INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

OPTIMIZATION METHODS

ASSIGNMENT 1

REPORT

Name: Deepti Rawat

Roll No: [redacted]

Date: 11 March 2024

1. Derive the Jacobians and Hessians for all the functions.

Trid Function

$$f(\bar{x}) = \sum_{i=1}^{d}(x_i - 1)^2 - \sum_{i=2}^{d} x_{i-1} \cdot x_i$$

Step 1: Generalizing for 'd'

- For **d = 1**: $f(\bar{x}) = (x_1 - 1)^2 - 0$
- For **d = 2**: $f(\bar{x}) = (x_1 - 1)^2 + (x_2 - 1)^2 - x_1 \cdot x_2$
- For **d = 3**: $f(\bar{x}) = (x_1 - 1)^2 + (x_2 - 1)^2 + (x_3 - 1)^2 - x_1 \cdot x_2 - x_2 \cdot x_3$
- For **d**: $f(\bar{x}) = (x_1 - 1)^2 + \cdots + (x_d - 1)^2 - x_1 \cdot x_2 - \cdots - x_{d-2} \cdot x_{d-1} - x_{d-1}x_d$

Step 2: Computing Jacobian

- $\frac{\partial f(\bar{x})}{\partial x_1} = 2(x_1 - 1) - x_0 - x_2$ $\quad [x_0 = 0]$
- $\frac{\partial f(\bar{x})}{\partial x_2} = 2(x_2 - 1) - x_1 - x_3$
- $\frac{\partial f(\bar{x})}{\partial x_3} = 2(x_3 - 1) - x_2 - x_4$
- $\frac{\partial f(\bar{x})}{\partial x_d} = 2(x_d - 1) - x_{d-1} - x_{d+1}$ $\quad [x_{d+1} = 0]$

$$\therefore \quad \nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \\ \vdots \\ \partial f(\bar{x})/\partial x_d \end{bmatrix} = \begin{bmatrix} 2(x_1 - 1) - x_2 \\ 2(x_2 - 1) - x_1 - x_3 \\ \vdots \\ 2(x_d - 1) - x_{d-1} - x_{d+1} \end{bmatrix}$$

Step 3: Computing Hessian

- $\frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} = 2 \qquad \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} = 2 \qquad \frac{\partial^2 f(\bar{x})}{\partial {x_3}^2} = 2 \qquad \cdots \qquad \frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} = 2$
- $\frac{\partial^2 f(\bar{x})}{\partial x_1 x_2} = -1 \qquad \frac{\partial^2 f(\bar{x})}{\partial x_2 x_3} = -1 \qquad \cdots \qquad \frac{\partial^2 f(\bar{x})}{\partial x_{d-1} x_d} = -1$
- $\frac{\partial^2 f(\bar{x})}{\partial x_2 x_1} = -1 \qquad \frac{\partial^2 f(\bar{x})}{\partial x_3 x_2} = -1 \qquad \frac{\partial^2 f(\bar{x})}{\partial x_4 x_3} = -1$

- $\frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 2 \ \ (for\ i = j) \qquad \frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = -1 \ \ (for\ i = j \pm 1) \qquad \frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 0 \ \ (otherwise)$

$$\therefore \quad H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \dfrac{\partial^2 f(\bar{x})}{\partial {x_1}^2} & \dfrac{\partial^2 f(\bar{x})}{\partial x_1 x_2} & \cdots & \dfrac{\partial^2 f(\bar{x})}{\partial x_1 x_d} \\ \dfrac{\partial^2 f(\bar{x})}{\partial x_2 x_1} & \dfrac{\partial^2 f(\bar{x})}{\partial {x_2}^2} & \cdots & \vdots \\ \vdots & \vdots & & \\ \dfrac{\partial^2 f(\bar{x})}{\partial x_d x_1} & \dfrac{\partial^2 f(\bar{x})}{\partial x_d x_2} & \cdots & \dfrac{\partial^2 f(\bar{x})}{\partial {x_d}^2} \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & & \\ 0 & -1 & 2 & & \vdots \\ 0 & 0 & -1 & \ddots & \\ \vdots & \vdots & \vdots & & 0 \\ & & & & -1 \\ 0 & 0 & & \cdots & 2 \end{bmatrix}$$

Three Hump Camel Function

$$f(\bar{x}) = 2{x_1}^2 - 1.05{x_1}^4 + \frac{1}{6}{x_1}^6 + x_1 x_2 + {x_2}^2$$

<u>Step 1</u>: Computing Jacobian

- $\frac{\partial f(\bar{x})}{\partial x_1} = 4x_1 - 4.2{x_1}^3 + {x_1}^5 + x_2$
- $\frac{\partial f(\bar{x})}{\partial x_2} = x_1 + 2x_2$

$$\therefore \quad \nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \end{bmatrix} = \begin{bmatrix} 4x_1 - 4.2{x_1}^3 + {x_1}^5 + x_2 \\ x_1 + 2x_2 \end{bmatrix}$$

<u>Step 2</u>: Computing Hessian

$$\therefore \quad H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \dfrac{\partial^2 f(\bar{x})}{\partial {x_1}^2} & \dfrac{\partial^2 f(\bar{x})}{\partial x_1 x_2} \\ \dfrac{\partial^2 f(\bar{x})}{\partial x_2 x_1} & \dfrac{\partial^2 f(\bar{x})}{\partial {x_2}^2} \end{bmatrix} = \begin{bmatrix} 4 + 5{x_1}^4 - 12.6{x_1}^2 & 1 \\ 1 & 2 \end{bmatrix}$$

Styblinski-Tang Function

$$f(\bar{x}) = \frac{1}{2}\sum_{i=1}^{d}({x_i}^4 - 16{x_i}^2 + 5x_i)$$

Step 1: Generalizing for 'd'

- For $\boldsymbol{d} = \mathbf{1}$: $f(\bar{x}) = \frac{1}{2}({x_1}^4 - 16{x_1}^2 + 5x_1)$
- For $\boldsymbol{d} = \mathbf{2}$: $f(\bar{x}) = \frac{1}{2}({x_1}^4 - 16{x_1}^2 + 5x_1) + \frac{1}{2}({x_2}^4 - 16{x_2}^2 + 5x_2)$
- For $\boldsymbol{d} = \mathbf{3}$: $f(\bar{x}) = \frac{1}{2}({x_1}^4 - 16{x_1}^2 + 5x_1) + \frac{1}{2}({x_2}^4 - 16{x_2}^2 + 5x_2) + \frac{1}{2}({x_3}^4 - 16{x_3}^2 + 5x_3)$
- For $\boldsymbol{d}$: $f(\bar{x}) = \frac{1}{2}({x_1}^4 - 16{x_1}^2 + 5x_1) + \frac{1}{2}({x_2}^4 - 16{x_2}^2 + 5x_2) + \cdots + \frac{1}{2}({x_d}^4 - 16{x_d}^2 + 5x_d)$

Step 2: Computing Jacobian

- $\frac{\partial f(\bar{x})}{\partial x_1} = \frac{1}{2}(4{x_1}^3 - 32x_1 + 5)$
- $\frac{\partial f(\bar{x})}{\partial x_2} = \frac{1}{2}(4{x_2}^3 - 32x_2 + 5)$
- $\frac{\partial f(\bar{x})}{\partial x_d} = \frac{1}{2}(4{x_d}^3 - 32x_d + 5)$

$$\therefore \quad \nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \\ \vdots \\ \partial f(\bar{x})/\partial x_d \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 4{x_1}^3 - 32x_1 + 5 \\ 4{x_2}^3 - 32x_2 + 5 \\ \vdots \\ 4{x_d}^3 - 32x_d + 5 \end{bmatrix}$$

Step 3: Computing Hessian

- $\frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} = \frac{1}{2}(12{x_1}^2 - 32) \qquad \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} = \frac{1}{2}(12{x_2}^2 - 32) \qquad \cdots \qquad \frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} = \frac{1}{2}(12{x_d}^2 - 32)$
- $\frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 0 \; for \; i \neq j$

$$H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_d} \\ \frac{\partial^2 f(\bar{x})}{\partial x_2 x_1} & \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} & \cdots & \vdots \\ \vdots & \vdots & & \\ \frac{\partial^2 f(\bar{x})}{\partial x_d x_1} & \frac{\partial^2 f(\bar{x})}{\partial x_d x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} \end{bmatrix}$$

$$\therefore \quad H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \frac{1}{2}(12{x_1}^2 - 32) & 0 & \cdots & 0 \\ 0 & \frac{1}{2}(12{x_2}^2 - 32) & & \\ & & \ddots & \vdots \\ 0 & 0 & & \\ \vdots & \vdots & & \\ 0 & 0 & \cdots & \frac{1}{2}(12{x_3}^2 - 32) \end{bmatrix}$$

Rosenbrock Function

$$f(\bar{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2]$$

<u>Step 1</u>: Generalizing for 'd'

- For $\boldsymbol{d} = \mathbf{1}$: $f(\bar{x}) = 0$
- For $\boldsymbol{d} = \mathbf{2}$: $f(\bar{x}) = [100(x_2 - {x_1}^2)^2 + (x_1 - 1)^2]$
- For $\boldsymbol{d} = \mathbf{3}$: $f(\bar{x}) = [100(x_2 - {x_1}^2)^2 + (x_1 - 1)^2] + [100(x_3 - {x_2}^2)^2 + (x_2 - 1)^2]$
- For $\boldsymbol{d}$ ($\boldsymbol{d} \neq 1$): $f(\bar{x}) = [100(x_2 - {x_1}^2)^2 + (x_1 - 1)^2] + [100(x_3 - {x_2}^2)^2 + (x_2 - 1)^2]$
  $+ [100(x_4 - {x_3}^2)^2 + (x_3 - 1)^2] + \cdots + [100(x_d - {x_{d-1}}^2)^2 + (x_{d-1} - 1)^2]$

