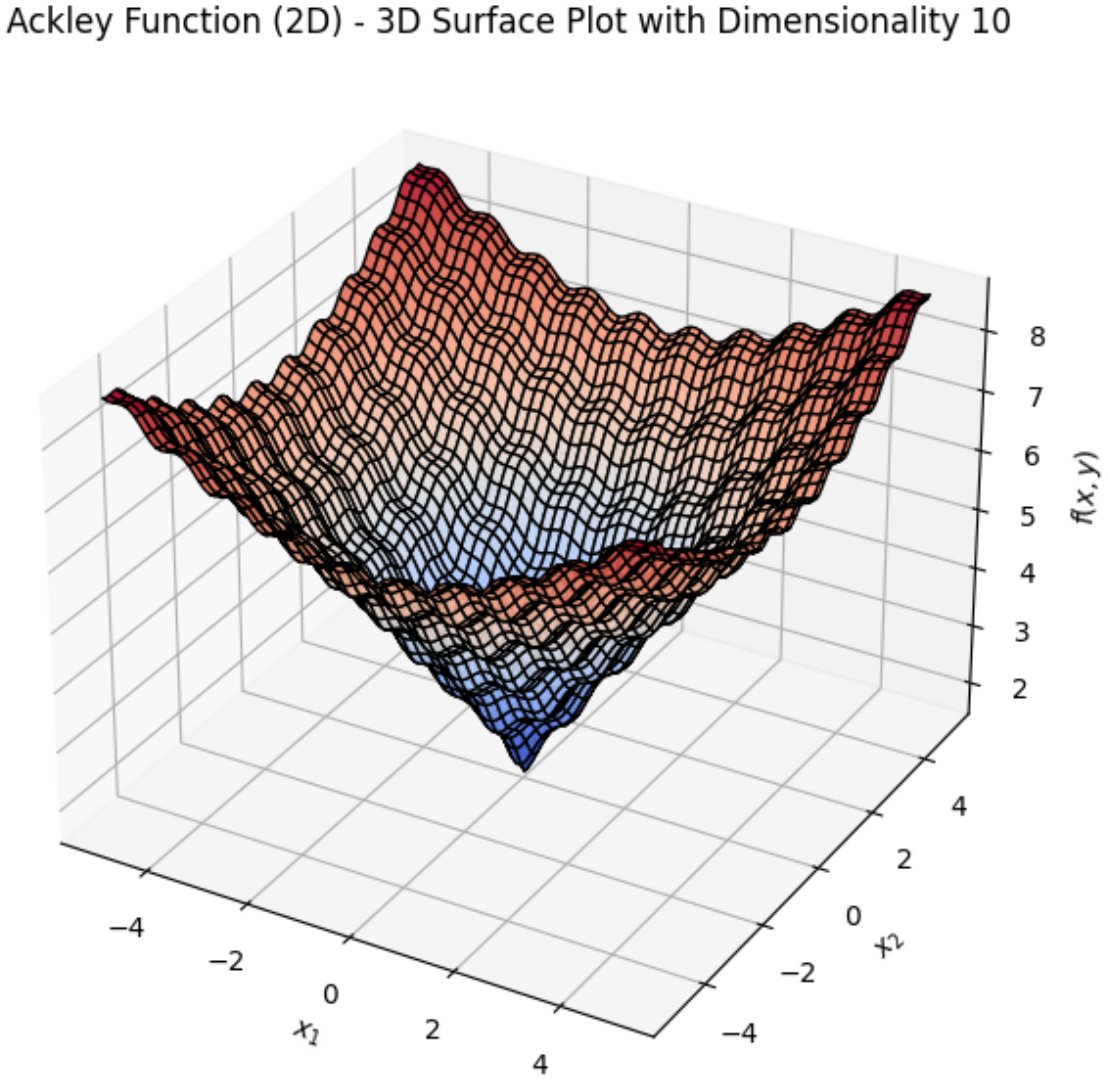

Ackley Function (2D) - Contour Plot with Dimensionality 3
x_2
x_1
4
2
0
−2
−4
−4
−2
0
2
4
12.00
10.75
9.50
8.25
7.00
5.75
4.50
3.25
2.00
0.75
Function Value

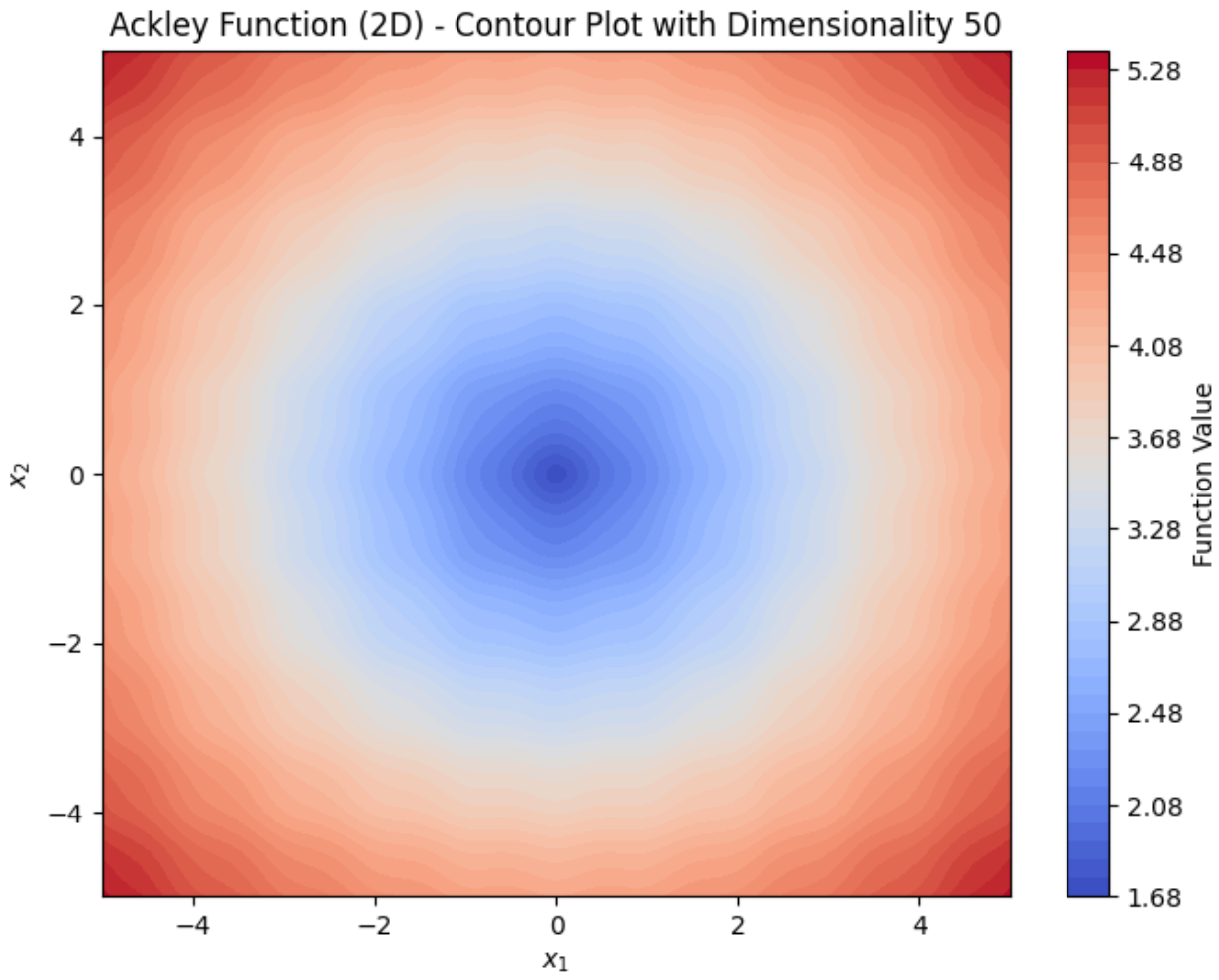

Ackley Function (2D) - 3D Surface Plot with Dimensionality 3
12
10
8
6
4
2
f(x, y)
−4
−2
0
2
4
x1
−4
−2
0
2
4
x2

