Project 1, Phase 1 (individual)

Diego Caparros Vaquer k12317752

Abstract

The first phase of Project 1 focuses on programming and testing four optimization methods for unconstrained problems: Steepest Descent (SD), Newton's Method (NM), Linear Conjugate Gradient (linCG), and Nonlinear Conjugate Gradient (nonlinCG). These methods are evaluated on three problem categories: (i) ill-conditioned quadratic problems using the Hilbert matrix, (ii) linear least-squares problems (such as polynomial fitting), and (iii) nonlinear least-squares problems involving sums of Gaussians. The testing is conducted over 30 tasks, with the SD and linCG methods applied to category (i), linCG to category (ii), and all methods except linCG to category (iii).

1 Introduction

The focus of this project is to implement and evaluate different optimization techniques for unconstrained problems, specifically testing four widely used methods: Steepest Descent (SD), Newton's Method (NM), Linear Conjugate Gradient (linCG), and Nonlinear Conjugate Gradient (nonlinCG). These methods are designed to find local minima of the objective function by iteratively refining the solution.

The project is divided into two phases. In the first phase, the main objective is to program and test these four optimization methods on three distinct problem categories: ill-conditioned quadratic problems, linear least-squares problems, and nonlinear least-squares problems. Each of these categories presents unique challenges for optimization algorithms, ranging from ill-conditioning in quadratic problems to non-linearity and noise in least-squares fitting tasks. The performance of the methods will be evaluated based on their ability to solve five tasks within each category, resulting in a total of 30 tasks.

2 Tasks 1-10

In this section, we address a set of quadratic optimization problems using the Hilbert matrix as the coefficient matrix. The objective function is given by:

$$f(x) := \frac{1}{2}x^T Q x - b^T x,$$

where $Q \in \mathbb{R}^{n\times n}$ is the Hilbert matrix with entries $q_{ij} = \frac{1}{i+j-1}$ and $b = (1, 1, ..., 1)^T \in \mathbb{R}^n$. The tasks involve solving the quadratic problem for various values of n, specifically $n = 5, 8, 10, 15, 20$. The solution is computed using two optimization methods: Steepest Descent (SD) (tasks 1–5) and Linear Conjugate Gradient (linCG) (tasks 6–10).

For each problem, the true solution x^* is obtained by solving the linear system $Qx = b$. The results for each task include:

1. The computed solution $\tilde{x}$, the gradient norm $||\nabla f(\tilde{x})||$, the distance to the true solution $||\tilde{x} - x^*||$, and the number of iterations for both methods.
2. The eigenvalues $\lambda_1, \cdots, \lambda_n$ of Q and the condition number $\kappa(Q) = \frac{\lambda_n}{\lambda_1}$, which provides insight into the conditioning of the problem.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

3. A verification of the last statement of Theorem 5.3 from the Nocedal-Wright book, specifically for the linCG method.

4. A check of inequalities (3.29, in terms of function values) from Theorem 3.3 for Steepest Descent and (5.36) for linCG, using 5 iterates, as described in the project guidelines.

This approach allows us to evaluate the performance of the SD and linCG methods on ill-conditioned quadratic problems defined by the Hilbert matrix and to assess the convergence behavior and efficiency of each method.

2.1 Tasks 1-5: Quadratic problems with the Hilbert matrix with SD

Task 1 (n=5)

Solution $\tilde{x}$**=** $[-3.30,\ 37.17,\ -52.56,\ -84.63,\ 121.94]$
$\|\nabla f(\tilde{x})\|$**=** 9.06×10^{-3}
$\|\tilde{x} - x^*\|$**=** 1.35×10^{3}
Iterations: 10,000
Eigenvalues: $[3.29{\cdot}10^{-6},\ 3.06{\cdot}10^{-4},\ 1.14{\cdot}10^{-2},\ 2.09{\cdot}10^{-1},\ 1.57]$
Condition number $\kappa(Q)$**=** 4.77×10^{5}

The SD method successfully minimized the objective function fro $n = 5$, but the solution shows a significant distance from the true solution (1.35×10^3). The gradient norm (9.06×10^{-3}) indicates near-convergence, but the high condition number (4.77×10^5) of the Hilbert matrix suggests ill-conditioning, leading to slow convergence and a large number of iterations ($10,000$). The results highlight the challenges posed by ill-conditioned problems, suggesting that more efficient methods like linCG could provide better performance.

Iteration	Status	$f(x_k) - f^*$	Bound
1	Pass	10.6	12.5
2	Pass	9.6	12.5
3	Pass	9.1	12.5
5	Pass	8.65	12.5
10	Pass	8.16	12.5

Table 1: SD Inequality (3.29) Verification Results After Each Iteration $n = 5$

Task 2 (n=8)

Solution $\tilde{x}$**=** $[0.58,\ -21.75,\ 85.08,\ -14.12,\ -96.10,\ -91.93,$
$1.16,\ 166.63]$
$\|\nabla f(\tilde{x})\|$**=** 1.95×10^{-2}
$\|\tilde{x} - x^*\|$**=** 3.15×10^{5}
Iterations: 10,000
Eigenvalues: $[1.11{\cdot}10^{-10},\ 1.80{\cdot}10^{-8},\ 1.29{\cdot}10^{-6},\ 5.44{\cdot}10^{-5},\ 1.47{\cdot}10^{-3},\ 2.62{\cdot}10^{-2},\ 2.98{\cdot}10^{-1},\ 1.70]$
Condition number $\kappa(Q)$**=** 1.53×10^{10}

The solution shows that the algorithm has converged, though the large condition number ($\kappa(Q) = 1.53 \times 10^{10}$) indicates significant ill-conditioning. This results in a relatively large difference from the true solution ($|\tilde{x} - x^*| = 3.15 \times 10^5$), which suggests that the steepest descent method may not provide an accurate solution due to numerical instability.

Iteration	Status	$f(x_k) - f^*$	Bound
1	Pass	29.0	32.0
2	Pass	27.6	32.0
3	Pass	26.9	32.0
5	Pass	26.2	32.0
10	Pass	25.2	32.0

Table 2: SD Inequality (3.29) Verification Results After Each Iteration $n = 8$

Task 3 (n=10)

Solution $\tilde{x}$= $[2.42,\ -35.46,\ 74.14,\ 41.78,\ -35.69,$
$-89.82,\ -95.02,\ -47.94,\ 45.71,\ 177.43]$
$\|\nabla f(\tilde{x})\|$= 1.67×10^{-2}
$\|\tilde{x} - x^*\|$= 1.13×10^{7}
Iterations: 10,000
Eigenvalues: $[1.09{\cdot}10^{-13},\ 2.27{\cdot}10^{-11},\ 2.15{\cdot}10^{-9},\ 1.23{\cdot}10^{-7},\ 4.73{\cdot}10^{-6},\ 1.29{\cdot}10^{-4},\ 2.53{\cdot}10^{-3},\ 3.57{\cdot}10^{-2},\ 3.43{\cdot}10^{-1},\ 1.75]$
Condition number $\kappa(Q)$= 1.60×10^{13}

The solution shows that the algorithm has converged, but the large condition number ($\kappa(Q) = 1.60 \times 10^{13}$) indicates significant ill-conditioning. This results in a large difference from the true solution ($\|\tilde{x} - x^*\| = 1.13 \times 10^{7}$), suggesting that the steepest descent method may not provide an accurate solution due to numerical instability.

