AM230 Course Project

Numerical Optimization Methods

Luke Catalano
UC Santa Cruz, M.S. Quantitative Economics

January 2026

Overview

This report studies numerical optimization methods for logistic regression, emphasizing how curvature and conditioning determine algorithmic convergence. We (i) derive gradient and Hessian expressions, (ii) establish convexity and strong convexity under ℓ_2 regularization, and (iii) compare fixed-step gradient descent, line-search variants, and Newton-type methods through controlled computational experiments.

Notation

Observations are (x^i, y^i) for $i = 1, \dots, N$ with $y^i \in \{0, 1\}$. The parameter is $\theta = [w^\top, \underline{\rbrack}^\top \in \mathbb{R}^3$ with score

$$s_\theta(x^i) = w^\top x^i + \underline{.}$$

Define the logistic function $\sigma(z) = (1 + e^{-z})^{-1}$. The per-sample logistic loss is

$$\ell^i(\theta) = \log\bigl(1 + e^{s_\theta(x^i)}\bigr) - y^i s_\theta(x^i),$$

and the empirical risk is

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell^i(\theta).$$

1. Analytical Properties of the Logistic Loss

1.1 Gradient of the per-sample loss

Let $s_\theta(x^i) = w_1 x_1^i + w_2 x_2^i + b$. Differentiating

$$\ell^i(\theta) = \log(1 + e^{s_\theta(x^i)}) - y^i s_\theta(x^i)$$

with respect to w_1, w_2, b and using the chain rule yields

$$\frac{\partial \ell^i}{\partial w_1} = x_1^i \left(\frac{e^{s_\theta(x^i)}}{1 + e^{s_\theta(x^i)}} - y^i \right), \quad \frac{\partial \ell^i}{\partial w_2} = x_2^i \left(\frac{e^{s_\theta(x^i)}}{1 + e^{s_\theta(x^i)}} - y^i \right), \quad \frac{\partial \ell^i}{\partial b} = \frac{e^{s_\theta(x^i)}}{1 + e^{s_\theta(x^i)}} - y^i.$$

Define $\sigma(s_\theta(x^i)) = \sigma(s_\theta(x^i)) = \frac{e^{s_\theta(x^i)}}{1+e^{s_\theta(x^i)}}$. Then

$$\nabla_\theta \ell^i(\theta) = \begin{bmatrix} x_1^i(\sigma(s_\theta(x^i)) - y^i) \\ x_2^i(\sigma(s_\theta(x^i)) - y^i) \\ (\sigma(s_\theta(x^i)) - y^i) \end{bmatrix}.$$

1.2 Gradient of the empirical risk

By linearity of differentiation,

$$\nabla_\theta L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \ell^i(\theta) = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} x_1^i(\sigma(s_\theta(x^i)) - y^i) \\ \frac{1}{N}\sum_{i=1}^{N} x_2^i(\sigma(s_\theta(x^i)) - y^i) \\ \frac{1}{N}\sum_{i=1}^{N}(\sigma(s_\theta(x^i)) - y^i) \end{bmatrix}.$$

1.3 Hessian factorization and positive semidefiniteness

The derivative of the logistic function is

$$\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big).$$

Let $x_i := [x_1^i,\, x_2^i,\, 1]^\top \in \mathbb{R}^3$. A standard computation shows the per-sample Hessian admits the rank-one factorization

$$\nabla_\theta^2 \ell^i(\theta) = \sigma'(s_\theta(x^i))\, x_i x_i^\top = \sigma(s_\theta(x^i))\big(1 - \sigma(s_\theta(x^i))\big)\, x_i x_i^\top.$$

Proposition 1 (Per-sample Hessian is PSD). *For every i and every $\theta \in \mathbb{R}^3$, the matrix $\nabla_\theta^2 \ell^i(\theta)$ is positive semidefinite.*

Proof. For any $v \in \mathbb{R}^3$,

$$v^\top \nabla_\theta^2 \ell^i(\theta)\, v = \sigma'(s_\theta(x^i))\, v^\top x_i x_i^\top v = \sigma'(s_\theta(x^i))\,(x_i^\top v)^2 \geq 0,$$

since $\sigma'(s_\theta(x^i)) \geq 0$ and $(x_i^\top v)^2 \geq 0$. □

Proposition 2 (Convexity of the empirical risk). *The empirical risk $L(\theta)$ is convex on $\mathbb{R}^3$.*

Proof. $\nabla_\theta^2 L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta^2 \ell^i(\theta)$ is an average of PSD matrices, hence PSD for all θ. Therefore L is convex. □

Proposition 3 (Strict convexity under full rank). *If the augmented feature matrix*

$$X = \begin{bmatrix} x_1^1 & x_2^1 & 1 \\ x_1^2 & x_2^2 & 1 \\ \vdots & \vdots & \vdots \\ x_1^N & x_2^N & 1 \end{bmatrix} \in \mathbb{R}^{N\times 3}$$

has full column rank, then $L(\theta)$ is strictly convex (and thus admits at most one minimizer).