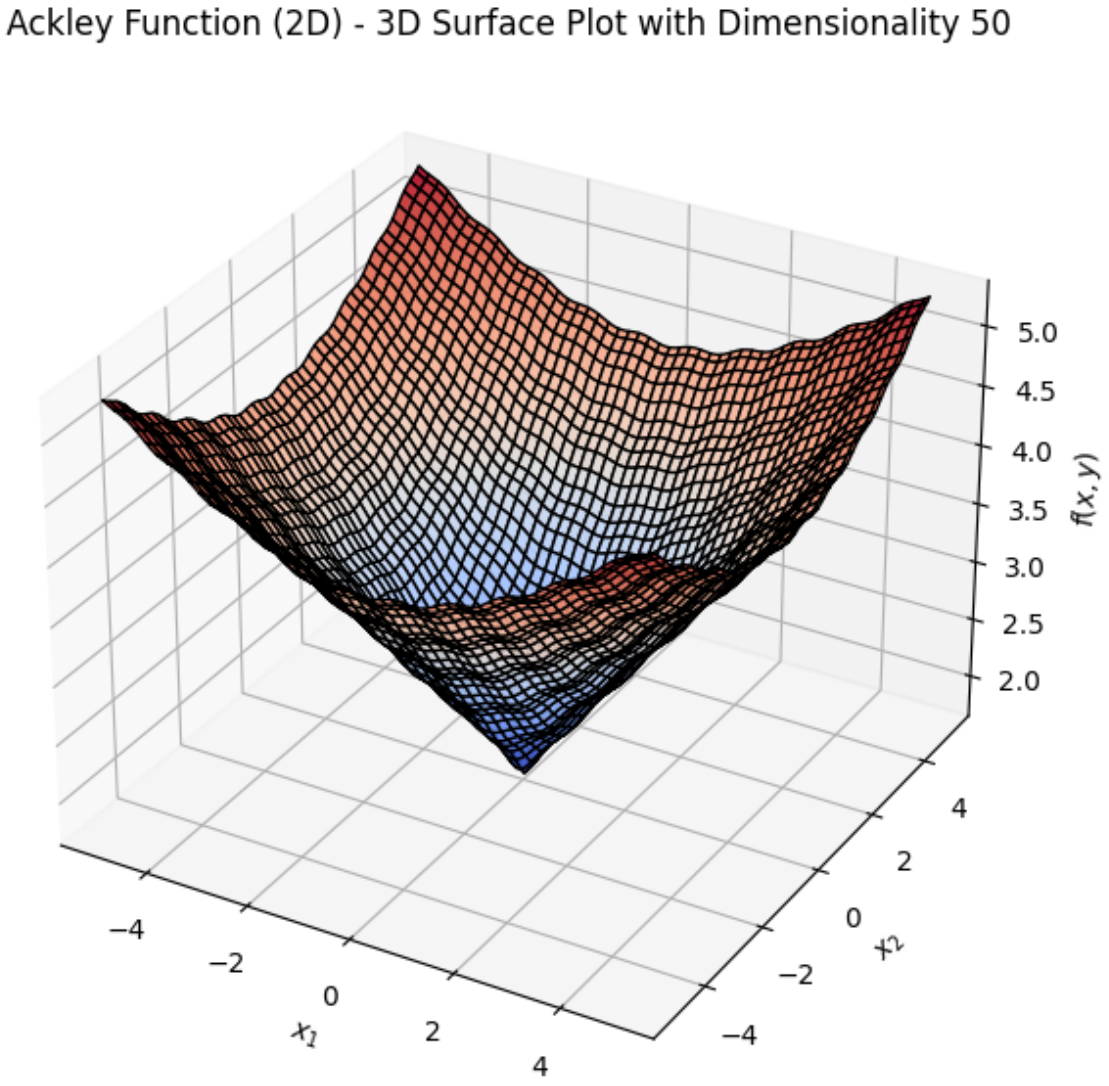

Ackley Function (2D) - Contour Plot with Dimensionality 5
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
10.2
9.2
8.2
7.2
6.2
5.2
4.2
3.2
2.2
1.2
Function Value

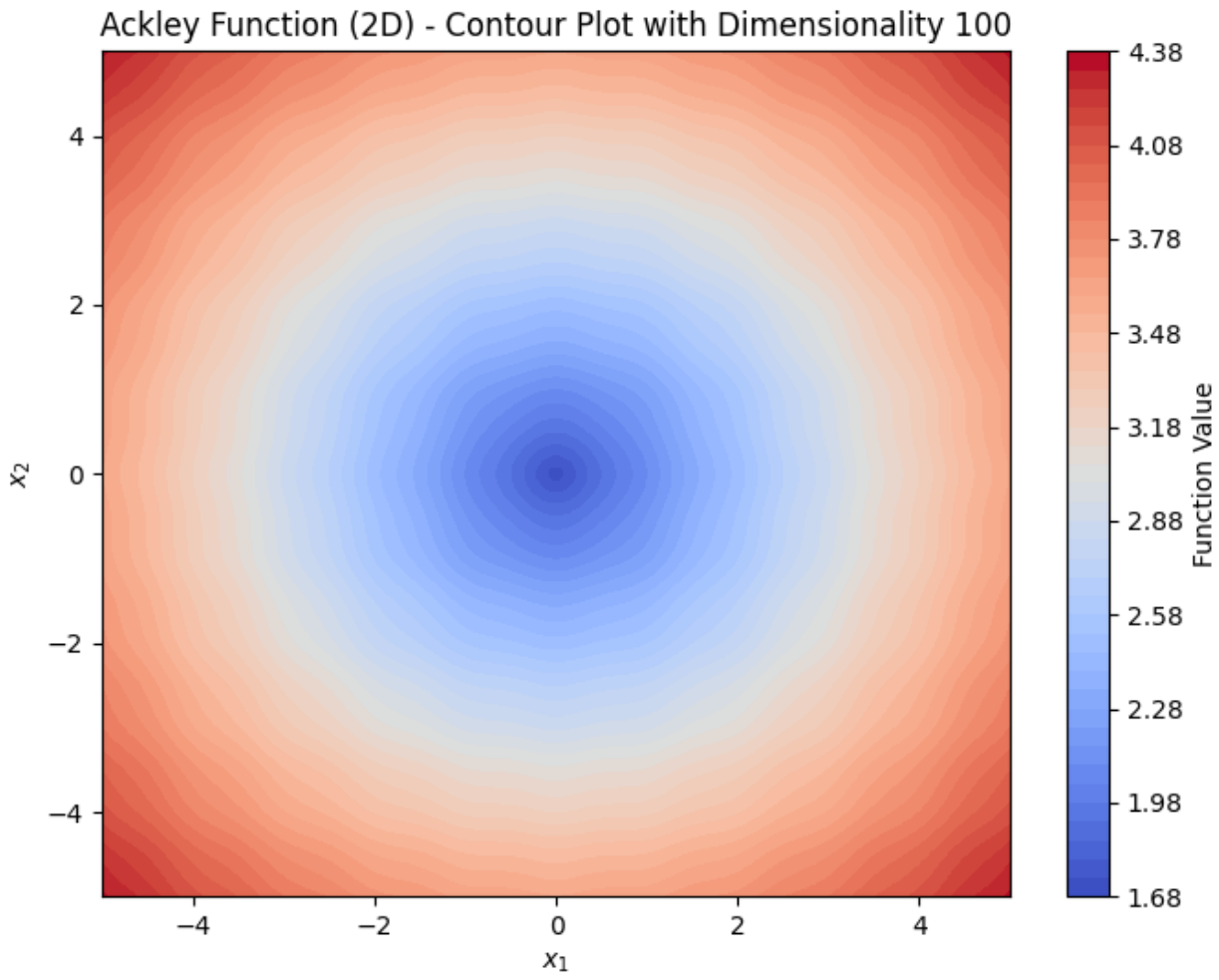

Ackley Function (2D) - 3D Surface Plot with Dimensionality 5
10
8
6
4
2
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

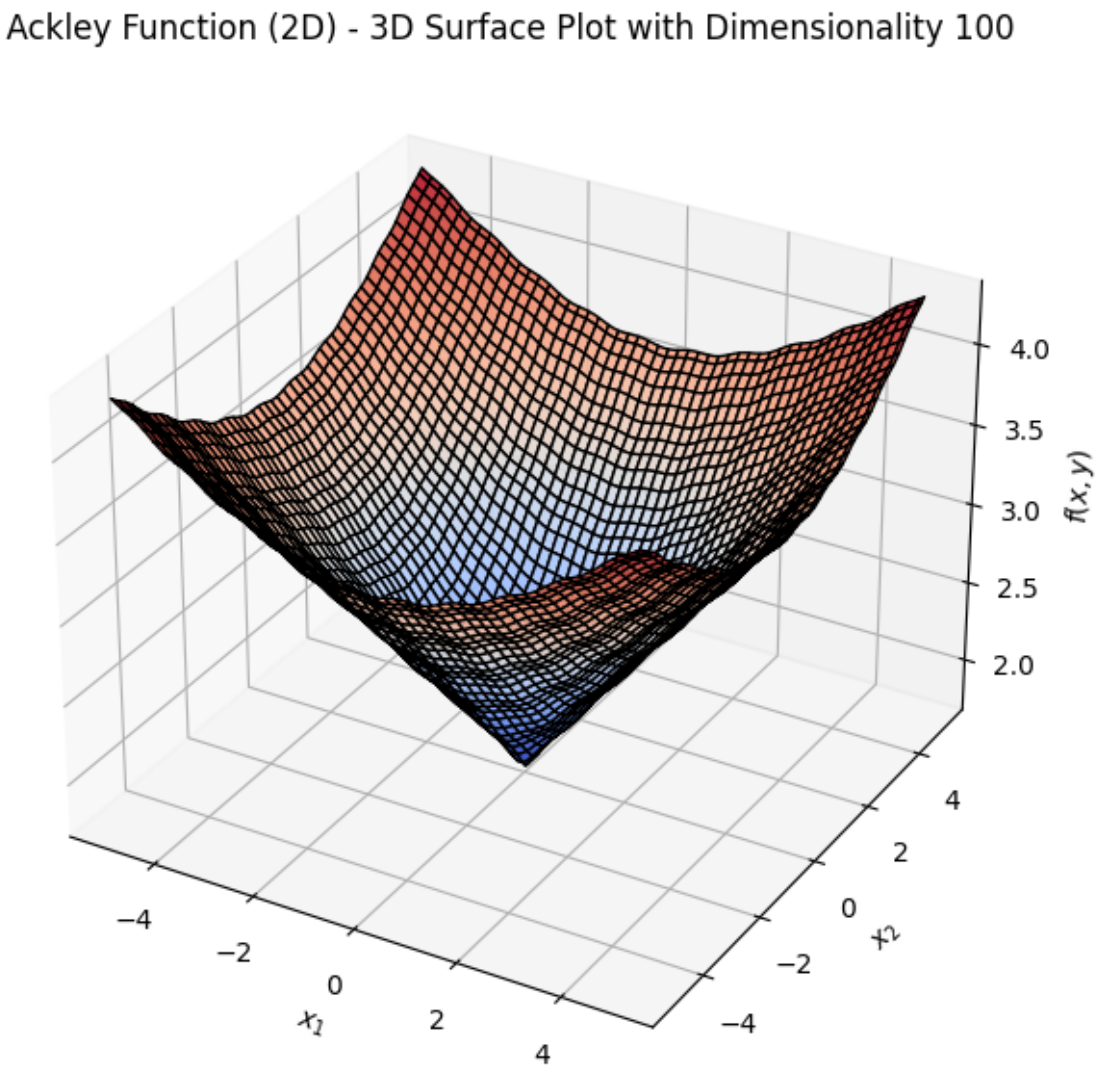

Ackley Function (2D) - Contour Plot with Dimensionality 10
x_1
x_2
Function Value
8.25
7.50
6.75
6.00
5.25
4.50
3.75
3.00
2.25
1.50