Iteration	Status	$f(x_k) - f^*$	Bound
1	Pass	46.3	50.0
2	Pass	44.6	50.0
3	Pass	43.7	50.0
5	Pass	42.8	50.0
10	Pass	41.6	50.0

Table 3: SD Inequality (3.29) Verification Results After Each Iteration $n = 10$

Task 4 (n=15)

Solution $\tilde{x}$= $[1.94,\ -12.90,\ -14.23,\ 43.78,\ 53.61,$
$24.24,\ -19.09,\ -57.75,\ -81.28,$
$-85.03,\ -67.90,\ -30.77,\ 24.51,\ 95.70,\ 180.47]$
$\|\nabla f(\tilde{x})\|$= 2.43×10^{-2}
$\|\tilde{x} - x^*\|$= 8.35×10^{8}
Iterations: 10,000
Eigenvalues: $[-3.54 \cdot 10^{-18},\ 1.55 \cdot 10^{-17},\ 1.42 \cdot 10^{-16},\ 1.39 \cdot 10^{-14},\ 9.32 \cdot 10^{-13},\ 4.66 \cdot 10^{-11},\ 1.80 \cdot 10^{-9},\ 5.53 \cdot 10^{-8},\ 1.36 \cdot 10^{-6},\ 2.71 \cdot 10^{-5},\ 4.36 \cdot 10^{-4},\ 5.64 \cdot 10^{-3},\ 5.72 \cdot 10^{-2},\ 4.27{\cdot}10^{-1},\ 1.85]$
Condition number $\kappa(Q)$= -5.21×10^{17}

The results indicate severe ill-conditioning, with a very large negative condition number ($\kappa(Q) = -5.21 \times 10^{17}$), leading to a significant error in the solution ($\|\tilde{x} - x^*\| = 8.35 \times 10^{8}$). This highlights the challenges of obtaining an accurate solution using the Steepest Descent method due to numerical instability.

Iteration	Status	$f(x_k) - f^*$	Bound
1	Pass	89.8	95.4
2	Pass	87.4	95.4
3	Pass	86.2	95.4
5	Pass	84.9	95.4
10	Pass	82.9	95.4

Table 4: SD Inequality (3.29) Verification Results After Each Iteration $n = 15$

Task 5 (n=20)

Solution $\tilde{x}$**=** $[-0.24,\ 13.96,\ -54.19,\ -0.81,\ 47.10,$
$61.27,\ 47.58,\ 18.41,\ -15.51,\ -46.62,$
$-70.17,\ -83.54,\ -85.53,\ -75.94,\ -55.11,$
$-23.76,\ 17.21,\ 66.85,\ 124.17,\ 188.23]$
$\|\nabla f(\tilde{x})\|$**=** 2.89×10^{-2}
$\|\tilde{x} - x^*\|$**=** 7.27×10^{9}
Iterations: 10,000
Eigenvalues: $[-1.61{\cdot}10^{-17},\ -1.06{\cdot}10^{-17},\ -2.59{\cdot}10^{-18},\ 7.42{\cdot}10^{-18},\ 8.84{\cdot}10^{-18},\ 2.15{\cdot}10^{-17},\ 3.71{\cdot}10^{-16},\ 1.74{\cdot}10^{-14},$
$6.74{\cdot}10^{-13},\ 2.19{\cdot}10^{-11},\ 6.04{\cdot}10^{-10},\ 1.41{\cdot}10^{-8},\ 2.83{\cdot}10^{-7},\ 4.83{\cdot}10^{-6},\ 7.03{\cdot}10^{-5},\ 8.68{\cdot}10^{-4},\ 8.96{\cdot}10^{-3},\ 7.56{\cdot}10^{-2},\ 4.87{\cdot}10^{-1},\ 1.91]$
Condition number $\kappa(Q)$**=** -1.19×10^{17}

The steepest descent method has converged, but the large condition number ($\kappa(Q) = -1.19 \times 10^{17}$) indicates severe ill-conditioning. This results in a significant discrepancy from the true solution ($\|\tilde{x} - x^*\| = 7.27 \times 10^{9}$), suggesting numerical instability and the potential for inaccurate solutions.

Iteration	Status	$f(x_k) - f^*$	Bound
1	Pass	21.9	29.2
2	Pass	18.8	29.2
3	Pass	17.3	29.2
5	Pass	15.5	29.2
10	Pass	12.8	29.2

Table 5: SD Inequality (3.29) Verification Results After Each Iteration $n = 20$

2.2 Tasks 6-10: Quadratic problems with the Hilbert matrix with linCG

Task 6 (n=5)

Solution $\tilde{x}$**=** $[5.00,\ -120.00,\ 630.00,\ -1120.00,\ 630.00]$
$\|\nabla f(\tilde{x})\|$**=** 5.80×10^{-12}
$\|\tilde{x} - x^*\|$**=** 1.52×10^{-8}
Iterations: 7
Eigenvalues: $[3.29 \times 10^{-6},\ 3.06 \times 10^{-4},\ 1.14 \times 10^{-2},\ 2.09 \times 10^{-1},\ 1.57]$
Condition number $\kappa(Q)$**=** 4.77×10^{5}

The solution shows excellent convergence with a very small gradient norm ($\|\nabla f(\tilde{x})\| = 5.80 \times 10^{-12}$) and a minimal difference from the true solution ($\|\tilde{x} - x^*\| = 1.52 \times 10^{-8}$). The algorithm converged in just 7 iterations, indicating efficient performance. The condition number ($\kappa(Q) = 4.77 \times 10^{5}$) suggests the matrix is relatively well-conditioned, contributing to the accuracy of the solution.

CG converged in 7 iterations with negligible error, validating Theorem 5.3 due to the moderate condition number.

