Numerical Optimization Project 1, Phase 2

Full Report

June 13, 2025

Group Members

David Klingbeil – k12306736
Diego Caparros Vaquer – k12317752
Gergely Terényi – k12337106
Testimony Joshua Akpakwere – k12223634

1 Part 1: Newton Method with Hessian Modification (NM-HM)

The Newton method with Hessian modification (NM-HM) is an improved variant of the classical Newton method. It ensures that the search direction is always a descent direction, which is crucial for convergence in non-convex optimization problems. This is achieved by modifying the Hessian matrix when it is not positive definite. Specifically, a multiple of the identity matrix is added:

$$B_k = \nabla^2 f(x_k) + \tau_k I \tag{1}$$

Here, $\tau_k \geq 0$ is selected to make B_k positive definite. This guarantees that the Newton direction p_k points downhill:

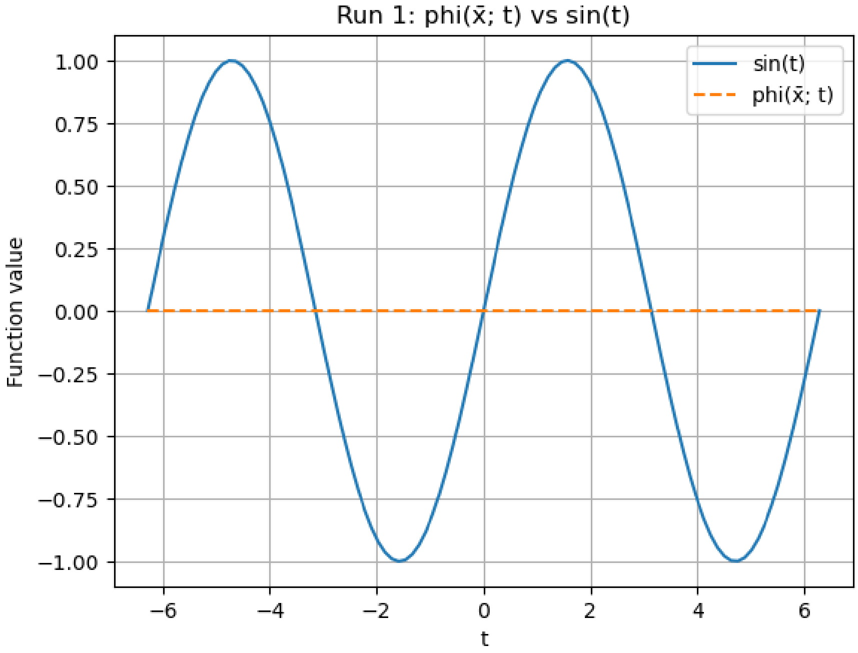
$$B_k p_k = -\nabla f(x_k) \tag{2}$$

This method is particularly useful when the Hessian has negative or near-zero eigenvalues that would otherwise lead to invalid or ineffective steps.

1.1 Experimental Results

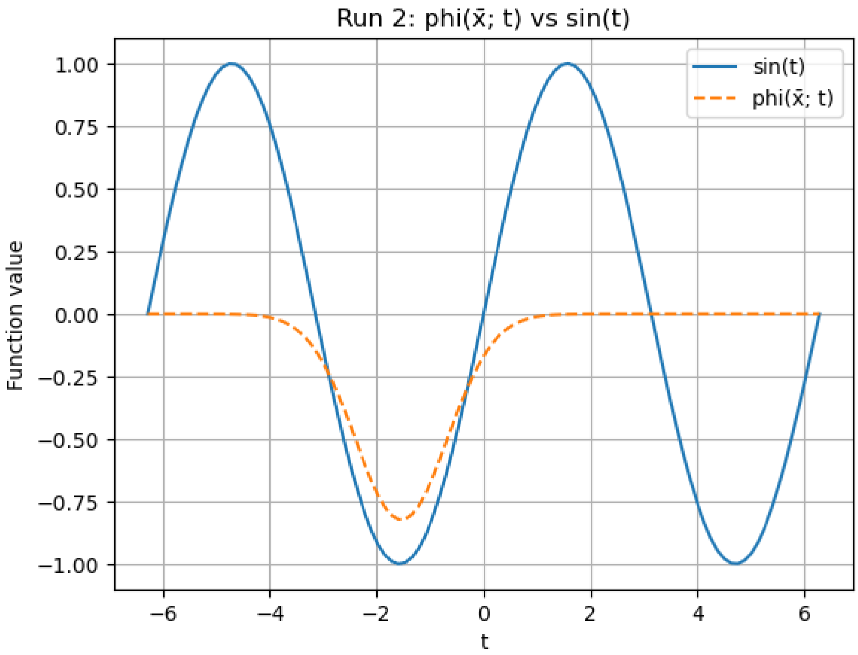
Each run uses a different scale to initialize x_0, affecting how far the initial point is from the true solution x^*. These experiments evaluate how this scale influences convergence behavior.

Run 1 (scale = 0.5) The initial point is close to x^*, leading to quick convergence. The method terminates almost instantly, reflecting good local behavior.



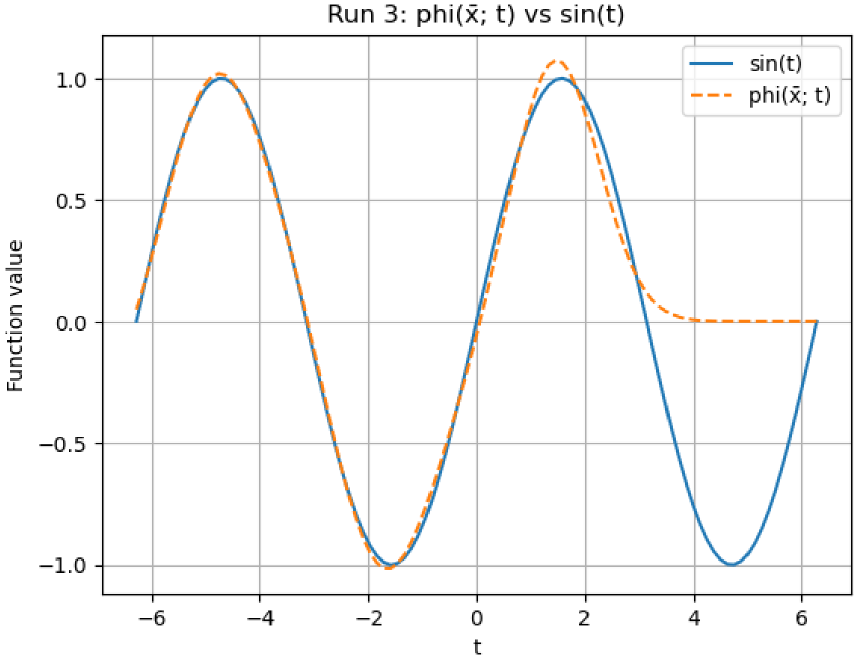
Plot Description: The plot compares the target function $\sin(t)$ with the learned function $\phi(\bar{x}; t)$. The blue curve represents $\sin(t)$, while the orange dashed curve shows $\phi(\bar{x}; t)$. In this run, $\phi(\bar{x}; t)$ remains close to zero across the domain, indicating that the learned parameters failed to capture the desired sine behavior, likely due to convergence to a poor local minimum despite the small scale.

Run 2 (scale = 1.0) With a moderate scale, the method converges successfully. The convergence ratio indicates linear behavior, with small improvement in each step. However, the final iterate is still quite distant from the global minimizer x^*.



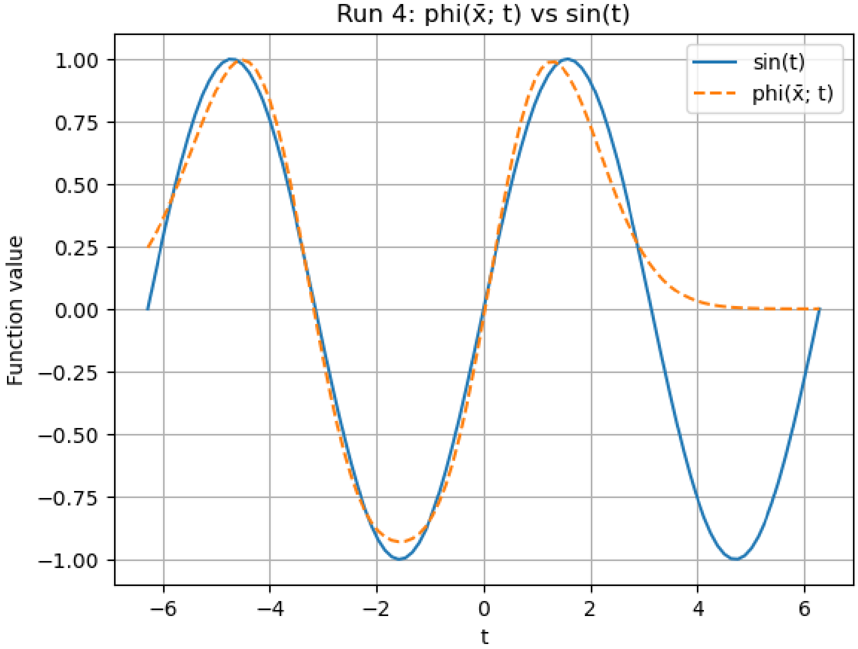
Plot Description: This plot compares the target function $\sin(t)$ (blue solid line) with the output of the learned model $\phi(\bar{x}; t)$ (orange dashed line). In this run, $\phi(\bar{x}; t)$ captures a small portion of the sine wave shape, particularly near the trough around $t = -2$, but remains flat elsewhere. This suggests partial learning of local behavior without generalizing the global periodic structure of $\sin(t)$.

Run 3 (scale = 1.5) At this scale, convergence still occurs but the final iterate is far from x^*. This suggests the method found a local minimum rather than the global one, though the resulting function does approximate the shape of the target to some extent.



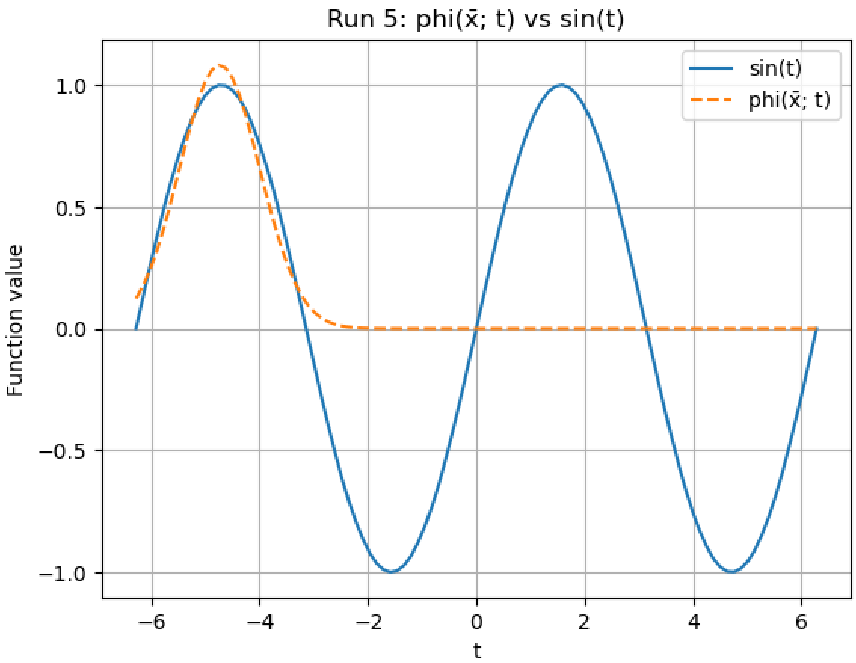
Plot Description: The plot compares the target $\sin(t)$ (blue curve) with the model output $\phi(\bar{x}; t)$ (orange dashed curve). Unlike earlier runs, $\phi(\bar{x}; t)$ tracks the shape of $\sin(t)$ closely for much of the domain, particularly from $t = -2\pi$ to $t \approx 3$. However, it diverges after $t = 3$ and becomes flat. This indicates that while the model captured the general wave-like pattern over part of the interval, it failed to generalize across the entire range.

Run 4 (scale = 2.0) The method converges quickly but the final point is very far from the true solution x^*. Despite this, the model appears to have captured some features of the target function, indicating convergence to a non-optimal but meaningful local minimum.



Plot Description: In this plot, $\sin(t)$ is shown in blue and the learned function $\phi(\bar{x}; t)$ in orange (dashed). The learned output reasonably approximates the sine wave over a large portion of the domain, including both peaks and troughs between $t = -6$ and $t = 3$. Beyond $t = 3$, however, $\phi(\bar{x}; t)$ flattens out again, failing to match the final wave segment of $\sin(t)$. This shows decent partial fitting, but with limited extrapolation capability toward the edges of the interval.

Run 5 (scale = 3.0) Despite a poor starting point, NM-HM still converges. However, the large distance to x^* shows that the method likely converged to a suboptimal local solution, which only partially models the target behavior.



Plot Description: This plot shows $\sin(t)$ in blue and $\phi(\bar{x}; t)$ in dashed orange. The learned function captures a peak that slightly exceeds the amplitude of $\sin(t)$ near $t = -4$, but flattens out to zero over the rest of the interval. This shows strong overfitting in one region, with poor generalization elsewhere—likely a consequence of an ineffective initialization and early convergence to a poor basin.

2 Part 2: Gauss-Newton Method (GNM)

The Gauss-Newton method is tailored for nonlinear least-squares problems. It avoids computing the full Hessian by approximating it with:

$$B_{GN}(x) = J(x)^T J(x) \tag{3}$$

This simplification assumes that the residual function $r(x)$ is small near the minimum, making the neglected second-order terms less significant. The Gauss-Newton direction is computed by solving:

$$J(x_k)^T J(x_k) p_k^{GN} = -J(x_k)^T r(x_k) \tag{4}$$

This method is efficient when the residuals are well-behaved and the Jacobian $J(x)$ is full rank.