<u>Step 2</u>: Computing Jacobian

- $\frac{\partial f(\bar{x})}{\partial x_1} = [200(x_2 - {x_1}^2) \cdot (-2x_1) + 2(x_1 - 1)]$ $\qquad [for\ i = 1]$
- $\frac{\partial f(\bar{x})}{\partial x_2} = [200(x_2 - {x_1}^2)] + [200(x_3 - {x_2}^2)(-2x_2) + 2(x_2 - 1)]$
- $\frac{\partial f(\bar{x})}{\partial x_d} = [200(x_d - {x_{d-1}}^2)] + [200(x_{d+1} - {x_d}^2)(-2x_d) + 2(x_d - 1)]$ $\qquad [for\ i = d]$
- Generalizing:
  - $\frac{\partial f(\bar{x})}{\partial x_i} = [200(x_i - {x_{i-1}}^2)] + [200(x_{i+1} - {x_i}^2)(-2x_i) + 2(x_i - 1)]$
  - $\Rightarrow \frac{\partial f(\bar{x})}{\partial x_i} = -400(x_{i+1} \cdot x_i - {x_i}^3) + 202x_i - 2 - 200{x_{i-1}}^2$
  - $\Rightarrow \frac{\partial f(\bar{x})}{\partial x_i} = 400{x_i}^3 + (202 - 400x_{i+1})x_i - 2(1 + 100x_{i-1}^2) \quad \forall\ \mathrm{i} \neq 1, \mathrm{d}$

$$\therefore \quad \nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \\ \vdots \\ \partial f(\bar{x})/\partial x_d \end{bmatrix} = \begin{bmatrix} [400{x_1}^3 + (2-400x_2)x_1 - 2] \\ \vdots \\ [400{x_1}^3 + (202-400x_{i+1})x_i - 2 - 200{x_{i-1}}^2] \\ \vdots \\ [200(x_d - {x_{d-1}}^2)] \end{bmatrix}$$

<u>Step 3</u>: Computing Hessian

- $\frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} = [1200{x_1}^2 - 400x_2 + 2]$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_1 x_2} = -400x_1$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_1 x_3} = 0$ ....
- $\frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} = [1200{x_2}^2 - 400x_3 + 202]$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_2 x_1} = -400x_1$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_2 x_3} = -400x_2$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_2 x_4} = 0 \ldots$
- $\frac{\partial^2 f(\bar{x})}{\partial {x_3}^2} = [1200{x_3}^2 - 400x_4 + 202]$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_3 x_1} = 0$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_3 x_2} = -400x_2$
- $\frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} = [1200{x_d}^2 - 400x_{d+1} + 202]$ $\quad \frac{\partial^2 f(\bar{x})}{\partial x_{d-1} x_d} = -400x_{d-1}$

$$\therefore \quad H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_d} \\ \frac{\partial^2 f(\bar{x})}{\partial x_2 x_1} & \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} & \cdots & \vdots \\ \vdots & \vdots & & \\ \frac{\partial^2 f(\bar{x})}{\partial x_d x_1} & \frac{\partial^2 f(\bar{x})}{\partial x_d x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} \end{bmatrix}$$

$$= \begin{bmatrix} 1200{x_1}^2 - 400x_2 + 2 & -400x_1 & & 0 & 0 \\ -400x_1 & 1200{x_2}^2 - 400x_3 + 2 & & & \\ 0 & -400x_2 & & \ddots & \vdots \\ \vdots & \vdots & & & \\ 0 & 0 & \cdots & & [1200{x_d}^2 - 400x_{d+1} + 202] \end{bmatrix}$$

$$\frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 1200{x_i}^2 - 400x_{i+1} + 2 \;\; (for\ i = j)$$

Root of Square Function

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

<u>Step 1</u>: Computing Jacobian

- $\frac{\partial f(\bar{x})}{\partial x_1} = \frac{1}{2}(1 + x_1^2)^{-\frac{1}{2}} \cdot (2x_1) = x_1(1 + x_1^2)^{-\frac{1}{2}} = \frac{x_1}{\sqrt{1+x_1^2}}$
- $\frac{\partial f(\bar{x})}{\partial x_2} = \frac{1}{2}(1 + x_2^2)^{-\frac{1}{2}} \cdot (2x_2) = x_2(1 + x_2^2)^{-\frac{1}{2}} = \frac{x_2}{\sqrt{1+x_2^2}}$

$$\therefore \quad \nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \end{bmatrix} = \begin{bmatrix} x_1(1 + x_1^2)^{-\frac{1}{2}} \\ x_2(1 + x_2^2)^{-\frac{1}{2}} \end{bmatrix}$$

<u>Step 2</u>: Computing Hessian

$$\therefore \quad H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_2} \\ \frac{\partial^2 f(\bar{x})}{\partial x_2 x_1} & \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} \end{bmatrix} = \begin{bmatrix} \frac{-x_1^2}{(1 + x_1^2)^{\frac{3}{2}}} + \frac{1}{(1 + x_1^2)^{\frac{1}{2}}} & 0 \\ 0 & \frac{-x_2^2}{(1 + x_2^2)^{\frac{3}{2}}} + \frac{1}{(1 + x_2^2)^{\frac{1}{2}}} \end{bmatrix}$$

2. Using the Jacobians and Hessians, calculate the minima for all functions except Rosenbrock.

Trid Function

Calculating minima for general d-dimension function:

- Stationary points by equating Jacobian to **0**:

$$\nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \\ \vdots \\ \partial f(\bar{x})/\partial x_d \end{bmatrix} = \begin{bmatrix} 2(x_1 - 1) - x_2 \\ 2(x_2 - 1) - x_1 - x_3 \\ \vdots \\ 2(x_d - 1) - x_{d-1} - x_{d+1} \end{bmatrix} = \mathbf{0}\ [For\ \partial f(\bar{x})/\partial x_d,\quad x_{d+1} = \mathbf{0}]$$

We can find roots by equating system of equations to 0:

- $2(x_1 - 1) - x_2 = 0 \ \Rightarrow x_2 = 2(x_1 - 1)$ ... (i)

- $2(x_2 - 1) - x_1 - x_3 = 0 \ \Rightarrow x_1 + x_3 = 2(x_2 - 1)$

  $\Rightarrow\ x_1 + x_3 = 2(2(x_1 - 1)\ - 1)$ $\quad Substituting\ x_2\ from\ (i)$

  $\Rightarrow\ x_1 + x_3 = 4x_1 - 4 - 2$

  $\Rightarrow x_3 = 3(x_1 - 2)$ ... (ii)

- $2(x_3 - 1) - x_2 - x_4 = 0 \ \Rightarrow x_2 + x_4 = 2(x_3 - 1)$

  $\Rightarrow\ 2(x_1 - 1) + x_4 = 2(3(x_1 - 2)\ - 1)$ $\quad Plugging\ x_2\ and\ x_3\ from\ (i)\ \&\ (ii)\ resp.$

  $\Rightarrow\ 2x_1 - 2 + x_4 = 6x_1 - 12 - 2)$ $\quad Substituting\ x_2\ from\ (i)$

  $\Rightarrow\ x_4 = 4(x_1 - 3)$ $\quad Substituting\ x_2\ from\ (i)$

- Similarly,

  $x_{d-1} = (d-1)(x_1 - (d - 1 - 1))$

  $= (d-1)(x_1 - d + 2)$ ... (iii)

  $x_d = d(x_1 - d + 1))$ ... (iv)

- $2(x_d - 1) = x_{d-1}$ $\quad Plugging\ \ x_{d-1}\ \ and\ x_d\ from\ (iii)\ and\ (iv)\ resp.$

  $\Rightarrow\ 2(d(x_1 - d + 1) - 1) = (d-1)(x_1 - d + 2)$

  $\Rightarrow dx_1 + x_1 = d^2 + d \ \ \Rightarrow (d+1)x_1 = (d+1)d \ \ \Rightarrow x_1 = d$ ... (v)

- For generalization, writing (i), (ii), (iii) and (iv) in terms of (v), we get:

  $x_2 = 2(d-1),\ \ x_3 = 3(d-2),\ \ x_{d-1} = 2(d-1),\ \ x_d = d$

- Therefore, the stationary points are:

  $x_1 = d, \;\; x_d = d$

  $x_i = i\big(d - (i-1)\big) \quad \forall\, i \neq 1, d$

- Validating minima by analysing Hessian:

$$\therefore \quad H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial x_1 x_d} \\ \frac{\partial^2 f(\bar{x})}{\partial x_2 x_1} & \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} & \cdots & \vdots \\ \vdots & \vdots & & \\ \frac{\partial^2 f(\bar{x})}{\partial x_d x_1} & \frac{\partial^2 f(\bar{x})}{\partial x_d x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & & \\ 0 & -1 & 2 & & \vdots \\ 0 & 0 & -1 & \ddots & \\ \vdots & \vdots & \vdots & & 0 \\ & & & & -1 \\ 0 & 0 & & \cdots & 2 \end{bmatrix}$$

We have,

$$\frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 2 \;\; (for\ i = j) \qquad \frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = -1 \;\; (for\ i = j \pm 1) \qquad \frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 0 \;\; (otherwise)$$

From above, we can observe that Hessian doesn't depend on variable 'd',
Therefore, using Sylvester criteria: If the determinant of all minors is positive, then the Hessian in positive definite.

- First minor: $\det(2) = 2 > 0$
- Second minor: $\det\left(\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\right) = 3 > 0$
- Third minor: $\det\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} = 4 > 0$
- Generalizing for kth minor: det(kth minor)=$k + 1 \; > \; 0 \qquad \forall\, k\ \epsilon\ [1,2, \dots, d]$

As we observe, determinant of all minors is positive and therefore, from Sylvester criteria, Hessian is positive definite.