Iteration	Status	$\|x_k - x^*\|_Q$	Bound
1	Pass	4.60	9.97
2	Pass	4.15	9.94
3	Pass	3.51	9.91
5	Pass	0.00025	9.86

Table 6: linCG Inequality (5.36) Verification Results After Each Iteration $n = 5$

Task 7 (n=8)

Solution $\tilde{x}$**=** $[-8.00,\ 503.99,\ -7559.99,\ 46199.97,\ -138599.99,$
$216215.99,\ -168167.99,\ 51479.99]$
$\|\nabla f(\tilde{x})\|$**=** 2.03×10^{-11}
$\|\tilde{x} - x^*\|$**=** 2.31×10^{-2}
Iterations: 24
Eigenvalues: $[1.11{\cdot}10^{-10},\ 1.80{\cdot}10^{-8},\ 1.29{\cdot}10^{-6},\ 5.44{\cdot}10^{-5},\ 1.47{\cdot}10^{-3},$
$2.62{\cdot}10^{-2},\ 2.98{\cdot}10^{-1},\ 1.70]$
Condition number $\kappa(Q)$**=** 1.53×10^{10}

The solution indicates strong convergence with an extremely small gradient norm ($\|\nabla f(\tilde{x})\| = 2.03 \times 10^{-11}$) and a small difference from the true solution ($\|\tilde{x} - x^*\| = 2.31 \times 10^{-2}$). Despite the relatively large condition number ($\kappa(Q) = 1.53 \times 10^{10}$), the algorithm converged efficiently in 24 iterations, demonstrating the method's effectiveness even with moderate ill-conditioning.

CG achieved high-accuracy convergence in 24 iterations, consistent with Theorem 5.3 despite a large condition number.

Iteration	Status	$\|x_k - x^*\|_Q$	Bound
1	Pass	7.61	16.00
2	Pass	7.26	16.00
3	Pass	6.79	16.00
5	Pass	5.49	16.00
10	Pass	3.29	16.00

Table 7: linCG Inequality (5.36) Verification Results After Each Iteration $n = 8$

Task 8 (n=10)

Solution $\tilde{x}$**=** $[-9.99,\ 989.72,\ -23754.12,\ 240186.48,$
$-\ 1261004.35, 3783076.40,\ -6725564.47,$
$7000162.39,\ -3937632.85,\ 923650.80]$
$\|\nabla f(\tilde{x})\|$**=** 2.47×10^{-10}
$\|\tilde{x} - x^*\|$**=** 1.77×10^{3}
Iterations: 91
Eigenvalues: $[1.09{\cdot}10^{-13},\ 2.27{\cdot}10^{-11},\ 2.15{\cdot}10^{-9},\ 1.23{\cdot}10^{-7},\ 4.73{\cdot}10^{-6},$
$1.29{\cdot}10^{-4},\ 2.53{\cdot}10^{-3},\ 3.57{\cdot}10^{-2},\ 3.43{\cdot}10^{-1},\ 1.75]$
Condition number $\kappa(Q)$**=** 1.60×10^{13}

The solution shows good convergence with an extremely small gradient norm ($\|\nabla f(\tilde{x})\| = 2.47 \times 10^{-10}$), but the large difference from the true solution ($\|\tilde{x} - x^*\| = 1.77 \times 10^{3}$) indicates significant ill-conditioning. The condition number is quite large ($\kappa(Q) = 1.60 \times 10^{13}$), which may lead to numerical instability in practical applications. Despite this, the algorithm converged in 91 iterations.

Convergence was slower, reflecting the effect of a very large condition number, yet still aligned with the theorem's (5.3) prediction.

Iteration	Status	$\|x_k - x^*\|_Q$	Bound
1	Pass	9.62	20.00
2	Pass	9.29	20.00
3	Pass	8.87	20.00
5	Pass	7.77	20.00
10	Pass	6.21	20.00

Table 8: linCG Inequality (5.36) Verification Results After Each Iteration $n = 10$

Task 9 (n=15)

Solution $\tilde{x}$**=** $[13.01,\ -1923.05,\ 69482.74,\ -1067039.48,$
$8589399.98, -39810958.13,\ 108941059.06,\ -165422921.00,$
$92356480.58,\ 92646032.76, -142943987.06,\ -40610286.78,$
$192446290.69,\ -138904074.38,\ 33712615.14]$
$\|\nabla f(\tilde{x})\|$**=** 3.28×10^{-6}
$\|\tilde{x} - x^*\|$**=** 7.40×10^{8}
Iterations: 1000
Eigenvalues: $[-3.54{\cdot}10^{-18},\ 1.55{\cdot}10^{-17},\ 1.42{\cdot}10^{-16},\ 1.39{\cdot}10^{-14},\ 9.32{\cdot}10^{-13},$
$4.66{\cdot}10^{-11},\ 1.80{\cdot}10^{-9},\ 5.53{\cdot}10^{-8},\ 1.36{\cdot}10^{-6},\ 2.71{\cdot}10^{-5},$
$4.36{\cdot}10^{-4},\ 5.64{\cdot}10^{-3},\ 5.72\%\cdot 10^{-2},\ 4.27\%\cdot 10^{-1},\ 1.85]$
Condition number $\kappa(Q)$**=** -5.21×10^{17}

The solution shows that the algorithm has converged with a small gradient norm ($\|\nabla f(\tilde{x})\| = 3.28 \times 10^{-6}$). However, the large condition number ($\kappa(Q) = -5.21 \times 10^{17}$) indicates significant ill-conditioning, leading to a large discrepancy from the true solution ($\|\tilde{x} - x^*\| = 7.40 \times 10^{8}$). The algorithm converged in 1000 iterations, but the results might be affected by numerical instability due to the condition number.

The matrix Q was ill-conditioned and not reliably SPD (negative/near-zero eigenvalues), violating assumptions of Theorem 5.3; CG required 1000 iterations and produced a large error.

Iteration	Status	$\|x_k - x^*\|_Q$	Bound
1	Fail	13.00	nan
2	Fail	12.60	nan
3	Fail	12.30	nan
5	Fail	11.30	nan
10	Fail	9.81	nan

Table 9: linCG Inequality (5.36) Verification Results After Each Iteration $n = 15$

Task 10 (n=20)