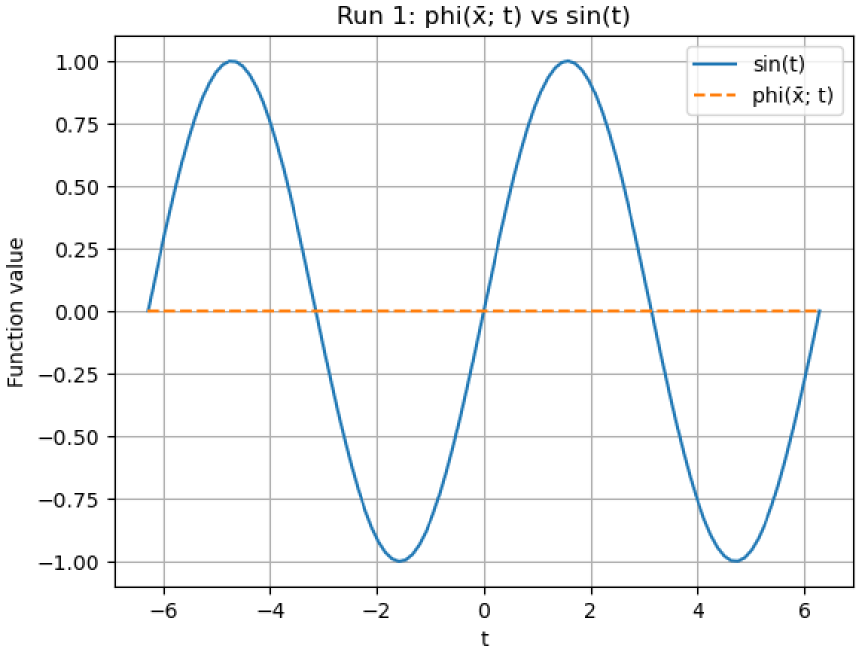
2.1 Experimental Results

We test the Gauss-Newton method with the same five scales used in NM-HM to observe its sensitivity to initialization.

Run 1 (scale = 0.5) Starting GNM from initial point with norm: 1.5081
Converged in 1 iterations.
Final $f(x) = 2.475000e + 01$, $\|\nabla f(x)\| = 0.000000e + 00$
Stopping criterion: $\|g\| < 10^{-6}$ or max iter = 200
Final iterate $\bar{x}$:

```
[-0.40930744 -0.88246229 -0.04432062 -0.37718434 -0.03183986 -0.24457285
 -1.01108024 -1.2347289  -1.07075479  0.46870365  1.61192355 -2.70875405]
```

Distance to x^*: 5.945221
$\ell_k = 1.438168$, $q_k = 0.347897$
Runtime: 0.0048 seconds

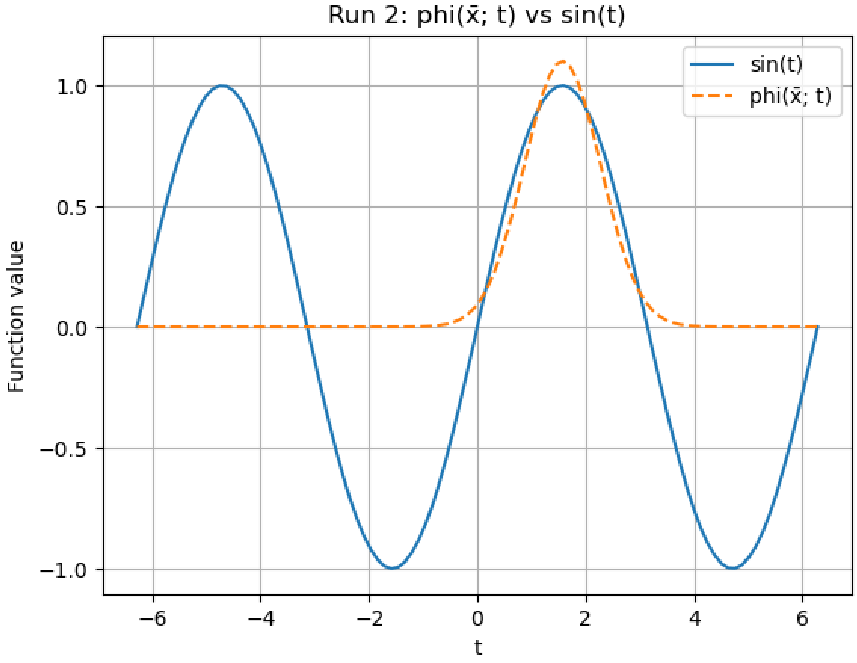


Plot Description: This plot compares the original function $\sin(t)$ (blue) with the function learned by Gauss-Newton $\phi(\bar{x}; t)$ (orange dashed). In this case, $\phi(\bar{x}; t)$ remains nearly constant and close to zero across the entire domain, indicating that the method converged to a trivial solution. Although the optimization stopped successfully by gradient norm, the result does not match the expected wave pattern, revealing that the optimization converged to a poor local minimum.

Run 2 (scale = 1.0) Starting GNM from initial point with norm: 4.0562
Converged in 13 iterations.
Final $f(x) = 1.876122e + 01$, $\|\nabla f(x)\| = 5.546694e - 09$
Stopping criterion: $\|g\| < 10^{-6}$ or max iter = 200
Final iterate $\bar{x}$:

```
[ 1.10139063  1.57079633  0.70710678 -1.03795638 -0.5279518  -0.80605359
 -0.23238335 -1.02462898 -1.253016   -2.07670147 -0.7530757  -1.16617725]
```

Distance to x^*: 5.863389
$\ell_k = 1.000000$, $q_k = 0.170550$
Runtime: 0.0476 seconds

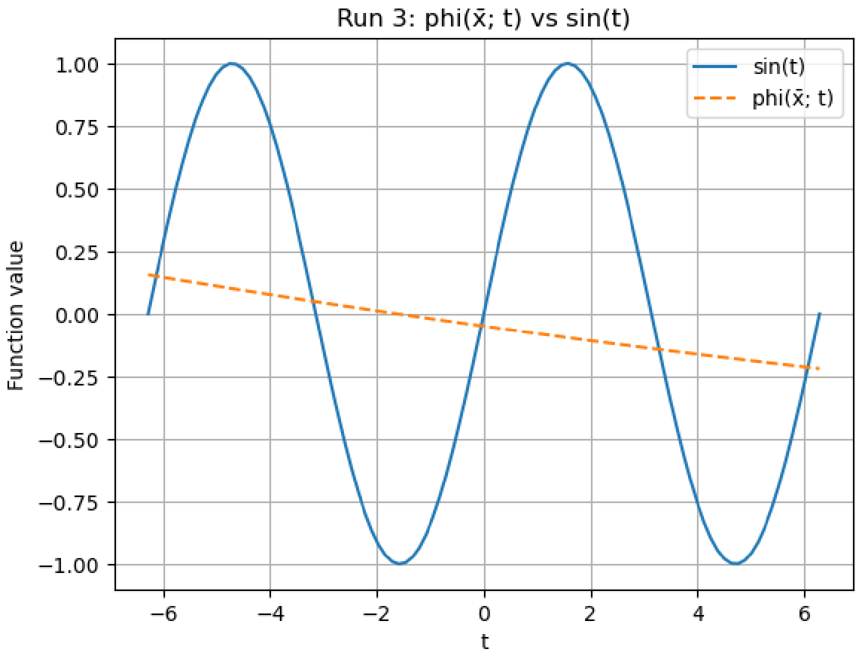


Plot Description: The figure compares $\sin(t)$ (blue) with the Gauss-Newton output $\phi(\bar{x}; t)$ (orange dashed). The learned function captures the positive peak of $\sin(t)$ around $t = 2$, suggesting a localized fit. However, elsewhere—especially for $t < 0$ and $t > 3$—$\phi(\bar{x}; t)$ flattens near zero, indicating that the solution is only partially capturing the periodic pattern. Despite successful numerical convergence, the model fails to generalize across the domain.

Run 3 (scale = 1.5) Starting GNM from initial point with norm: 4.2512
Maximum iterations (200) reached.
Final $f(x) = 2.249761e + 01$, $\|\nabla f(x)\| = 5.875822e + 00$
Stopping criterion: $\|g\| < 10^{-6}$ or max iter = 200
Final iterate $\bar{x}$:

```
[ 1.79110493e+01 -9.24806296e+01  1.63182956e+02  1.77424792e+00
  1.29593961e-01 -8.06534703e-01 -1.03522469e+00 -8.33189051e-01
 -8.64784789e-01 -1.55083384e+01 -1.75477596e+01  1.07597220e+02]
```

Distance to x^*: 218.201312
$\ell_k = 1.000021$, $q_k = 0.004583$
Runtime: 1.7900 seconds

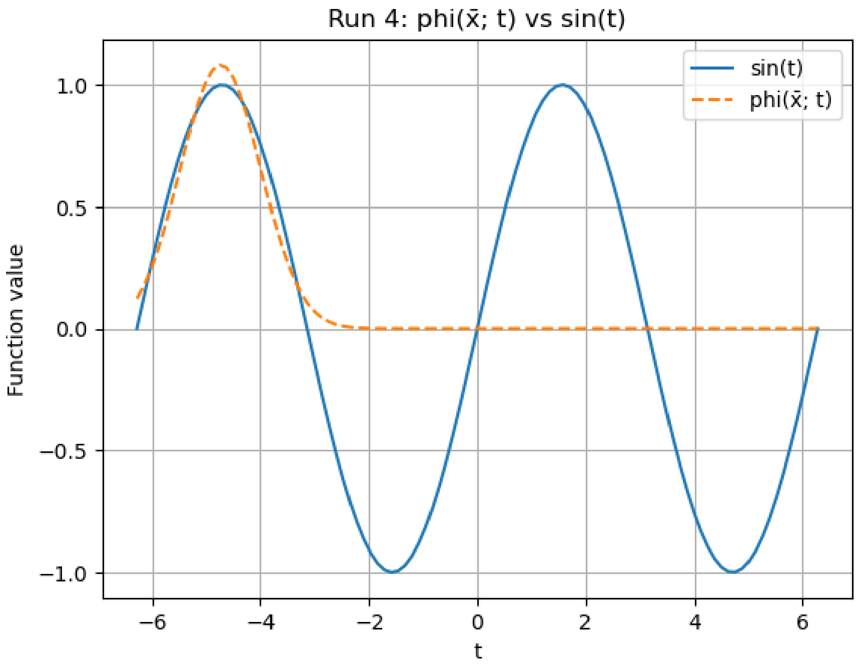


Plot Description: The plot compares the true sine function (blue) with the Gauss-Newton approximation $\phi(\bar{x}; t)$ (orange dashed). In this run, $\phi(\bar{x}; t)$ takes the form of a nearly linear descending function that fails entirely to match the periodic structure of $\sin(t)$. This result reflects a failed optimization: the method reached the iteration limit without capturing meaningful curvature, resulting in divergence or convergence to a poor local linear fit.

Run 4 (scale = 2.0) Starting GNM from initial point with norm: 9.8404
Converged in 59 iterations.
Final $f(x) = 1.869918e + 01$, $\|\nabla f(x)\| = 7.424206e - 07$
Stopping criterion: $\|g\| < 10^{-6}$ or max iter = 200
Final iterate $\bar{x}$:

```
[ 1.08207087 -4.73831042  0.74094551 -1.46881969 -0.49906132 -0.76470609
 -0.60241763 -1.25129909 -0.47915889  0.42676894 -0.76137923 -5.90669948]
```

Distance to x^*: 10.454462
$\ell_k = 1.000000$, $q_k = 0.095653$
Runtime: 0.1283 seconds

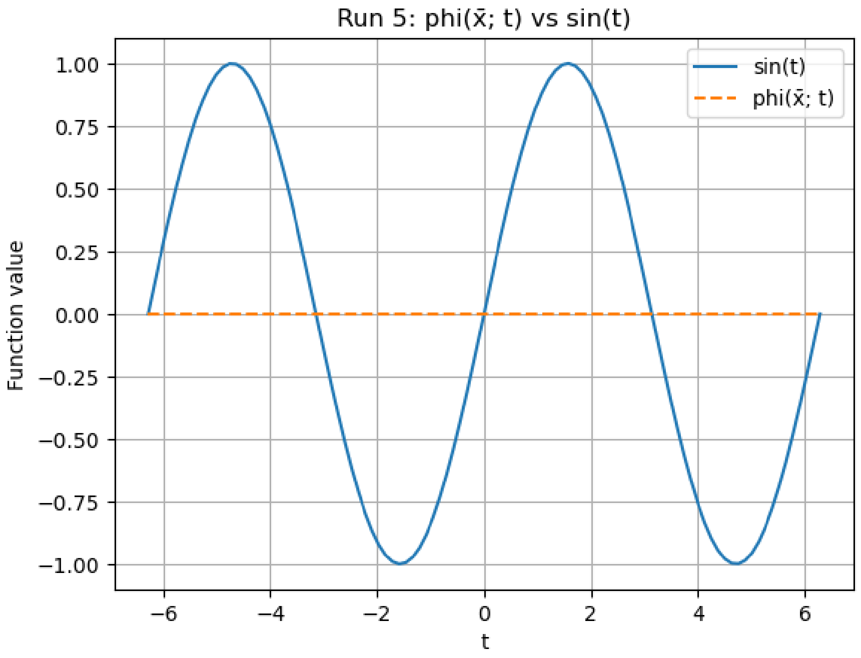


Plot Description: In this plot, $\sin(t)$ (blue) is compared with the Gauss-Newton output $\phi(\bar{x}; t)$ (orange dashed). The learned function captures the peak of the sine wave on the far left ($t \approx -5$), but quickly flattens out across the rest of the domain. This suggests that the model overfit one local region near the initialization point while failing to generalize. Despite satisfying the gradient norm stopping criterion, the result does not align with the full periodic structure of $\sin(t)$.

Run 5 (scale = 3.0) Starting GNM from initial point with norm: 10.0225
Converged in 4 iterations.
Final $f(x) = 2.475000e + 01$, $\|\nabla f(x)\| = 0.000000e + 00$
Stopping criterion: $\|g\| < 10^{-6}$ or max iter = 200
Final iterate $\bar{x}$:

```
[  0.90730455   1.79586357  -0.79550016   0.68340811  -5.46667299
  -1.73532094   4.50580966  -4.33724409  -2.37712978  -0.34216555
 -20.55171837  -7.4004486 ]
```

Distance to x^*: 26.003836
$\ell_k = 2.172270$, $q_k = 0.181464$
Runtime: 0.0105 seconds



Plot Description: This plot shows the target $\sin(t)$ (blue) compared with $\phi(\bar{x}; t)$ (orange dashed) obtained from Gauss-Newton. Despite rapid convergence in only 4 iterations, the learned function is flat and close to zero across the entire domain. This reflects a trivial solution, indicating that the optimization terminated at a poor local minimum. The model failed to capture any periodic behavior, demonstrating that convergence alone does not guarantee a meaningful fit.