Since, Hessian is positive definite, all stationary points are local minima.

- Hessian is positive definite. Hence, the stationary points $x_1 = d, \;\; x_d = d,$

  $x_i = i\big(d - (i-1)\big) \quad \forall\, i \neq 1, d$ is a minima.

Three Hump Camel Function

- Stationary points:

$$\nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \end{bmatrix} = \begin{bmatrix} 4x_1 - 4.2{x_1}^3 + {x_1}^5 + x_2 \\ x_1 + 2x_2 \end{bmatrix} = \mathbf{0}$$

- We can find roots by equating system of equations to 0.
  - $4x_1 - 4.2{x_1}^3 + {x_1}^5 + x_2 = 0$ ... (1)
  - $x_1 + 2x_2 = 0 \Rightarrow x_1 = -2x_2$ ... (2)
  - Substituting (2) in (1):
    - $4(-2x_2) - 4.2(-2x_2)^3 + (-2x_2)^5 + x_2 = 0$
    - $\Rightarrow -7x_2 + 33.6{x_2}^3 - 32{x_2}^5 = 0$
    - $\Rightarrow x_2(-7 + 33.6{x_2}^2 - 32{x_2}^4) = 0$ ... (3)
    - $x_2 = 0$ (root) ... (4)
    - Plugging (4) in (2):
    - $x_1 = 0$
    - First pair of stationary points (0,0)
  - Solving the other part of eq. (3), i.e., $(-7 + 33.6{x_2}^2 - 32{x_2}^4) = 0$
    - Substituting $y = {x_2}^2$ in the above equation:
    - $-32y^2 + 33.6y - 7 = 0$ ... (5)
    - $y = 0.286$ and $y = 0.763$ ... (6)
    - We know $y = {x_2}^2 \quad \Rightarrow \quad x_2 = \pm\sqrt{y}$ ... (7)
    - Plugging the values from eq. (6) to eq. (7):
    - $x_2 = \pm 0.5347$ and $x_2 = \pm 0.8734$ ... (8)
    - Plugging (8) in (2):
    - For $x_2 = 0.5347$, $x_1 = -1.0694$
    - For $x_2 = -0.5347$, $x_1 = 1.0694$
    - For $x_2 = 0.8734$, $x_1 = -1.7468$
    - For $x_2 = -0.8734$, $x_1 = 1.7468$
- Stationary points in the form of (x_1, x_2) are:
  - $(0,0)$, $(-1.0694, 0.5347)$, $(1.0694, -0.5347)$, $(-1.7468, 0.8734)$, $(1.7468, -0.8734)$
- Validating minima by plugging values of stationary points in the Hessian:
- $H = \begin{bmatrix} 4 + 5{x_1}^4 - 12.6{x_1}^2 & 1 \\ 1 & 2 \end{bmatrix}$
- Plugging the stationary points in the above Hessian matrix, we find that Hessian is positive definite for the stationary points: $(0,0)$, $(-1.7468, 0.8734)$, $(1.7468, -0.8734)$
- Therefore, the minima for this function are: $(0,0)$, $(-1.7468, 0.8734)$, $(1.7468, -0.8734)$

Styblinski-Tang Function

Calculating minima for general d-dimension function:

- Stationary point:

$$\nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \\ \vdots \\ \partial f(\bar{x})/\partial x_d \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 4{x_1}^3 - 32x_1 + 5 \\ 4{x_2}^3 - 32x_2 + 5 \\ \vdots \\ 4{x_d}^3 - 32x_d + 5 \end{bmatrix} = \mathbf{0}$$

- We can find roots by equating system of equations to 0:
  - $\partial f(\bar{x})/\partial x_i = \frac{1}{2}4{x_i}^3 - 32x_i + 5 = 0$ ... (i)

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

$$\Rightarrow {x_i}^3 - 8x_i + \frac{5}{4} = 0$$

$$\Rightarrow {x_i}^3 - 8\left(x_i - \frac{5}{32}\right) = 0$$

  - Solving for the roots of cubic equation in (i) using scientific calculator:
    - $x_i = -2.9035 \;\; or \;\; 0.1567 \;\; or \;\; 2.7468$

- Validating minima by plugging values of stationary points in the Hessian:

$$H = \nabla^2 f(\bar{x}) = \begin{bmatrix} \frac{1}{2}(12{x_1}^2 - 32) & 0 & \cdots & 0 \\ 0 & \frac{1}{2}(12{x_1}^2 - 32) & & \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & & \\ 0 & 0 & \cdots & \frac{1}{2}(12{x_1}^2 - 32) \end{bmatrix}$$

For the above Hessian matrix, we observe that:

$$\frac{\partial^2 f(\bar{x})}{\partial {x_1}^2} = \frac{1}{2}(12{x_1}^2 - 32) \qquad \frac{\partial^2 f(\bar{x})}{\partial {x_2}^2} = \frac{1}{2}(12{x_2}^2 - 32) \quad \cdots \quad \frac{\partial^2 f(\bar{x})}{\partial {x_d}^2} = \frac{1}{2}(12{x_d}^2 - 32)$$

i.e., $\frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = \frac{\partial^2 f(\bar{x})}{\partial {x_i}^2} = \frac{1}{2}(12{x_i}^2 - 32) \quad for\ i = j$ $\qquad \frac{\partial^2 f(\bar{x})}{\partial x_i x_j} = 0 \quad for\ i \neq j$

- Stationary points have to satisfy, second-order necessary condition (SONC) to qualify as local minima. SONC is satisfied if Hessian is positive i.e., $H = \nabla^2 f(\bar{x}) > 0$.
- SONC is satisfied for the given Hessian matrix, if all its diagonal elements are positive, i.e., $\frac{\partial^2 f(\bar{x})}{\partial {x_i}^2} > 0$.

$$\Rightarrow \frac{1}{2}(12{x_i}^2 - 32) > 0 \;\; \Rightarrow {x_i}^2 > \frac{32}{12} \;\; \Rightarrow {x_i}^2 > \frac{8}{3}$$

$$\Rightarrow -\frac{8}{3} > x_i > \frac{8}{3} \Rightarrow -2.667 > x_i > 2.667 \qquad \ldots (ii)$$

- From (ii), we know that stationary points -2.9035 *and* 2.7468 SONC. Therefore, they are the minima.

Root of Square Function

- Stationary point:

$$\nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \end{bmatrix} = \begin{bmatrix} x_1(1+x_1^2)^{-\frac{1}{2}} \\ x_2(1+x_2^2)^{-\frac{1}{2}} \end{bmatrix} = \mathbf{0}$$

$$\Rightarrow \frac{x_1}{\sqrt{1+x_1^2}} = 0 \Rightarrow x_1 = 0$$

$$\Rightarrow \frac{x_2}{\sqrt{1+x_2^2}} = 0 \Rightarrow x_2 = 0$$

Stationary point: $(x_1, x_2) = (0,0)$

- Validating minima by plugging values of stationary points in the Hessian:

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

$$\Rightarrow det(H) > 0 \ (positive\ definite)$$

- Hessian is positive definite. Hence the stationary point $(0,0)$ is a strict local minimum.

3. State which algorithms failed to converge and under which circumstances.
4. Plot f(x) vs iterations and |f′(x)| vs iterations.
5. Make a contour plot with arrows indicating the direction of updates for all 2-d functions.

Test case	Backtracking	Bisection	Pure	Damped	Levenberg-Marquardt	Combined
0	[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]
1	[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]	[2. 2.]
2	[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]	[-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]	[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]
4	[0. 0.]	[1.748 -0.874]	[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]	[-1.748 0.874]
5	[-0. -0.]	[-1.748 0.874]	[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]	[1.748 -0.874]
6	[1. 1. 1. 0.999]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]
7	[1. 1. 1. 0.999]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]
8	[1. 1. 1. 0.999]	[1. 1. 1. 1.]	[-0.776 0.613 0.382 0.146]	[-0.776 0.613 0.382 0.146]	[-0.776 0.613 0.382 0.146]	[-0.776 0.613 0.382 0.146]
9	[1. 1. 1. 0.999]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]	[1. 1. 1. 1.]
10	[-2.904 -2.904 -2.904 -2.904]	[-2.904 -2.904 -2.904 -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]
11	[-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]	[2.747 2.747 2.747 2.747]
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]	[-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]	[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]	[2.747 -2.904 2.747 -2.904]
14	[0. 0.]	[0. 0.]	[-27. -27.]	[-0. -0.]	[-27. -27.]	[-0. -0.]
15	[0. -0.]	[-0. 0.]	[0. -0.]	[0. -0.]	[0. -0.]	[0. -0.]
16	[0. -0.]	[0. -0.]	[-7.8815639e+04 2.0000000e-03]	[-0. 0.]	[-7.8815639e+04 2.0000000e-03]	[-0. 0.]

- The aforementioned snapshot is the terminal output after running all six algorithms for all 17 test cases.
- We have computed the minima for different functions using the general equations computed in the last question.