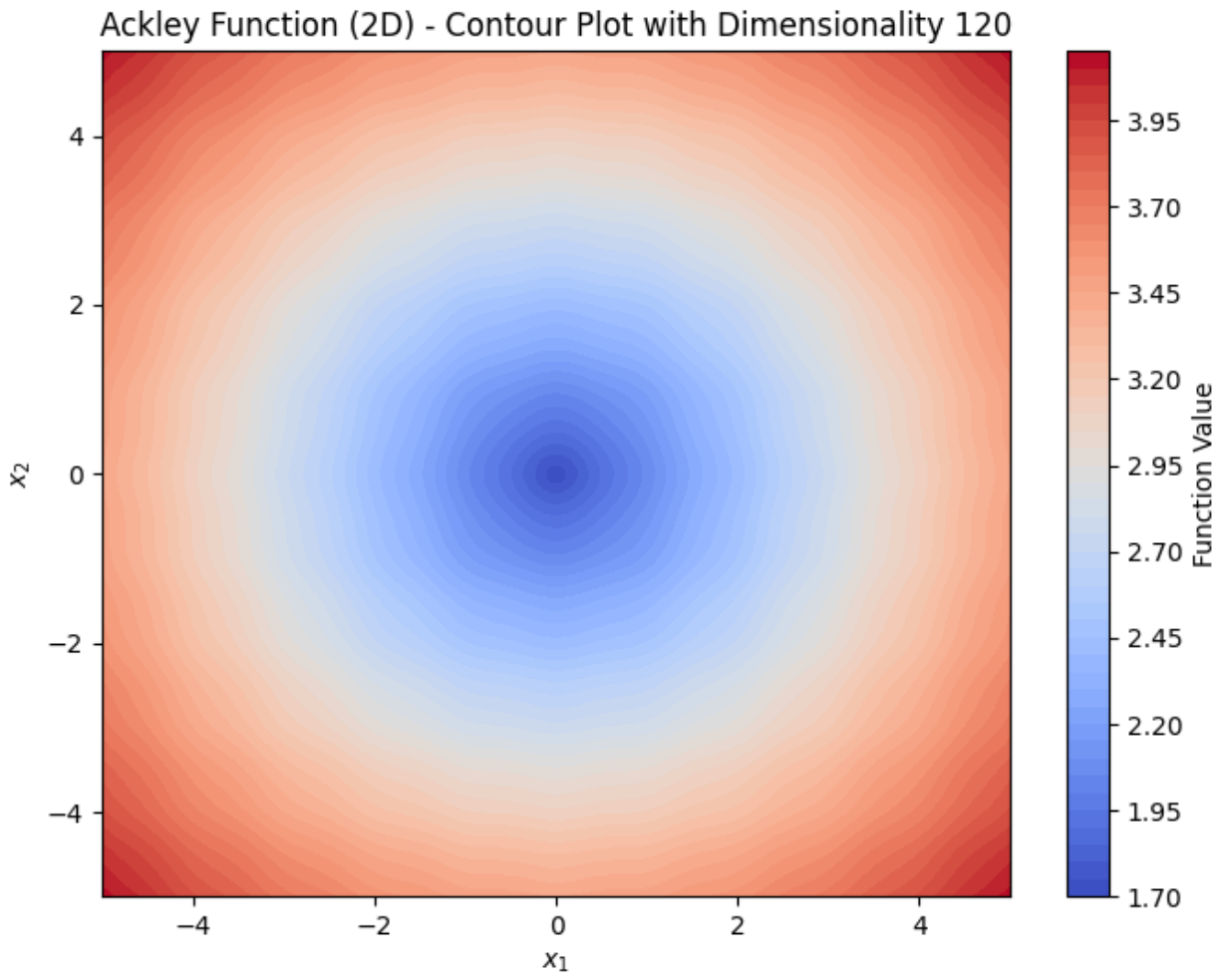

Ackley Function (2D) - 3D Surface Plot with Dimensionality 10
8
7
6
5
4
3
2
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

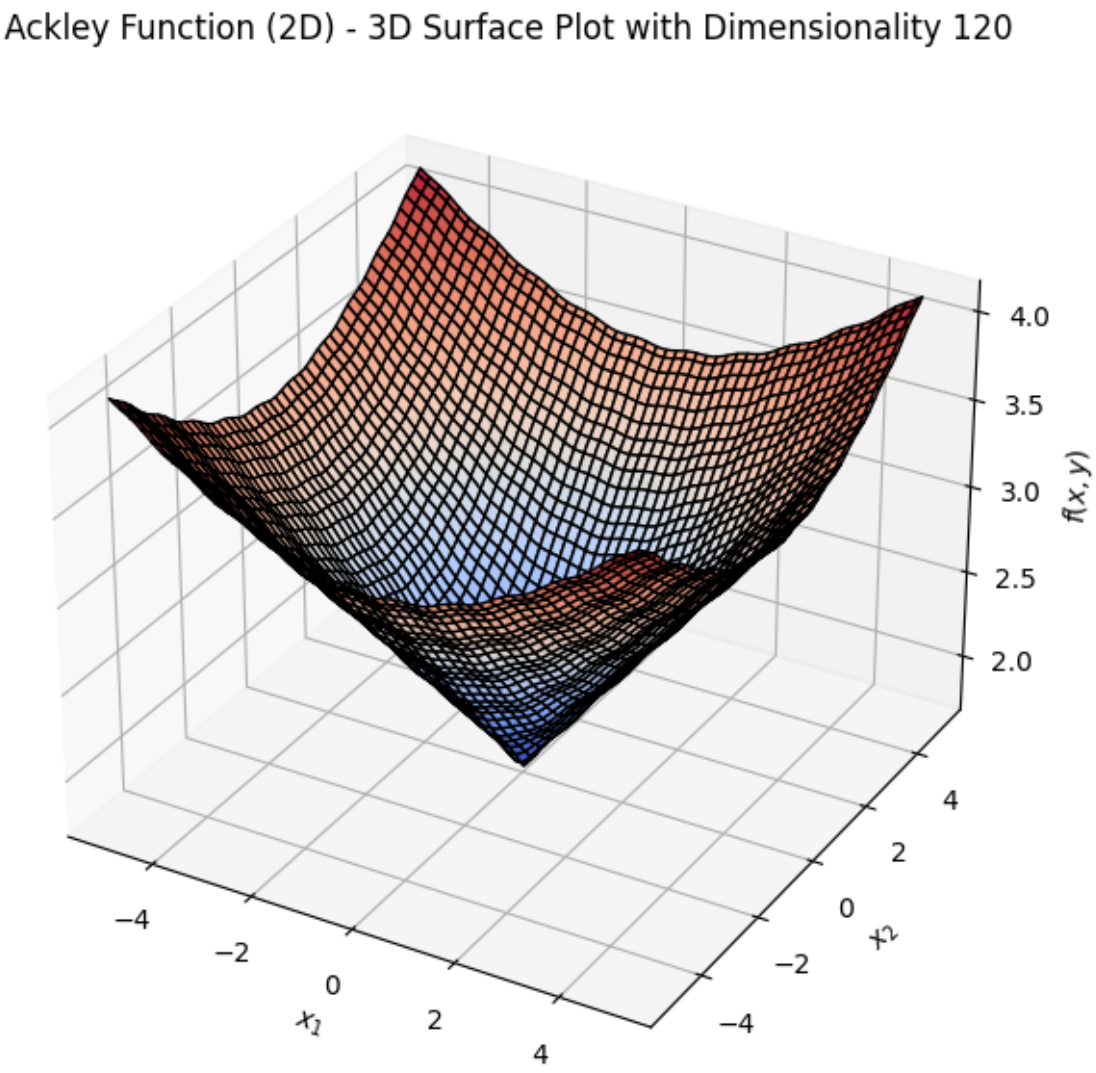

Ackley Function (2D) - Contour Plot with Dimensionality 50
x_1
x_2
Function Value
5.28
4.88
4.48
4.08
3.68
3.28
2.88
2.48
2.08
1.68

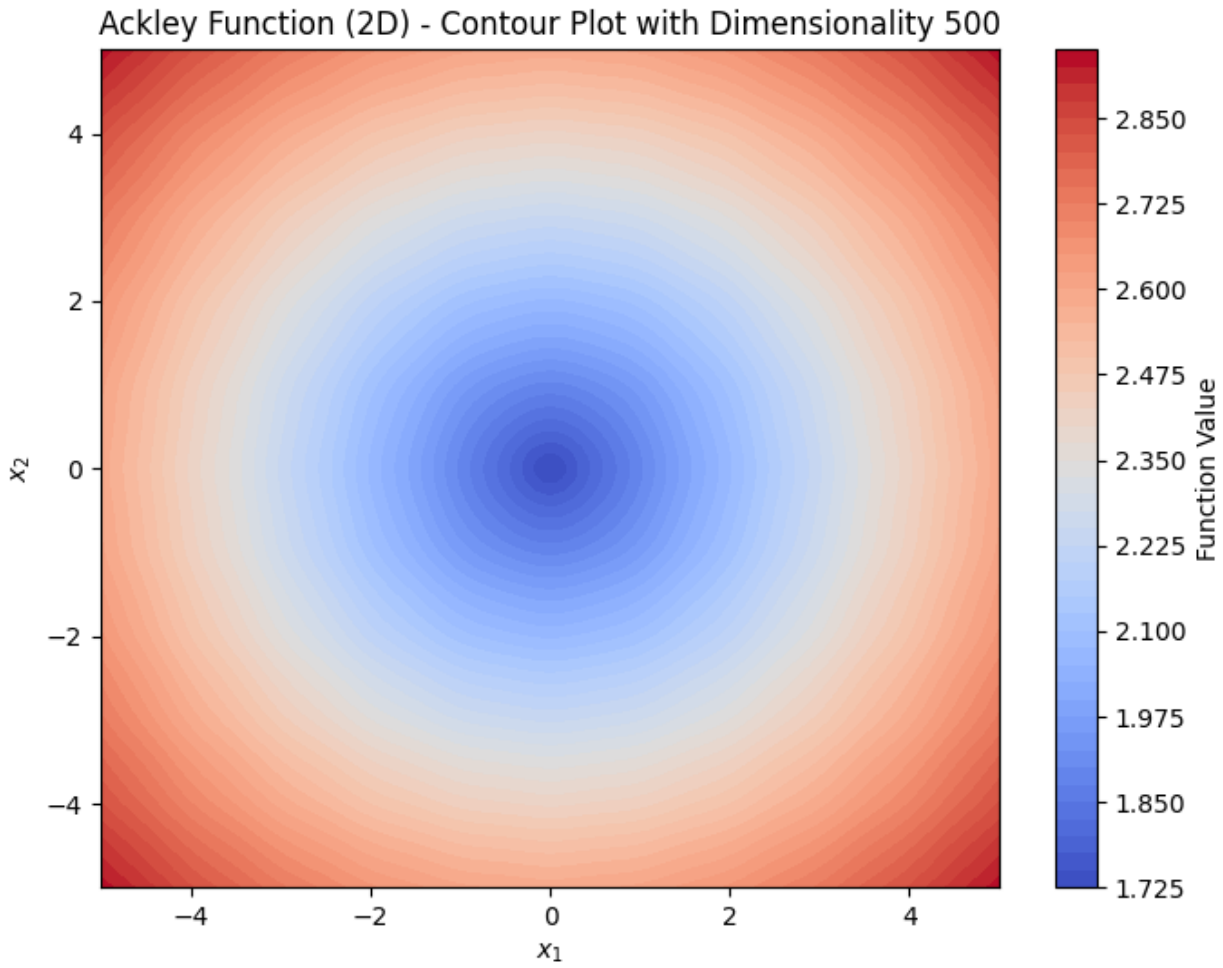

Ackley Function (2D) - 3D Surface Plot with Dimensionality 50
5.0
4.5
4.0
3.5
3.0
2.5
2.0
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