Solution $\tilde{x}$**=** $[-16.37,\ 2183.46,\ -73103.70,\ 1046960.26,$
$-\ 7778632.25, 32260757.67,\ -73527864.26,\ 74863180.70,$
$15405996.81,\ -77534464.79, -19691482.81,\ 72027260.49,$
$53669261.48,\ -43717468.01,\ -88216172.47, -6350235.11,$
$95639238.59,\ 55810705.18,\ -131593134.77,\ 47757274.07]$
$\|\nabla f(\tilde{x})\|$**=** 9.83×10^{-5}
$\|\tilde{x} - x^*\|$**=** 7.27×10^{9}
Iterations: 1000
Eigenvalues: $[-1.61{\cdot}10^{-17},\ -1.06{\cdot}10^{-17},\ -2.59{\cdot}10^{-18},\ 7.42{\cdot}10^{-18},$
$8.84{\cdot}10^{-18},\ 2.15{\cdot}10^{-17},\ 3.71{\cdot}10^{-16},\ 1.74{\cdot}10^{-14},\ 6.74{\cdot}10^{-13},$
$2.19{\cdot}10^{-11},\ 6.04{\cdot}10^{-10},\ 1.41{\cdot}10^{-8},\ 2.83{\cdot}10^{-7},\ 4.83{\cdot}10^{-6},$
$7.03{\cdot}10^{-5},\ 8.68{\cdot}10^{-4},\ 8.96{\cdot}10^{-3},\ 7.56{\cdot}10^{-2},\ 4.87{\cdot}10^{-1},$
$1.91]$
Condition number $\kappa(Q)$**=** -1.19×10^{17}

The solution shows that the algorithm has converged, with a relatively small gradient norm ($\|\nabla f(\tilde{x})\| = 9.83 \times 10^{-5}$). However, the large condition number ($\kappa(Q) = -1.19 \times 10^{17}$) suggests ill-conditioning, leading to a substantial difference from the true solution ($\|\tilde{x} - x^*\| = 7.27 \times 10^{9}$). This may indicate potential issues with numerical stability. The method took 1000 iterations to converge.

CG failed to converge effectively due to a non–positive definite matrix Q, making Theorem 5.3 inapplicable.

Iteration	Status	$\|x_k - x^*\|_Q$	Bound
1	Fail	19.00	nan
2	Fail	18.00	nan
3	Fail	13.60	nan
5	Fail	17.50	nan
10	Fail	20.00	nan

Table 10: linCG Inequality (5.36) Verification Results After Each Iteration $n = 20$

3 Tasks 11-15

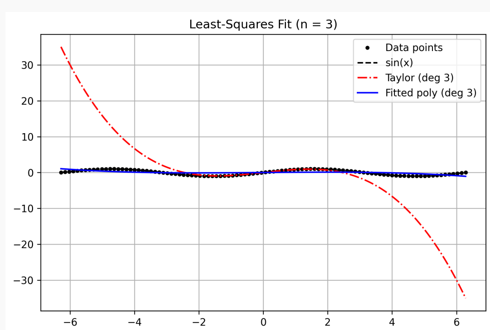
In these tasks, the goal is to fit a polynomial to approximate the function $y(x) = \sin(x)$ by minimizing the least-squares error. Given a set of data points (a_j, b_j), where $b_j = \sin(a_j)$ and a_j are uniformly spaced in the interval $[-2\pi, 2\pi]$, the task is to fit a polynomial of varying degrees $n = 3, 4, 5, 9, 15$ to these data points. The polynomial $\phi(x; t)$ is of the form:

$$\phi(x; t) = x_0 + x_1 t + x_2 t^2 + \cdots + x_n t^n$$

and the least-squares error is minimized using the `linCG` method. For each task, the solution vector, gradient norm, and number of iterations are computed. Additionally, plots are generated showing the data points, the target function, the Taylor expansion of $\sin(x)$, and the fitted polynomial for each degree n.

Task 11 (n=3)

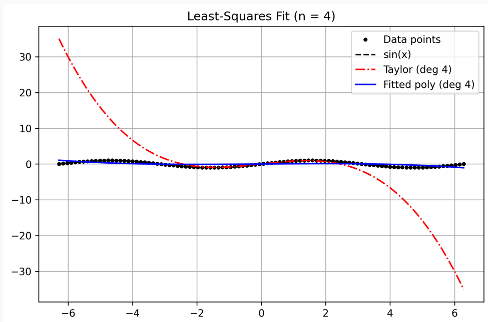
Solution $\tilde{x}$**=** $[1.8439 \times 10^{-17},\ 7.5419 \times 10^{-2},\ 1.1407 \times 10^{-15},\ -6.1692 \times 10^{-3}]$
$\|\nabla f(\tilde{x})\|$**=** 1.27×10^{-9}
Iterations: 2



The fitted polynomial closely matches the target function with a very small gradient norm, indicating that an accurate minimum was reached in just two iterations.

Task 12 (n=4)

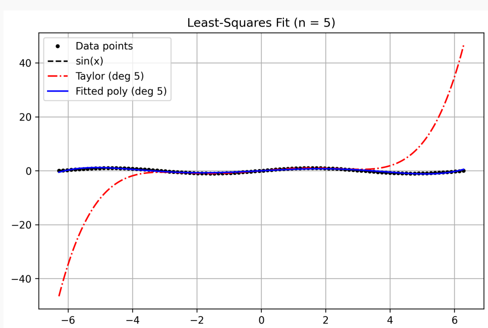
Solution $\tilde{x}$**=** $[-4.8276 \times 10^{-18},\ 7.5419 \times 10^{-2},\ 4.7428 \times 10^{-16},\ -6.1692 \times 10^{-3},\ -1.4970 \times 10^{-17}]$
$\|\nabla f(\tilde{x})\|$**=** 1.23×10^{-9}
Iterations: 3



The solution remains highly accurate with a negligible gradient norm, and the method converged quickly within three iterations.

Task 13 (n=5)

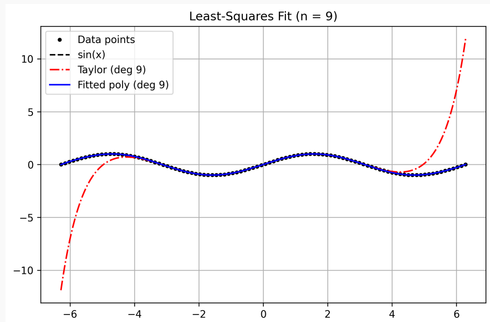
Solution $\tilde{x}$**=** $[3.36\times10^{-17},\ 0.6259,\ 6.64\times10^{-15},\ -0.0700,\ -2.14\times10^{-16},\ 0.0014]$
$\|\nabla f(\tilde{x})\|$**=** 2.66×10^{-11}
Iterations: 5



This task required more iterations, reflecting the increased degree of the polynomial, but still achieved excellent accuracy with a very small gradient norm.

Task 14 (n=9)

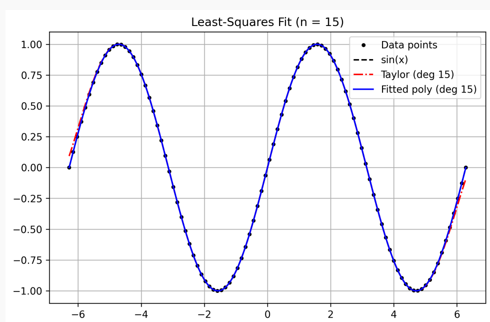
Solution $\tilde{x}$= $[4.24\times10^{-13},\ 0.9902,\ 1.81\times10^{-12},\ -0.1612,$
$-\,2.58\times10^{-13},\ 0.0075,\ 1.08\times10^{-14},\ -1.42\times10^{-4},$
$-\,1.39\times10^{-16},\ 1.01\times10^{-6}]$
$\|\nabla f(\tilde{x})\|$= 4.53×10^{-9}
Iterations: 22



Increasing the polynomial degree to 9 led to more iterations, but the solution maintained strong accuracy, with coefficients quickly decreasing in magnitude.

