Gradient and Hessian of Selected Functions

Naitik Kariwal

16-03-2025

1 Trid Function

The Trid function is defined as:

$$f(\mathbf{x}) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1} \tag{1}$$

1.1 Gradient

$$\frac{\partial f}{\partial x_i} = \begin{cases} 2(x_1 - 1) - x_2, & i = 1 \\ 2(x_i - 1) - x_{i-1} - x_{i+1}, & 1 < i < n \\ 2(x_n - 1) - x_{n-1}, & i = n \end{cases} \tag{2}$$

1.2 Hessian

The Hessian matrix H is tridiagonal:

$$H_{i,j} = \begin{cases} 2, & i = j \\ -1, & |i - j| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

1.3 Finding the Minima

To find the minima, we solve $\nabla f = 0$:

Step 1: Solve for x_2

$$2(x_1 - 1) - x_2 = 0 \tag{4}$$

$$\implies x_2 = 2(x_1 - 1) \tag{5}$$

Step 2: General Recurrence Relation

$$2(x_i - 1) - x_{i-1} - x_{i+1} = 0, \quad 2 \leq i \leq n - 1 \tag{6}$$

$$\implies x_{i+1} = 2(x_i - 1) - x_{i-1} \tag{7}$$

Step 3: Identify the Pattern

Upon recursively substituting values in terms of x_1, we get a pattern:

$$x_i = ix_1 - i(i - 1) \tag{8}$$

Step 4: Solve for x_1

$$2(x_n - 1) - x_{n-1} = 0 \tag{9}$$

$$\implies 2(nx_1 - n(n - 1) - 1) = (n - 1)x_1 - (n - 1)(n - 2) \tag{10}$$

$$\implies x_1 = n \tag{11}$$

The Hessian, given by (3), is a symmetric, tridiagonal Toeplitz matrix with diagonal entries equal to 2 and off-diagonal entries equal to -1. The eigenvalues of this type of matrix are given by:

$$\lambda_k = 2 - 2\cos\left(\frac{k\pi}{n+1}\right), \quad k = 1, 2, \ldots, n. \tag{12}$$

Since $-1 < \cos\left(\frac{k\pi}{n+1}\right) < 1$, we conclude that all eigenvalues are positive. This confirms that the Hessian is positive definite, ensuring that the critical point found is a local minimum.

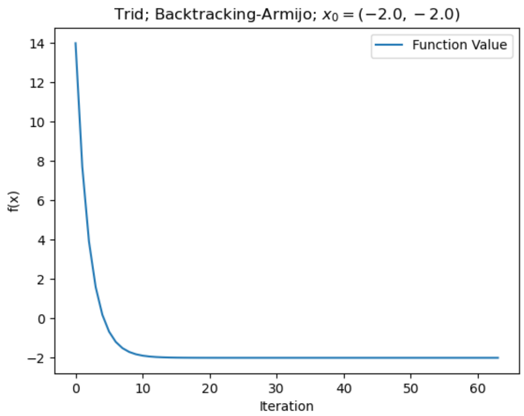
Thus, the **unique, global minimum** is:

$$\boxed{x_i = i(n+1-i) \quad \forall\ i \in [1, n]} \tag{13}$$

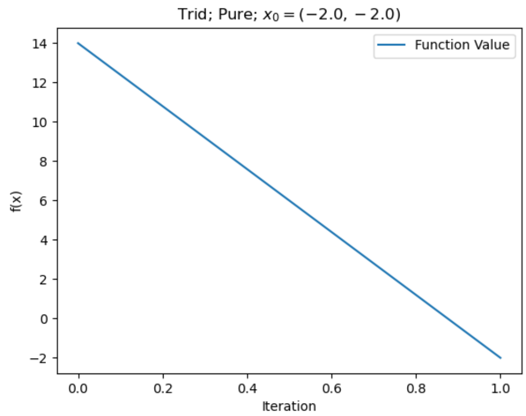
1.4 Plots

Below are the plots illustrating the optimization process:

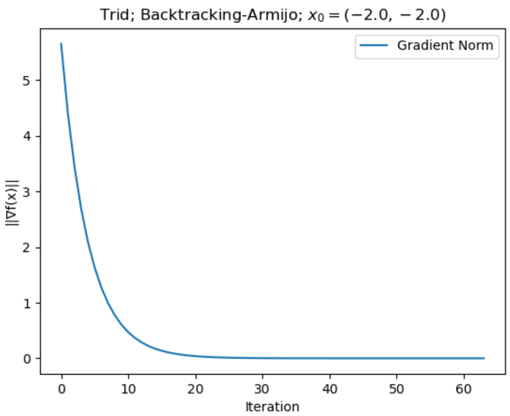
1. Plot of function values $f(x)$ vs. iterations.
2. Plot of gradient norm $\|\nabla f(x)\|$ vs. iterations.
3. Contour plot with the optimization path.