- Trid Function, for **d** = 2, minima = (2,2) (explanation in supplementary)
- Three Hump Camel function, for **d** = 2, minima = (0,0), $(-1.7468, 0.8734)$, $(1.7468, -0.8734)$ (explanation in supplementary)
- For Rosenbrock function, for **d**=4, minima = (1, 1, 1, 1)
- Styblinski-Tang function, for **d** = 4, minima $-2.667 > x_i > 2.667$
- Root of Square function, for **d** = 2, minima = (0,0)

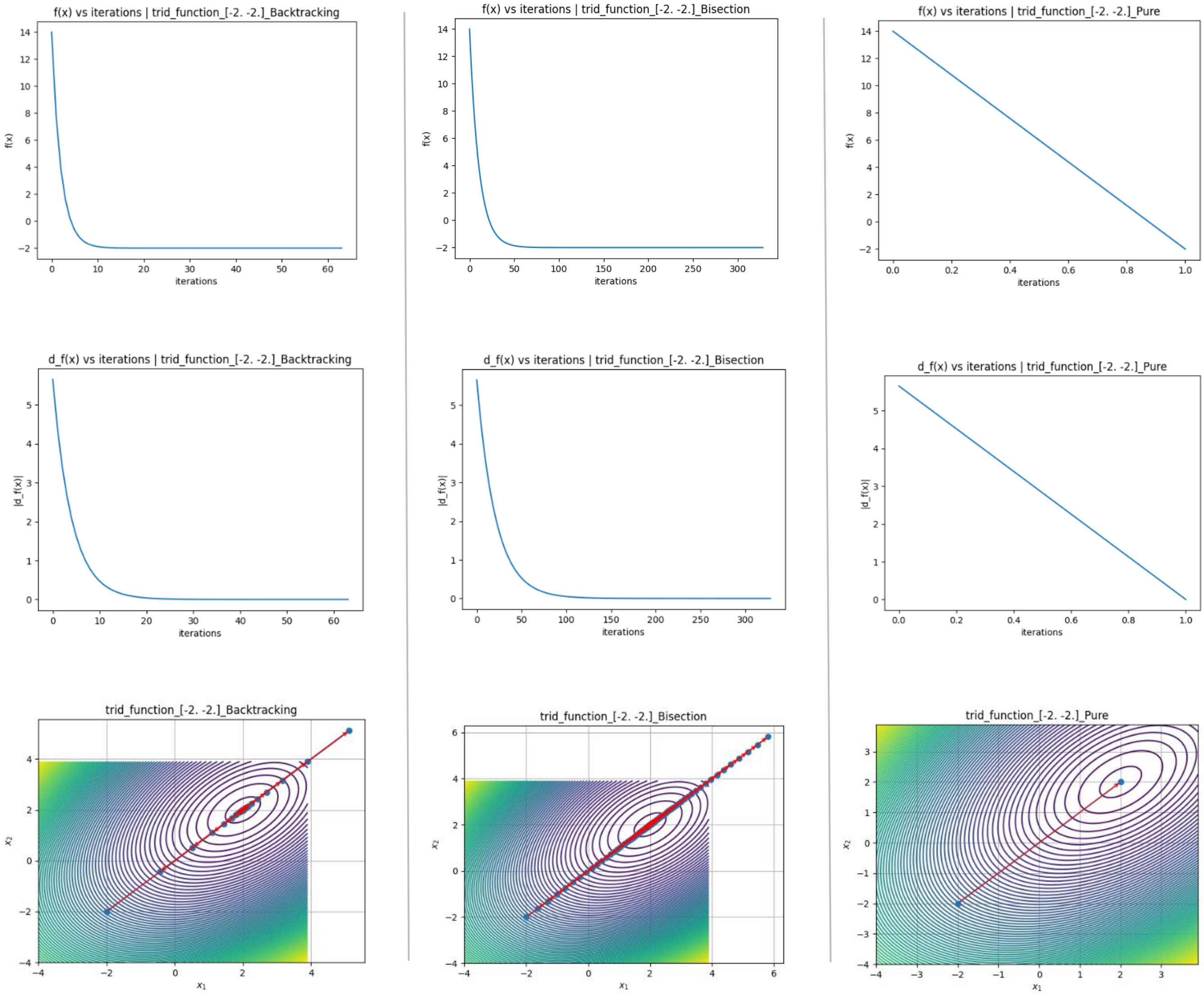
- Comparing the computed minima values with the values in the table, we can conclude that the following functions fail to converge:

Test Case	Function name	Condition	Initial Point
8	Rosenbrock Function	Pure Newton	[-2.0, 2, 2, 2]
8	Rosenbrock Function	Damped Newton	[-2.0, 2, 2, 2]
8	Rosenbrock Function	Levenberg-Marquardt	[-2.0, 2, 2, 2]
8	Rosenbrock Function	Combined	[-2.0, 2, 2, 2]
10	Styblinski-Tang function	Pure Newton	[0, 0, 0, 0]
10	Styblinski-Tang function	Damped Newton	[0, 0, 0, 0]
14	Root of Square function	Pure Newton	[3.0, 3]
14	Root of Square function	Levenberg-Marquardt	[3.0, 3]
16	Root of Square function	Pure Newton	[-3.5, 0.5]
16	Root of Square function	Levenberg-Marquardt	[-3.5, 0.5]

Note: Plots of only some test cases are included in the report. Plots for rest of the test cases are stored in the plots directory.

For test case 0 – 1:

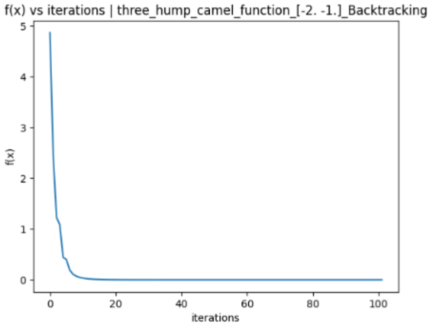
- From the terminal output, we can observe that the six algorithms (Backtracking, Bisection, Pure-Newton, Damped-Newton, Levenberg-Marquardt, and Combined) converges to minima (2, 2) for Trid function initialized at two different values.

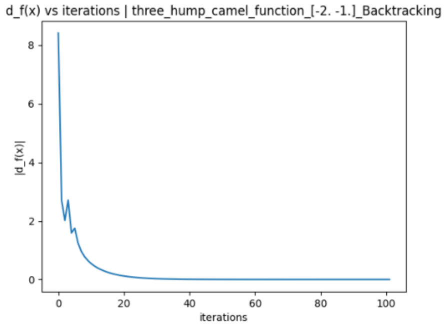


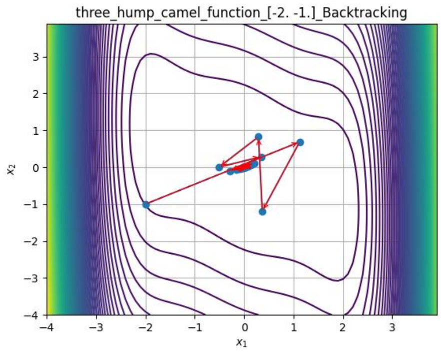
For test case 2 – 5:

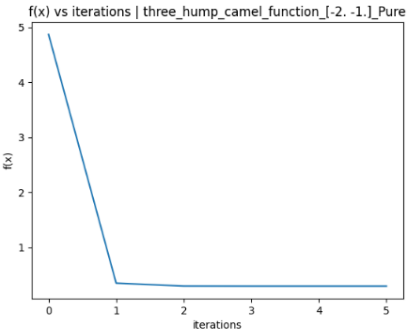
- From the terminal output, we can observe that the six algorithms (Backtracking, Bisection, Pure-Newton, Damped-Newton, Levenberg-Marquardt, and Combined) converges to minima for Three Hump Camel function initialized at four different initial values.

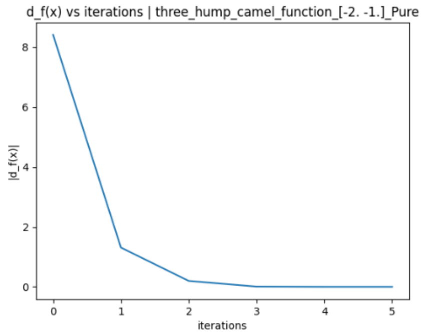
Including the graphs of all 6 functions for one of the test cases of Three hump camel function

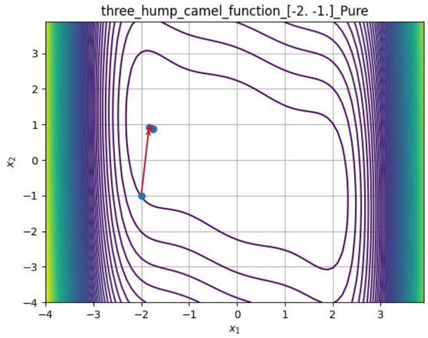

f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Backtracking
f(x)
iterations

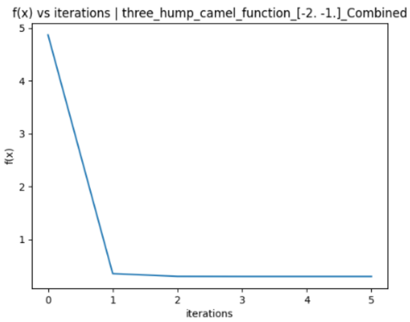

d_f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Backtracking
|d_f(x)|
iterations