Proof. Using the factorization,

$$\nabla_\theta^2 L(\theta) = \frac{1}{N} X^\top D X,$$

where D is diagonal with $D_{ii} = \sigma'(s_\theta(x^i)) > 0$ for all finite θ. Thus D is positive definite. If X has full column rank, then $Xv \neq 0$ for all $v \neq 0$. Hence

$$v^\top X^\top D X v = (Xv)^\top D (Xv) > 0 \quad \text{for all } v \neq 0,$$

so $\nabla_\theta^2 L(\theta)$ is positive definite and L is strictly convex. □

1.4 Positivity and separability

Lemma 1 (Strict positivity for finite parameters)**.** *For any finite θ, $\ell^i(\theta) > 0$ for every i, hence $L(\theta) > 0$.*

Proof. If $y^i = 0$, then $\ell^i(\theta) = \log(1 + e^{s_\theta(x^i)}) > 0$. If $y^i = 1$, then

$$\ell^i(\theta) = \log(1 + e^{s_\theta(x^i)}) - s_\theta(x^i) = \log(1 + e^{-s_\theta(x^i)}) > 0.$$

Averaging preserves strict positivity. □

Linear separability implies infimum 0 is not attained

Assume the data are linearly separable, so there exists $\tilde{\theta}$ such that $\tilde{s}(x^i) = s_{\tilde{\theta}}(x^i)$ satisfies $\tilde{s}(x^i) > 0$ when $y^i = 1$ and $\tilde{s}(x^i) < 0$ when $y^i = 0$. Consider $t\tilde{\theta}$ with $t \to \infty$:

$$L(t\tilde{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \Big[\log\big(1 + e^{t\tilde{s}(x^i)}\big) - y^i \, t\tilde{s}(x^i)\Big].$$

If $y^i = 1$, then the summand becomes $\log(1 + e^{-t\tilde{s}(x^i)}) \to 0$. If $y^i = 0$, then $\log(1 + e^{t\tilde{s}(x^i)}) \to 0$ since $t\tilde{s}(x^i) \to -\infty$. Hence

$$\lim_{t\to\infty} L(t\tilde{\theta}) = 0.$$

Combined with the lemma above ($L(\theta) > 0$ for finite θ), the infimum of L is 0 but is not attained at any finite θ; thus no minimizer exists in the separable case.

2. Gradient Descent with Fixed Step Size

2.1 Implementation

Gradient descent updates

$$\theta_{k+1} = \theta_k - \alpha \nabla L(\theta_k),$$

with fixed $\alpha =$ `opts.alpha_fixed`. The descent direction is $p_k = -\nabla L(\theta_k)$, and $\theta_{k+1} = \theta_k + \alpha p_k$.

2.2 Experiment: fixed step size, $\alpha = 1$

Initialize $\theta_0 = [0, 0, 0]^\top$ and iterate for $k = 500$.

Results (500 iterations)

After 500 iterations:

$$[w_1, w_2, b] = [4.1799,\ 9.1322,\ -8.0352].$$

Training accuracy was 96.69% (1 misclassification).

Long-run behavior (10,000 iterations)

With $k = 10{,}000$, the loss decreases toward 0 and $\|\nabla L(\theta_k)\|_2$ decreases, while $\|\theta_k\|_2$ grows without stabilizing. This behavior is consistent with the separable-data result: the empirical risk can be driven arbitrarily close to 0 along rays $t\tilde{\theta}$, but no finite minimizer exists.