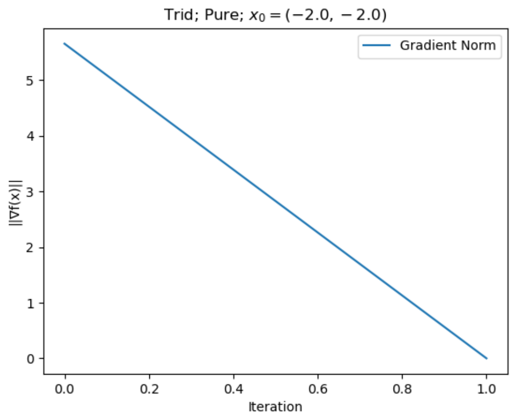
(a) $f(x)$ vs. iterations (Backtracking Armijo)



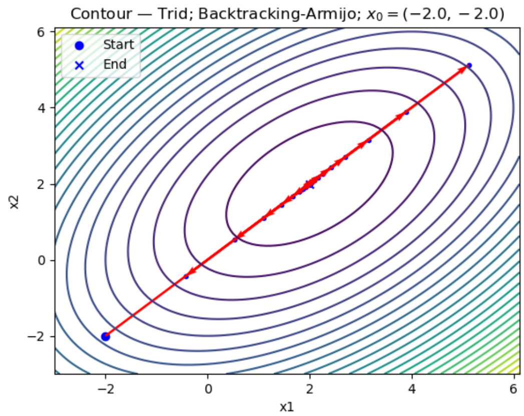
(b) $f(x)$ vs. iterations (Pure Newton's)



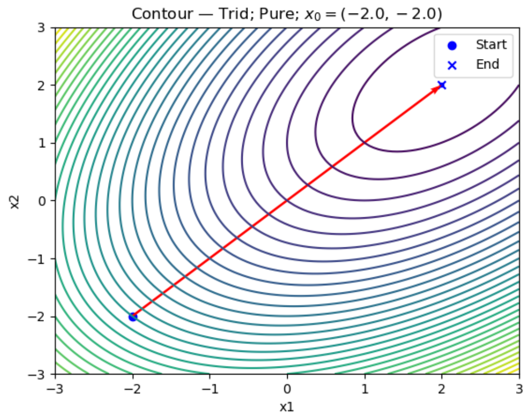
(a) $\|\nabla f(x)\|$ vs. iterations (Backtracking Armijo)



(b) $\|\nabla f(x)\|$ vs. iterations (Pure Newton's)



(a) Contour plot (Backtracking Armijo)



(b) Contour plot (Pure Newton's)

1.5 Results

Below are the minimizers across different methods and initial points:

Test case	Backtracking-Armijo	Backtracking-Goldstein	Bisection
0	[2. 2.]	[2. 2.]	[2. 2.]
1	[2. 2.]	[2. 2.]	[2. 2.]

Pure	Damped	Levenberg-Marquardt	Combined
[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]
[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]

Comments: All the algorithms led to convergence to the unique minimum.

2 Three-Hump Camel Function

The function is defined as:

$$f(x_1, x_2) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2 \tag{14}$$

2.1 Gradient

$$\nabla f = \begin{bmatrix} 4x_1 - 4.2x_1^3 + x_1^5 + x_2 \\ x_1 + 2x_2 \end{bmatrix} \tag{15}$$

2.2 Hessian

$$H = \begin{bmatrix} 4 - 12.6x_1^2 + 5x_1^4 & 1 \\ 1 & 2 \end{bmatrix} \tag{16}$$

2.3 Finding the Minima

Step 1: Solve for x_2

$$x_2 = -\frac{x_1}{2} \tag{17}$$

Step 2: Substitute into the first equation

$$4x_1 - 4.2x_1^3 + x_1^5 = \frac{x_1}{2} \tag{18}$$

$$\frac{7x_1}{2} = 4.2x_1^3 - x_1^5 \tag{19}$$

Factoring out x_1:

$$x_1 \left(\frac{7}{2} - 4.2x_1^2 + x_1^4 \right) = 0 \tag{20}$$

This gives solutions:

$$x_1 = 0 \quad \text{or} \quad \frac{7}{2} - 4.2x_1^2 + x_1^4 = 0 \tag{21}$$

For $x_1 = 0$:

$$x_2 = -\frac{0}{2} = 0 \tag{22}$$

Thus, a critical point is $(0, 0)$. Now, solving the equation:

$$x_1^4 - 4.2x_1^2 + \frac{7}{2} = 0 \tag{23}$$

Let $y = x_1^2$, so we rewrite it as:

$$y^2 - 4.2y + \frac{7}{2} = 0 \tag{24}$$

Using the quadratic formula $y = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$:

$$y = \frac{4.2 \pm \sqrt{(-4.2)^2 - 4(1)(7/2)}}{2(1)} \tag{25}$$

$$= \frac{4.2 \pm \sqrt{17.64 - 14}}{2} \tag{26}$$

$$= \frac{4.2 \pm \sqrt{3.64}}{2} \tag{27}$$

Approximating $\sqrt{3.64} \approx 1.91$:

$$y = \frac{4.2 \pm 1.91}{2} \tag{28}$$

giving two values:

$$y_1 = \frac{4.2 + 1.91}{2} = \frac{6.11}{2} = 3.055 \tag{29}$$

$$y_2 = \frac{4.2 - 1.91}{2} = \frac{2.29}{2} = 1.145 \tag{30}$$

Since $y = x_1^2$, we take square roots:

$$x_1 = \pm\sqrt{3.055}, \quad x_1 = \pm\sqrt{1.145} \tag{31}$$

Approximating:

$$x_1 \approx \pm 1.75, \quad x_1 \approx \pm 1.07 \tag{32}$$

From $x_2 = -\frac{x_1}{2}$:

$$x_2 \approx \mp 0.875, \quad x_2 \approx \mp 0.535 \tag{33}$$

Thus, the critical points are:

$$(0, 0), \quad (1.75, -0.875), \quad (-1.75, 0.875), \quad (1.07, -0.535), \quad (-1.07, 0.535) \tag{34}$$

For $x_1 = \pm 1.75$, all the leading principal minors of H were positive, confirming local minima. However, for $x_1 = \pm 1.07$, the first leading principal minor $H_{1,1}$ was negative. Therefore, the following are the **local minima**:

$$\boxed{(0, 0), (1.75, -0.875) \text{ and } (-1.75, 0.875)} \tag{35}$$

We now evaluate $f(x_1, x_2)$ at these points:

- At $(0, 0)$:

$$f(0, 0) = 2(0)^2 - 1.05(0)^4 + \frac{(0)^6}{6} + 0 \cdot 0 + (0)^2 = 0. \tag{36}$$

- At $(1.75, -0.875)$:

$$\begin{aligned} f(1.75, -0.875) &= 2(1.75)^2 - 1.05(1.75)^4 + \frac{(1.75)^6}{6} + (1.75)(-0.875) + (-0.875)^2 \\ &\approx 6.125 - 9.8479 + 4.79 - 1.5313 + 0.7656 \\ &\approx 0.3015. \end{aligned} \tag{37}$$

- At $(-1.75, 0.875)$, by symmetry we obtain:

$$f(-1.75, 0.875) \approx 0.3015. \tag{38}$$

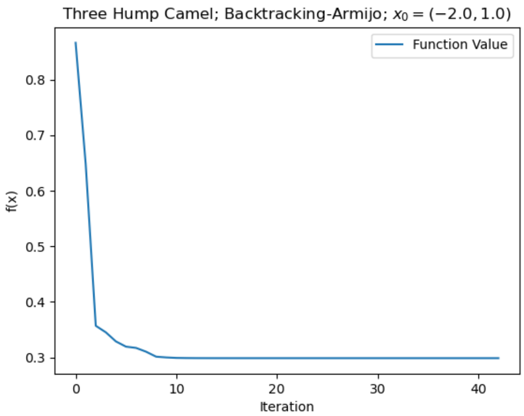
Since $f(0, 0) = 0$ is lower than $f(1.75, -0.875)$ and $f(-1.75, 0.875)$, the **global minimum** is achieved at:

$$\boxed{(0, 0)}. \tag{39}$$

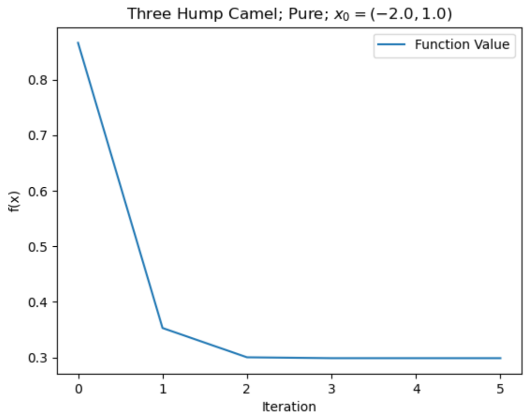
2.4 Plots

Below are the plots illustrating the optimization process:

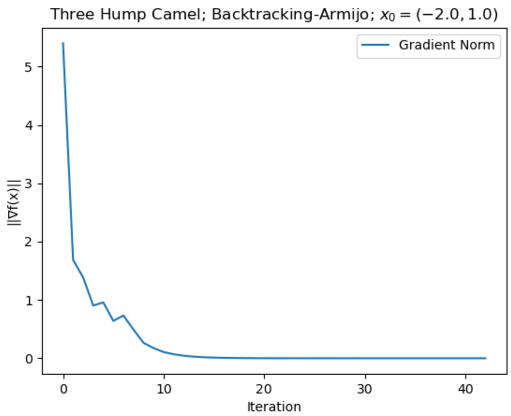
1. Plot of function values $f(x)$ vs. iterations.
2. Plot of gradient norm $\|\nabla f(x)\|$ vs. iterations.
3. Contour plot with the optimization path.