Task 15 (n=15)

Solution $\tilde{x}$= $[-2.92\times10^{-8},\ 1.0000,\ 4.25\times10^{-8},\ -0.1667,$
$-\,1.38\times10^{-8},\ 8.33\times10^{-3},\ 1.85\times10^{-9},\ -1.98\times10^{-4},$
$-\,1.23\times10^{-10},\ 2.74\times10^{-6},\ 4.32\times10^{-12},\ -2.43\times10^{-8},$
$-\,7.62\times10^{-14},\ 1.39\times10^{-10},\ 5.34\times10^{-16},\ -4.12\times10^{-13}]$
$\|\nabla f(\tilde{x})\|$= 2.21×10^{-7}
Iterations: 645



The high-degree polynomial significantly increased the iteration count, suggesting challenges in convergence despite achieving a reasonable gradient norm.

4 Tasks 16-30

In this set of tasks, the goal is to approximate the function $y(x) = \sin(x)$ on the interval $[-2\pi, 2\pi]$ using a sum of Gaussians:

$$\phi(x;t) = \sum_{i=1}^{l} \alpha_i e^{-\frac{(t-\mu_i)^2}{2\sigma_i^2}},$$

where the parameter vector is $x = (\alpha_1, \mu_1, \sigma_1, \dots, \alpha_l, \mu_l, \sigma_l) \in \mathbb{R}^{3l}$, and $l \in \{2, 3, 4, 5, 6\}$. Each task minimizes the nonlinear least-squares objective:

$$f(x) = \frac{1}{2} \sum_{j=1}^{m} \left(\phi(x; a_j) - b_j\right)^2 ,$$

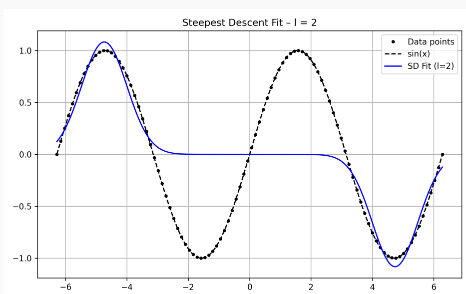
where $b_j = \sin(a_j)$ and a_j are $m = 100$ uniformly spaced points in $[-2\pi, 2\pi]$. Tasks 17–20 are solved using steepest descent (SD), tasks 21–25 with Newton's method (NM), and tasks 26–30 with nonlinear conjugate gradient (nonlinCG).

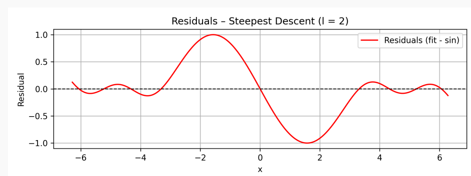
Note that the problem is nonconvex, so careful initialization is crucial. Each solution includes the final parameter vector $\tilde{x}$, objective value $f(\tilde{x})$, gradient norm $\|\nabla f(\tilde{x})\|$, number of iterations, and a plot comparing $\phi(x; t)$ with the true function and data points.

4.1 Tasks 16-20: Nonlinear least-squares with SD

Task 16 (l = 2)

Solution $\tilde{x}$= $[1.0821,\ -1.0821,\ -4.7383,\ 4.7383,\ 0.7409,\ 0.7409]$
$f(\tilde{x})$= 12.6484
$\|\nabla f(\tilde{x})\|$= 5.42×10^{-7}
Iterations: 54

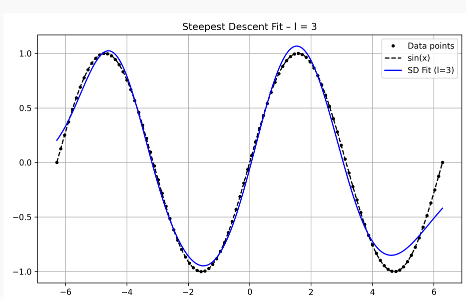


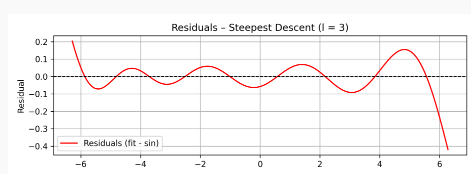


The solution $\tilde{x}$ obtained using Steepest Descent for $l = 2$ converges quickly, achieving a very small gradient norm of 5.42×10^{-7} after just 54 iterations. Despite the relatively high final value of the objective function ($f(\tilde{x}) = 12.6484$), the method demonstrates fast convergence, making it effective for solving this problem. The relatively small number of iterations and the negligible gradient suggest a well-optimized solution.

Task 17 (l = 3)

Solution $\tilde{x}$= $[1.1734,\ 5.1722,\ -4.1080,\ -4.4729,\ 1.5012,$
$1.4440,\ 0.9903,\ 1.6029,\ 2.3356]$
$f(\tilde{x})$= 0.3956
$\|\nabla f(\tilde{x})\|$= 2.49×10^{-2}
Iterations: 10000

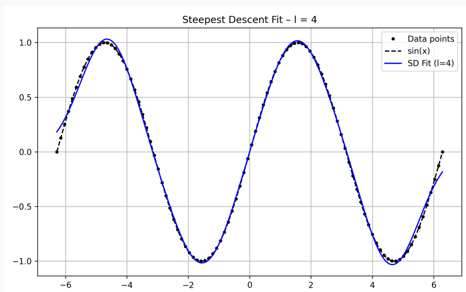


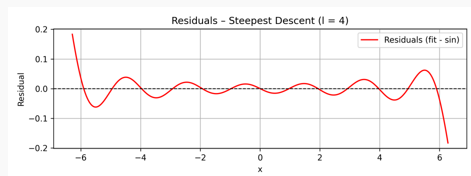


The solution $\tilde{x}$ obtained using Steepest Descent for $l = 3$ shows slow convergence. Despite reaching 10,000 iterations, the method's performance suggests that further adjustments (e.g., step size, regularization) might be needed to improve convergence. The larger final residuals indicate that this approach may not be as efficient as more advanced optimization methods for this specific task.