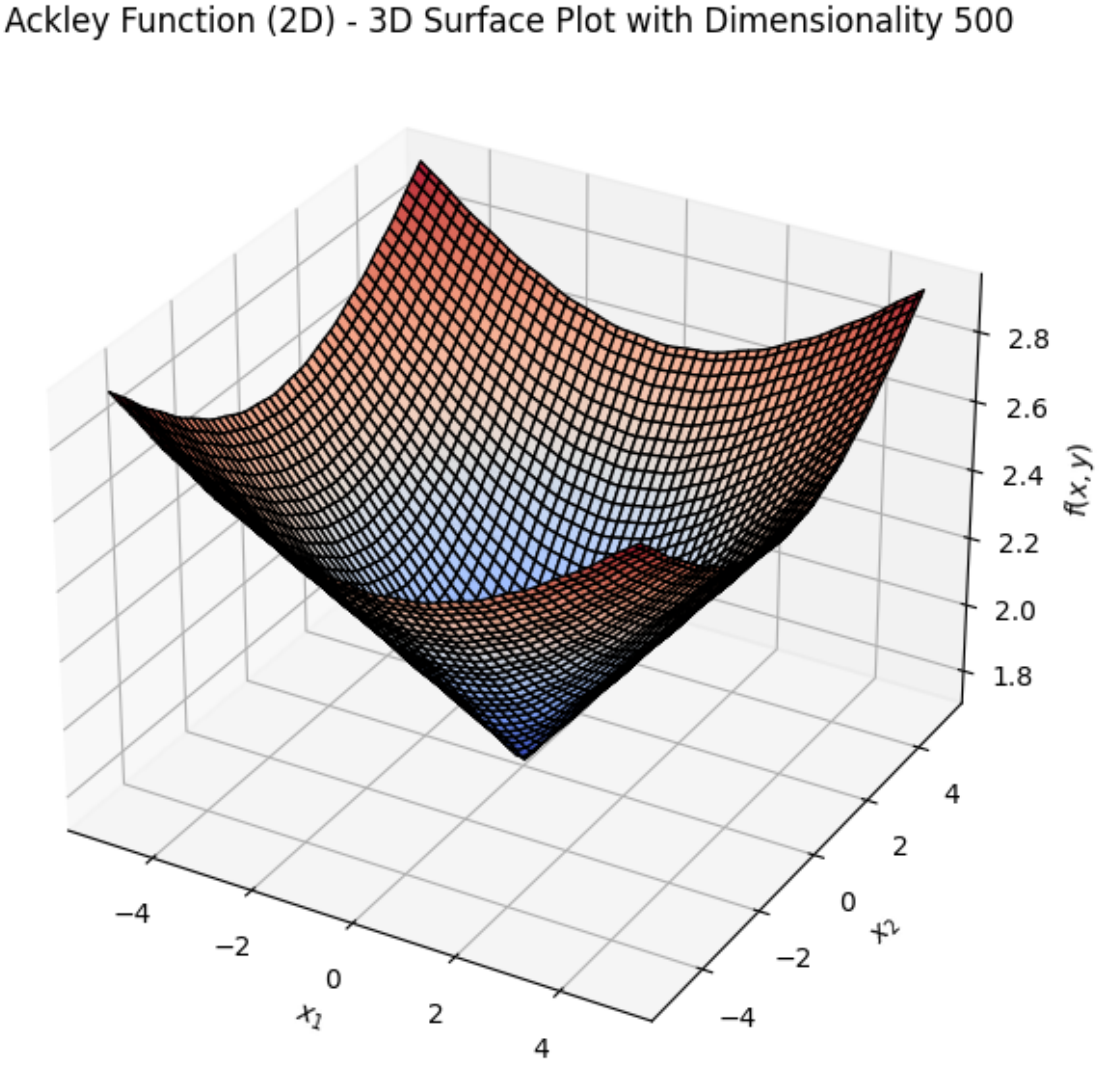

Ackley Function (2D) - Contour Plot with Dimensionality 100
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
4.38
4.08
3.78
3.48
3.18
2.88
2.58
2.28
1.98
1.68
Function Value

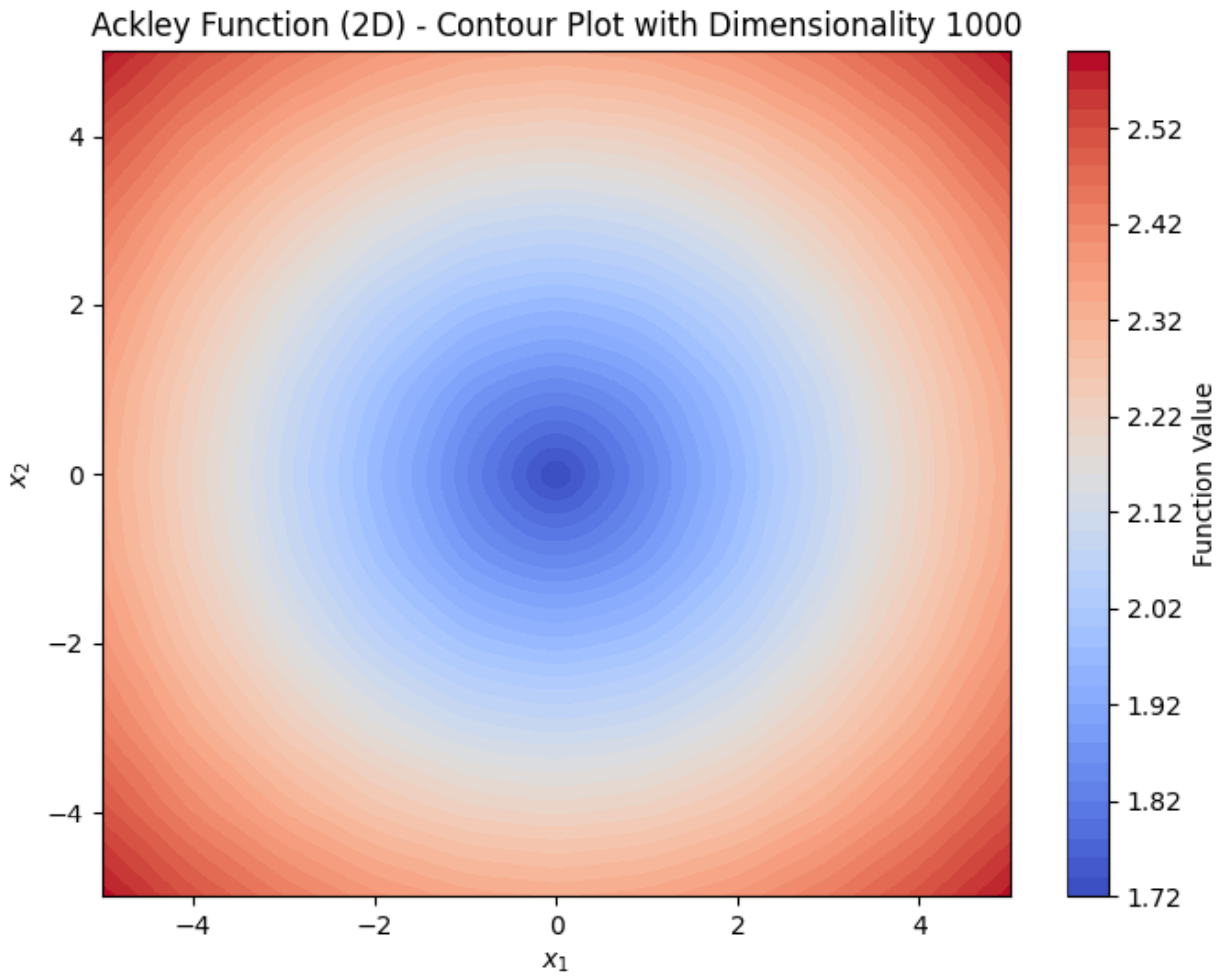

Ackley Function (2D) - 3D Surface Plot with Dimensionality 100
4.0
3.5
3.0
2.5
2.0
f(x, y)
−4
−2
0
2
4
x1
−4
−2
0
2
4
x2