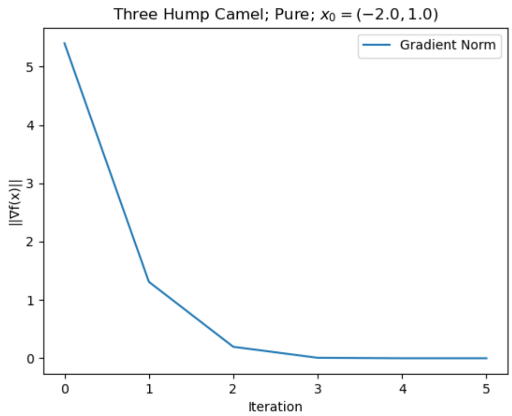
(a) $f(x)$ vs. iterations (Backtracking Armijo)



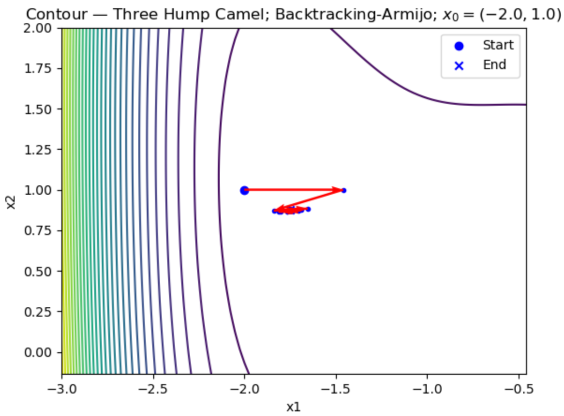
(b) $f(x)$ vs. iterations (Pure Newton's)



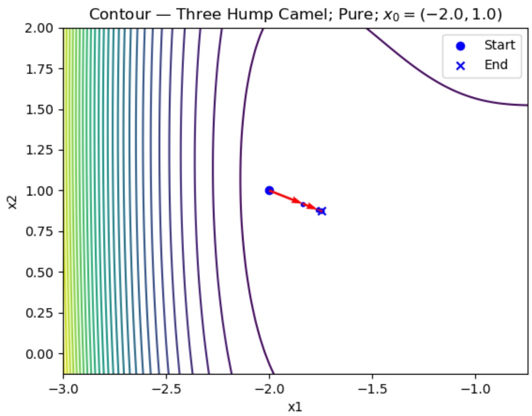
(a) $\|\nabla f(x)\|$ vs. iterations (Backtracking Armijo)



(b) $\|\nabla f(x)\|$ vs. iterations (Pure Newton's)



(a) Contour plot (Backtracking Armijo)



(b) Contour plot (Pure Newton's)

2.5 Results

Below are the minimizers across different methods and initial points:

2	[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]
3	[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]
4	[0. 0.]	[0. 0.]	[1.748 -0.874]
5	[-0. -0.]	[-0. -0.]	[-1.748 0.874]

[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]	[0. -0.]
[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]	[-0. 0.]
[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]
[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]

Comments: All the algorithms led to convergence to the either of the three minimizers.

3 Styblinski-Tang Function

The function is defined as:

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i) \tag{40}$$

3.1 Gradient

$$\frac{\partial f}{\partial x_i} = \frac{1}{2}(4x_i^3 - 32x_i + 5) \tag{41}$$

3.2 Hessian

$$H_{i,j} = \begin{cases} \frac{1}{2}(12x_i^2 - 32), & i = j \\ 0, & otherwise \end{cases} \tag{42}$$

3.3 Finding the Minima

Taking the gradient and setting it to zero:

$$\nabla f = 0 \tag{43}$$

Solving for x_i:

$$4x_i^3 - 32x_i + 5 = 0 \tag{44}$$

Using numerical methods, the three real roots are found to be:

$$x_1 \approx -2.903, \tag{45}$$
$$x_2 \approx\ 0.157, \tag{46}$$
$$x_3 \approx\ 2.747. \tag{47}$$
$$\tag{48}$$

Thus, for an n-dimensional input, each coordinate x_i can take one of these three values.

For the n-dimensional function, the Hessian is a diagonal matrix with the ith diagonal entry given by:

$$\frac{1}{2}(12x_i^2 - 32).$$

We now evaluate one of the Hessian's eigen values at the critical points:

- For $x \approx -2.903$:

$$\frac{1}{2}\Big(12 \cdot (-2.903)^2 - 32\Big) \approx \frac{1}{2}\Big(12 \cdot 8.426 - 32\Big) \approx \frac{1}{2}(101.112 - 32) > 0. \tag{49}$$

- For $x \approx 0.157$:

$$\frac{1}{2}\Big(12 \cdot (0.157)^2 - 32\Big) \approx \frac{1}{2}\Big(12 \cdot 0.0246 - 32\Big) \approx \frac{1}{2}(0.2952 - 32) < 0. \tag{50}$$

- For $x \approx 2.747$:

$$\frac{1}{2}\Big(12 \cdot (2.747)^2 - 32\Big) \approx \frac{1}{2}\Big(12 \cdot 7.548 - 32\Big) \approx \frac{1}{2}(90.576 - 32) > 0. \tag{51}$$