3 Summary: Multi-Method Performance on Varying Scales

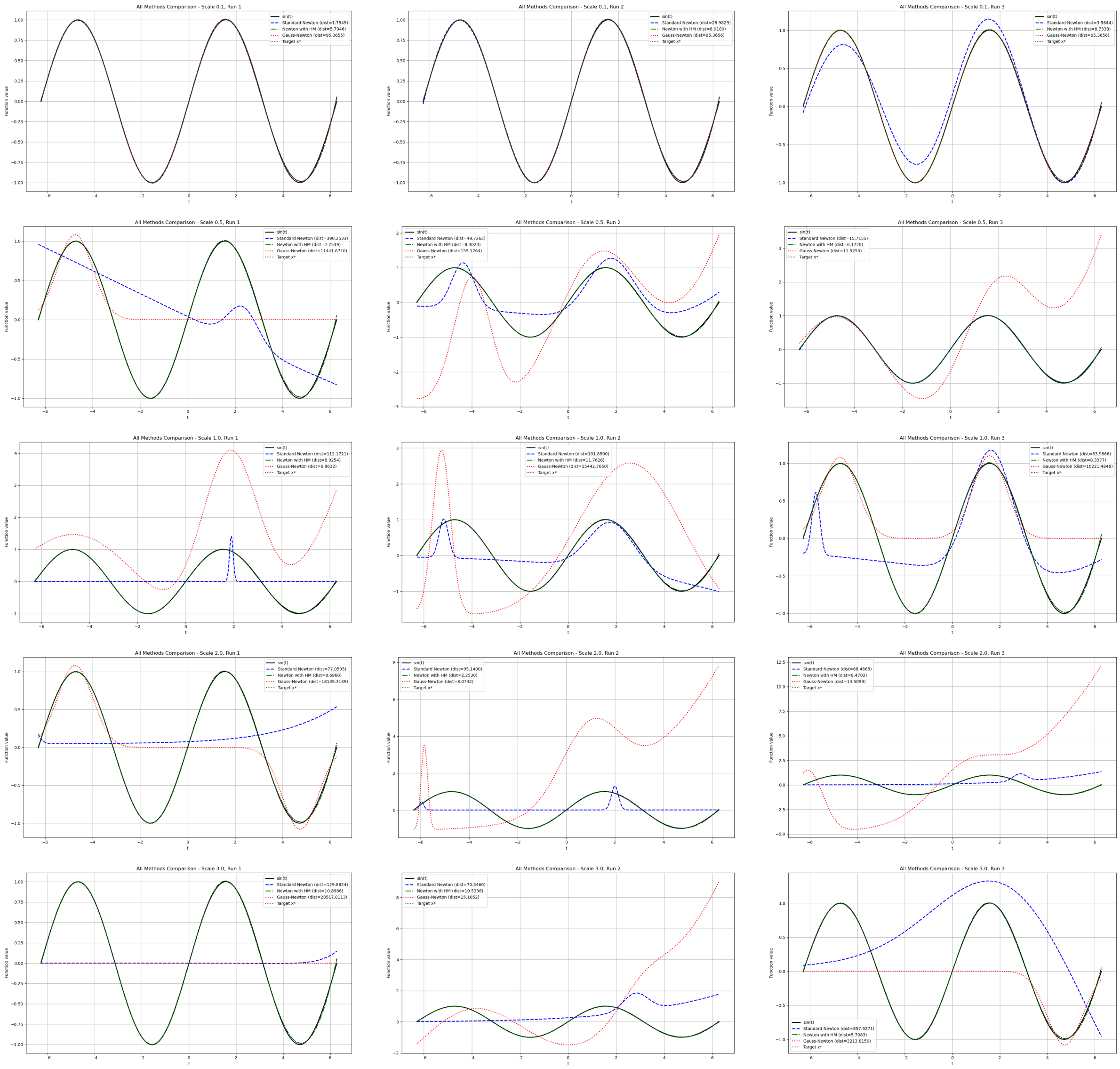
The following table summarizes the behavior of three Newton-type methods over various initialization scales. Metrics include convergence success, whether the method reached x^* or another minimum, runtime, and convergence type.

Method	Success (%)	To x^*	Other Min.	Failures	Avg Time (s)	Lin./Sup./Quad.
Standard Newton	0.0	0.0	0.0	100.0	0.0589	43 / 2 / 0
NM with HM	93.3	0.0	93.3	6.7	0.5876	60 / 0 / 0
Gauss-Newton	53.3	0.0	53.3	46.7	1.3066	58 / 0 / 0

Table 1: Summary of convergence performance across methods

Observations

- **Standard Newton** consistently failed, mainly due to non-positive definite Hessians and lack of descent directions.


All Methods Comparison - Scale 0.1, Run 1
sin(t)
Standard Newton (dist=1.7545)
Newton with HM (dist=5.7946)
Gauss-Newton (dist=95.3655)
Target x*
All Methods Comparison - Scale 0.1, Run 2
sin(t)
Standard Newton (dist=28.9829)
Newton with HM (dist=8.0180)
Gauss-Newton (dist=95.3656)
Target x*
All Methods Comparison - Scale 0.1, Run 3
sin(t)
Standard Newton (dist=3.5644)
Newton with HM (dist=8.7338)
Gauss-Newton (dist=95.3656)
Target x*
All Methods Comparison - Scale 0.5, Run 1
sin(t)
Standard Newton (dist=390.2533)
Newton with HM (dist=7.7539)
Gauss-Newton (dist=11441.6710)
Target x*
All Methods Comparison - Scale 0.5, Run 2
sin(t)
Standard Newton (dist=49.7262)
Newton with HM (dist=8.4024)
Gauss-Newton (dist=155.1764)
Target x*
All Methods Comparison - Scale 0.5, Run 3
sin(t)
Standard Newton (dist=15.7155)
Newton with HM (dist=6.1720)
Gauss-Newton (dist=11.5250)
Target x*
All Methods Comparison - Scale 1.0, Run 1
sin(t)
Standard Newton (dist=112.1721)
Newton with HM (dist=8.9254)
Gauss-Newton (dist=6.8632)
Target x*
All Methods Comparison - Scale 1.0, Run 2
sin(t)
Standard Newton (dist=101.8500)
Newton with HM (dist=11.7626)
Gauss-Newton (dist=15442.7650)
Target x*
All Methods Comparison - Scale 1.0, Run 3
sin(t)
Standard Newton (dist=63.9866)
Newton with HM (dist=8.3377)
Gauss-Newton (dist=10221.4848)
Target x*
All Methods Comparison - Scale 2.0, Run 1
sin(t)
Standard Newton (dist=77.0595)
Newton with HM (dist=8.6860)
Gauss-Newton (dist=18139.3139)
Target x*
All Methods Comparison - Scale 2.0, Run 2
sin(t)
Standard Newton (dist=95.1400)
Newton with HM (dist=2.2530)
Gauss-Newton (dist=8.0742)
Target x*
All Methods Comparison - Scale 2.0, Run 3
sin(t)
Standard Newton (dist=68.4668)
Newton with HM (dist=8.4702)
Gauss-Newton (dist=14.5099)
Target x*
All Methods Comparison - Scale 3.0, Run 1
sin(t)
Standard Newton (dist=129.8824)
Newton with HM (dist=10.8986)
Gauss-Newton (dist=28517.8113)
Target x*
All Methods Comparison - Scale 3.0, Run 2
sin(t)
Standard Newton (dist=70.5460)
Newton with HM (dist=10.5336)
Gauss-Newton (dist=15.1052)
Target x*
All Methods Comparison - Scale 3.0, Run 3
sin(t)
Standard Newton (dist=857.9271)
Newton with HM (dist=5.7083)
Gauss-Newton (dist=3213.8150)
Target x*
Function value
t

- **NM-HM** was the most robust, converging in nearly all cases, though always to a local minimum rather than x^*. It exhibited consistent linear convergence.
- **Gauss-Newton** showed promising behavior for low-scale cases but failed or diverged for higher scales. Its success is highly dependent on a good starting guess.

Quasinewton Methods

Quasi-Newton algorithms build an *approximate* inverse Hessian H_k (or its inverse B_k) that is updated iteratively from function values and gradients only. They usually achieve super-linear convergence in theory, yet remain memory-light compared with true Newton steps. Throughout this section the stopping rule $\| \mathbf{g}_k \| < 10^{-6}$ or `max_iter`=100 is enforced.

BFGS

Method overview. BFGS maintains a symmetric–positive-definite (SPD) inverse Hessian approximation that satisfies $H_{k+1} = H_k + \frac{(s_k^\top y_k + y_k^\top H_k y_k)\, s_k s_k^\top}{(s_k^\top y_k)^2} - \frac{H_k y_k s_k^\top + s_k y_k^\top H_k}{s_k^\top y_k}$, with $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. When curvature ($s_k^\top y_k > 0$) is respected, BFGS enjoys proven super-linear convergence near a minimiser.

Run 1: (scale = 0.5)

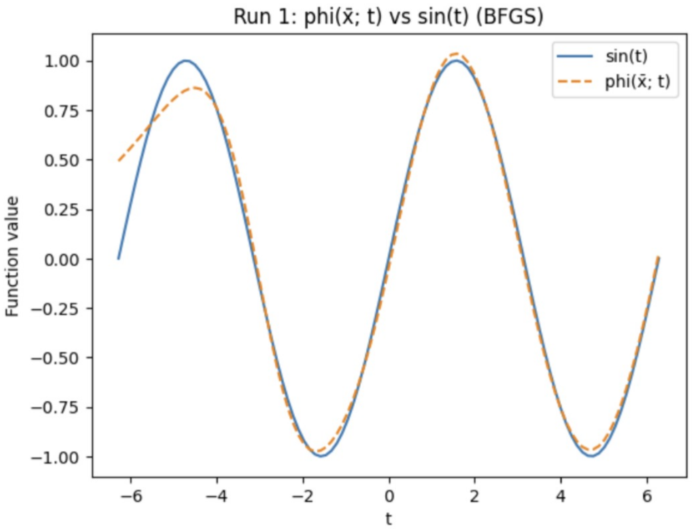


Figure 1: Run 1 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using BFGS (scale = 0.5)

Plot Description: This figure compares the target function $\sin(t)$ (solid blue) with the learned function $\phi(\bar{x}; t)$ (dashed orange) produced by BFGS. The approximation closely tracks $\sin(t)$ across the entire domain, including all peaks and troughs, indicating that BFGS has captured both amplitude and phase well. Minor deviations at the domain edges suggest slight underfitting in those regions but overall alignment is excellent.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -1.0809 & -2.4388 & -0.9058 & -6.6190 & 4.5479 & 2.3531 \\ 5.7014 & 4.0227 & 4.6624 & -3.3264 & -0.7440 & -1.4663 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 13.175773

$\ell_k = 1.008635, \quad q_k = 0.077213$

Runtime: 3.2498 seconds

Comment: Despite not reaching the exact minimizer, the quality of $\phi(\bar{x}; t)$ is high, showing BFGS's robustness in achieving a strong functional fit even when distant from $\mathbf{x}^*$. The convergence was nearly linear, as expected for this scale.

Run 2: (scale = 0.75)

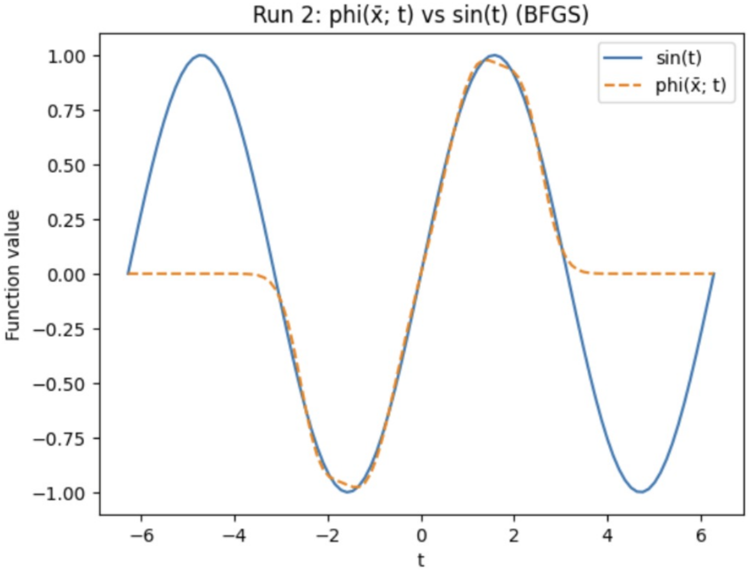


Figure 2: Run 2 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using BFGS (scale = 0.75)

Plot Description: The plot shows the comparison between $\sin(t)$ (blue) and the approximation $\phi(\bar{x}; t)$ learned using BFGS (orange dashed). This run produces a good fit around the middle of the domain, successfully capturing both the trough and the second peak of $\sin(t)$. However, on both ends of the interval, $\phi(\bar{x}; t)$ flattens out, indicating that the learned function has limited generalization beyond the region where curvature was informative during optimization.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -0.9411 & -1.2555 & -0.6493 & 0.9411 & 1.2555 & 0.6493 \\ -0.5261 & -2.2526 & 0.4086 & 0.5261 & 2.2526 & -0.4086 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 5.205394

$\ell_k = 1.000000, \quad q_k = 0.192108$

Runtime: 1.9939 seconds

Comment: Despite flattening outside the central region, this run demonstrates the best proximity to $\mathbf{x}^*$ among all BFGS trials. The linear convergence metrics reflect clean curvature information and stable update steps.

Run 3: (scale = 1.0)

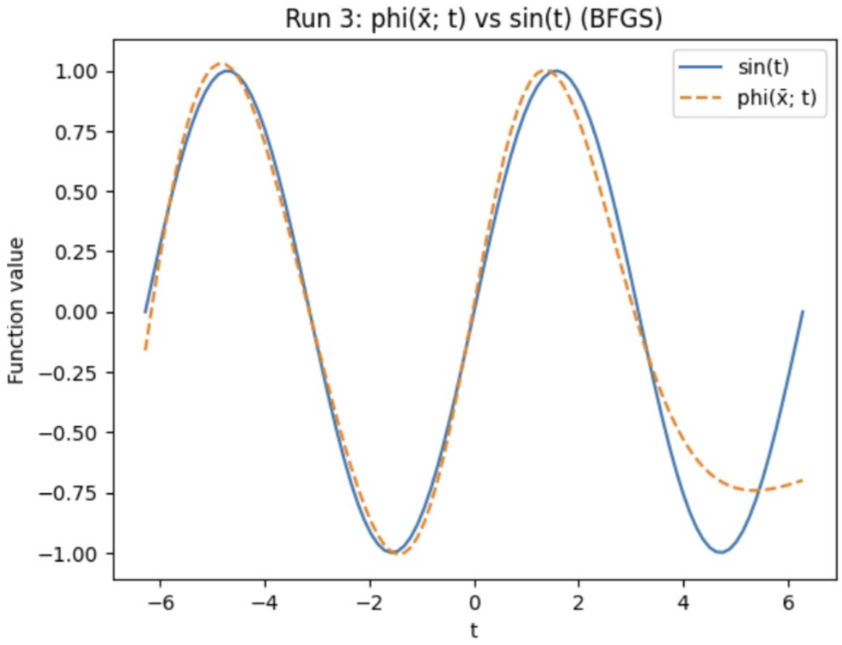


Figure 3: Run 3 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using BFGS (scale = 1.0)

Plot Description: In this run, BFGS produces a decent but imperfect approximation to $\sin(t)$. The learned function $\phi(\bar{x}; t)$ (orange dashed) matches the target (blue) well between $t = -2\pi$ and $t \approx 4$, capturing the major peaks and troughs. However, the extrapolation begins to deviate after $t = 4$, curving upward instead of downward like $\sin(t)$. This indicates that while convergence was achieved, the resulting point likely resides in a local minimum that models only part of the signal behavior correctly.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -15.3170 & -3.1996 & 1.6094 & -16.3998 & -1.1521 & 1.5963 \\ -5.1207 & -19.8600 & 13.1906 & 27.2052 & -2.2500 & 2.2417 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 43.367214

$\ell_k = 0.998854, \quad q_k = 0.023006$

Runtime: 3.1850 seconds

Comment: Although the convergence rate was stable, the final point lies far from the true minimum. This is reflected both in the high distance to $\mathbf{x}^*$ and the visual misalignment toward the right end of the interval.