3. ℓ_2 Regularization: Strong Convexity and Conditioning

A standard remedy for separability is quadratic regularization:

$$\bar{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\Big(\log(1+e^{s_\theta(x^i)}) - y^i s_\theta(x^i)\Big) + \mu\|\theta\|_2^2, \quad \mu > 0.$$

3.1 Gradient and Hessian

Using Section 1 and differentiating the penalty:

$$\nabla \bar{L}(\theta) = \nabla L(\theta) + 2\mu\theta,$$

$$\nabla^2 \bar{L}(\theta) = \nabla^2 L(\theta) + 2\mu I = \frac{1}{N}\sum_{i=1}^{N} \sigma'(s_\theta(x^i))\, x_i x_i^\top + 2\mu I.$$

Proposition 4 (Strong convexity). *$\bar{L}$ is 2μ-strongly convex, hence admits a unique global minimizer.*

Proof. The first term of $\nabla^2 \bar{L}$ is PSD. Therefore $\nabla^2 \bar{L}(\theta) \succeq 2\mu I$ for all θ, which is exactly 2μ-strong convexity. Strong convexity implies a unique minimizer. □

3.2 Experiment: fixed-step GD on $\bar{L}$

With $\theta_0 = 0$ and $\alpha = 1$, compare $\mu \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

- $\mu = 10^{-2}$: 66.67% training accuracy; 10 misclassified samples. $\|\theta_k\|_2$ stabilizes.
- $\mu = 10^{-3}$: 90.00% training accuracy; 3 misclassified samples. $\|\theta_k\|_2$ stabilizes.
- $\mu = 10^{-4}$: 96.67% training accuracy; 1 misclassified sample. $\|\theta_k\|_2$ stabilizes.

Regularization prevents parameter blow-up and ensures a finite minimizer, at the cost of introducing bias that may reduce training accuracy.

4. Step Size Stability and Optimal Linear Rate (Empirical Hessian)

Throughout this section, set $\mu = 10^{-4}$.

4.1 Fixed step size $\alpha = 10$

After an initial transient, the semilog plot of $\|\nabla \bar{L}(\theta_k)\|_2$ is approximately linear, indicating linear convergence.

4.2 Eigenvalues at the numerical minimizer

At the numerical minimizer $\theta^\star$, the Hessian eigenvalues were approximately

$$(\lambda_{\min}, \ldots, \lambda_{\max}) \approx (4 \times 10^{-4},\ 3.6 \times 10^{-3},\ 1.249 \times 10^{-1}).$$

For (local) strongly convex objectives, a standard linearized rate heuristic for fixed-step GD is

$$r(\alpha) = \max_i |1 - \alpha\lambda_i|.$$

With $\alpha = 10$, this suggests $r \approx 0.9956$, consistent with observed slow but linear decay.

4.3 Stability threshold

A common stability condition is $0 < \alpha < 2/\lambda_{\max}$, giving

$$\alpha_{\text{crit}} \approx \frac{2}{0.1249} \approx 16.01.$$

Empirically:

Step-size stability test (predicted $\alpha_{\text{crit}} \approx 16.01$)

α	f_{end}	$\|\nabla \bar{L}\|_{\text{end}}$	Status
15.00	1.667×10^{-1}	9.955×10^{-9}	Converged
15.50	1.667×10^{-1}	9.967×10^{-9}	Converged
16.00	1.667×10^{-1}	9.998×10^{-9}	Converged
16.50	1.675×10^{-1}	1.393×10^{-2}	Not converged
17.00	1.711×10^{-1}	3.082×10^{-2}	Not converged

4.4 Best fixed-step rate for a quadratic model

For a quadratic model with spectrum in $[\lambda_{\min}, \lambda_{\max}]$, the optimal constant step is

$$\alpha^{\star} = \frac{2}{\lambda_{\min} + \lambda_{\max}}, \qquad r^{\star} = \frac{\kappa - 1}{\kappa + 1}, \quad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

Using $\lambda_{\min} \approx 0.0004$ and $\lambda_{\max} \approx 0.1249$:

$$\alpha^{\star} \approx \frac{2}{0.1253} \approx 15.96, \qquad \kappa \approx 312.25, \qquad r^{\star} \approx 0.9936.$$

Empirically, the fastest convergence occurred near $\alpha \approx 16$, consistent with this heuristic.

5. Variable Step Sizes: Line Search

5.1 Implementation interface

The gradient descent solver uses:

- fixed step size when `opts.useLineSearch = false` with $\alpha =$ `opts.alpha_fixed`;
- line search when `opts.useLineSearch = true`, calling `steplength` with `opts.ls`.