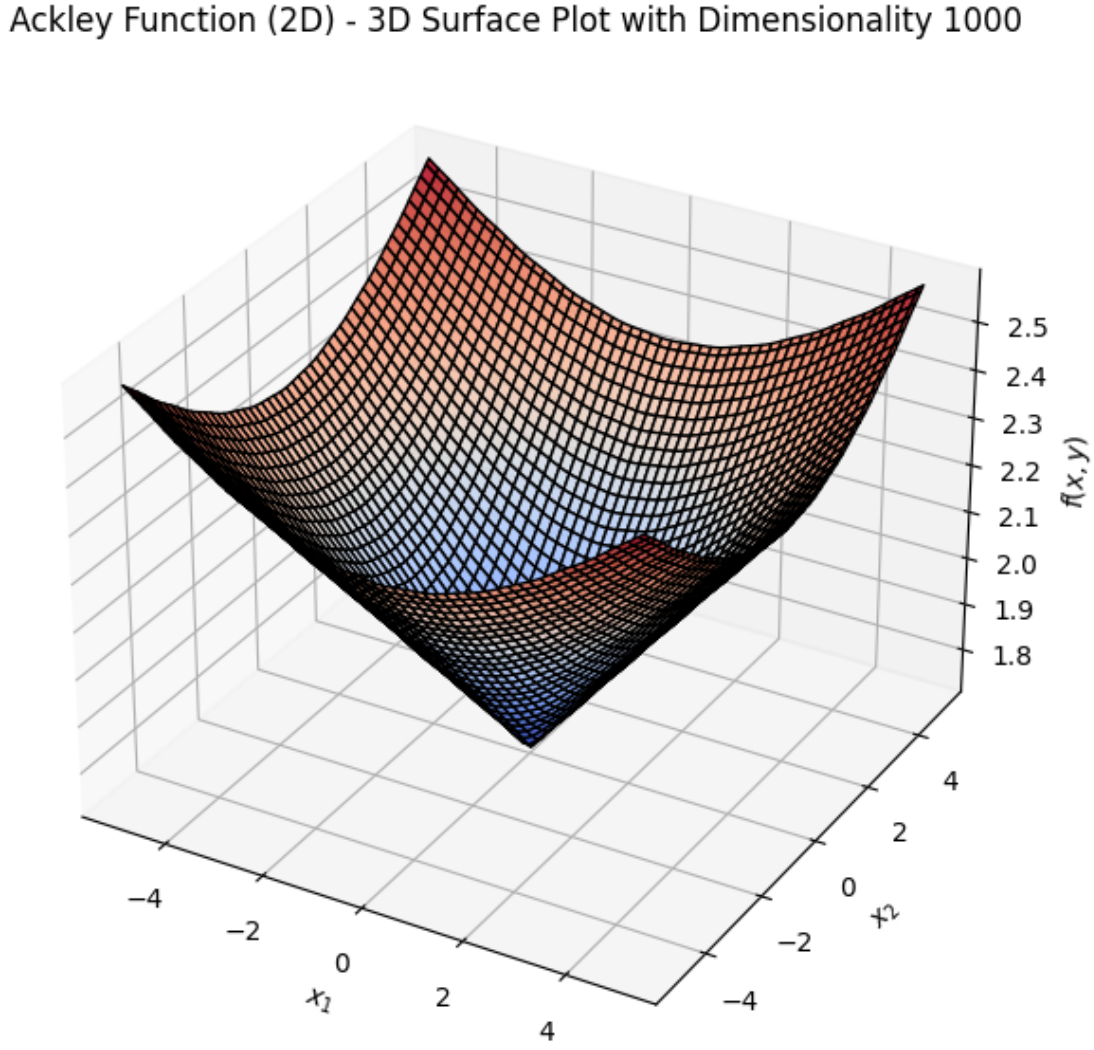

Ackley Function (2D) - Contour Plot with Dimensionality 120
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
3.95
3.70
3.45
3.20
2.95
2.70
2.45
2.20
1.95
1.70
Function Value

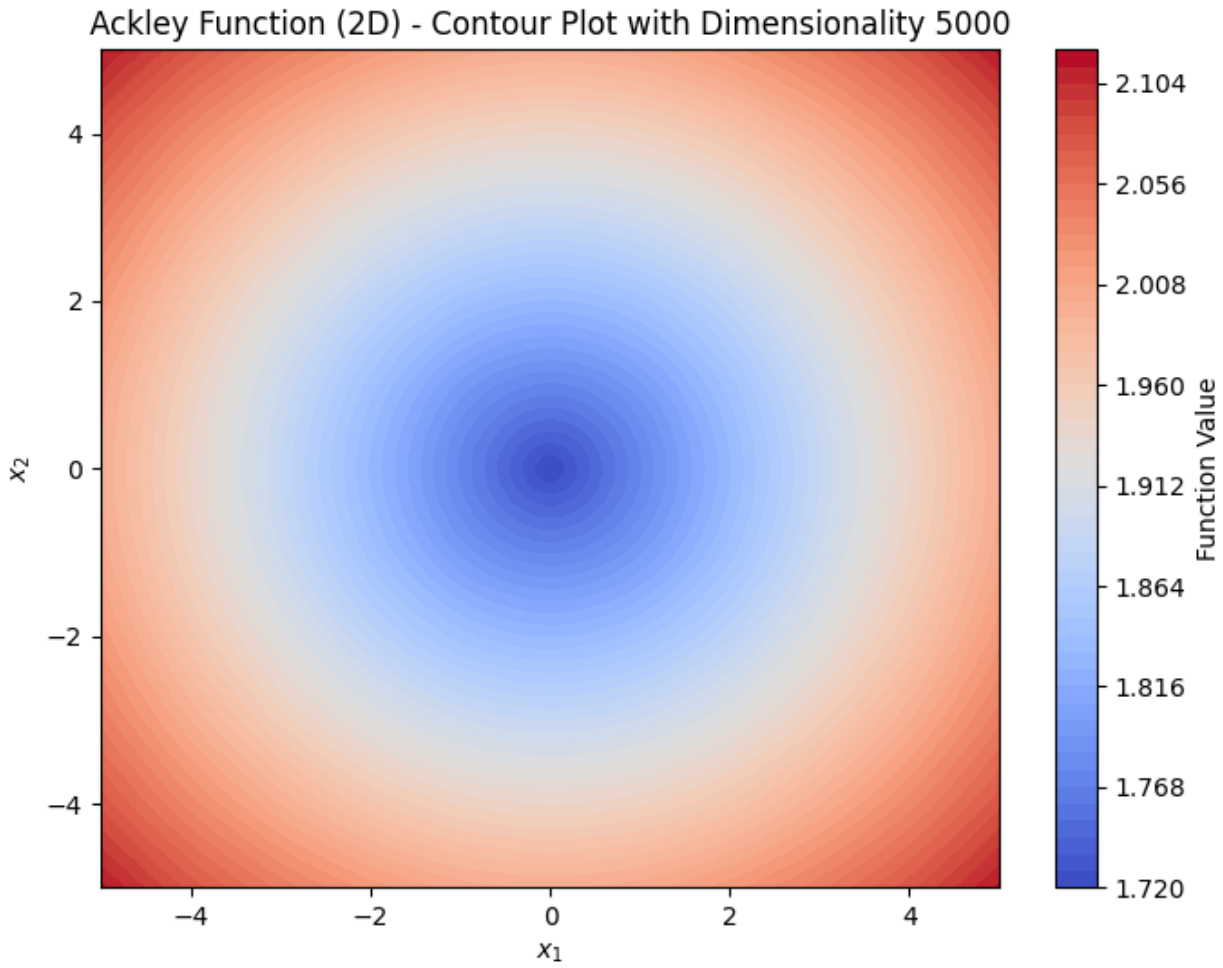

Ackley Function (2D) - 3D Surface Plot with Dimensionality 120
4.0
3.5
3.0
2.5
2.0
f(x, y)
−4
−2
0
2
4
x_1
−4
−2
0
2
4
x_2

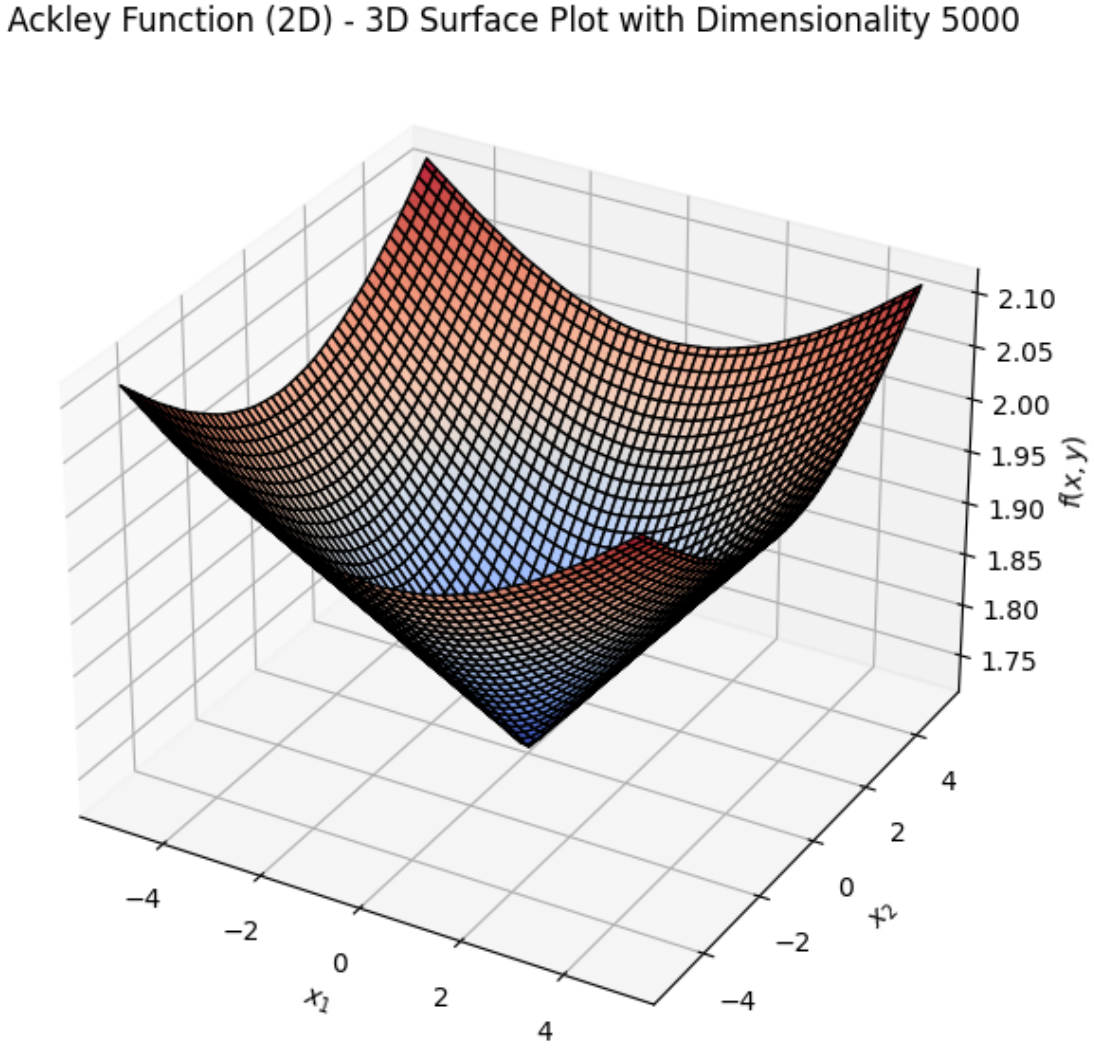

Ackley Function (2D) - Contour Plot with Dimensionality 500
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
2.850
2.725
2.600
2.475
2.350
2.225
2.100
1.975
1.850
1.725
Function Value