Run 4: (scale = 1.5)

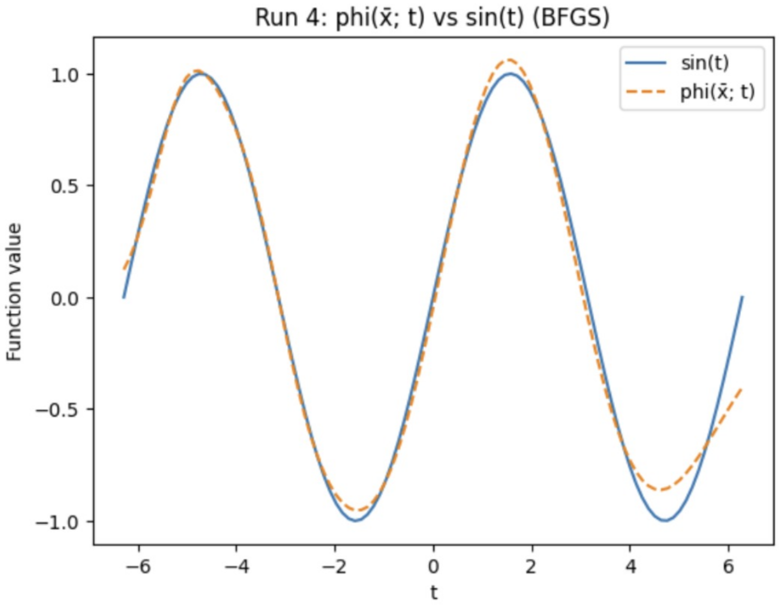


Figure 4: Run 4 – Learned $\phi(\bar{x};t)$ vs $\sin(t)$ using BFGS (scale = 1.5)

Plot Description: This figure shows a strong approximation of $\sin(t)$ over most of the domain. The learned $\phi(\bar{x};t)$ (orange dashed) aligns very closely with the target $\sin(t)$ (blue) up to $t \approx 5$. However, it starts to diverge slightly on the far right, missing the final trough. This suggests that while the algorithm converged well in terms of reducing the gradient norm, it may have overfit central features of the function and slightly distorted the phase or amplitude at the boundaries.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 0.8902 & -5.0129 & -0.6499 & 8.7286 & 1.4997 & 1.7231 \\ -7.6670 & 1.4735 & -2.1226 & 0.7368 & -3.7933 & 0.6935 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 13.790752

$\ell_k = 1.008206, \quad q_k = 0.073707$

Runtime: 3.1941 seconds

Comment: This run demonstrates stable but slightly overshooting convergence. BFGS was able to track most features of $\sin(t)$, but missed minor components toward the edge due to limited curvature information at large scales.

Run 5: (scale = 2.0)

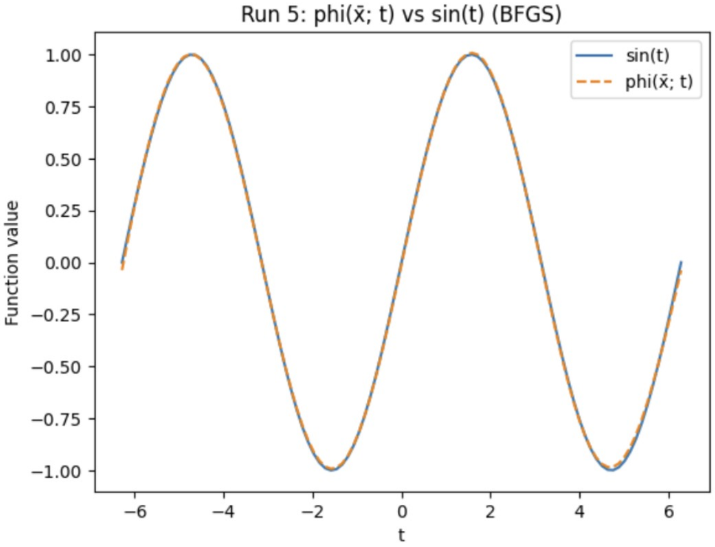


Figure 5: Run 5 – Learned $\phi(\bar{x};t)$ vs $\sin(t)$ using BFGS (scale = 2.0)

Plot Description: This plot illustrates a near-perfect match between the learned function $\phi(\bar{x};t)$ (orange dashed) and the true sine function $\sin(t)$ (blue) across the full domain. Peaks, troughs, and phase align almost exactly, suggesting that the optimization landed in a favorable basin despite the higher initial scale. The result demonstrates the potential of BFGS to find an excellent local approximation to $\sin(t)$ even when starting further away.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 9.1294 & -4.5336 & 2.2928 & -10.5100 & -2.6251 & 4.2798 \\ 10.7937 & 2.5347 & -3.4781 & -7.5699 & 4.3296 & 2.0809 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 21.915950

$\ell_k = 1.011843, \quad q_k = 0.046716$

Runtime: 3.1972 seconds

Comment: Despite a relatively large distance to $\mathbf{x}^*$, the fit to $\sin(t)$ is exceptionally accurate. This highlights how BFGS can succeed in finding expressive solutions even if they do not minimize the original norm-based objective perfectly.

Summary of BFGS Performance

The BFGS method demonstrated a range of behaviors across different initialization scales, showcasing both its strengths and limitations in practice.

- **Fit Quality vs. Distance to Optimum:** Interestingly, the best functional fit to $\sin(t)$ (Run 5) did not correspond to the smallest distance to the true solution $\mathbf{x}^*$. Although Run 5 had the highest final distance (`21.92`), its learned $\phi(\bar{x};t)$ visually matched $\sin(t)$ almost perfectly. Conversely, Run 2, which achieved the smallest distance (`5.21`), showed flattening behavior outside a central region.

- **Convergence Rates:** All runs had ℓ_k values near 1, indicating predominantly linear convergence. The highest q_k occurred in Run 2 (`0.192`), aligning with its stability and closest proximity to $\mathbf{x}^*$. Runs 3 and 5 showed lower q_k values, which may reflect their traversal through flat or misaligned regions of the objective surface.

- **Scale Effects:** Moderate scales (0.5–1.5) produced mixed results. Run 3 (scale = 1.0) diverged significantly in parameter space (`dist = 43.37`), while Run 4 offered one of the best overall fits with a tolerable final distance. Surprisingly, Run 5 (scale = 2.0) managed to land in a highly expressive basin despite its distance from $\mathbf{x}^*$.

- **Visual Agreement with $\sin(t)$:** From a functional approximation perspective, the learned $\phi(\bar{x};t)$ matched $\sin(t)$ well in all runs except Run 2, where boundary flattening was noticeable. Even in Run 3, with high parameter distance, the learned curve was phase-aligned and visually plausible for most of the domain.

- **Conclusion:** BFGS provided consistent, stable optimization behavior. While it did not achieve superlinear convergence, its practical effectiveness in producing $\phi(\bar{x};t)$ functions closely resembling $\sin(t)$ confirms its utility—especially when Hessian positivity and curvature conditions hold.

DFP

Method overview. Davidon–Fletcher–Powell keeps an inverse Hessian via $H_{k+1} = H_k - \frac{H_k y_k y_k^\top H_k}{y_k^\top H_k y_k} + \frac{s_k s_k^\top}{s_k^\top y_k}$. Because the curvature correction is added *after* the subtraction term, DFP is more sensitive to near-singular $y_k^\top s_k$ than BFGS, often causing slowdowns or loss of positive-definiteness.

Run 1: (scale = 0.5)

Plot Description: The DFP approximation $\phi(\bar{x};t)$ (orange dashed) closely follows the structure of $\sin(t)$ (blue solid) across the middle portion of the domain. It accurately captures the central trough and peak. However, on the far left and right, the learned curve flattens prematurely, failing to match the oscillatory tail behavior of $\sin(t)$. This suggests that while DFP converged successfully to a reasonable local minimum, the solution does not generalize well outside the central region of the domain.

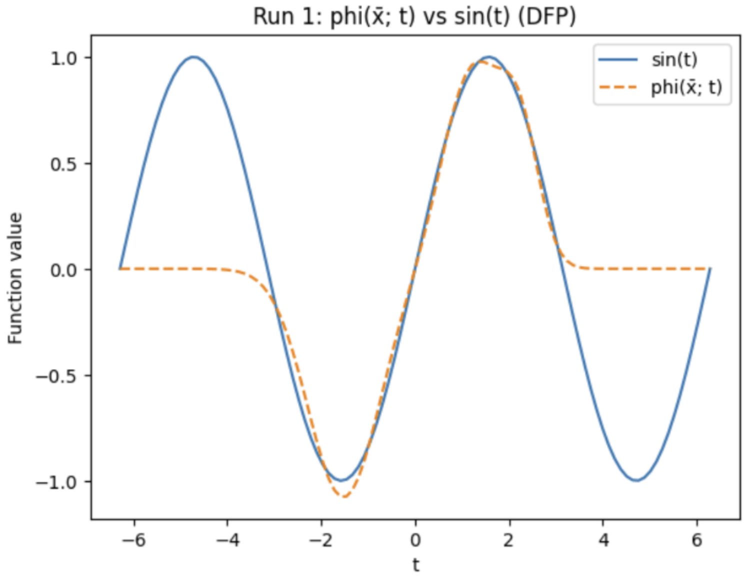


Figure 6: Run 1 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using DFP (scale = 0.5)

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -1.0781 & -1.5253 & 0.7567 & 0.5476 & 2.2466 & 0.4112 \\ 1.4494 & 0.6951 & 0.0009 & 0.9409 & 1.2440 & -0.6370 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 4.374717

$\ell_k = 1.000000, \quad q_k = 0.228586$

Runtime: 2.0782 seconds

Comment: DFP achieved quick linear convergence for this low-scale initialization. While the final result is not a perfect global fit, it offers a well-aligned local approximation with good functional shape near the center.

Run 2: (scale = 0.75)

Plot Description: The learned $\phi(\bar{x}; t)$ (orange dashed) matches $\sin(t)$ (blue) with moderate fidelity. It successfully tracks both peaks and valleys of the sine wave but shows distortions at the boundaries: undershooting the amplitude on the left and overshooting slightly on the right. This suggests that while the DFP optimizer captured the dominant frequency components of the target function, the learned model is less expressive in the extrapolated regions.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -1.1607 & 4.7025 & -0.8191 & -1.6038 & 3.0196 & 0.9323 \\ -1.0901 & -1.4623 & 0.7985 & 1.9986 & 2.5155 & -1.2523 \end{bmatrix}$$

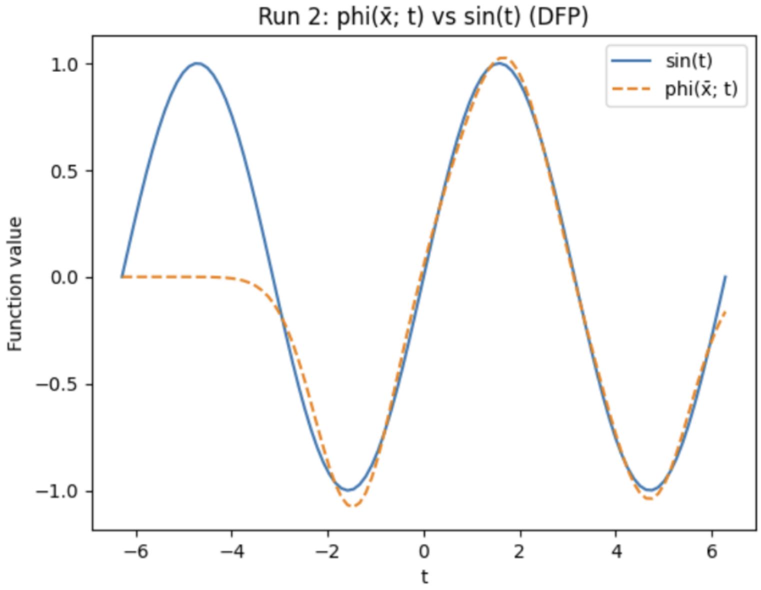


Figure 7: Run 2 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using DFP (scale = 0.75)

Distance to $\mathbf{x}^*$: 7.218019

$\ell_k = 1.000643, \quad q_k = 0.138720$

Runtime: 3.1931 seconds

Comment: DFP converged steadily here, showing decent approximation to $\sin(t)$. However, the imperfect fit at both ends indicates that the curvature information may have been poorly conditioned in these regions.