Default strong Wolfe line search parameters:

```
opts.ls = struct( ...
  'c1', 1e-4, ...
  'c2', 0.9, ...
  'alpha0', 1.0, ...
  'alpha_max', 100, ...
  'alpha_min', 1e-6, ...
  'maxIter', 50, ...
  'maxZoom', 50 );
```

5.2 Strong Wolfe line search (default parameters)

With $\mu = 10^{-4}$ and $\theta_0 = 0$, the gradient norm decreases monotonically but can be slower per iteration than the tuned fixed-step method due to conservative accepted step sizes and per-iteration overhead (multiple function/gradient evaluations).

5.3 Approximate "exact" line search

Using $c_1 = 10^{-8}$ and $c_2 = 10^{-7}$, the gradient norm exhibits stable decay; after an initial transient, the semilog plot becomes approximately linear. Empirically, selected step sizes clustered near a near-optimal constant step (e.g., $\alpha \approx 8.4$ in one run), yielding convergence behavior close to tuned fixed-step GD but with additional per-iteration cost.

5.4 Comparison: fixed vs Wolfe vs approximate exact

- **Optimal fixed step ($\alpha^\star$):** cheapest per iteration; fastest linear decrease once calibrated.
- **Strong Wolfe:** robust step selection; higher per-iteration cost; sometimes more conservative steps.
- **Approximate "exact":** can approach calibrated fixed-step performance when it chooses near-optimal steps; still incurs line-search overhead.

6. Conditioning vs Regularization Strength μ

6.1 Strong Wolfe GD across μ

For $\mu \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, gradient descent with line search yields the following Hessian spectrum at $\theta^\star(\mu)$:

μ	$\lambda_{\min}$	$\lambda_{\max}$	$\kappa(\mu) = \lambda_{\max}/\lambda_{\min}$	Iterations
10^{-6}	0.0013366	0.22913	171.42	5000
10^{-5}	0.0013522	0.22974	169.90	5000
10^{-4}	0.0015073	0.23576	156.41	5000
10^{-3}	0.0029069	0.28403	97.71	5000
10^{-2}	0.0125460	0.38795	30.923	5000

As μ increases, $\lambda_{\min}$ increases noticeably while $\lambda_{\max}$ changes more moderately, substantially improving conditioning. Improved conditioning accelerates first-order methods; the trade-off is increased regularization bias.

6.2 Optimal fixed-step GD across μ

When using $\alpha^\star(\mu)$ computed from local curvature, the convergence curves across μ become more similar, consistent with the fact that an optimal step size adapts to curvature and partially offsets poor conditioning.

7. Newton and Damped Newton

7.1 Implementation

A Newton solver (`solve_newton.m`) was implemented supporting:

- pure Newton updates ($\alpha = 1$),
- damped Newton with strong Wolfe line search.

7.2 Pure Newton

From some initializations Newton converges rapidly; from others it can behave poorly, illustrating that pure Newton has excellent local convergence but does not guarantee robust global behavior without safeguarding.

7.3 Damped Newton with strong Wolfe line search

The Newton direction is

$$p_k = -\left(\nabla^2 \bar{L}(\theta_k)\right)^{-1} \nabla \bar{L}(\theta_k).$$

Since $\bar{L}$ is strongly convex, $\nabla^2 \bar{L}(\theta_k)$ is positive definite. Let $g_k = \nabla \bar{L}(\theta_k)$ and $H_k = \nabla^2 \bar{L}(\theta_k)$. Then

$$g_k^\top p_k = -g_k^\top H_k^{-1} g_k < 0 \quad \text{whenever } g_k \neq 0,$$

so p_k is a descent direction. A strong Wolfe line search provides a stable globalization mechanism, yielding reliable convergence from all tested initializations.

7.4 Sensitivity to conditioning

Newton's method is substantially less sensitive to conditioning than gradient descent: changing μ strongly affects first-order convergence, while Newton iterations remain relatively stable across μ, reflecting curvature-correcting updates.

Appendix: Figures

Figures referenced in the text are included below as available. (If compiling locally, ensure the image filenames match exactly.)