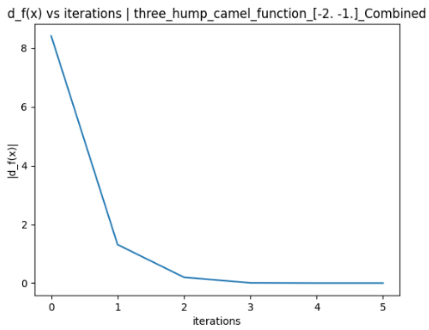

three_hump_camel_function_[-2. -1.]_Backtracking
x_1
x_2

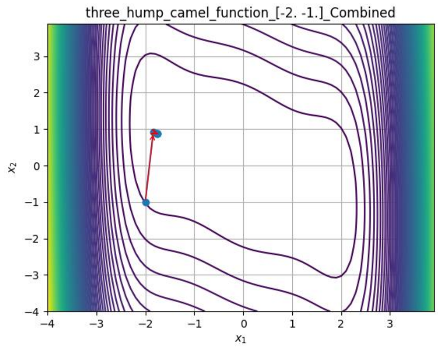

f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Pure
f(x)
iterations

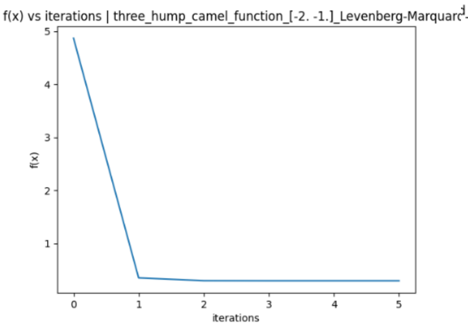

d_f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Pure
|d_f(x)|
iterations

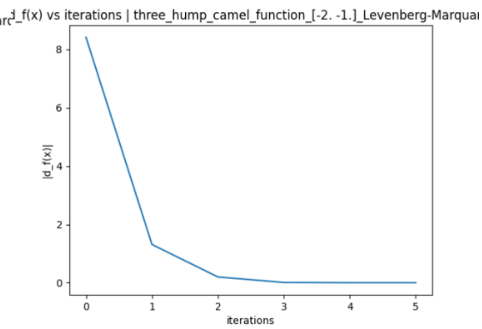

three_hump_camel_function_[-2. -1.]_Pure
x_1
x_2

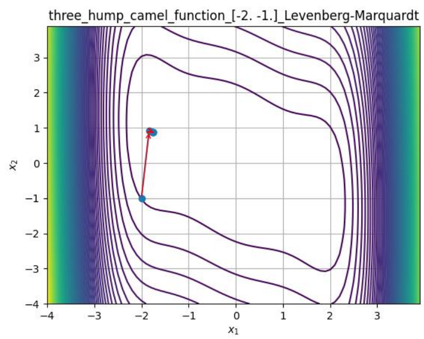

f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Combined
f(x)
iterations

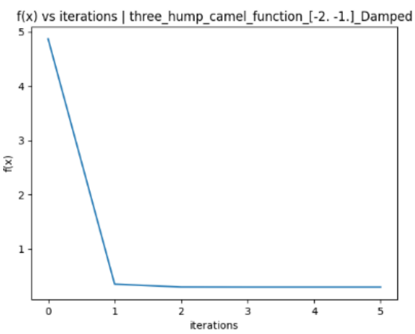

d_f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Combined
|d_f(x)|
iterations

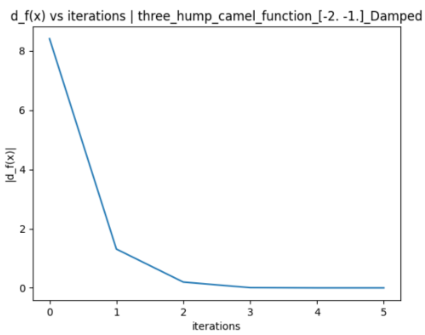

three_hump_camel_function_[-2. -1.]_Combined
x_1
x_2

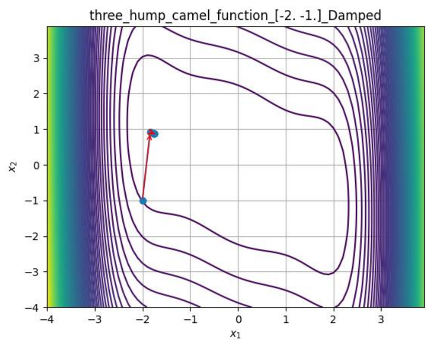

f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Levenberg-Marquarc
f(x)
iterations

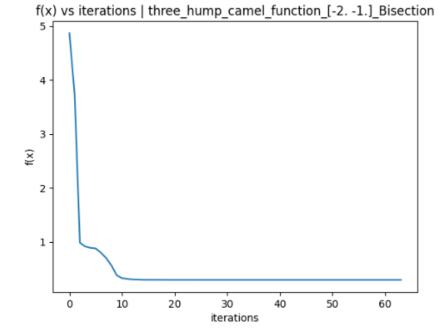

_f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Levenberg-Marquai
|d_f(x)|
iterations

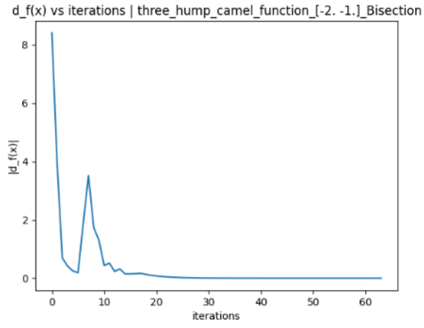

three_hump_camel_function_[-2. -1.]_Levenberg-Marquardt
x_1
x_2

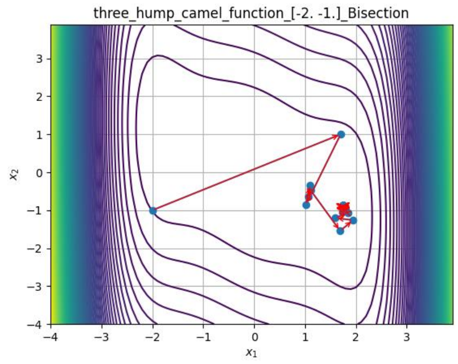

f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Damped
f(x)
iterations


d_f(x) vs iterations | three_hump_camel_function_[-2. -1.]_Damped
|d_f(x)|
iterations


three_hump_camel_function_[-2. -1.]_Damped
x_1
x_2







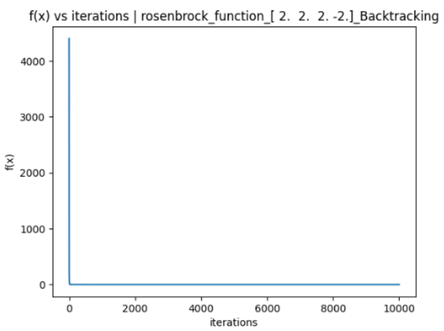
For test case 6 – 9:

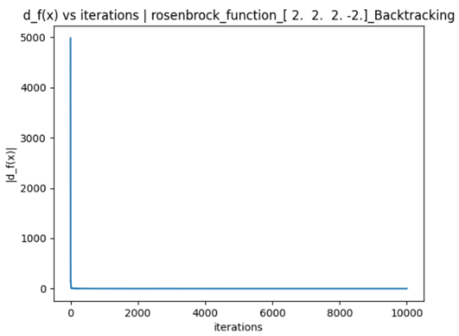
- From the terminal output, we can observe that the two algorithms (Backtracking, Bisection) converges to minima for Rosenbrock for all 4 test cases. However, the rest of the 4 alogorithms (Pure-Newton, Damped-Newton, Levenberg-Marquardt, and Combined) converges to minima for Rosenbrock, only when initialized at points **[2.0, 2, 2, -2]** (test case 6), **[2.0, -2, -2, 2]** (test case 7), **[3.0, 3, 3, 3]** (test case 9), and fail to converge when initialized to **[-2.0, 2, 2, 2]** (test case 8).

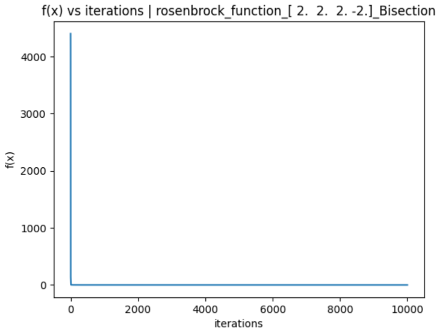
Including the graphs of test case 6 (where all algorithms converge)

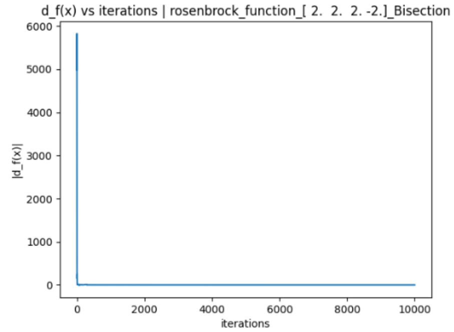
and test case 8 where 2 algorithms converge and 4 algorithms fail to converge