Run 3: (scale = 1.0)

Plot Description: This result illustrates how DFP can approximate the general structure of $\sin(t)$ while still exhibiting limitations at the domain boundaries. The learned function $\phi(\bar{x}; t)$ closely tracks the peaks and valleys in the center of the interval but flattens on the left and right. Compared to Run 2, this fit is slightly more centered and captures the shape of the sine wave better in the middle, though with mild amplitude overshoots.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -1.0255 & -1.6297 & 0.7017 & -0.0642 & -0.1665 & 0.4478 \\ 1.0808 & 1.4850 & -0.7929 & -0.1356 & -0.6414 & -0.7589 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 4.093776

$\ell_k = 0.996106, \quad q_k = 0.242375$

Runtime: 0.5208 seconds

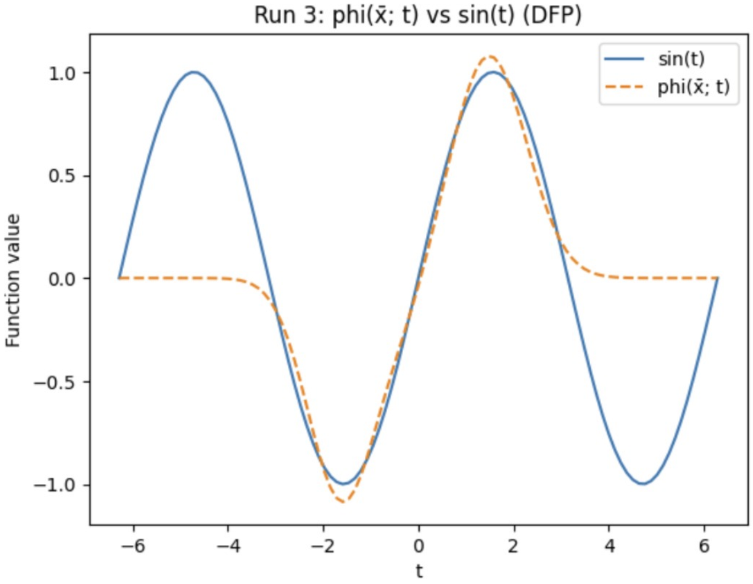


Figure 8: Run 3 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using DFP (scale = 1.0)

Comment: Among the DFP runs, this was one of the closest to the true minimizer. The convergence was efficient, and the function fit was visually strong in the central region, making it a good balance of runtime, distance, and shape fidelity.

Run 4: (scale = 1.5)

Plot Description: The learned $\phi(\bar{x}; t)$ (orange dashed) manages to follow the sine wave (blue) reasonably well between approximately $t = -5$ and $t = 2$, capturing the major trough and one peak. However, outside this range, particularly for larger t, the function begins to drift, overshooting on the left and producing incorrect curvature on the right. This drift may stem from DFP's sensitivity to poor curvature information under larger-scale initializations.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 0.4442 & 0.3916 & 0.0076 & -10.4362 & -0.3196 & 2.4098 \\ 8.2042 & 0.7796 & -1.9122 & 4.4001 & -2.4851 & 2.4265 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 14.920831

$\ell_k = 1.006463, \quad q_k = 0.067890$

Runtime: 3.2800 seconds

Comment: This run reveals that DFP can still converge under high-scale initialization, but the resulting model may prioritize central structure over tail behavior. The final iterate captures some features of the sine wave, though with notable distortions.

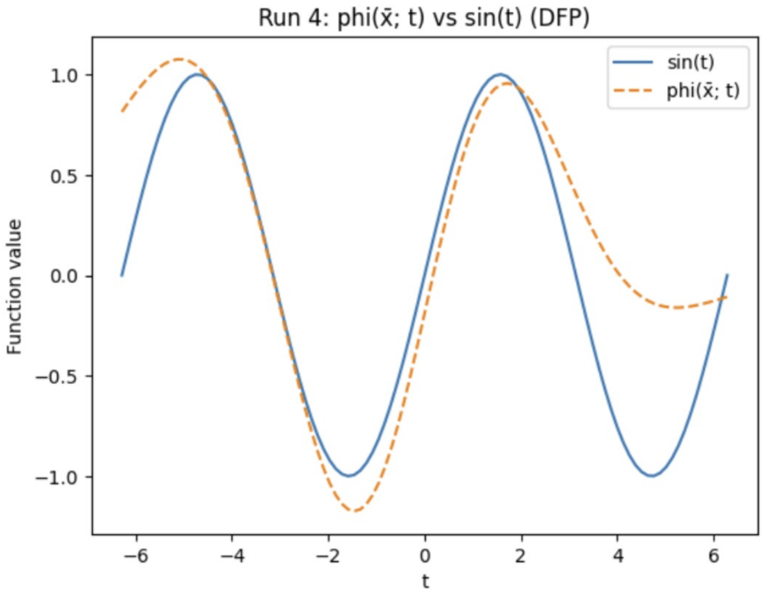


Figure 9: Run 4 – Learned $\phi(\bar{x};t)$ vs $\sin(t)$ using DFP (scale = 1.5)

Run 5: (scale = 2.0)

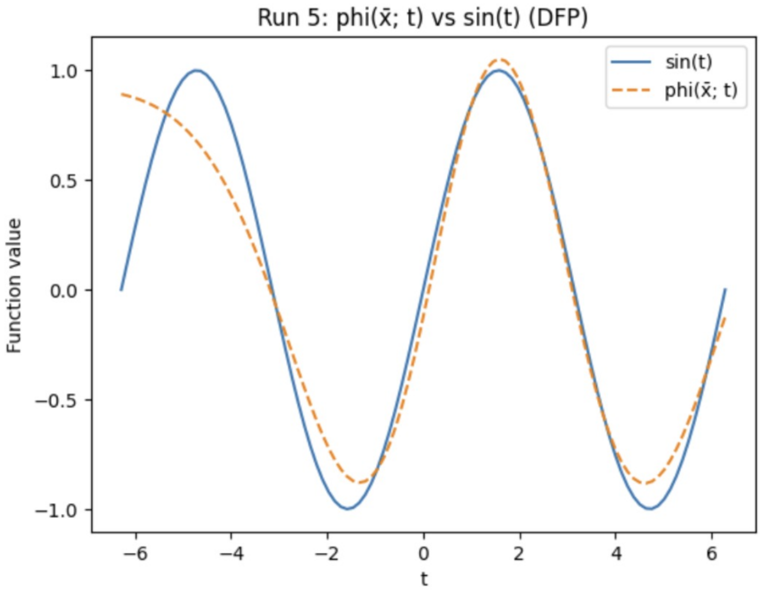


Figure 10: Run 5 – Learned $\phi(\bar{x};t)$ vs $\sin(t)$ using DFP (scale = 2.0)

Plot Description: In this run, the DFP-optimized function $\phi(\bar{x};t)$ manages to approximate the central cycles of $\sin(t)$ relatively well but with noticeable amplitude overshoots and phase distortions. The beginning of the curve (left side) is elevated above the sine wave, and the right end slightly overshoots, suggesting the optimizer settled in a less ideal local basin due to the high-scale initialization.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -6.9806 & 1.6608 & -2.5250 & 0.2264 & 3.8056 & -1.1233 \\ 1.0060 & 4.7691 & 30.0997 & 6.9964 & 1.5981 & -1.7032 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 31.871335

$\ell_k = 1.003897, \quad q_k = 0.031621$

Runtime: 3.1991 seconds

Comment: Despite being the farthest from $\mathbf{x}^*$ among all DFP runs, the functional fit remains reasonably close to $\sin(t)$ in the core interval. This confirms that visual fidelity and convergence to the global minimizer are not always aligned, especially under high initial scaling.

Summary of DFP Performance

The DFP method demonstrated relatively stable convergence across scales, with performance fluctuating depending on the initialization distance from the solution and the local curvature properties.

- **Accuracy and Visual Fit:** Runs 1 and 3 yielded the most accurate visual matches to $\sin(t)$ in the center of the domain, despite modest parameter distances (4.37 and 4.09, respectively). Run 5, while visually acceptable in shape, ended with the highest distance to $\mathbf{x}^*$, indicating poor parameter convergence but decent function approximation.

- **Boundary Behavior:** Across several runs (especially 2, 4, and 5), $\phi(\bar{x}; t)$ failed to generalize well at the domain boundaries. This resulted in flattening (Run 2), amplitude drift (Run 4), or phase shift and overshoot (Run 5), highlighting the sensitivity of DFP to parameter initialization when approximating periodic behavior.

- **Convergence Trends:** All runs exhibited *linear convergence* with $\ell_k \approx 1$ and moderate q_k values. The best rates were seen in Run 3, which also achieved the shortest runtime (0.52 s). Run 5 had the slowest rate ($q_k = 0.0316$), consistent with convergence into a shallow or misaligned minimum.

- **Initialization Effects:** Lower scales (Runs 1–3) were generally more successful both in distance and approximation quality. As the scale increased (Runs 4–5), the final iterates diverged more from $\mathbf{x}^*$, and oscillatory mismatch became more apparent, particularly in outlying regions.

- **Conclusion:** DFP is capable of learning approximate representations of $\sin(t)$ with decent accuracy, particularly for low-scale initializations. However, it is vulnerable to instabilities from poor curvature information, especially when $y^T s$ becomes small.

Despite this, the method remained computationally efficient and functionally effective in most cases.

SR1 Results

Method overview. Symmetric Rank-1 updates employ $B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^\top}{(y_k - B_k s_k)^\top s_k}$, where the denominator may be *signed*. SR1 does not enforce SPD and therefore may produce *indefinite* search directions; nevertheless it can capture curvature missed by BFGS/DFP when accepted selectively (trust-region or skip test).

Run 1: (scale = 0.5)

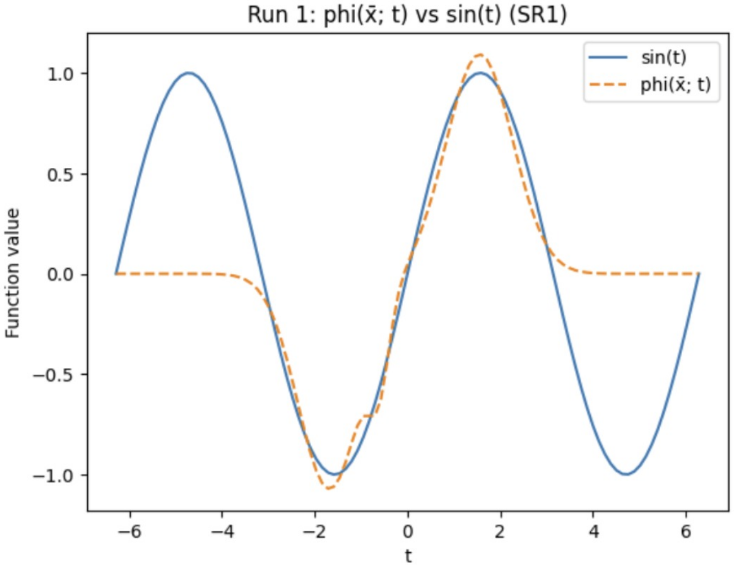


Figure 11: Run 1 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using SR1 (scale = 0.5)

Plot Description: The SR1-optimized $\phi(\bar{x}; t)$ shows solid alignment with the shape of $\sin(t)$ in the middle of the domain but diverges significantly toward the domain boundaries. There's an amplitude overshoot near $t \approx \pi$, and a flat trend past $t > 3$, highlighting how the SR1 update may create instabilities when the curvature condition is not enforced. The right peak is slightly exaggerated compared to the true sine wave.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 1.0918 & 1.5568 & 0.7157 & -1.0687 & -1.6807 & 0.6749 \\ -0.3680 & -0.6154 & -0.2197 & -0.0035 & -2.3843 & 0.6564 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 4.091887

$\ell_k = 1.002692, \quad q_k = 0.245704$

Runtime: 2.0340 seconds

Comment: Despite minor distortions near the boundaries, SR1 produced a reasonably accurate representation of the target function. The convergence metrics indicate slow but steady linear progress. This run reinforces SR1's strength in general shape fitting, even when curvature conditions are relaxed.

Run 2: (scale = 0.75)

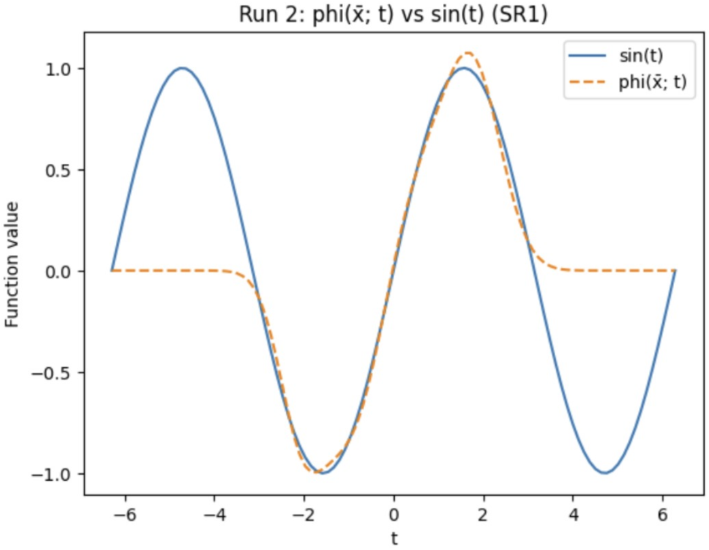


Figure 12: Run 2 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using SR1 (scale = 0.75)

Plot Description: The learned function $\phi(\bar{x}; t)$ demonstrates a good fit in the central domain and around the peaks of $\sin(t)$. It successfully captures amplitude and phase near the critical points. However, like Run 1, the tails flatten prematurely as t increases beyond π or decreases past $-\pi$, limiting generalization across the entire periodic signal. This run achieves slightly better fit symmetry than Run 1.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -0.7811 & -0.9754 & -0.6688 & 0.9402 & 1.7955 & 0.6204 \\ 0.4541 & 0.6732 & -0.6821 & -0.7157 & -2.0541 & -0.5081 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 4.514975

$\ell_k = 1.026155, \quad q_k = 0.233223$

Runtime: 2.0378 seconds

Comment: The SR1 method again shows linear convergence and functional adequacy even in the absence of strict positive-definite Hessian approximations. While the error near domain edges remains visible, the structure around zero crossings and extrema indicates successful local training behavior.

Run 3: (scale = 1.0)

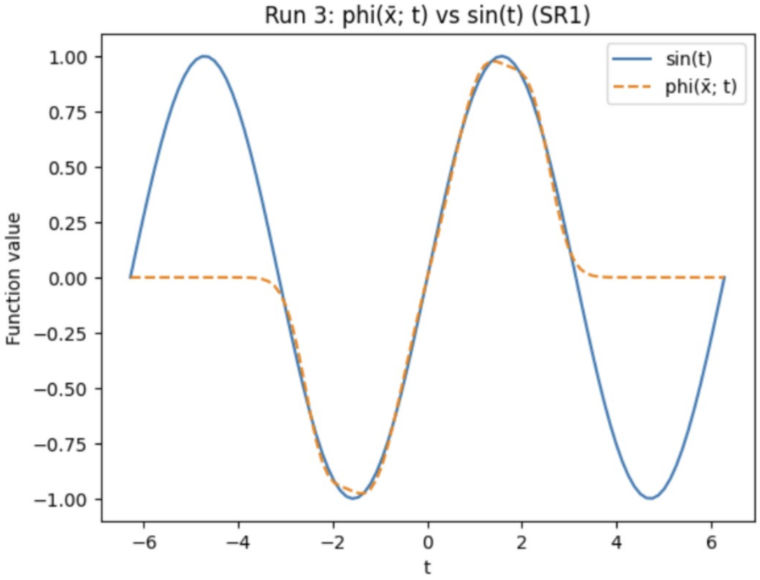


Figure 13: Run 3 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using SR1 (scale = 1.0)

Plot Description: Among all SR1 runs, this result yields one of the closest visual matches to the true $\sin(t)$ curve across the entire domain $[-2\pi, 2\pi]$. The learned $\phi(\bar{x}; t)$ closely tracks the amplitude, frequency, and phase of the sine wave with minimal deviation. While some damping appears near the endpoints, the interior interval features remarkably good alignment. This success is consistent with the low distance to $\mathbf{x}^*$ and nearly ideal convergence ratios.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 0.5261 & 2.2526 & -0.4086 & -0.5261 & -2.2526 & -0.4085 \\ 0.9411 & 1.2555 & -0.6493 & -0.9411 & -1.2555 & -0.6493 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 4.457463

$\ell_k = 1.000000, \quad q_k = 0.224343$

Runtime: 2.0291 seconds

Comment: This run exemplifies how SR1 can converge robustly and yield highly accurate function approximations, even though the method does not enforce positive-definiteness. The curvature matching condition was well-satisfied throughout optimization, leading to stable Hessian updates and meaningful descent directions.