This Hessian is positive definite if and only if each x_i is chosen from $\{-2.903,\ 2.747\}$. Thus, the **local minima** of $f(\mathbf{x})$ occur at:

$$\boxed{\mathbf{x} = (x_1, x_2, \ldots, x_n) \quad \text{with} \quad x_i \in \{-2.903,\ 2.747\} \quad \text{for all } i.}$$

Since

$$f_i(-2.903) \approx -39.17 \quad \text{and} \quad f_i(2.747) \approx -25.02,$$

the **global minimum** is achieved when $x_i = -2.903$ for all i:

$$\boxed{\mathbf{x}^* \approx (-2.903, -2.903, \ldots, -2.903)}$$

with the global function value approximately

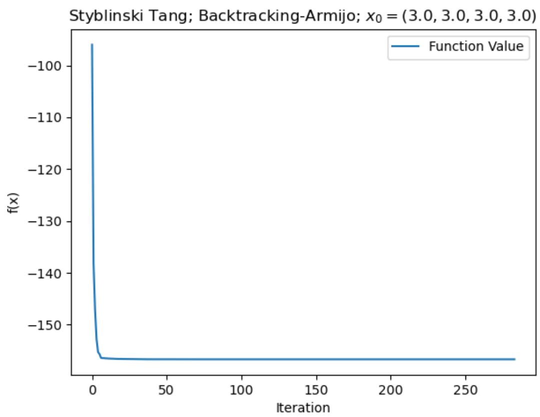
$$f(\mathbf{x}^*) \approx n \cdot (-39.17).$$

Any other combination (using both -2.903 and 2.747) yields a higher function value, although **those combinations are also local minima**.

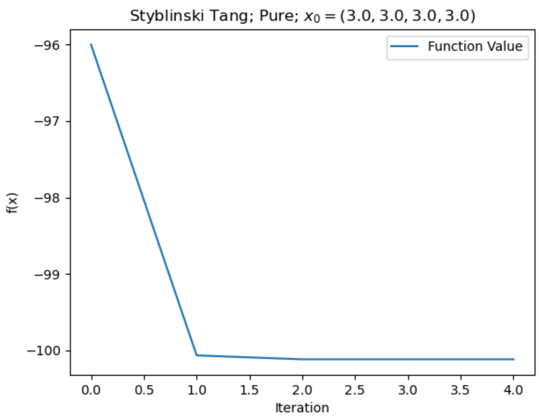
3.4 Plots

Below are the plots illustrating the optimization process:

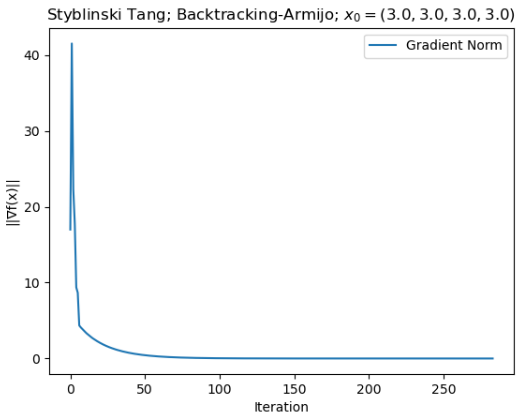
1. Plot of function values $f(x)$ vs. iterations.
2. Plot of gradient norm $\|\nabla f(x)\|$ vs. iterations.
3. Contour plot with the optimization path.



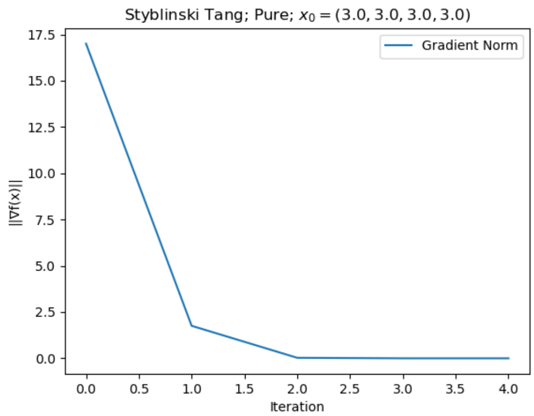
(a) $f(x)$ vs. iterations (Backtracking Armijo)



(b) $f(x)$ vs. iterations (Pure Newton's)



(a) $\|\nabla f(x)\|$ vs. iterations (Backtracking Armijo)



(b) $\|\nabla f(x)\|$ vs. iterations (Pure Newton's)

3.5 Results

Below are the minimizers across different methods and initial points:

10	[-2.904 -2.904 -2.904 -2.904]	[-0. -0. -0. -0.]	[-2.904 -2.904 -2.904 -2.904]
11	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]
12	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]
13	[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]

[0.157 0.157 0.157 0.157]	[0. 0. 0. 0.]	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]
[2.747 2.747 2.747 2.747]	[2.747 2.747 2.747 2.747]	[2.747 2.747 2.747 2.747]	[2.747 2.747 2.747 2.747]
[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -2.904]
[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]

Comments:

- With initial point $[0, 0, 0, 0]$ and Backtracking with Armijo-Goldstein and Damped Newton's algorithms, we observe that the update is **negligible**. Note that the origin is not a minima. Very tiny step sizes (or updates) can be mitigated by increasing the update factor, ρ from 0.75 to say, 0.99.
- With initial point $[0, 0, 0, 0]$ and Pure Newton's algorithm, the convergence stops at $(0.157, 0.157, 0.157, 0.157)$, a local maxima. Newton's method solves for stationary points, but it does not distinguish whether the point is a minimum, maximum, or saddle. The update step does not guarantee movement in a descent direction unless the Hessian is positive definite. Methods like Levenberg-Marquardt adds regularization to modify the Hessian.