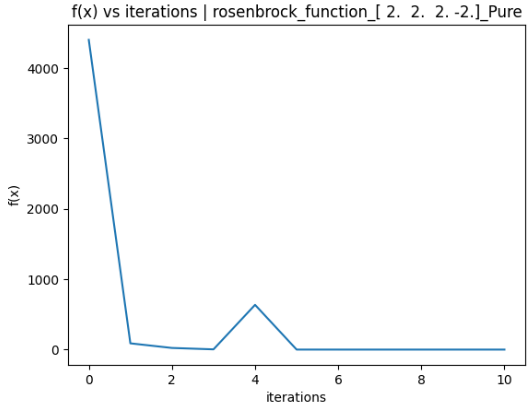
Test case 6

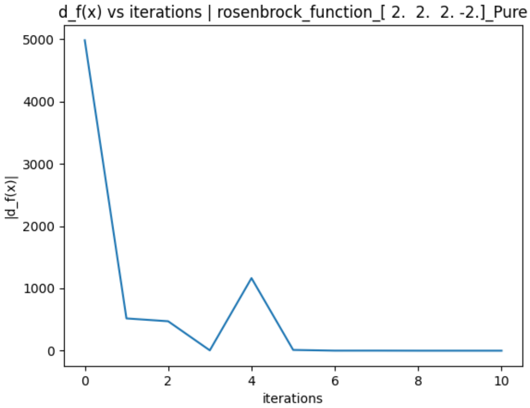


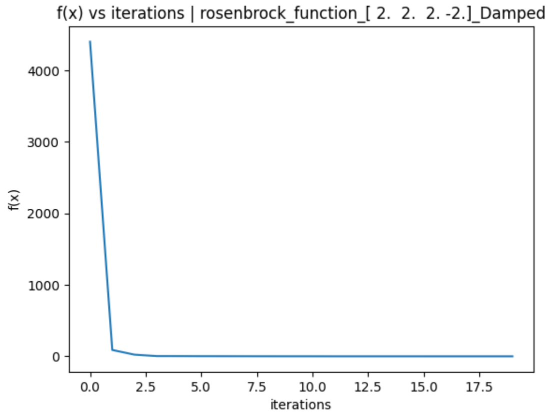


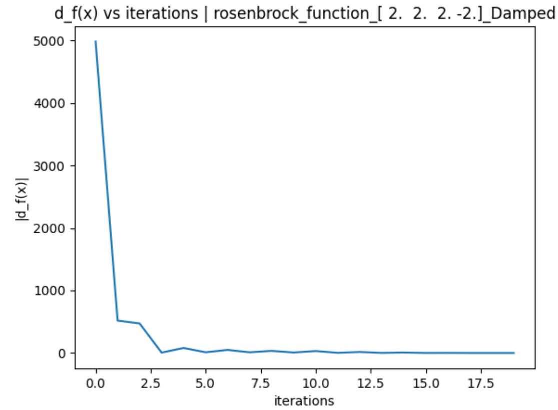


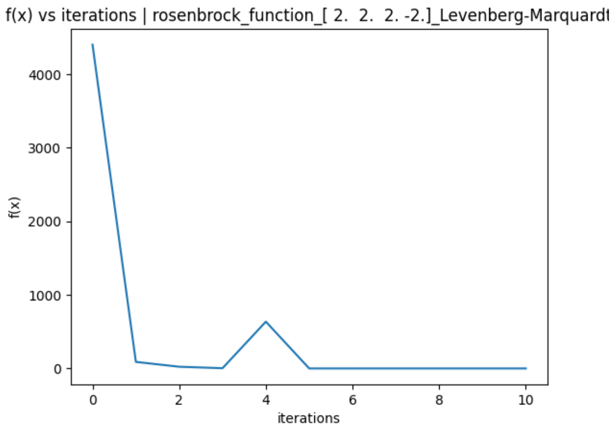


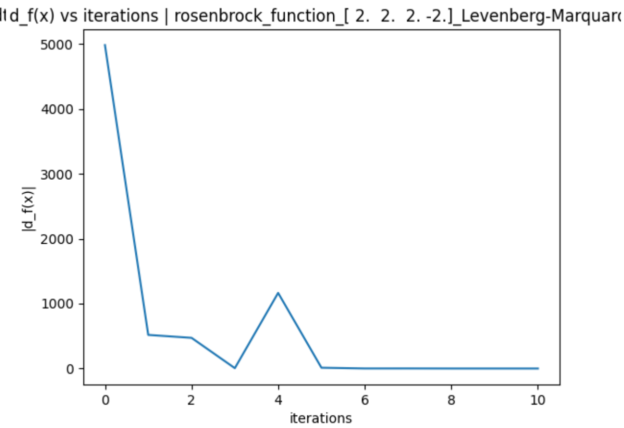

f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Pure
f(x)
iterations
0 1000 2000 3000 4000
0 2 4 6 8 10

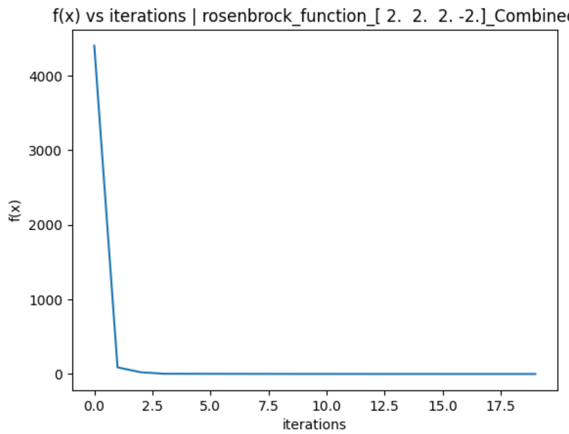

d_f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Pure
|d_f(x)|
iterations
0 1000 2000 3000 4000 5000
0 2 4 6 8 10

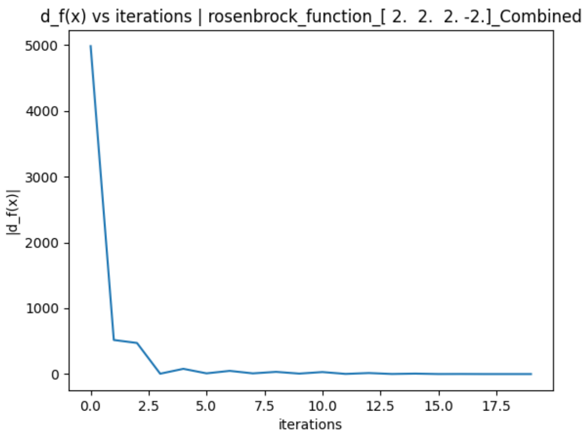

f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Damped
f(x)
iterations
0 1000 2000 3000 4000
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5


d_f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Damped
|d_f(x)|
iterations
0 1000 2000 3000 4000 5000
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5

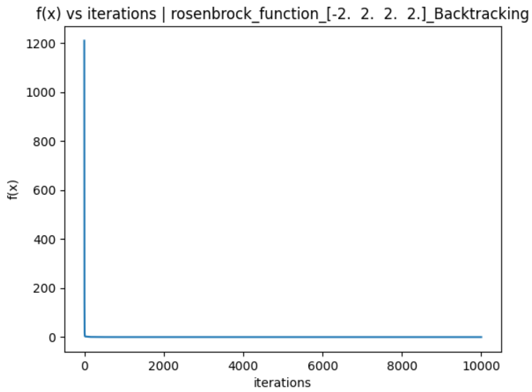

f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Levenberg-Marquardt
f(x)
iterations
0 1000 2000 3000 4000
0 2 4 6 8 10

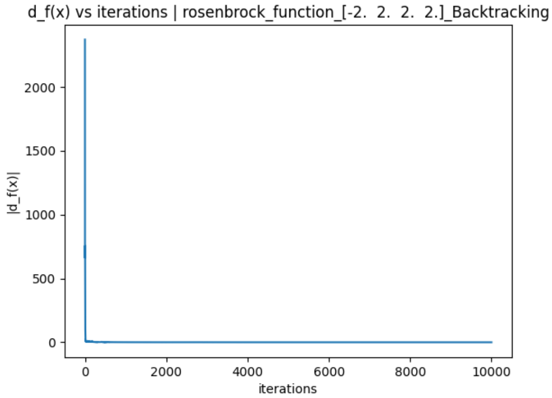

d_f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Levenberg-Marquar
|d_f(x)|
iterations
0 1000 2000 3000 4000 5000
0 2 4 6 8 10

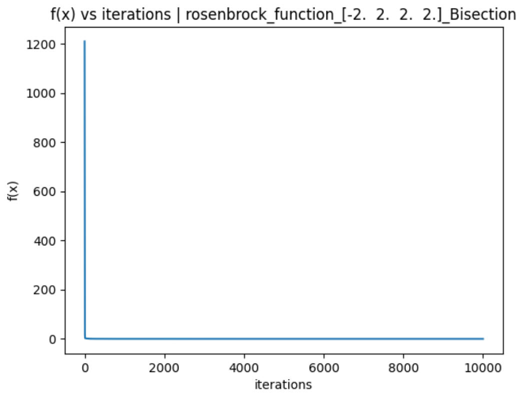

f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Combined
f(x)
iterations
0 1000 2000 3000 4000
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5

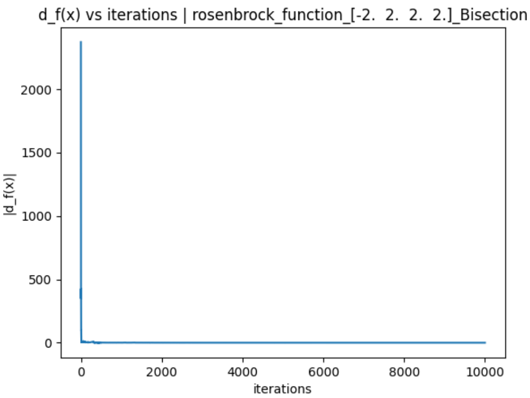

d_f(x) vs iterations | rosenbrock_function_[2. 2. 2. -2.]_Combined
|d_f(x)|
iterations
0 1000 2000 3000 4000 5000
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5

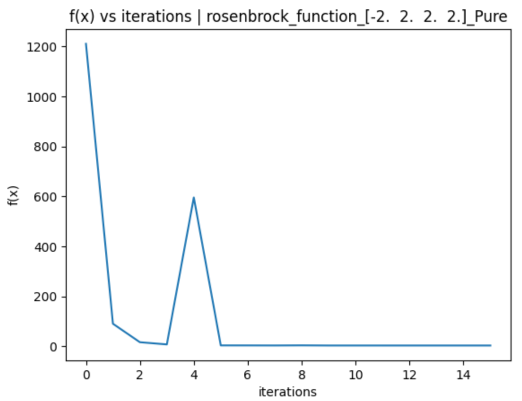
Test case 8

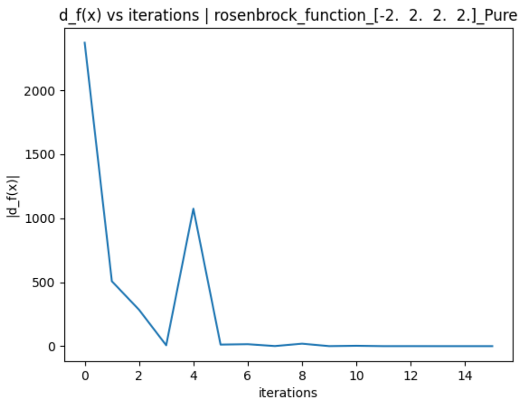

f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Backtracking
f(x)
1200
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000
iterations