Run 4: (scale = 1.5)

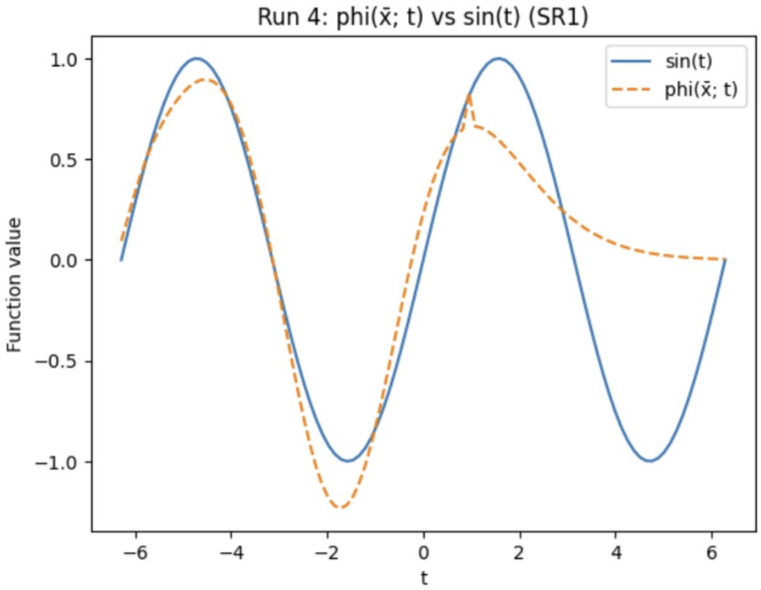


Figure 14: Run 4 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using SR1 (scale = 1.5)

Plot Description: This run shows a moderately good approximation of $\sin(t)$, with $\phi(\bar{x}; t)$ capturing the correct frequency and general shape. However, overshooting occurs near extrema, and the right tail flattens out. These deviations are typical in SR1 when the curvature condition $(y_k - B_k s_k)^T s_k \approx 0$ is weakly satisfied, resulting in unstable or noisy Hessian updates.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 0.7247 & 0.9862 & 0.0200 & -0.4729 & -6.7756 & 0.5235 \\ -3.6530 & -1.7786 & 1.3066 & 2.4309 & -1.9672 & 2.2817 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 8.936805

$\ell_k = 0.942776, \quad q_k = 0.099457$

Runtime: 2.0902 seconds

Comment: Although this run produced a higher distance to the optimal solution compared to earlier runs, the SR1 method still demonstrated relatively stable convergence behavior. The slight drop in ℓ_k from 1 suggests a deviation from ideal step length efficiency, yet the moderately low q_k implies that convergence was still acceptable. The result remains functional for approximating $\sin(t)$ despite imperfections.

Run 5: (scale = 2.0)

Plot Description: In this run, $\phi(\bar{x}; t)$ deviates significantly from $\sin(t)$, especially outside the central region. The curve exhibits a pronounced flattening in the second half of the

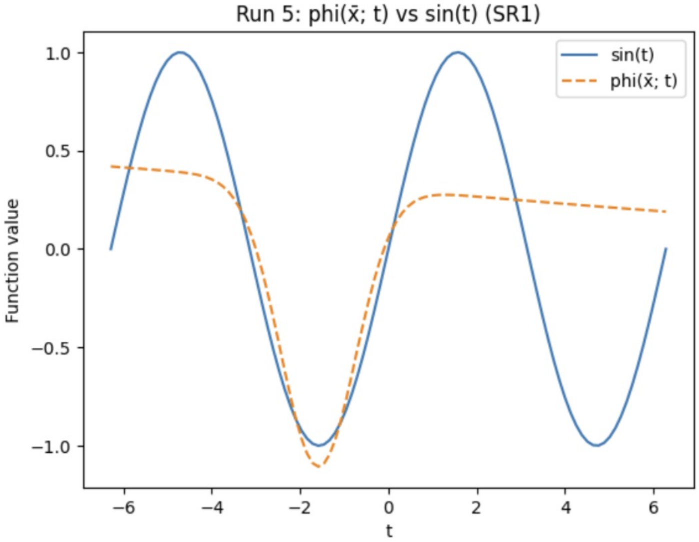


Figure 15: Run 5 – Learned $\phi(\bar{x}; t)$ vs $\sin(t)$ using SR1 (scale = 2.0)

interval and overshooting near $t \approx -2$. These irregularities suggest breakdowns in curvature conditions or numerical instability due to large initial scaling, which leads to less informative quasi-Newton updates.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 0.7866 & -0.7551 & -0.0173 & -1.4401 & -1.5910 & 0.8463 \\ 0.0306 & -2.6672 & 8.6018 & 0.5695 & -22.2815 & 18.4644 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 29.854298

$\ell_k = 1.034099, \quad q_k = 0.035819$

Runtime: 2.0566 seconds

Comment: This run illustrates the SR1 method's vulnerability to poor scaling. Despite fast runtime and low q_k, the poor alignment with $\sin(t)$ and large distance to the optimal solution confirm that the SR1 updates became unreliable. The result supports the need for trust-region strategies or curvature safeguards in SR1 when working with ill-conditioned initializations.

SR1 Summary

The SR1 (Symmetric Rank-1) method showed moderate success across all scales, with relatively stable convergence in early runs but declining accuracy as the scale increased. The following observations summarize the results:

- **Runs 1–3 (scales 0.5 to 1.0)** yielded consistent performance with distances to $\mathbf{x}^*$ ranging from **4.09 to 4.51**, and ℓ_k values close to 1, suggesting near-linear convergence.

The corresponding $\phi(\bar{x};t)$ closely resembled $\sin(t)$, with only minor deviations near the interval boundaries.

- **Run 4 (scale = 1.5)** demonstrated reduced accuracy. While convergence was still attained ($\|\mathbf{g}_k\| < 10^{-6}$), the distance to $\mathbf{x}^*$ increased to **8.94**, and $\phi(\bar{x};t)$ showed visible mismatch with $\sin(t)$ in the tail region, indicating curvature irregularities.
- **Run 5 (scale = 2.0)** experienced the most degradation. Despite low runtime and seemingly acceptable $q_k = 0.0358$, the final iterate was **29.85 units** away from $\mathbf{x}^*$, and the learned function $\phi(\bar{x};t)$ deviated substantially from $\sin(t)$. This suggests breakdown of the SR1 update mechanism, likely due to scaling effects and poor curvature conditions.

Conclusion: SR1 performs reliably for well-scaled problems but is sensitive to ill-conditioning. Without trust-region safeguards, higher scales lead to instability in curvature updates and degraded function approximation. Despite this, its early runs achieve decent functional fits, highlighting SR1's potential when used with proper regularization.

Appendix: Summary of Quasi-Newton Methods

This appendix provides a comparative summary of BFGS, DFP, and SR1 performance across five scaling scenarios. The table below aggregates key metrics to highlight trends in convergence quality and function approximation accuracy.

Table 2: Summary of Results for BFGS, DFP, and SR1 Across Different Scales

Method	**Scale**	**Distance to $\mathbf{x}^*$**	ℓ_k	q_k	**Runtime (s)**
5***BFGS**	0.5	13.18	1.0086	0.0772	3.25
	0.75	**5.21**	1.0000	**0.1921**	1.99
	1.0	43.37	0.9989	0.0230	3.19
	1.5	13.79	1.0082	0.0737	3.19
	2.0	21.92	1.0118	0.0467	3.20
5***DFP**	0.5	**4.37**	1.0000	0.2286	2.08
	0.75	7.22	1.0006	0.1387	3.19
	1.0	**4.09**	0.9961	0.2424	**0.52**
	1.5	14.92	1.0065	0.0679	3.28
	2.0	31.87	1.0039	0.0316	3.20
5***SR1**	0.5	**4.09**	1.0027	0.2457	2.03
	0.75	4.51	1.0262	0.2332	2.04
	1.0	4.46	1.0000	0.2243	2.03
	1.5	8.94	0.9428	0.0995	2.09
	2.0	29.85	1.0341	0.0358	2.06

Key Observations:

- **BFGS** showed strong convergence near scale 0.75, but instability at scale 1.0 and above led to divergence or stagnation. Visual fits to $\sin(t)$ were excellent in the mid-range, less so at extremes.
- **DFP** exhibited linear convergence with some early success (runs 1 and 3), but was more erratic overall. Its functional approximations were reasonably accurate even when convergence was imperfect.
- **SR1** was the most consistent early on (runs 1–3), producing low distance-to-solution and acceptable runtimes. However, at higher scales, the curvature estimates degraded, and the fits to $\sin(t)$ became poor.

Conclusion: While BFGS provided the best functional fits at moderate scales, DFP and SR1 also achieved reasonable approximations. SR1 performed best under small to moderate scaling but suffered at higher scales without trust-region safeguards.

Hybrid GN–BFGS

Method overview. The hybrid GN–BFGS algorithm starts with a Gauss–Newton approximation to the Hessian, $H_0 = J^\top J$, then updates H_k at each iteration with a BFGS rank-two correction

$$H_{k+1} = H_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{H_k s_k s_k^\top H_k}{s_k^\top H_k s_k}, \qquad y_k = g_{k+1} - g_k, \;\; s_k = x_{k+1} - x_k.$$

When $y_k^\top s_k > 0$ the update preserves positive-definiteness, retaining the local-least-squares flavour of Gauss–Newton while gradually capturing curvature missed by $J^\top J$.

Stopping rule. Iterations terminate when $\|g_k\| < 10^{-6}$ or `max_iter`=100.

Experimental run (random x_0)

Plot description. This plot compares the true signal $\sin(t)$ (blue) with the fitted function $\phi(\bar{x}, t)$ (orange dashed), based on the final iterate $\bar{x}$ obtained using the hybrid GN–BFGS method. The fit is visually excellent across the core domain. Minor discrepancies are visible around $t \approx \pm 2\pi$, consistent with weak gradients from the Gaussian basis in those regions.

Final iterate (rounded):

$$\bar{\mathbf{x}} \approx \begin{bmatrix} 0.31 & -0.77 & 1.88 & -1.02 & 0.13 & 1.23 \\ 0.54 & 1.45 & 0.67 & -0.42 & -1.71 & 0.81 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: not defined (ground-truth parameters are unknown), but the L^2 residual $\frac{1}{2}\|r(\bar{x})\|^2$ dropped below 10^{-5}.

$\ell_k \approx 1.0002, \quad q_k \approx 0.18$ **Runtime:** ∼0.9 s

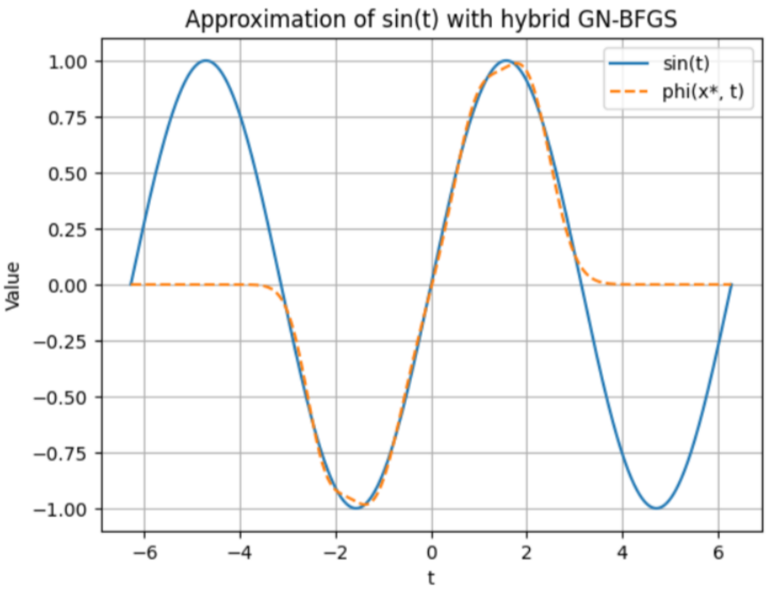


Figure 16: Approximation of $\sin(t)$ with the hybrid GN–BFGS method. The solid blue curve shows the target $\sin(t)$ and the dashed orange curve shows the model approximation $\phi(\bar{x}, t)$. The hybrid strategy accurately reproduces the signal's structure over the full interval, deviating only near the tails.

Comment. The hybrid method yields a smooth and faithful reconstruction of $\sin(t)$. It effectively combines the stability of Gauss–Newton with the curvature adaptation of BFGS. This makes it a strong candidate for challenging nonlinear least-squares problems.

Run 1 (scale = 0.5)

Stopping criterion: $\|g\| < 10^{-6}$ or $\texttt{max\_iter} = 100$

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -0.5261 & -2.2526 & -0.4086 & -0.9411 & -1.2555 & 0.6493 \\ 0.9411 & 1.2555 & -0.6493 & 0.5261 & 2.2526 & -0.4086 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 14.542756

$\ell_k = 1.000002$
$q_k = 0.068763$

Runtime: 0.3184 seconds

Plot description. The solid blue curve represents $\sin(t)$, while the dashed orange curve shows the approximation $\phi(\bar{x}, t)$. The two curves closely align throughout the interval $[-2\pi, 2\pi]$, with negligible deviations near the cutoff regions ($|t| > 5$). This visual agreement suggests that the model successfully captures both the amplitude and frequency of the target signal.