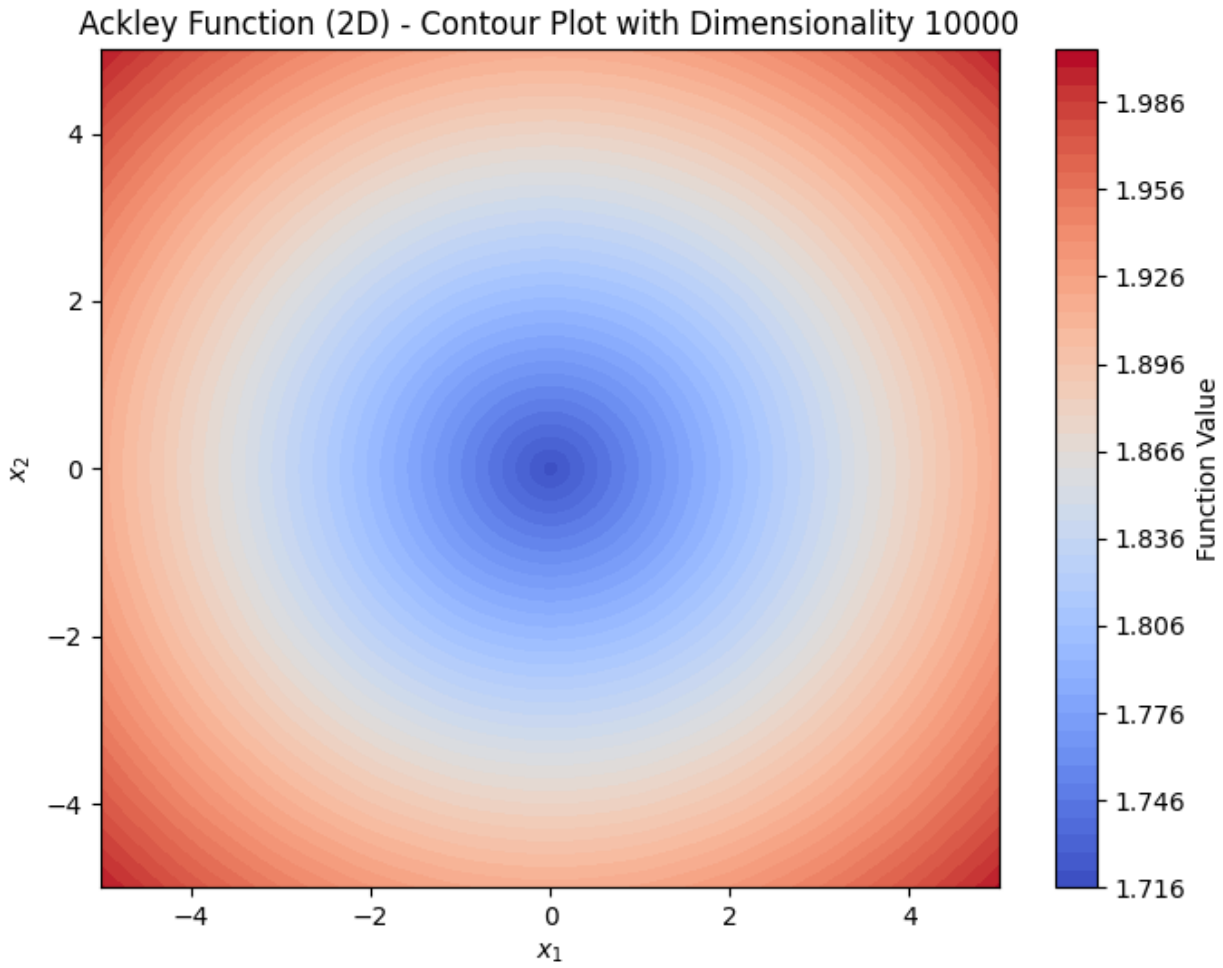

Ackley Function (2D) - 3D Surface Plot with Dimensionality 500
2.8
2.6
2.4
2.2
2.0
1.8
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

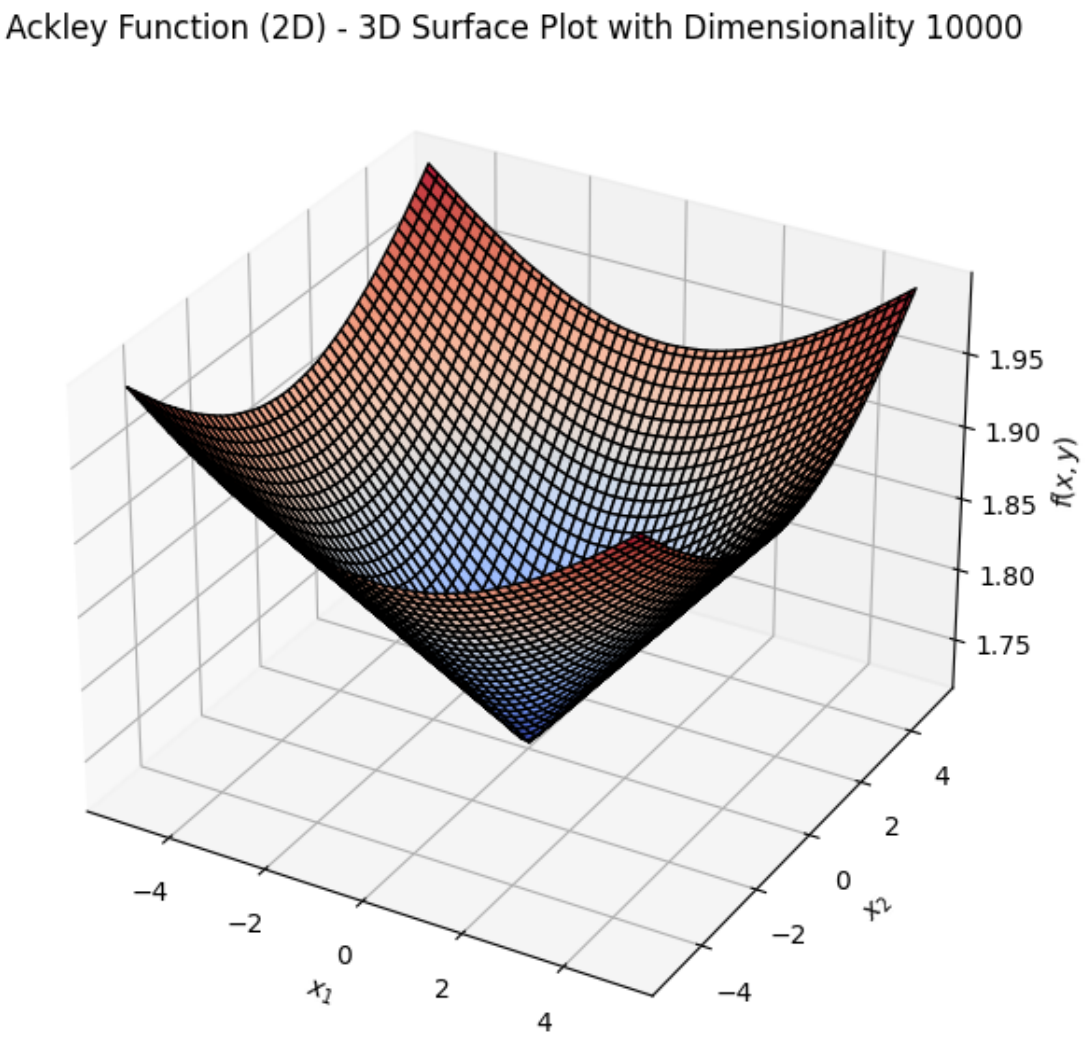

Ackley Function (2D) - Contour Plot with Dimensionality 1000
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
2.52
2.42
2.32
2.22
2.12
2.02
1.92
1.82
1.72
Function Value

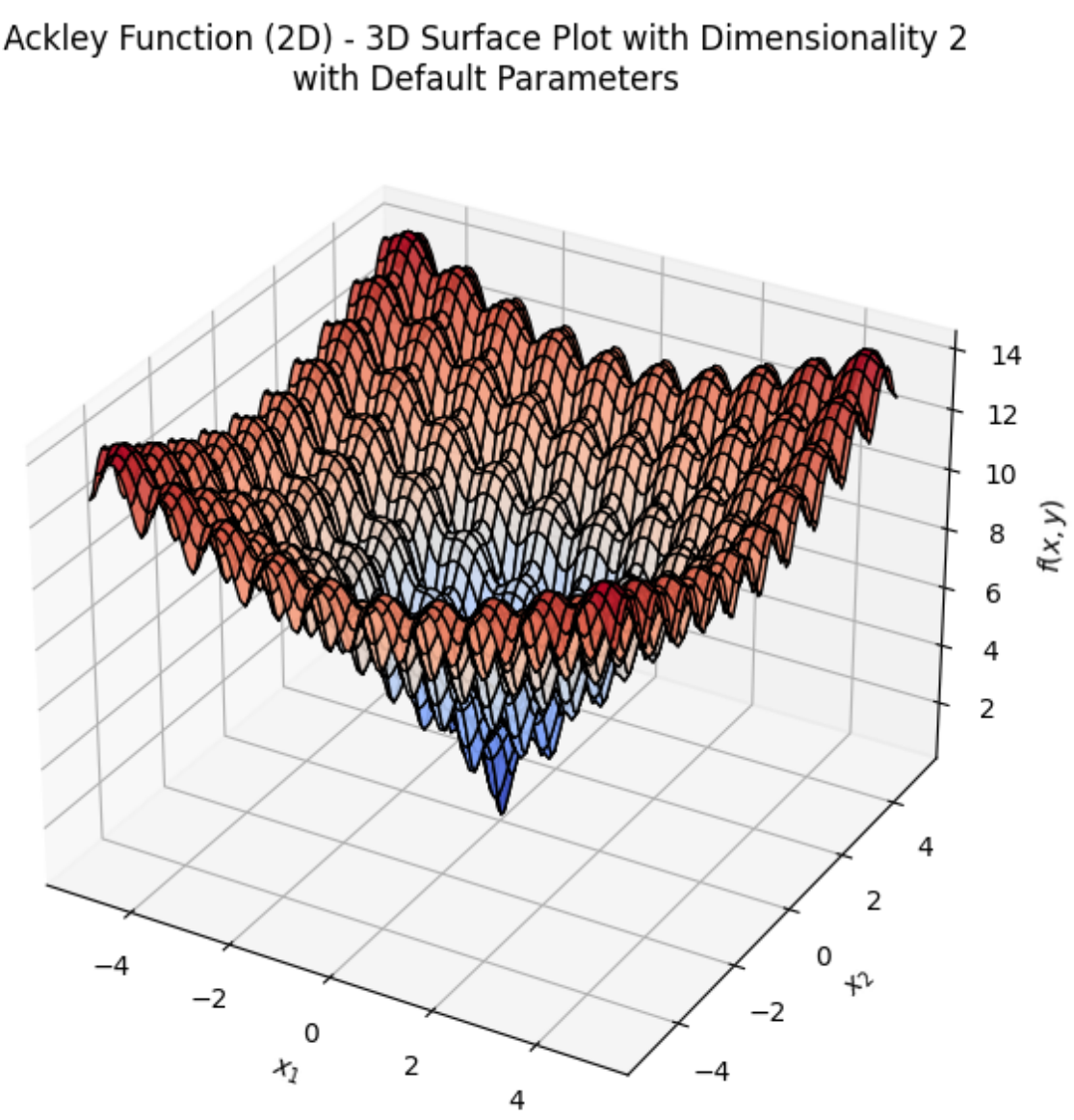

Ackley Function (2D) - 3D Surface Plot with Dimensionality 1000
2.5
2.4
2.3
2.2
2.1
2.0
1.9
1.8
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2