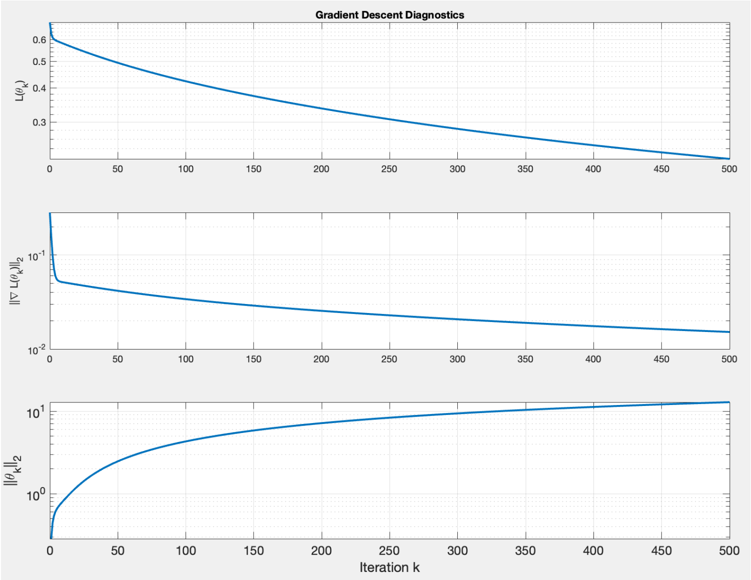

Figure 1: Gradient descent implementation output (fixed step).

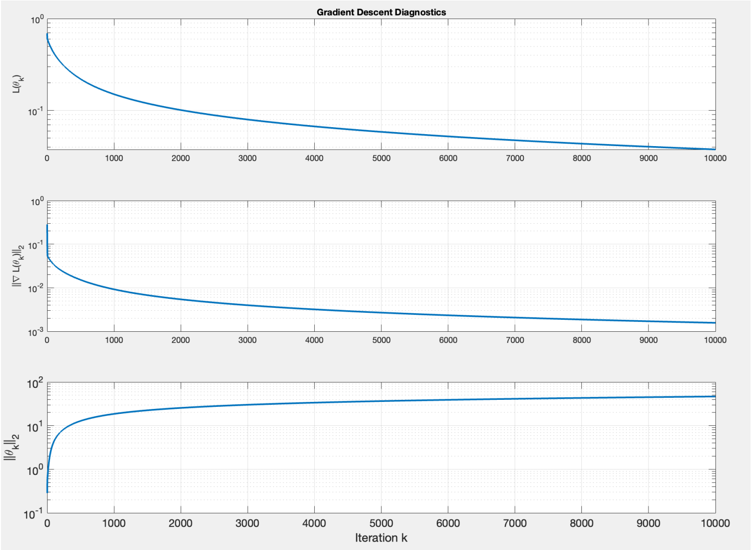

Figure 2: Loss, gradient norm, and parameter norm vs iterations (log scale).