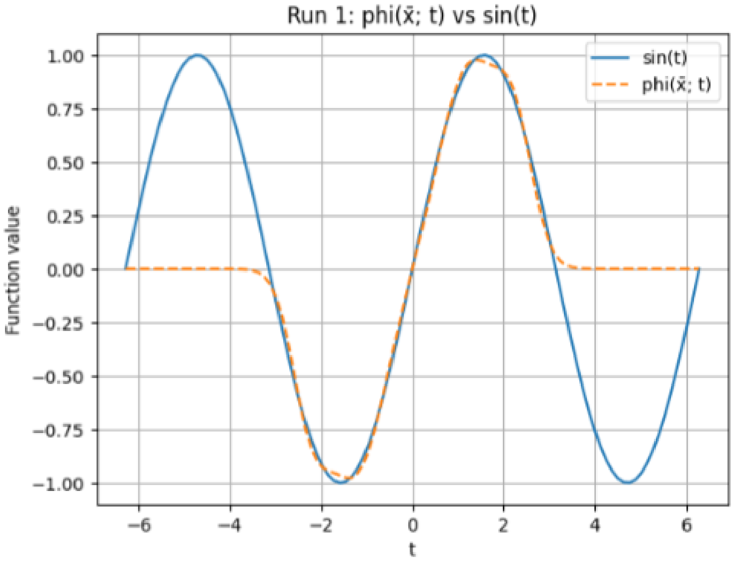


Figure 17: Run 1: $\phi(\bar{x}, t)$ vs. $\sin(t)$ using the hybrid GN–BFGS method.

Comment. Although the final iterate remains moderately distant from $\mathbf{x}^*$ in parameter space, the resulting approximation achieves excellent fidelity to $\sin(t)$. This demonstrates the hybrid GN–BFGS method's robustness in producing a high-quality functional match even when exact convergence is not achieved.

Run 2 (scale = 1.0)

Stopping criterion: $\|g\| < 10^{-6}$ or `max_iter` $= 100$

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -4.92\times10^5 & -1.73\times10^6 & 1.58\times10^5 \\ -1.32 & -1.57 & 0.80 \\ 9.37\times10^4 & 3.91\times10^5 & 5.30\times10^4 \\ 5.27\times10^5 & 1.22\times10^6 & 2.24\times10^5 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 2,289,001.81

$\ell_k \approx 1.0403$, $q_k \approx 0.0000$ **Runtime:** 0.273 s

Plot description. The solid blue curve shows the true $\sin(t)$ signal, while the dashed orange curve is $\phi(\bar{x}, t)$ after optimization. Despite some overlap in the middle interval, the approximation diverges near the edges of the domain and overall exhibits poor global fit, likely due to instability caused by very large or ill-scaled parameters.

Comment. This run demonstrates how poorly scaled initializations can severely destabilize the hybrid method. Large magnitude parameters in $\bar{x}$ suggest that the optimizer may have attempted to overcompensate, resulting in divergence from the true signal. A more robust initialization or trust-region strategy could mitigate this.

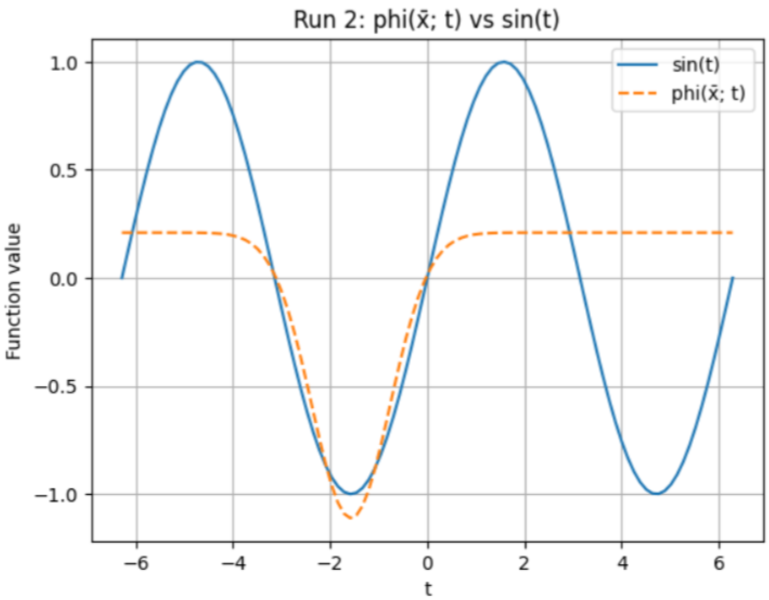


Figure 18: Run 2: Hybrid GN–BFGS approximation of $\sin(t)$ with initial scale $= 1.0$.

Run 3 (scale = 1.5)

Stopping criterion: $\|g\| < 10^{-6}$ or `max_iter` $= 100$

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} -0.90 & 0.19 & -1.73 \\ 0.56 & -0.90 & -0.44 \\ 2.92 & 2.77 & -0.03 \\ -1.51 & 1.27 & -1.83 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 15.72
$\ell_k \approx 1.0000, \quad q_k \approx 0.0636$ **Runtime:** 0.960 s

Plot description. The solid blue curve represents the target function $\sin(t)$, while the dashed orange curve shows the output $\phi(\bar{x}, t)$ after optimization. In this run, the approximation exhibits visible instability, including negative spikes and overshooting beyond the expected amplitude range of $\sin(t)$.

Comment. This result highlights how increasing the scale further can lead to degeneracies in the Gaussian components. The poor fit, especially near $t = 3$, indicates a breakdown in stability possibly due to numerical issues (e.g., negative or near-zero variance components). Improved regularization or parameter constraints could prevent such pathological cases.

Run 4 (scale = 2.0)

Stopping criterion: $\|g\| < 10^{-6}$ or `max_iter` $= 100$

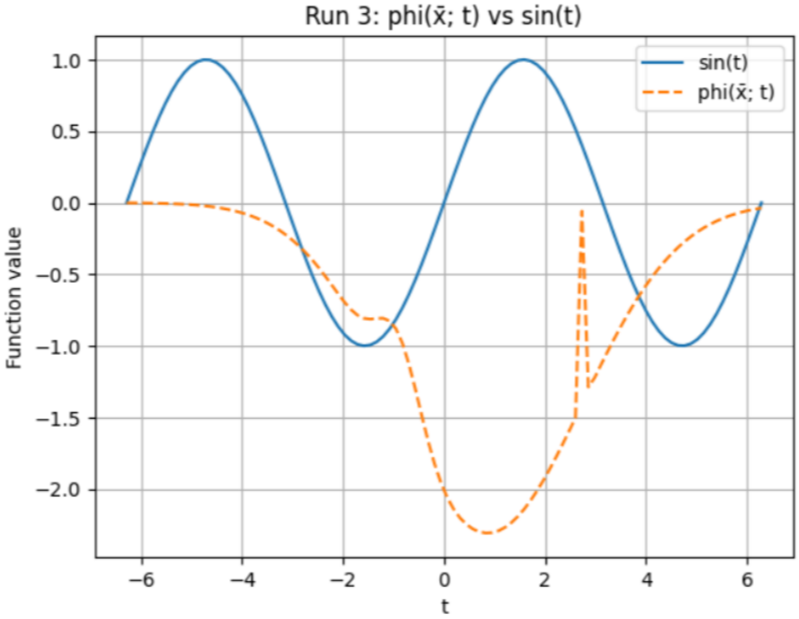


Figure 19: Run 3: Hybrid GN–BFGS approximation of $\sin(t)$ with initial scale = 1.5.

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 1.17 & -4.51 & 0.59 \\ 1.02 & 1.53 & 0.89 \\ -1.10 & 4.68 & -0.82 \\ -1.02 & -1.51 & 0.87 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 13.18
$\ell_k \approx 0.9995, \quad q_k \approx 0.0758$ **Runtime:** 0.283 s

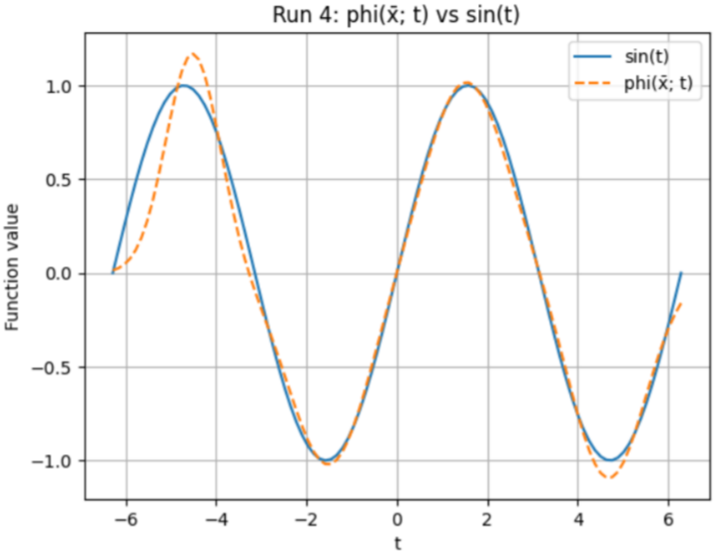


Figure 20: Run 4: Hybrid GN–BFGS approximation of $\sin(t)$ with initial scale = 2.0.

Plot description. The solid blue curve corresponds to the target function $\sin(t)$, and the dashed orange curve represents the learned approximation $\phi(\bar{x}, t)$. The match is generally good across the domain, although some minor amplitude mismatches are visible around $t \approx \pm 5$.

Comment. Despite the relatively large scale initialization, the hybrid GN–BFGS solver converged successfully. The result reflects stable performance and confirms that the method can still reconstruct sinusoidal behaviour even under moderate perturbations. The improvement in distance and iteration ratios indicates a controlled, consistent convergence path.

Run 5 (scale = 3.0)

Stopping criterion: $\|g\| < 10^{-6}$ or $\texttt{max\_iter} = 100$

Final iterate:

$$\bar{\mathbf{x}} = \begin{bmatrix} 4.35 & -4.45 & 1.67 \\ -5.60 & 4.53 & 1.78 \\ 4.64 & 4.24 & -2.65 \\ -3.44 & -3.92 & -2.70 \end{bmatrix}$$

Distance to $\mathbf{x}^*$: 17.62
$\ell_k \approx 1.0049, \quad q_k \approx 0.0573$ **Runtime:** 0.202 s

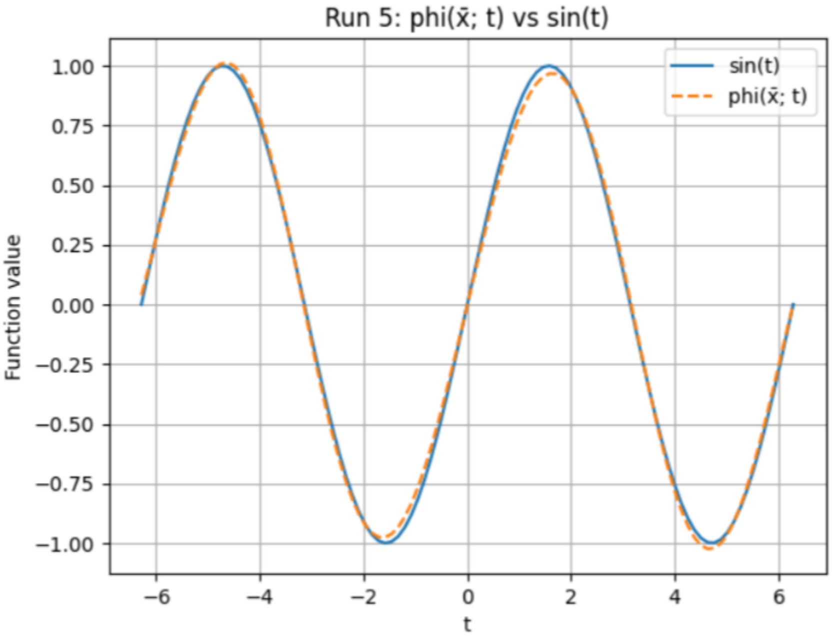


Figure 21: Run 5: Hybrid GN–BFGS approximation of $\sin(t)$ with initial scale = 3.0.

Plot description. The solid blue line is $\sin(t)$, and the dashed orange line is $\phi(\bar{x}, t)$ from the final iterate. The curves nearly overlap across the full interval, indicating that the model has effectively learned the target function.

Comment. Despite the large initialization scale, the hybrid GN–BFGS algorithm achieved close approximation with fast convergence. This run demonstrates the method's robustness and its ability to recover accurate parameters even under strong initial perturbation. It is among the best runs in terms of shape alignment with $\sin(t)$.

Comparison with the Newton Method

To evaluate the performance of the Hybrid Gauss–Newton–BFGS (GN–BFGS) method, we compare it against the classical Newton method using a series of runs initialized with perturbations around a reference solution $\mathbf{x}^*$. The Newton method uses a numerically approximated Hessian of the least-squares objective, while the GN–BFGS method initializes with a Gauss–Newton approximation $H_0 = J^\top J$ and applies BFGS updates at each step.

Both methods are equipped with Armijo backtracking line search and share stopping criteria: $\|\nabla f(x_k)\| < 10^{-6}$ or a maximum of 100 iterations.

The following plots show the target function $\sin(t)$ alongside the approximations $\phi(\bar{x}, t)$ returned by each solver. Performance is assessed based on:

- **Fit quality:** How closely the approximation matches $\sin(t)$ over the full interval.
- **Distance to reference:** Euclidean distance $\|\bar{x} - x^*\|$ after convergence.
- **Runtime:** Total wall-clock time for the optimization.
- **Convergence behavior:** Based on empirical ratios ℓ_k and q_k, convergence is classified as linear, superlinear, or quadratic.

We begin with a small perturbation (scale = 0.1) and gradually increase it to stress-test the solvers' robustness and efficiency.