4 Rosenbrock Function

The Rosenbrock function is given by:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right) \tag{52}$$

4.1 Gradient

$$\frac{\partial f}{\partial x_i} = \begin{cases} -400x_i(x_{i+1} - x_i^2) + 2(x_i - 1), & i = 1 \\ 200(x_i - x_{i-1}^2), & i = n \\ -400x_i(x_{i+1} - x_i^2) + 200(x_i - x_{i-1}^2) + 2(x_i - 1), & otherwise \end{cases} \tag{53}$$

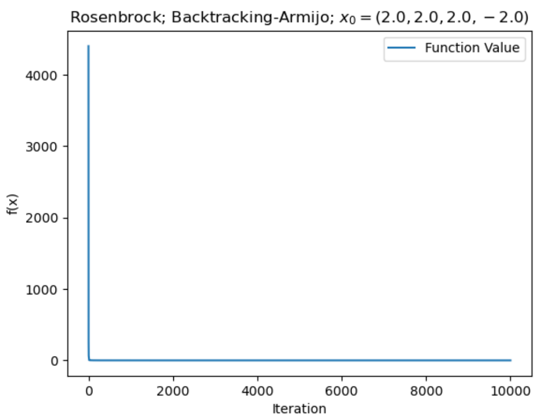
4.2 Hessian

$$H_{i,j} = \begin{cases} 2 - 400(x_2 - 3x_1^2), & i = j = 1 \\ 200, & i = j = n \\ 202 - 400(x_{i+1} - 3x_i^2), & i = j \text{ and } 1 < i < n \\ -400x_i, & j = i + 1 \text{ and } 1 \leq i < n \\ -400x_j, & j = i - 1 \text{ and } 1 < i \leq n \\ 0, & \text{otherwise} \end{cases} \tag{54}$$

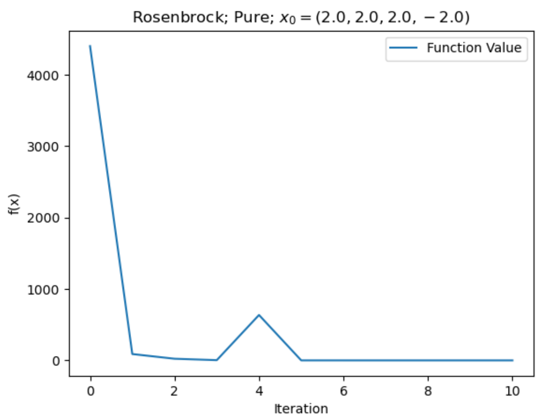
4.3 Plots

Below are the plots illustrating the optimization process:

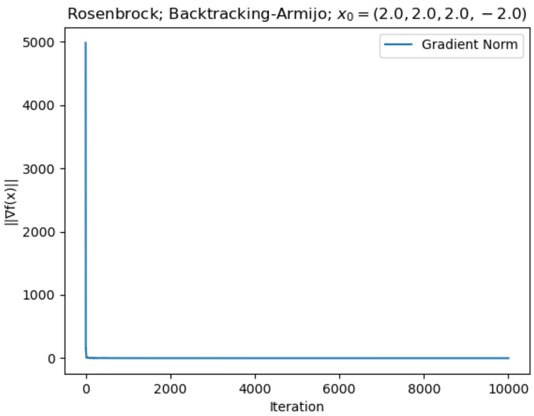
1. Plot of function values $f(x)$ vs. iterations.
2. Plot of gradient norm $\|\nabla f(x)\|$ vs. iterations.
3. Contour plot with the optimization path.



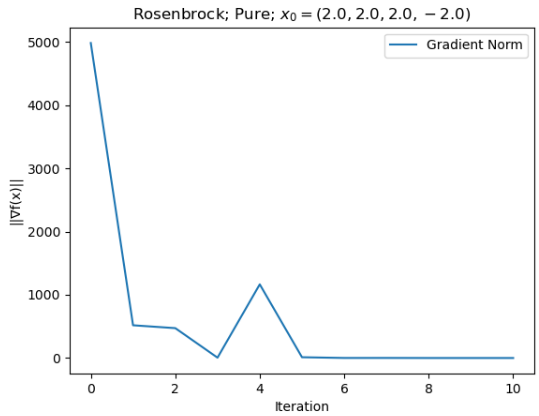
(a) $f(x)$ vs. iterations (Backtracking Armijo)



(b) $f(x)$ vs. iterations (Pure Newton's)



(a) $\|\nabla f(x)\|$ vs. iterations (Backtracking Armijo)



(b) $\|\nabla f(x)\|$ vs. iterations (Pure Newton's)

4.4 Results

Below are the minimizers across different methods and initial points:

6	[1. 1. 1. 0.999]	[1. 1. 1. 0.999]	[1. 1. 1. 1.]
7	[1. 1. 1. 0.999]	[1. 1. 1. 0.999]	[1. 1. 1. 1.]
8	[1. 1. 1. 0.999]	[1. 1. 1. 0.999]	[1. 1. 1. 1.]
9	[1. 1. 1. 0.999]	[1. 1. 1. 0.999]	[1. 1. 1. 1.]