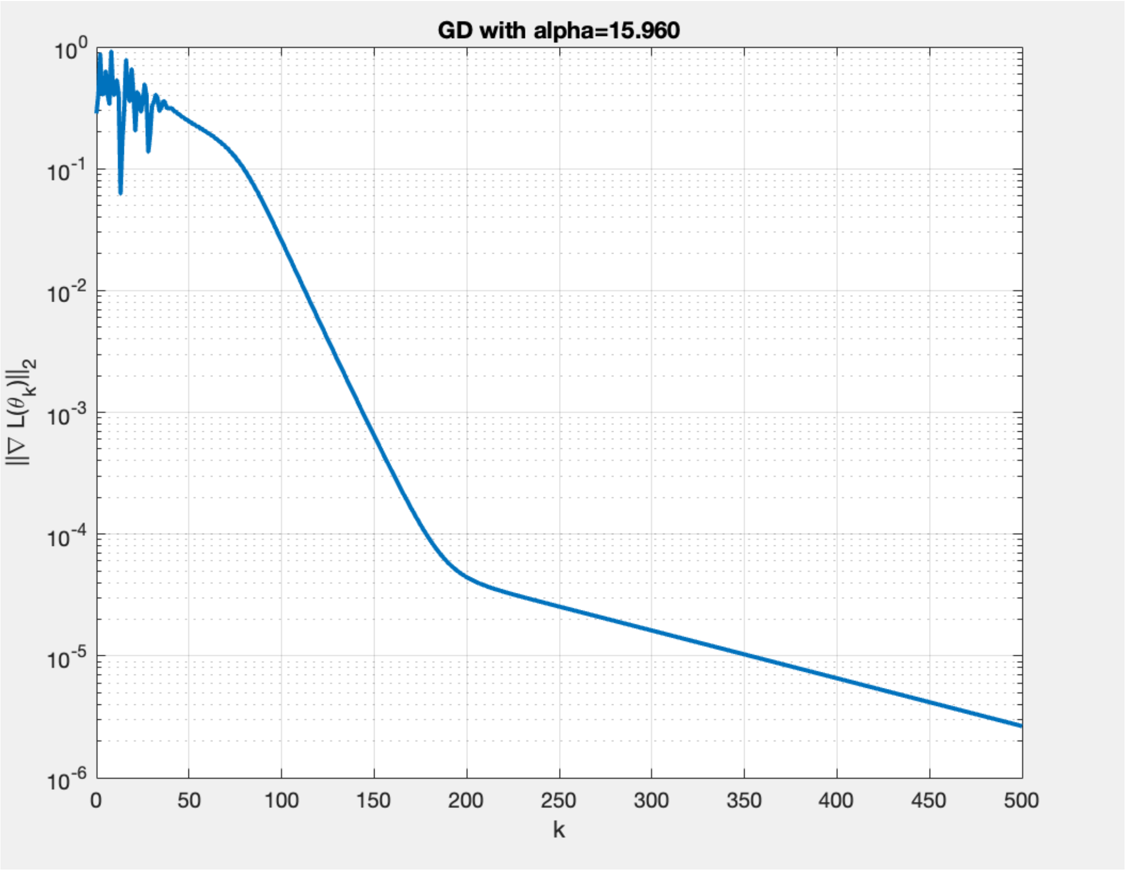

Figure 3: Empirical verification near the optimal constant step size.

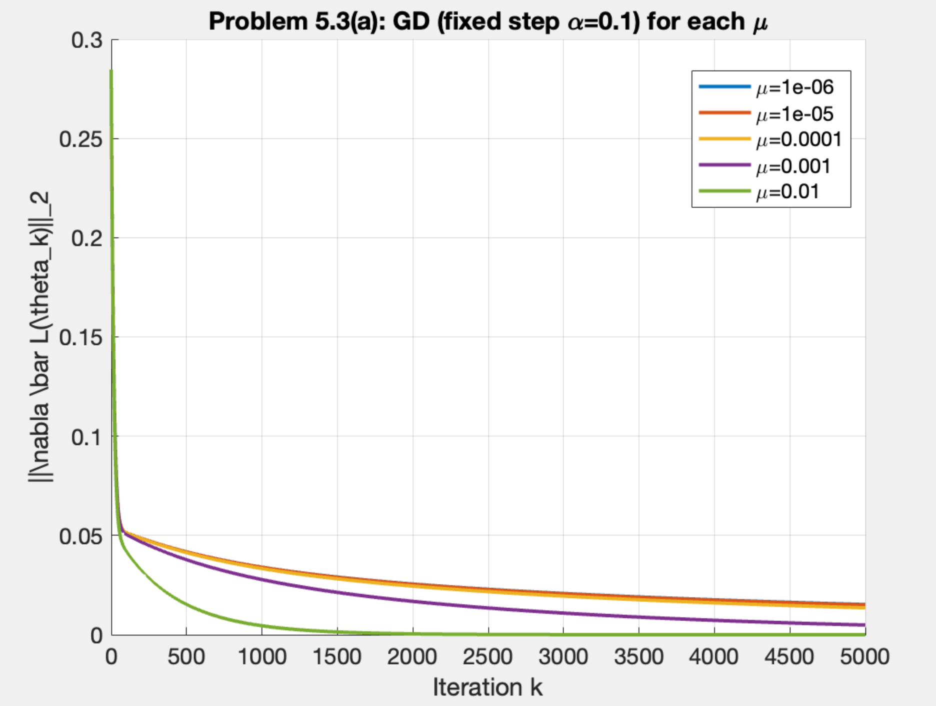

Figure 4: Gradient norms (strong Wolfe) across values of μ.