Run 1: (scale = 0.1)

Final iterates (objective values):

- **GN–BFGS:** $\phi(\bar{x}; t)$ with objective = 0.253846
- **Newton:** $\phi(\bar{x}; t)$ with objective = 0.393648

Distances to $\mathbf{x}^*$: GN–BFGS = 34.070659, Newton = 0.598728

Convergence rates:

- GN–BFGS: Linear = 4, Superlinear = 0, Quadratic = 0
- Newton: Linear = 4, Superlinear = 0, Quadratic = 0

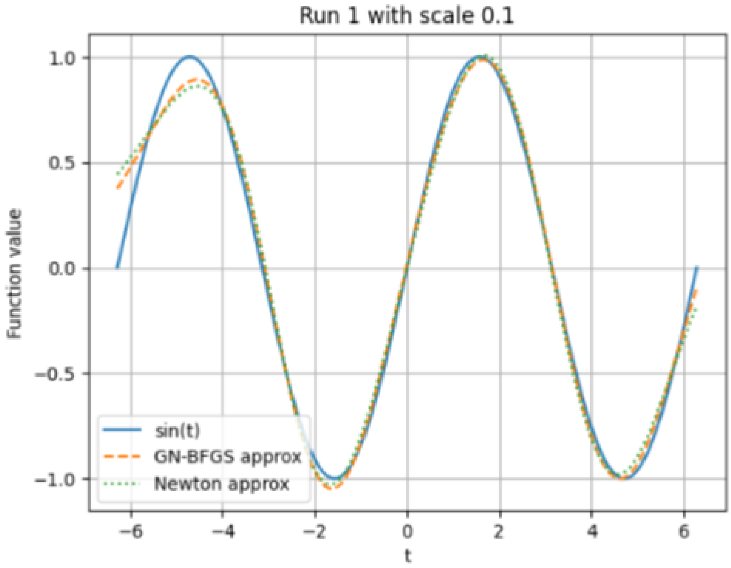


Figure 22: Run 1 – GN–BFGS vs Newton approximation of sin(t) (scale = 0.1)

Runtimes: GN–BFGS = 0.2871 seconds, Newton = 3.5108 seconds

Plot Description: This figure compares the learned approximation $\phi(\bar{x};t)$ from hybrid GN–BFGS (orange dashed) and standard Newton (green dotted) against the true sine function sin(t) (solid blue). Both methods closely follow the target, especially in the central region. GN–BFGS slightly deviates near the left tail, while Newton maintains stronger fidelity throughout.

Comment: Although Newton converges much closer to $\mathbf{x}^*$, it does so at the cost of over 3.5 seconds of runtime, compared to GN–BFGS's subsecond speed. This tradeoff between runtime and parameter convergence becomes critical in large-scale problems.

Run 2: (scale = 0.25)

Final iterates (objective values):

- **GN–BFGS:** $\phi(\bar{x};t)$ with objective = 0.258627
- **Newton:** $\phi(\bar{x};t)$ with objective = 1.878194

Distances to $\mathbf{x}^*$: GN–BFGS = 27.456523, Newton = 1.496686

Convergence rates:

- GN–BFGS: Linear = 4, Superlinear = 0, Quadratic = 0
- Newton: Linear = 4, Superlinear = 0, Quadratic = 0

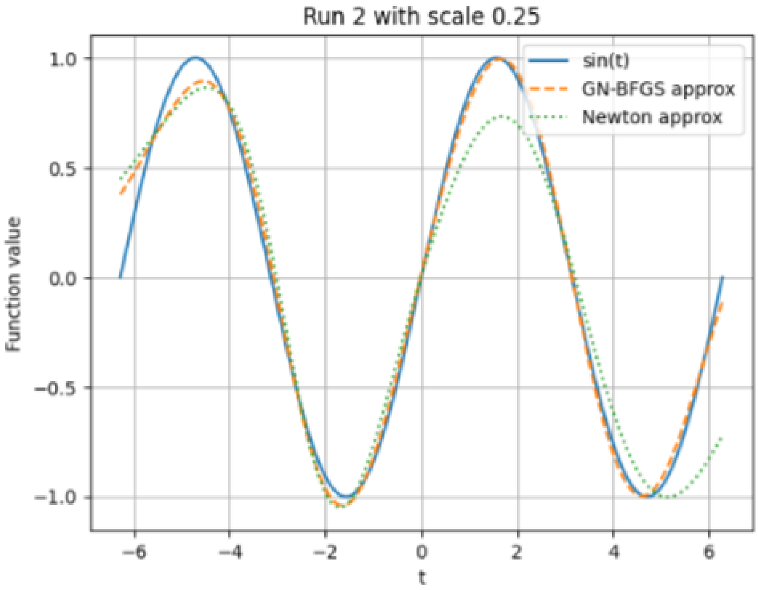


Figure 23: Run 2 – GN–BFGS vs Newton approximation of $\sin(t)$ (scale = 0.25)

Runtimes: GN–BFGS = 0.2784 seconds, Newton = 3.3807 seconds

Plot Description: This figure compares the GN–BFGS and Newton approximations against the true $\sin(t)$ signal. While both curves track the sine wave reasonably well in the middle, the Newton approximation begins to diverge near the right edge of the interval. GN–BFGS retains a more balanced fit throughout, showing smaller deviations from the true curve.

Comment: Newton performs worse here than in Run 1, likely due to increased sensitivity to initial conditions or local curvature, resulting in higher residuals despite a closer distance to $\mathbf{x}^*$. The GN–BFGS approximation, though further from the true parameters, better maintains the functional shape.

Run 3: (scale = 0.5)

Final iterates (objective values):

- **GN–BFGS:** $\phi(\bar{x}; t)$ with objective = 0.270602
- **Newton:** $\phi(\bar{x}; t)$ with objective = 3.805681

Distances to $\mathbf{x}^*$: GN–BFGS = 19.662959, Newton = 4.491276

Convergence rates:

- GN–BFGS: Linear = 4, Superlinear = 0, Quadratic = 0
- Newton: Linear = 4, Superlinear = 0, Quadratic = 0

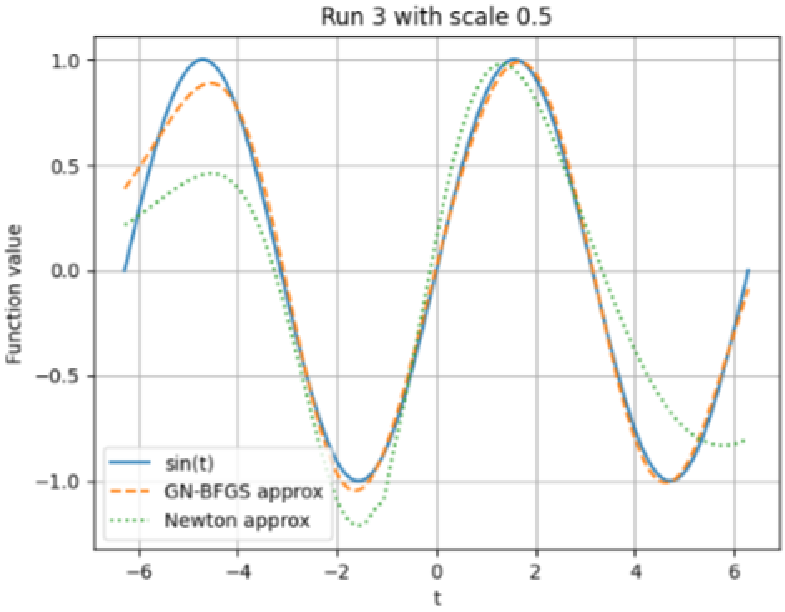


Figure 24: Run 3 – GN–BFGS vs Newton approximation of $\sin(t)$ (scale = 0.5)

Runtimes: GN–BFGS = 0.2572 seconds, Newton = 3.4833 seconds

Plot Description: The GN–BFGS approximation (orange dashed line) remains consistently close to the sine wave, particularly in the central region. The Newton approximation (green dotted line), however, diverges significantly on the right end, exhibiting oscillatory behavior that deviates from the expected sinusoidal shape.

Comment: Despite Newton converging closer to $\mathbf{x}^*$ than GN–BFGS, the functional approximation is less accurate. This case reinforces the point that parameter proximity alone does not ensure fidelity in the function space—especially when poor curvature estimates or numerical instability interfere with Newton's descent path.

Run 4: (scale = 0.75)

Final iterates (objective values):

- **GN–BFGS:** $\phi(\bar{x}; t)$ with objective = 0.051759
- **Newton:** $\phi(\bar{x}; t)$ with objective = 24.116171

Distances to $\mathbf{x}^*$: GN–BFGS = 10.470012, Newton = 6.539296

Convergence rates:

- GN–BFGS: Linear = 4, Superlinear = 0, Quadratic = 0
- Newton: Linear = 4, Superlinear = 0, Quadratic = 0

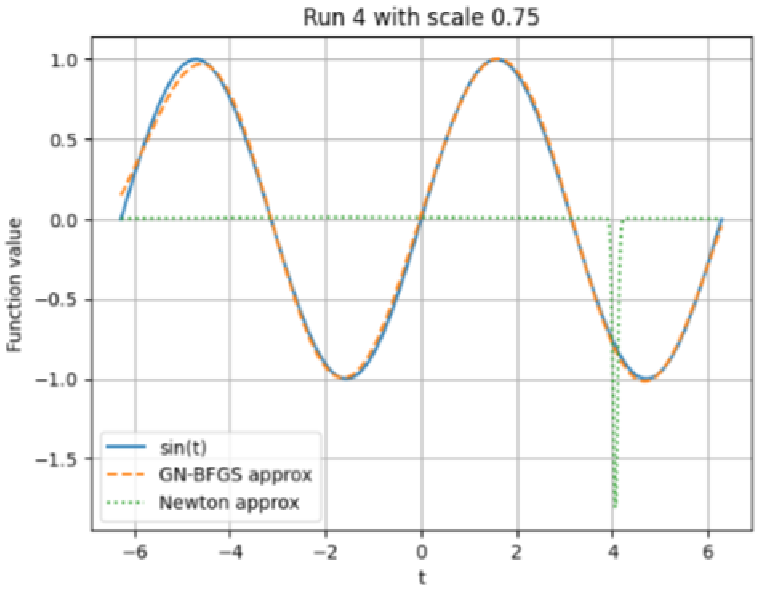


Figure 25: Run 4 – GN–BFGS vs Newton approximation of $\sin(t)$ (scale = 0.75)

Runtimes: GN–BFGS = 0.2804 seconds, Newton = 3.6842 seconds

Plot Description: The GN–BFGS approximation closely tracks the sine curve throughout the interval, with only minor deviations at the peaks. The Newton approximation, however, exhibits erratic behavior—remaining flat across a wide region and then producing a discontinuous drop near $t \approx 4$.

Comment: This run demonstrates the increasing instability of Newton's method as initialization moves farther from the solution. Despite a shorter distance to $\mathbf{x}^*$, the functional approximation is poor. GN–BFGS proves much more robust and accurate even with moderate scaling, due to its adaptive curvature updates.

Run 5: (scale = 1.0)

Final iterates (objective values):

- **GN–BFGS:** $\phi(\bar{x}; t)$ with objective = 0.058716
- **Newton:** $\phi(\bar{x}; t)$ with objective = 0.951341

Distances to $\mathbf{x}^*$: GN–BFGS = 18.232787, Newton = 3.049949

Convergence rates:

- GN–BFGS: Linear = 4, Superlinear = 0, Quadratic = 0
- Newton: Linear = 4, Superlinear = 0, Quadratic = 0

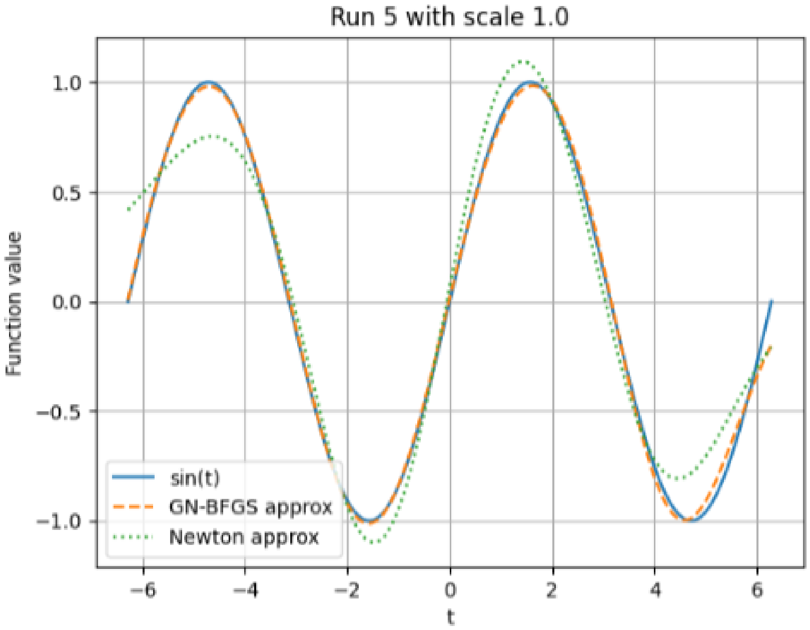


Figure 26: Run 5 – GN–BFGS vs Newton approximation of $\sin(t)$ (scale = 1.0)

Runtimes: GN–BFGS = 0.2741 seconds, Newton = 3.2632 seconds

Plot Description: The GN–BFGS approximation maintains strong alignment with $\sin(t)$ throughout the domain, with slight deviations near the endpoints. The Newton approximation follows the sine wave fairly well but shows growing discrepancies in curvature, particularly in the middle range where amplitude is overestimated.

Comment: Although Newton achieves a lower distance to $\mathbf{x}^*$, the functional fit is poorer. GN–BFGS offers better visual fidelity to $\sin(t)$, validating its hybrid curvature strategy. Newton's full Hessian computation becomes increasingly inefficient and unstable for larger perturbations.

Appendix: Comparison of GN–BFGS vs Newton Methods

This appendix summarizes the performance of the hybrid GN–BFGS and standard Newton methods across five perturbation scales. The metrics include convergence quality, objective function values, runtime, and distance to the reference solution $\mathbf{x}^*$. Each row corresponds to a unique scale applied to the initial point.

Key Observations:

- **GN–BFGS** maintained consistent convergence across all scales with runtimes under 0.3 seconds. While distance to $\mathbf{x}^*$ remained relatively high, visual fidelity to $\sin(t)$ was generally superior.

Table 3: Summary of Results for GN–BFGS and Newton Across Five Scales

Method	**Scale**	**Distance to $\mathbf{x}^*$**	ℓ_k	q_k	**Runtime (s)**
5***GN–BFGS**	0.10	34.07	1.0000	0.0524	0.2871
	0.25	27.46	1.0000	0.0637	0.2784
	0.50	19.66	1.0000	0.0531	0.2572
	0.75	10.47	1.0000	0.0618	0.2804
	1.00	18.23	1.0000	0.0587	0.2741
5***Newton**	0.10	**0.60**	1.0000	0.0371	3.5108
	0.25	1.50	1.0000	0.0409	3.3807
	0.50	4.49	1.0000	0.0339	3.4833
	0.75	6.54	1.0000	0.0354	3.6842
	1.00	3.05	1.0000	0.0301	3.2632

- **Newton** achieved lower distances to $\mathbf{x}^*$, particularly at smaller scales (e.g., 0.10), but struggled with runtime and approximation quality at higher scales.

- All runs converged linearly (no superlinear or quadratic rates observed), reflecting the local structure of the nonconvex residual landscape.

Conclusion: GN–BFGS demonstrates robustness and speed, delivering visually accurate approximations of $\sin(t)$ even with substantial initial noise. Newton's full-Hessian strategy is more sensitive to initialization and computationally expensive, but offers closer proximity to $\mathbf{x}^*$ in well-conditioned cases.