[1. 1. 1. 1.] [1. 1. 1. 1.] [-0.776 0.613 0.382 0.146] [1. 1. 1. 1.]	[1. 1. 1. 1.] [1. 1. 1. 1.] [-0.776 0.613 0.382 0.146] [1. 1. 1. 1.]	[1. 1. 1. 1.] [1. 1. 1. 1.] [-0.776 0.613 0.382 0.146] [1. 1. 1. 1.]	[1. 1. 1. 1.] [1. 1. 1. 1.] [-0.776 0.613 0.382 0.146] [1. 1. 1. 1.]

Comments:

- With initial point $(-2, 2, 2, 2)$ and any of the Newton methods - Pure, Damped, Levenberg-Marquardt and Combined, the process stopped at $(-0.776, 0.613, 0.382, 0.146)$. We know that $(1, 1, 1, 1)$ is the global minima, but there are other local minimas like $(-0.776, 0.613, 0.382, 0.146)$ (this was verified by computing the gradient and function values near the point). So, all the methods converged to some local minima for Rosenbrock function.

5 Root of Square Function

The function is defined as:

$$f(\mathbf{x}) = \sqrt{x_1^2 + 1} + \sqrt{x_2^2 + 1} \tag{55}$$

5.1 Gradient

$$\nabla f = \begin{bmatrix} \frac{x_1}{\sqrt{x_1^2+1}} \\ \frac{x_2}{\sqrt{x_2^2+1}} \end{bmatrix} \tag{56}$$

5.2 Hessian

$$H = \begin{bmatrix} \frac{1}{(x_1^2+1)^{3/2}} & 0 \\ 0 & \frac{1}{(x_2^2+1)^{3/2}} \end{bmatrix} \tag{57}$$

5.3 Finding the Minima

To find the critical point(s), we set the gradient equal to zero:

$$\frac{x_1}{\sqrt{x_1^2+1}} = 0 \quad \text{and} \quad \frac{x_2}{\sqrt{x_2^2+1}} = 0.$$

Since the denominators are always positive, the only solution is:

$$x_1 = 0 \quad \text{and} \quad x_2 = 0.$$

Thus, the only critical point is $(0, 0)$.

Evaluating the Hessian at $(0, 0)$:

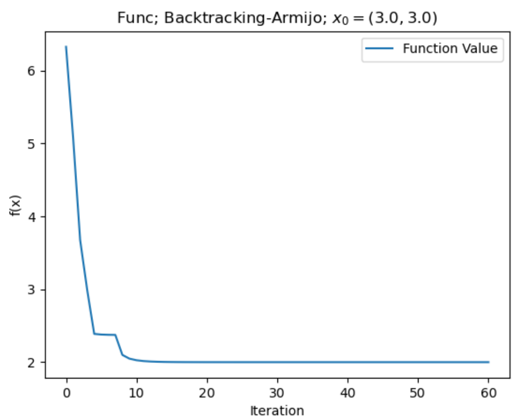
$$H(0,0) = \begin{bmatrix} \frac{1}{(0^2+1)^{3/2}} & 0 \\ 0 & \frac{1}{(0^2+1)^{3/2}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since this matrix is positive definite, the critical point $(0, 0)$ **is a local minimum**.

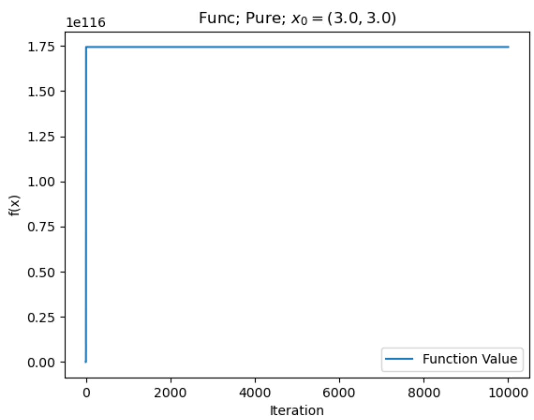
5.4 Plots

Below are the plots illustrating the optimization process:

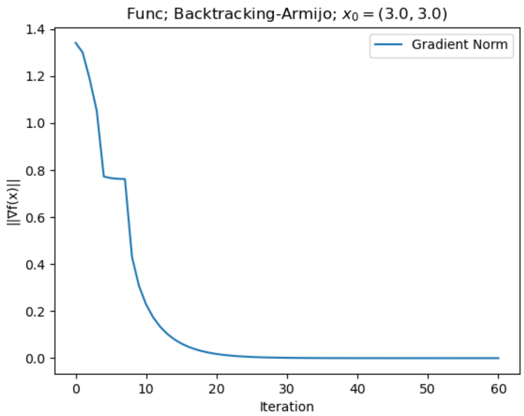
1. Plot of function values $f(x)$ vs. iterations.
2. Plot of gradient norm $\|\nabla f(x)\|$ vs. iterations.
3. Contour plot with the optimization path.