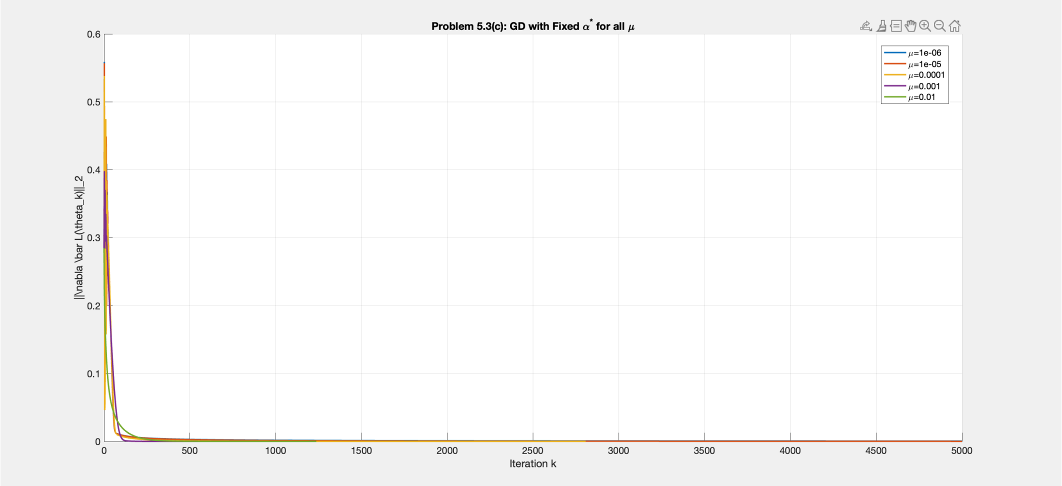

Figure 5: Gradient norms (optimal constant step) across values of μ.

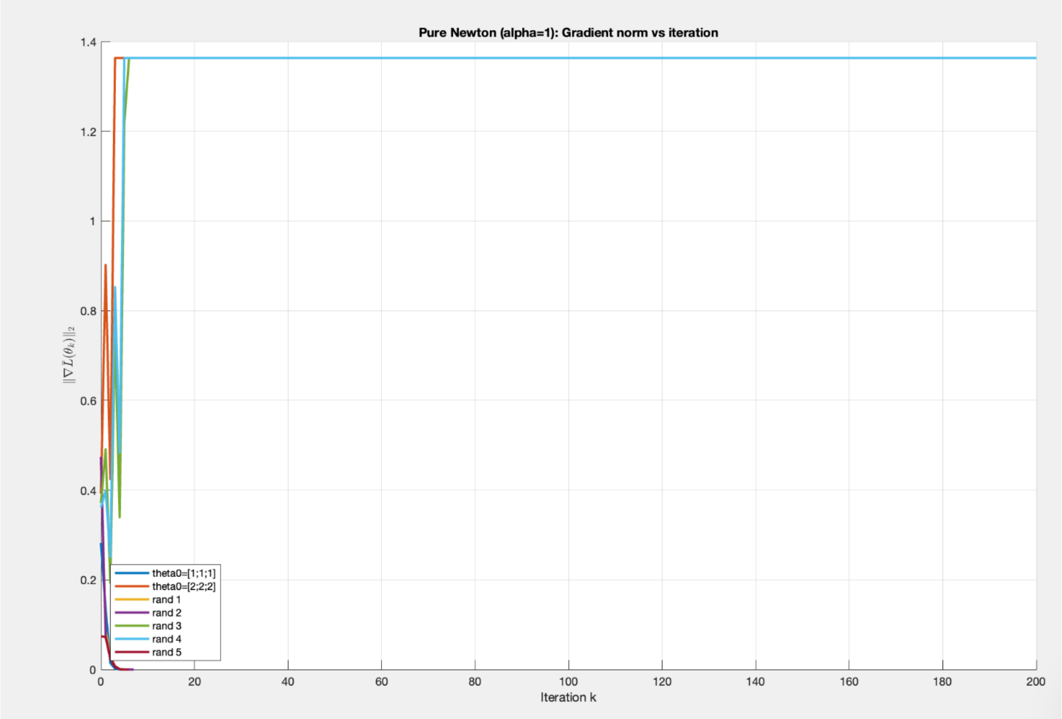

Figure 6: Pure Newton ($\alpha = 1$): gradient norm vs iteration.