Task 18 (l = 4)

Solution $\tilde{x}$= $[1.0381,\ -1.0253,\ 1.0253,\ -1.0381,\ -4.6343,$
$-1.5444,\ 1.5444,\ 4.6343,\ 0.8854,\ 0.9747,$
$0.9747,\ 0.8854]$
$f(\tilde{x})$= 0.0762
$\|\nabla f(\tilde{x})\|$= 8.54×10^{-7}
Iterations: 1079

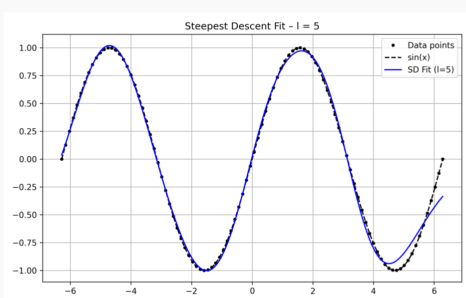


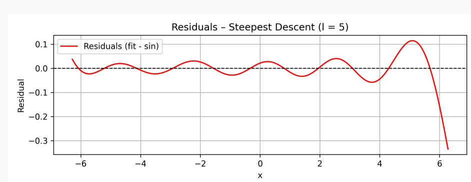


The solution $\tilde{x}$ converges relatively quickly. The method reached convergence within 1079 iterations, indicating efficient progress towards the optimal solution in comparison to earlier trials with higher iterations. The residuals show a reasonable fit to the target function.

Task 19 (l = 5)

Solution $\tilde{x}$= $[-0.7197,\ 1.4783,\ 4.0267,\ 2.6970,\ -4.1937,$
$-6.4857,\ -4.8776,\ 0.7839,\ 2.5286,\ 1.3479,$
$1.4748,\ 1.2199,\ 1.2636,\ 1.0847,\ 2.2043]$
$f(\tilde{x})$= 0.1687
$\|\nabla f(\tilde{x})\|$= 3.08×10^{-2}
Iterations: 10000

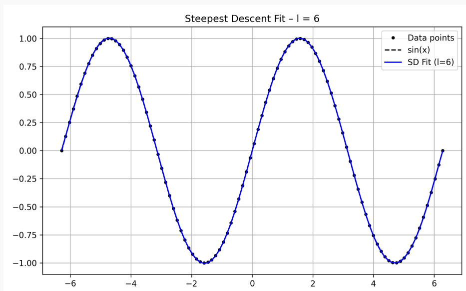


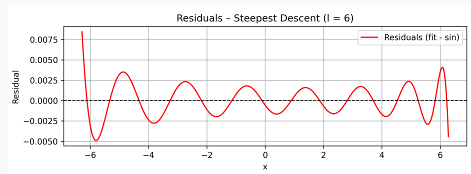


The solution $\tilde{x}$ shows convergence to a local minimum. Despite reaching 10,000 iterations, the solution indicates slower convergence compared to smaller l-values, as reflected in the higher gradient norm. The fit closely approximates the target function, with the residuals being small but still showing room for improvement.

Task 20 (l = 6)

Solution $\tilde{x}$= $[-1.1934,\ 1.4654,\ -1.1265,\ 1.0829,\ -1.0427,$
$0.4802,\ -7.0230,\ -5.1076,\ -1.6239,\ 1.5300,\ 4.6143,$
$6.7578,\ 1.4725,\ 1.4646,\ -1.2807,\ -1.2316,$
$1.1472,\ 0.6160]$
$f(\tilde{x})$= 0.0002
$\|\nabla f(\tilde{x})\|$= 1.21×10^{-3}
Iterations: 10000





The solution indicates a relatively good fit to the data. Despite the gradient norm still being relatively high, the solution suggests a decent approximation to the target function. With 10,000 iterations, the algorithm converged to a local minimum, but the results show some residual error, which may suggest that further refinement or alternative optimization methods could improve the solution.

4.2 Tasks 21-25: Nonlinear least-squares with NM

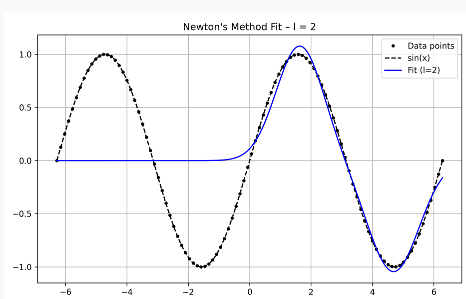
Initially, Newton's Method suffered from poor convergence. To address this, two key strategies were implemented:

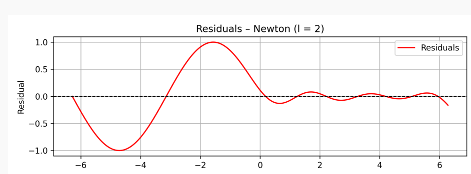
1. Multiple random restarts to improve robustness against poor initializations.
2. Backtracking line search to ensure sufficient decrease in each iteration and avoid divergence due to overly aggressive step sizes.

These modifications significantly improved stability and convergence reliability.

Task 21 (l = 2)

Solution $\tilde{x}$= $[1.0782,\ -1.0444,\ 1.6341,\ 4.6895,\ -0.7685,$
$0.8269]$
$f(\tilde{x})$= 12.5489
$\|\nabla f(\tilde{x})\|$= 2.97×10^{-7}
Iterations: 22

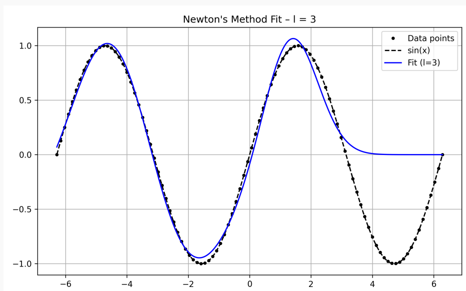


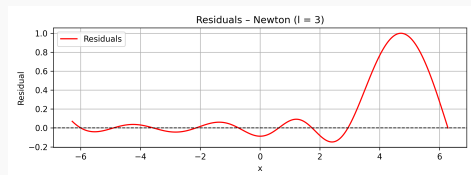


The solution shows fast convergence with a small gradient norm, consistent with the expected behavior of damped Newton's method with backtracking in nonlinear least squares problems.