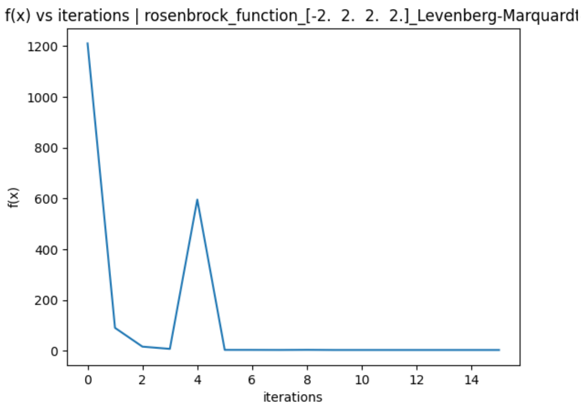

d_f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Backtracking
|d_f(x)|
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
iterations

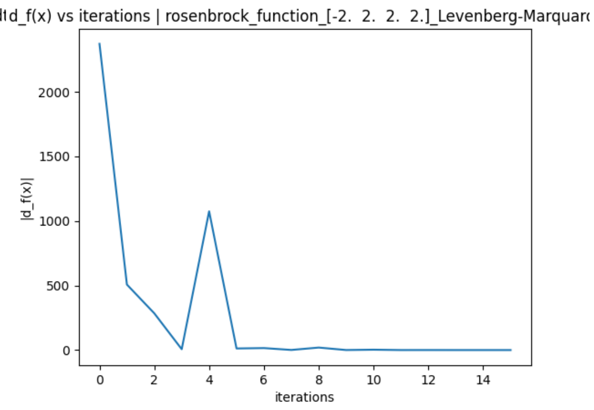

f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Bisection
f(x)
1200
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000
iterations

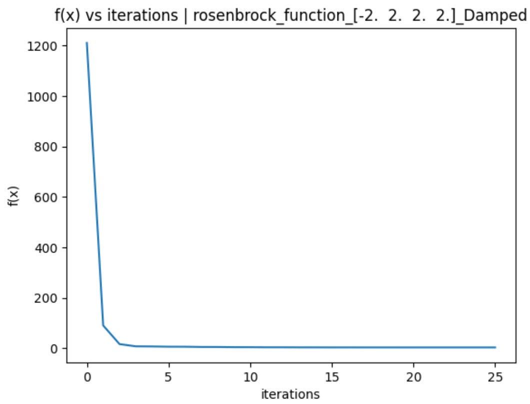

d_f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Bisection
|d_f(x)|
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
iterations

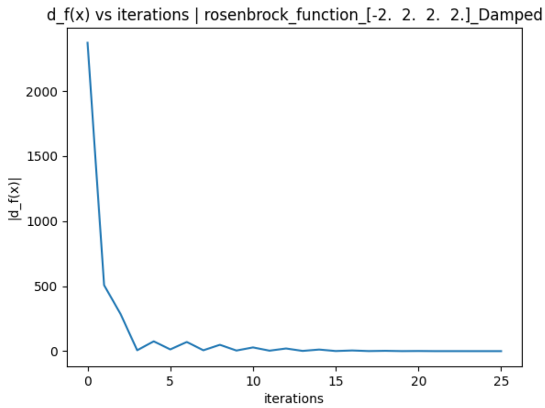

f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Pure
f(x)
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
14
iterations

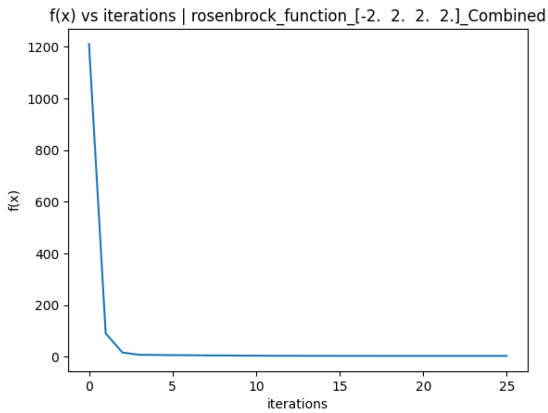

d_f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Pure
|d_f(x)|
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
iterations

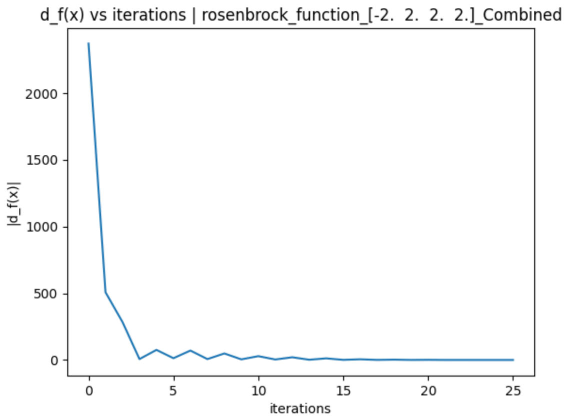

f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Levenberg-Marquardt
f(x)
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
14
iterations


d_f(x) vs iterations | rosenbrock_function_[-2. 2. 2. 2.]_Levenberg-Marquar
|d_f(x)|
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
iterations









- In the test case 8, Pure Newton, Damped Newton, Levenberg-Marquardt, and Combined failed to converge because the gradient in the direction of Hessian inverse approaches zero, satisfying the stopping condition before reaching the minima.

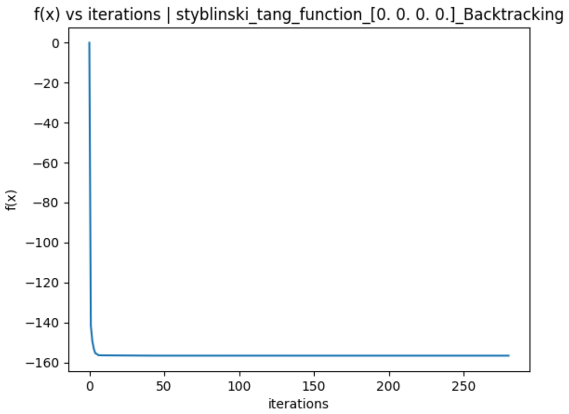
For test case 10 – 13:

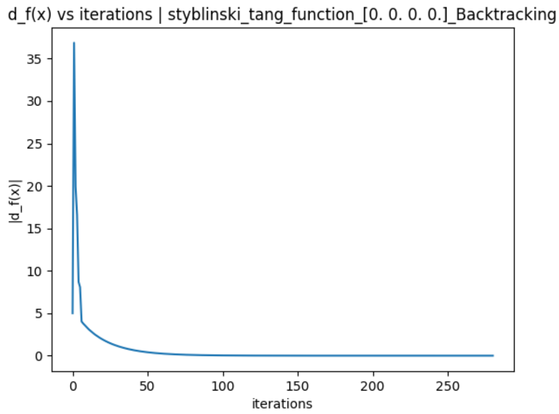
- From the terminal output, we can observe for Styblinski Tang function, Pure Newton method and Damped Newton method fail to converge for test case 10, where the initial point is **[0, 0, 0, 0]** because $d_k = -(Hessian^{-1} \times Gradient)$, evaluated at the initial point turns out not to be a descent direction, resulting in the algorithms to not converge.

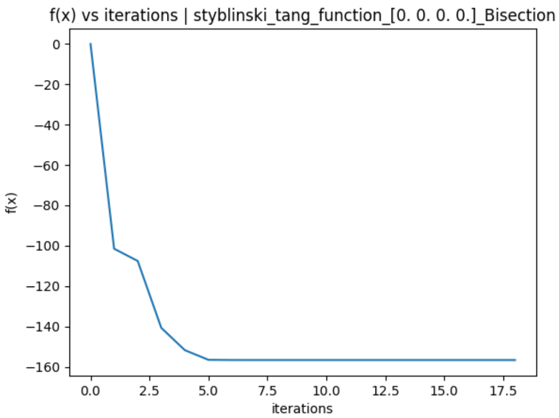
Including the graphs of test case 10

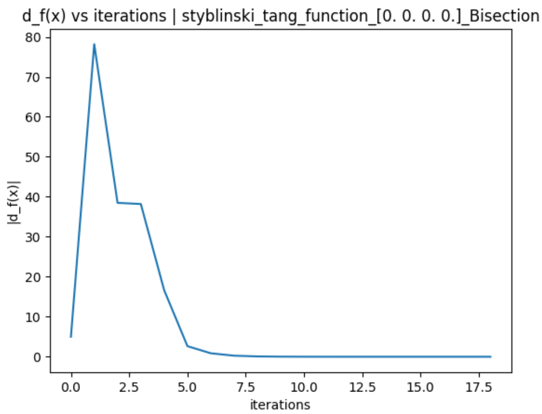
(where all algorithms converge except Pure Newton method and Damped Newton)

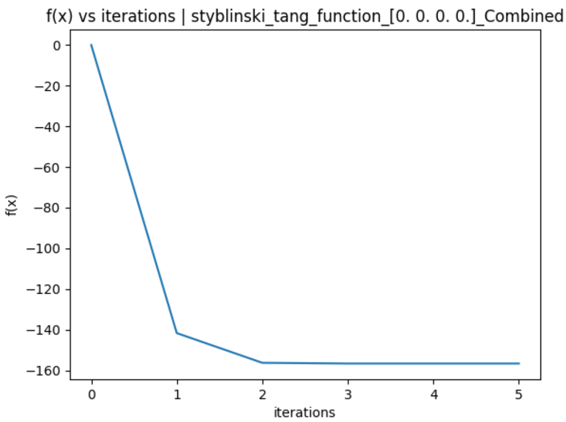
Test case 10

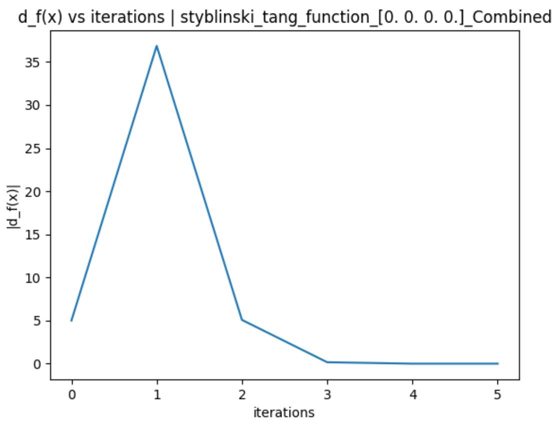

f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Backtracking
f(x)
iterations