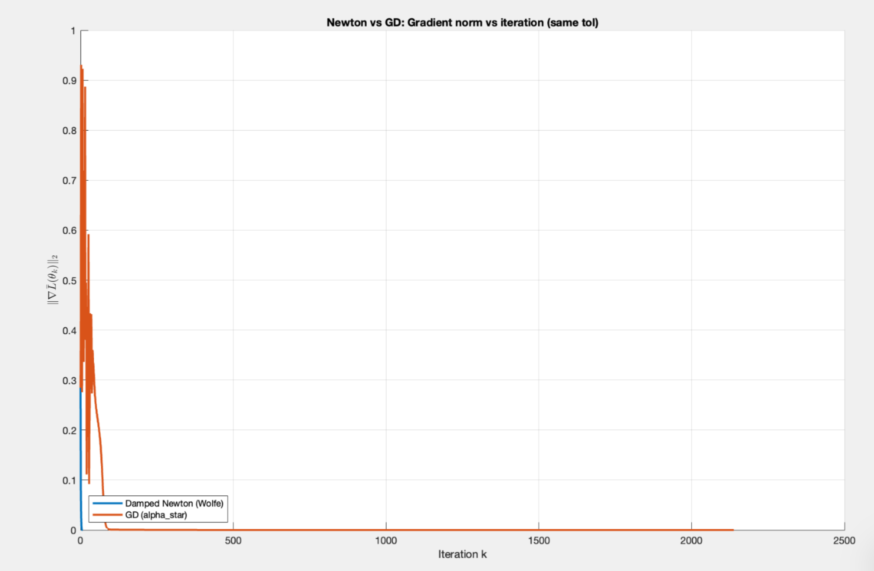

Figure 7: Newton vs Gradient Descent (same tolerance).

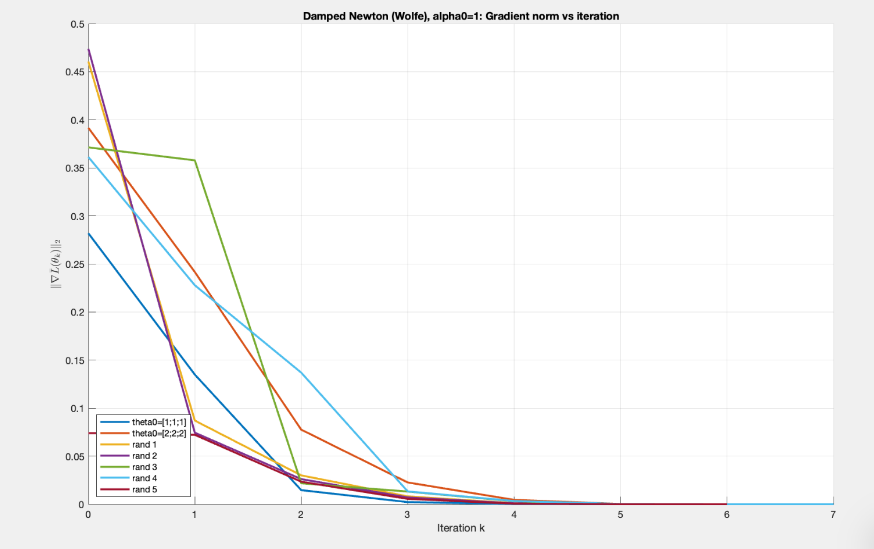

Figure 8: Damped Newton (Wolfe), $\alpha_0 = 1$.

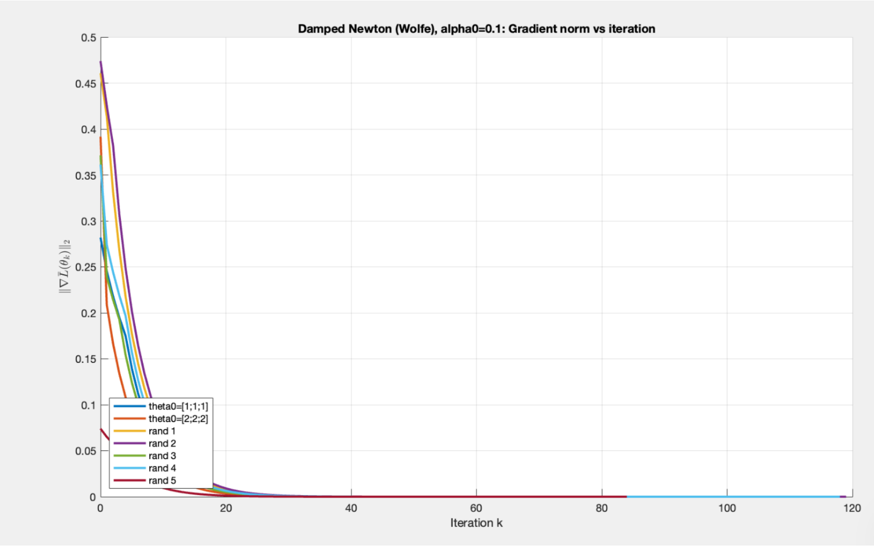

Figure 9: Damped Newton (Wolfe), $\alpha_0 = 0.1$.