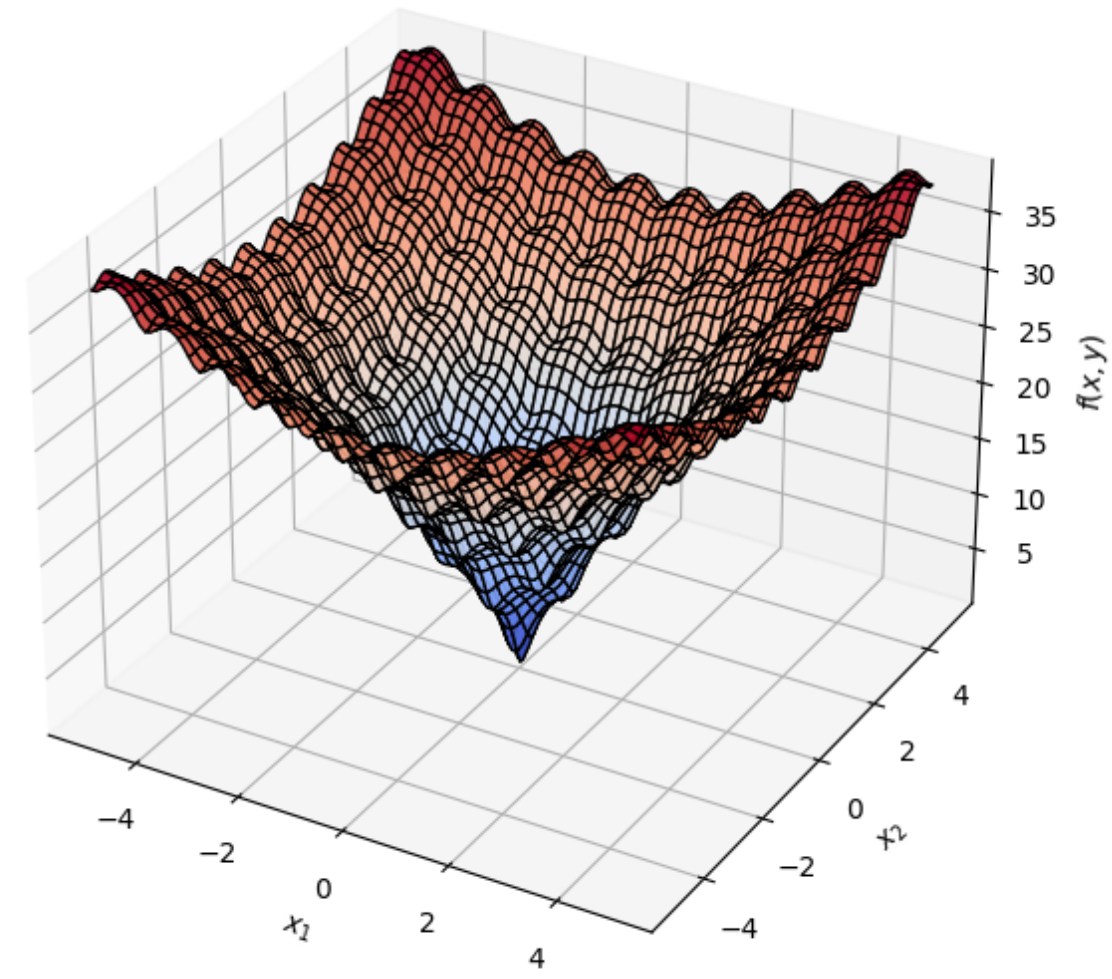

Ackley Function (2D) - Contour Plot with Dimensionality 5000
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
2.104
2.056
2.008
1.960
1.912
1.864
1.816
1.768
1.720
Function Value

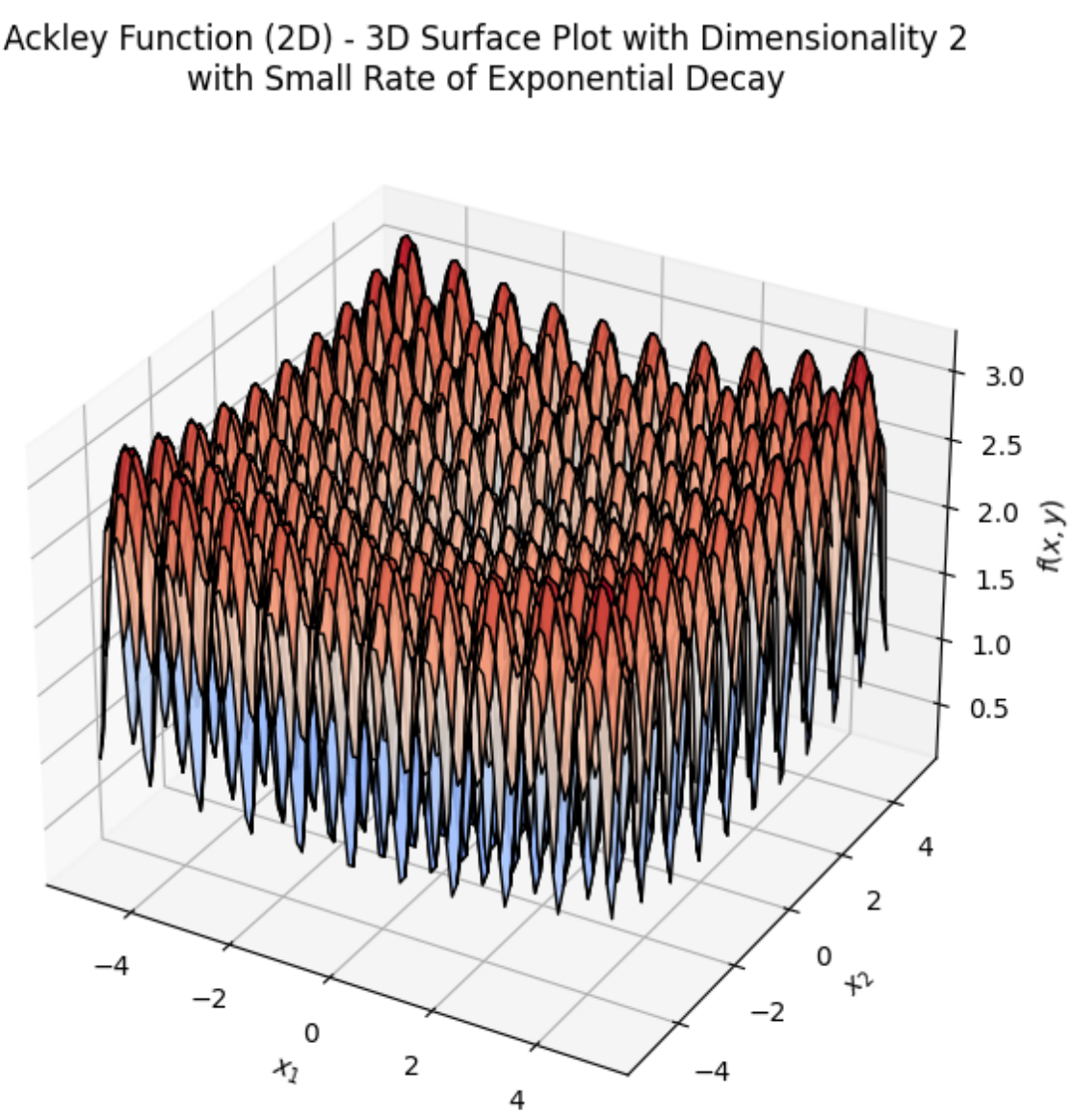

Ackley Function (2D) - 3D Surface Plot with Dimensionality 5000
2.10
2.05
2.00
1.95
1.90
1.85
1.80
1.75
f(x, y)
−4
−2
0
2
4
x1
−4
−2
0
2
4
x2