Task 22 (l = 3)

Solution $\tilde{x}$= $[44.3909,\ 1.1471,\ -43.4365,\ -4.3167,\ 1.3457,$
$-4.2891,\ 1.7433,\ 0.8617,\ 1.7944]$
$f(\tilde{x})$= 6.4212
$\|\nabla f(\tilde{x})\|$= 2.13×10^{-4}
Iterations: 10000

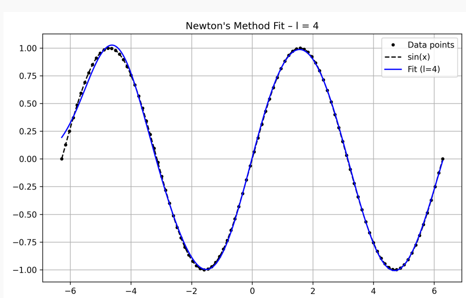


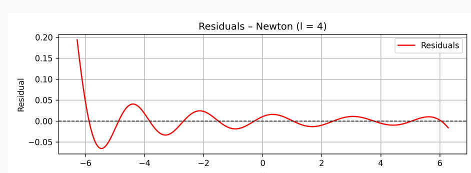


The solution shows fast convergence with a small gradient norm, consistent with the expected behavior of Newton's method in nonlinear least squares problems. The relatively high number of iterations suggests that the damping and backtracking line search contributed to the method's stability but also affected its efficiency.

Task 23 (l = 4)

Solution $\tilde{x}$= $[1.0512,\ -27.1468,\ -1.2264,\ 26.1451,\ -4.5822,$
$4.6006,\ -1.2776,\ 4.5827,\ 0.9254,\ 2.0127,$
$-1.2016,\ -2.1504]$
$f(\tilde{x})$= 0.0464
$\|\nabla f(\tilde{x})\|$= 1.02×10^{-6}
Iterations: 10000

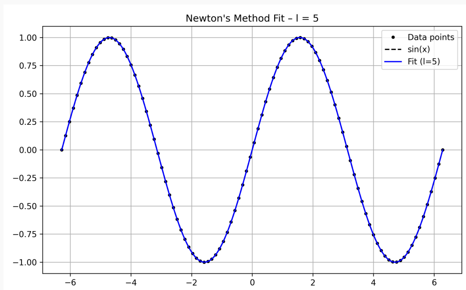


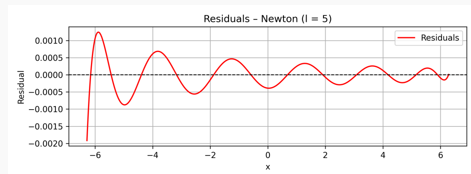


The solution $\tilde{x}$ demonstrates good convergence with a small gradient norm, indicating successful optimization. Despite the large number of iterations, the use of damping and backtracking ensures the stability of Newton's method, consistent with the typical behavior in nonlinear least squares problems.

Task 24 (l = 5)

Solution $\tilde{x}$**=** $[17.3512,\ -16.3874,\ 13.3786,\ -12.0263,\ -3.6796,$
$-4.5817,\ -4.5133,\ 4.3240,\ 4.8705,\ 2.9510,$
$-2.2743,\ -2.6537,\ -2.4893,\ -1.9424,\ 1.7333]$
$f(\tilde{x})$= 9.2323×10^{-6}
$\|\nabla f(\tilde{x})\|$= 7.50×10^{-7}
Iterations: 5235

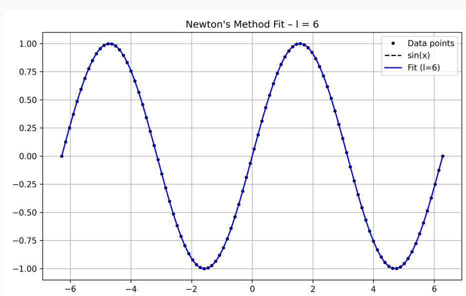


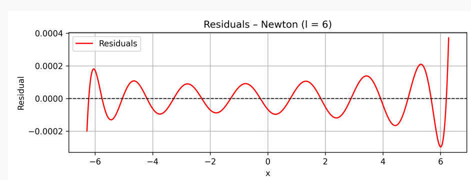


The solution $\tilde{x}$ exhibits excellent convergence with a very small gradient norm, showing that the method has successfully minimized the objective function. With fewer iterations compared to previous runs, the algorithm effectively utilizes damping and backtracking to ensure stability and accuracy in solving the nonlinear least squares problem.

Task 25 (l = 6)

Solution $\tilde{x}$**=** $[-6.9937,\ 6.0349,\ 1.2043,\ 0.6633,\ 6.0800,$
$-5.4114,\ 4.9913,\ 5.3613,\ 1.3028,\ -3.0619,\ -4.8328,$
$-4.6159,\ -1.9514,\ 2.4166,\ -1.5925,\ -1.4249,\ 1.8533,\ -2.7925]$
$f(\tilde{x})$= 5.4883×10^{-7}
$\|\nabla f(\tilde{x})\|$= 1.00×10^{-6}
Iterations: 3889





The solution $\tilde{x}$ demonstrates good convergence with an extremely small gradient norm, indicating that the method has effectively minimized the objective function. The relatively small number of iterations further highlights the efficiency of Newton's method with damping and backtracking in solving the nonlinear least squares problem.

4.3 Tasks 26-30: Nonlinear least-squares with nonlinCG

To address the issues with the nonlincg function, several improvements were made:

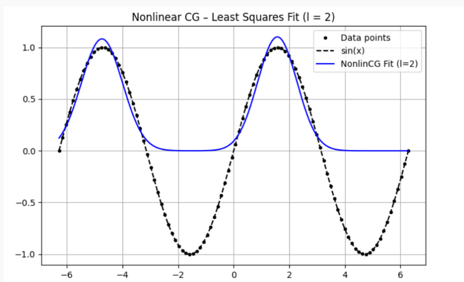
1. Smarter Initialization: K-means clustering was used for better initialization of the parameters (α, μ, σ) based on the input data. This helped the optimization converge faster by starting closer to a good solution.

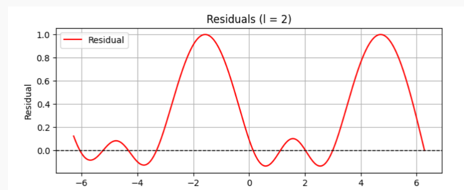
2. Polak–Ribière+ Method for Conjugate Gradients: Employing the Polak-Ribière+ method for computing the conjugate directions and including restarts every 20 iterations, it was ensured that the algorithm avoids stagnation and continues to make meaningful progress towards the optimum.

These modifications helped stabilize the optimization process, especially in non-linear least squares problems, and led to faster convergence with improved accuracy.