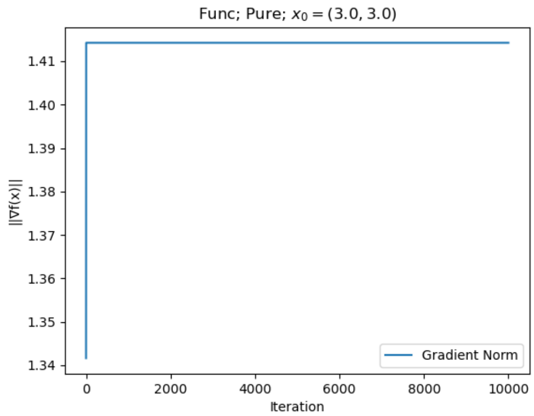
(a) $f(x)$ vs. iterations (Backtracking Armijo)



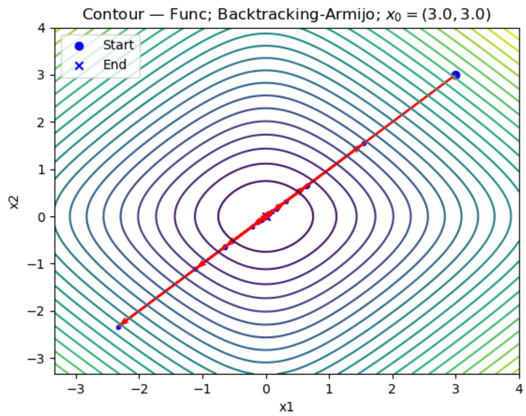
(b) $f(x)$ vs. iterations (Pure Newton's)



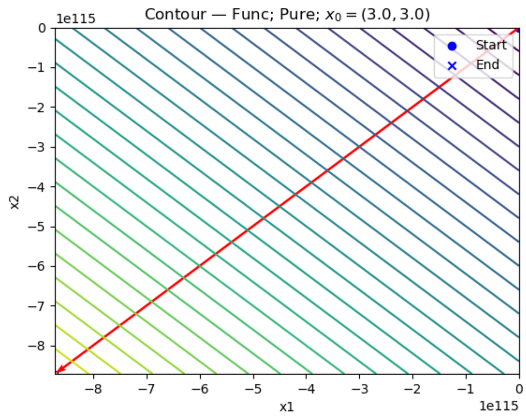
(a) $\|\nabla f(x)\|$ vs. iterations (Backtracking Armijo)



(b) $\|\nabla f(x)\|$ vs. iterations (Pure Newton's)



(a) Contour plot (Backtracking Armijo)



(b) Contour plot (Pure Newton's)

5.5 Results

Below are the minimizers across different methods and initial points:

14 15 16	[0. 0.] [0. -0.] [0. -0.]	[0. 0.] [0. -0.] [0. -0.]	[-0. -0.] [0. -0.] [0. -0.]

[-8.71896425e+115 -8.71896425e+115] [0. -0.] [1.61634765e+132 0.00000000e+000]	[-0. -0.] [0. -0.] [-0. 0.]	[-8.71896425e+115 -8.71896425e+115] [0. -0.] [1.61634765e+132 0.00000000e+000]	[0. 0.] [0. -0.] [0. -0.]

Comments:

- With initial point $(3, 3)$ and either of Pure Newton's or Levenberg-Marquardt algorithm, the optimization stops at (-∞, -∞). Following are some observations. In case of Pure Newton's, **singular** Hessian is encountered in almost all iterations. Besides, the process divergingly oscillates around the minima $(0, 0)$. Also, the gradient explodes within a few iterations in the process and remains unchanged for subsequent iterations. A probable reason for divergence is **negative eigenvalues** of Hessian, pushing the update toward a maximum (ill-conditioned).

- With initial point $(-3.5, 0.5)$ and either of Pure Newton's or Levenberg-Marquardt algorithm, the optimization stops at (∞, 0). While the reason for divergence can be derived from previous point, the other coordinate converged to 0. The initial value (x_2) is already near the minimum at 0. In that region, the gradient is small and the Hessian is **more stable**. The update step is **moderate** and thus, it quickly converged to 0.

- To mitigate the effect of gradient explosion and allow convergence to minimum, **gradient clipping** can be used. For instance, with Pure Newton method, clipping gradient norm to 1.1 helped the algorithm to converge to (0, 0). However, using larger limits like 10, did not help converge, though it helped contain coordinates to finite range.