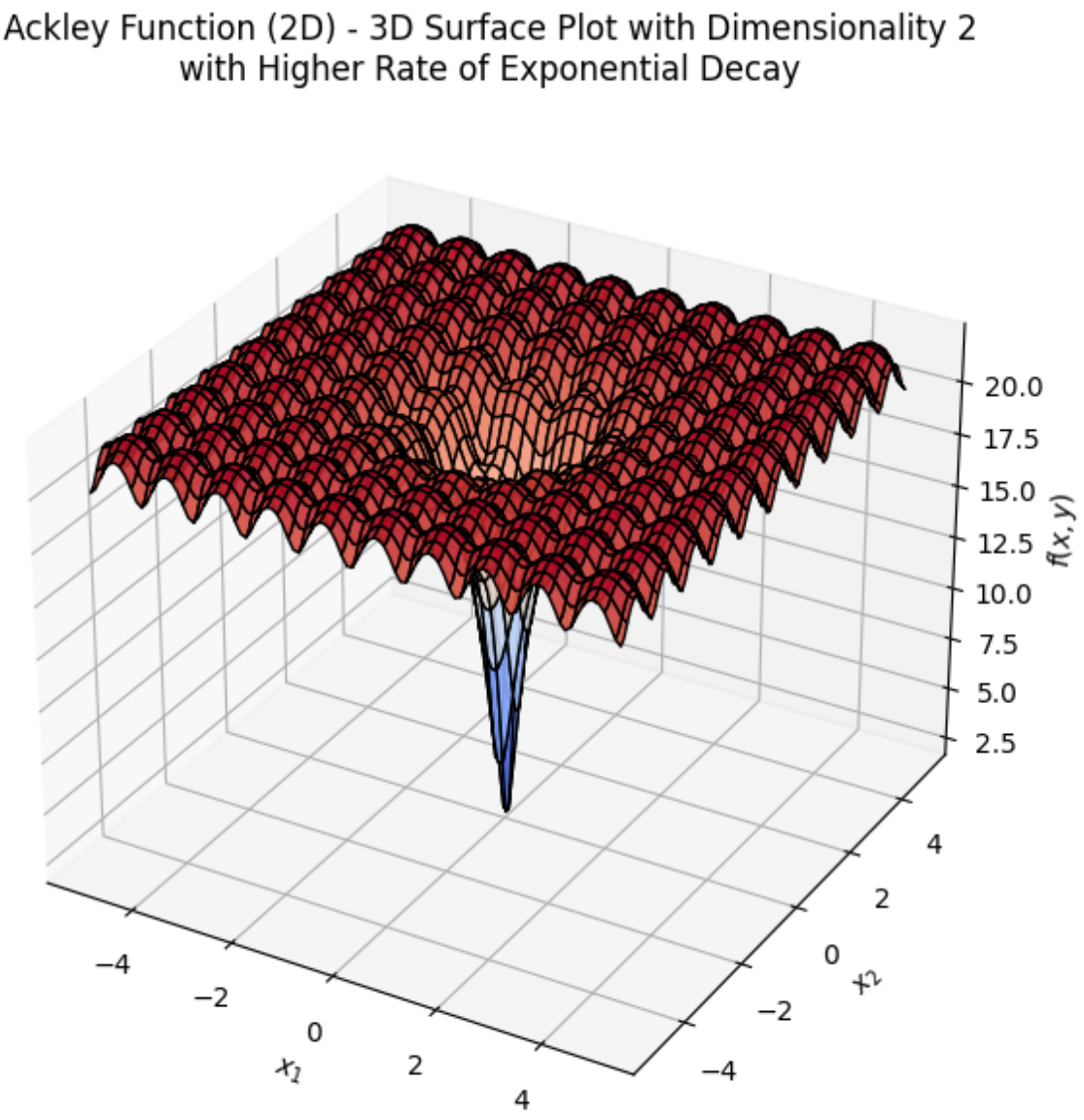

Ackley Function (2D) - Contour Plot with Dimensionality 10000
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
1.986
1.956
1.926
1.896
1.866
1.836
1.806
1.776
1.746
1.716
Function Value

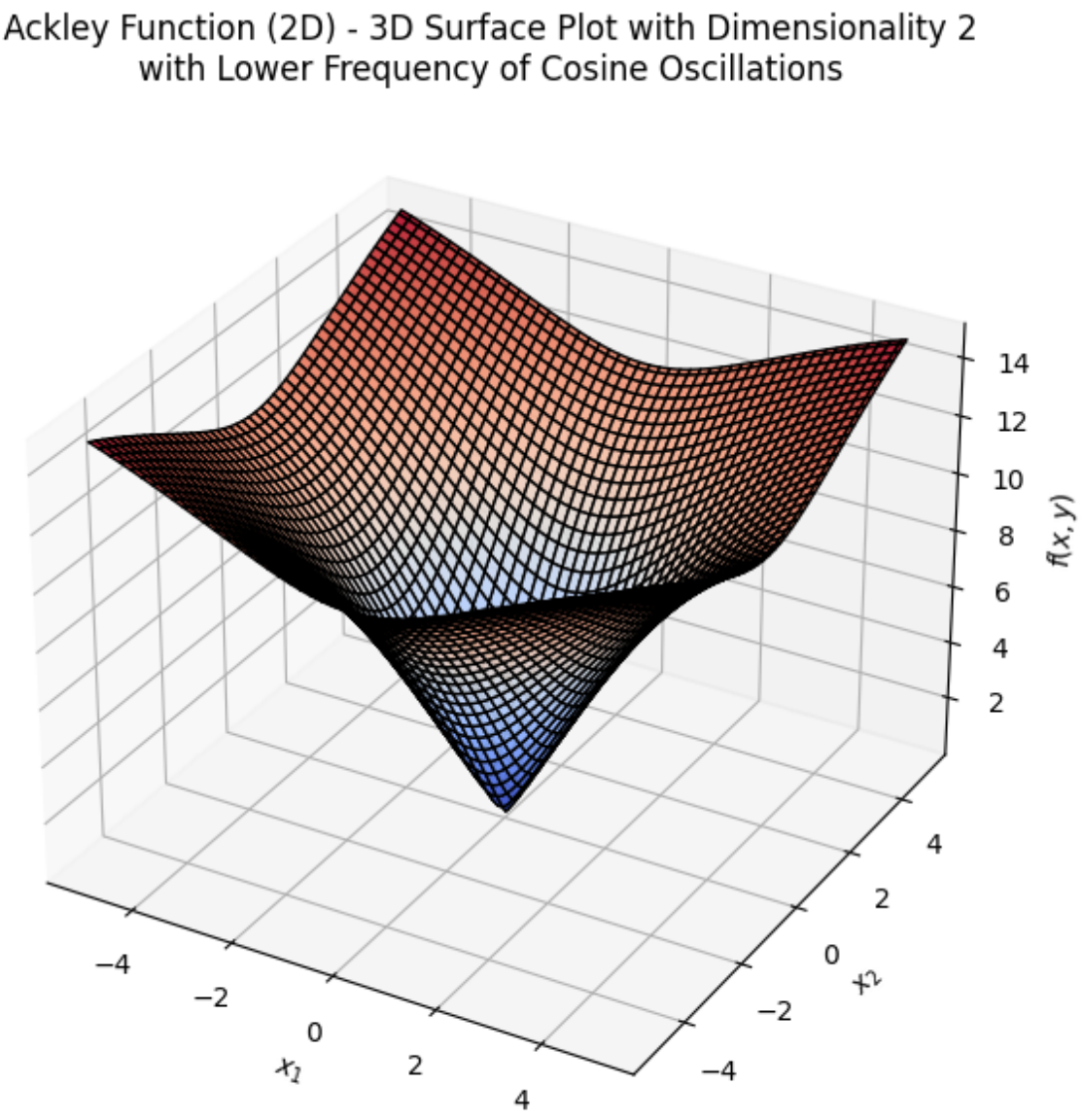

Ackley Function (2D) - 3D Surface Plot with Dimensionality 10000
1.95
1.90
1.85
1.80
1.75
f(x, y)
−4
−2
0
2
4
x1
−4
−2
0
2
4
x2

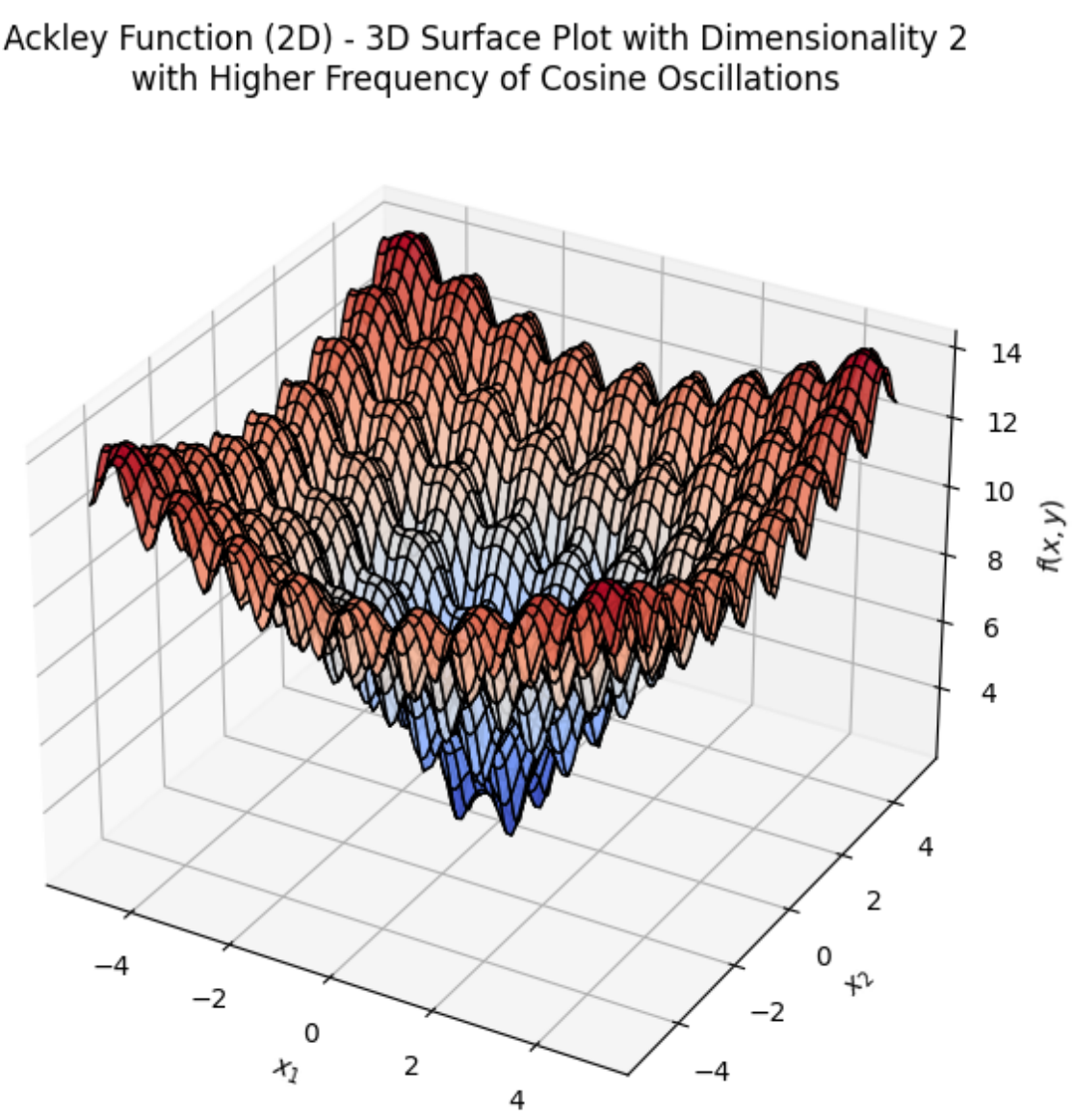

Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
with Default Parameters
14
12
10
8
6
4
2
f(x, y)
−4
−2
0
2
4
x1
4
2
0
−2
−4
x2


Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
with Increased Height
35
30
25
20
15
10
5
f(x, y)
−4
−2
0
2
4
x1
−4
−2
0
2
4
x2


Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
with Small Rate of Exponential Decay
3.0
2.5
2.0
1.5
1.0
0.5
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2


Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
with Higher Rate of Exponential Decay
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2


Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
with Lower Frequency of Cosine Oscillations
14
12
10
8
6
4
2
f(x, y)
−4
−2
0
2
4
x1
4
2
0
−2
−4
x2


Ackley Function (2D) - 3D Surface Plot with Dimensionality 2
with Higher Frequency of Cosine Oscillations
14
12
10
8
6
4
f(x, y)
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2