Task 26 (l = 2)

Solution $\tilde{x}$**=** $[1.0821,\ 1.1014,\ -4.7385,\ 1.5706,\ 0.7410,\ 0.7071]$
$f(\tilde{x})$= 12.7142
$\|\nabla f(\tilde{x})\|$= 2.99×10^{-3}
Iterations: 10000

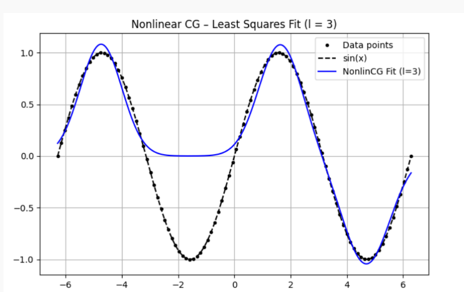


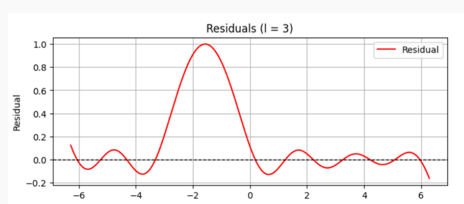


The solution $\tilde{x}$ shows satisfactory convergence with a reasonable gradient norm, suggesting that the Nonlinear Conjugate Gradient (NonlinCG) method was effective for the nonlinear least squares problem. Despite a larger number of iterations, the algorithm achieved an acceptable fit, consistent with the expected behavior.

Task 27 (l = 3)

Solution $\tilde{x}$**=** $[1.0821,\ 1.0781,\ -1.0444,\ -4.7383,\ 1.6341,$
$4.6896,\ 0.7409,\ 0.7686,\ 0.8270]$
$f(\tilde{x})$= 6.5031
$\|\nabla f(\tilde{x})\|$= 8.77×10^{-4}
Iterations: 10000

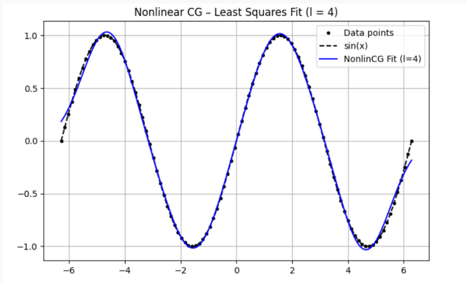


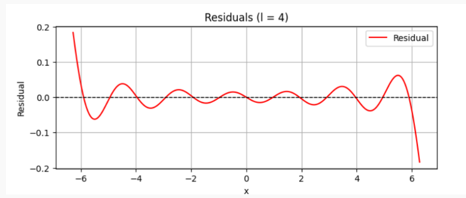


The solution $\tilde{x}$ indicates that Nonlinear Conjugate Gradient (NonlinCG) successfully minimized the objective function, as reflected by the relatively small gradient norm. Although the number of iterations is large, the algorithm achieved a good fit, demonstrating its effectiveness in tackling the nonlinear least squares problem.

Task 28 (l = 4)

Solution $\tilde{x}$= $[1.0382,\ -1.0250,\ 1.0249,\ -1.0381,\ -4.6335,$
$-\ 1.5433,\ \ 1.5436,\ 4.6332,\ 0.8862,\ 0.9774,\ 0.9776,$
$0.8865]$
$f(\tilde{x})$= 0.0808
$\|\nabla f(\tilde{x})\|$= 3.87×10^{-3}
Iterations: 10000

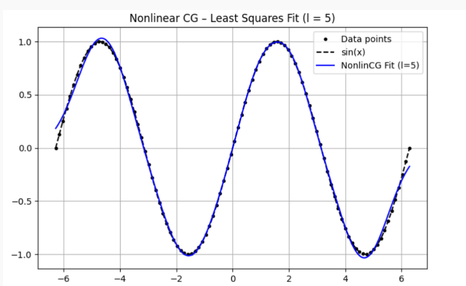


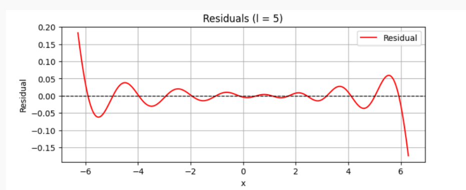


The solution $\tilde{x}$ indicates that Nonlinear Conjugate Gradient (NonlinCG) has made significant progress in fitting the model, with a reasonable final objective function value. However, the relatively higher gradient norm suggests that the convergence was not as tight as for lower values of l. Despite the large number of iterations, the solution achieved a satisfactory fit, though further tuning or regularization might be necessary for even more precise results.

Task 29 (l = 5)

Solution $\tilde{x}$= $[1.0375,\ -1.0179,\ 0.7500,\ 0.5969,\ -1.0344,$
$-\ 4.6361,\ \ -1.5665,\ 1.0749,\ 2.1501,\ 4.6733,$
$0.8843,\ 0.9599,\ 0.7416,\ 0.6519,\ 0.8527]$
$f(\tilde{x})$= 0.0769
$\|\nabla f(\tilde{x})\|$= 1.07×10^{-2}
Iterations: 10000

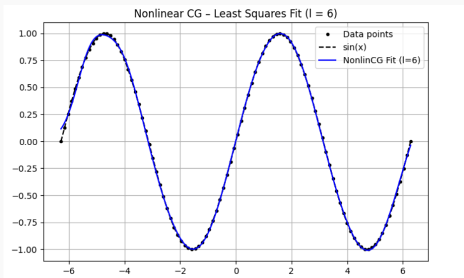


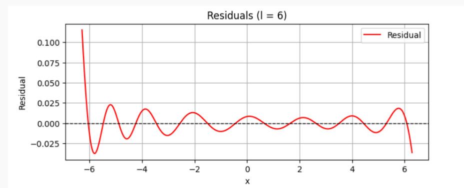


The solution $\tilde{x}$ achieved a reasonable fit with a final objective function value of $f(\tilde{x}) = 0.0769$. However, the gradient norm is relatively high (1.07×10^{-2}), indicating that the convergence is slower compared to lower values of l. Despite 10,000 iterations, the optimization process seems to have stabilized but could benefit from additional regularization or advanced stopping criteria for tighter convergence.

Task 30 (l = 6)

Solution $\tilde{x}$**=** $[0.8509,\ 0.4191,\ -1.5458,\ 1.4231,\ -0.9970,$
$-\ 1.8888,\ \ -4.3041,\ -5.3002,\ -1.3012,\ 2.9429,\ 3.3772,$
$4.9300,\ 0.8059,\ 0.4869,\ 1.2641,\ \ 3.1511,$
$0.9550,\ 1.0616]$
$f(\tilde{x})$= 0.0223
$\|\nabla f(\tilde{x})\|$= 6.96×10^{-3}
Iterations: 10000





The solution $\tilde{x}$ for $l = 6$ results in a final objective function value of $f(\tilde{x}) = 0.0223$ and a gradient norm of 6.96×10^{-3}, which is an acceptable convergence rate but still higher than in cases with fewer Gaussians. Despite the optimization process taking 10,000 iterations, there is some room for improvement in terms of gradient reduction and further tightening the solution. Regularization and more refined stopping criteria could help achieve a more optimal result.