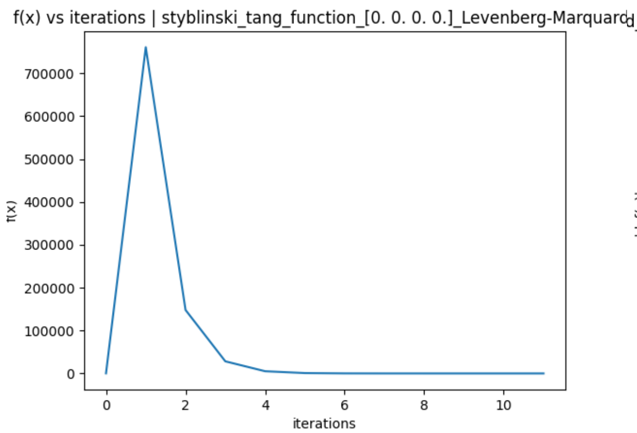

d_f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Backtracking
|d_f(x)|
iterations

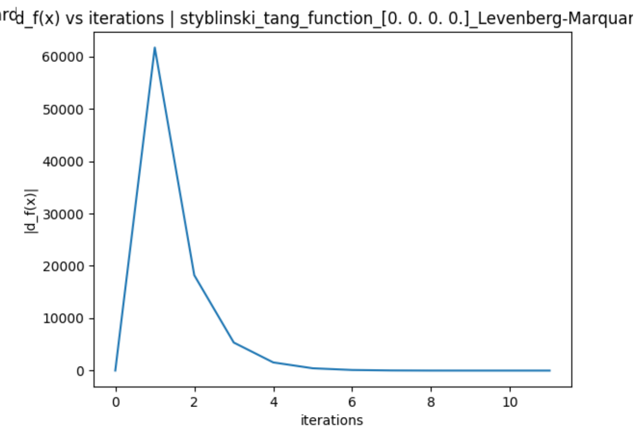

f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Bisection
f(x)
iterations

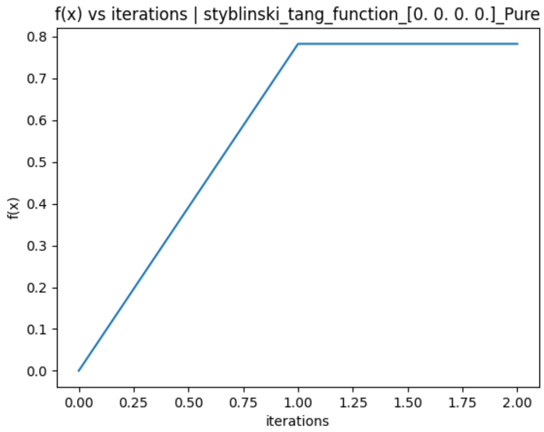

d_f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Bisection
|d_f(x)|
iterations

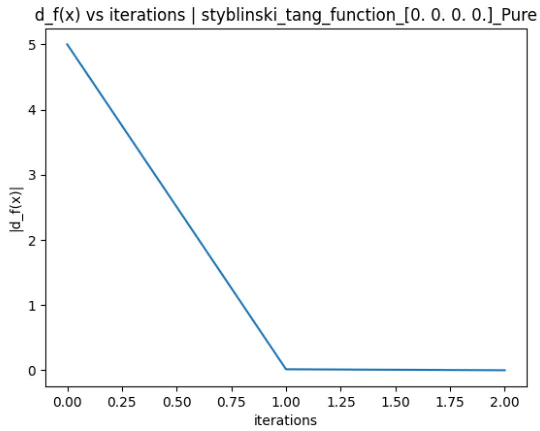

f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Combined
f(x)
iterations

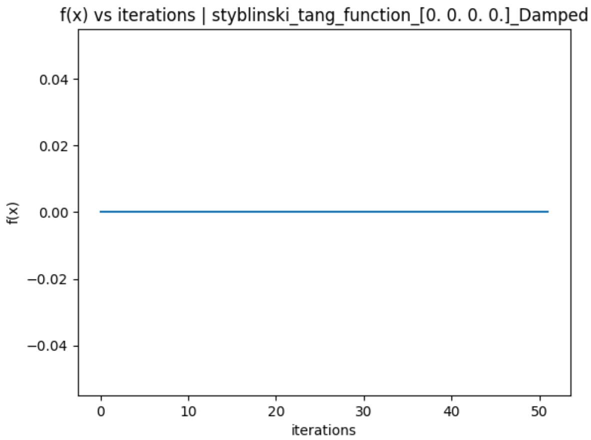

d_f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Combined
|d_f(x)|
iterations

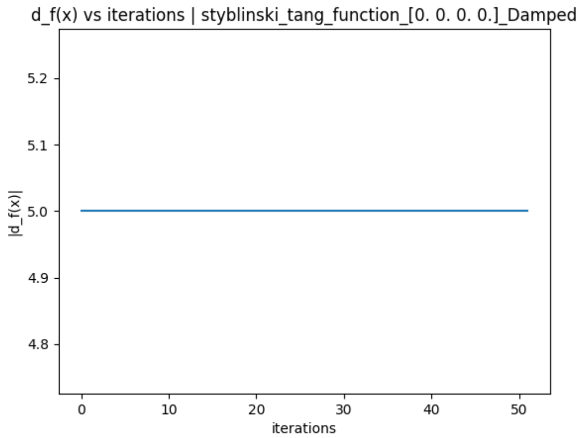

f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Levenberg-Marquard
f(x)
iterations


d_f(x) vs iterations | styblinski_tang_function_[0. 0. 0. 0.]_Levenberg-Marquar
|d_f(x)|
iterations







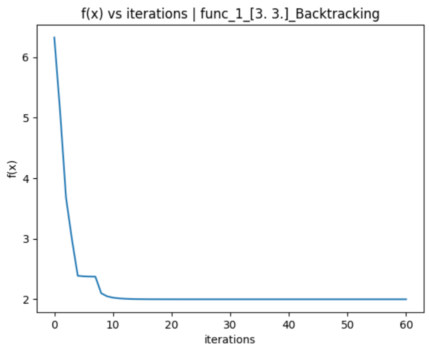


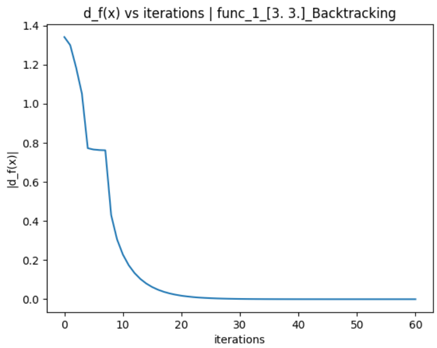
For test case 14 – 16:

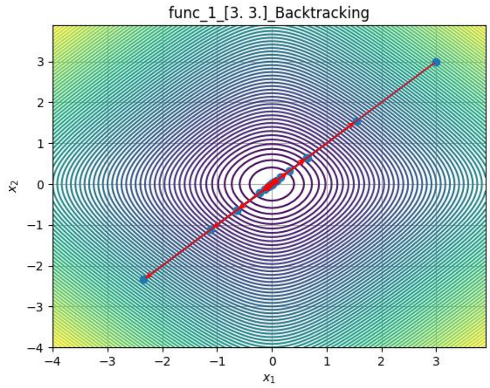
- From the terminal output, we can observe for Root of Square function, Pure Newton method and Levenberg Marquardt Newton method fail to converge for test case 14 and 16, where the initial points are **[3.0, 3]** and **[-3.5, 0.5]** respectively, because the determinant of the Hessian is tending to zero so the inverse of the Hessian gradually tends to a very very large value, leading to the diverging updates.

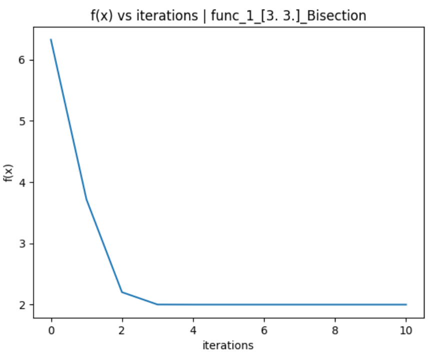
Including all the graphs of test case 14 and failure case graphs of test case 16

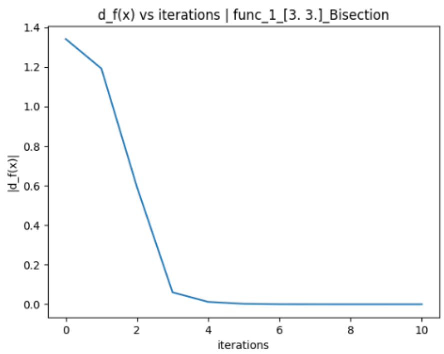
Test case 14

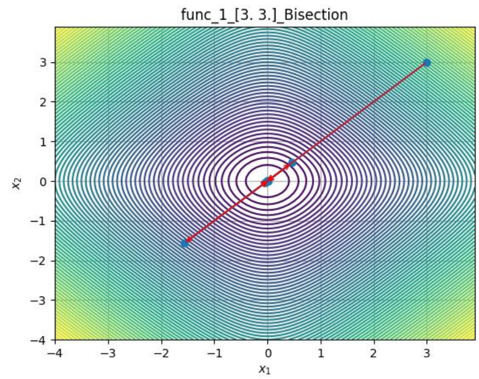

f(x) vs iterations | func_1_[3. 3.]_Backtracking
f(x)
iterations

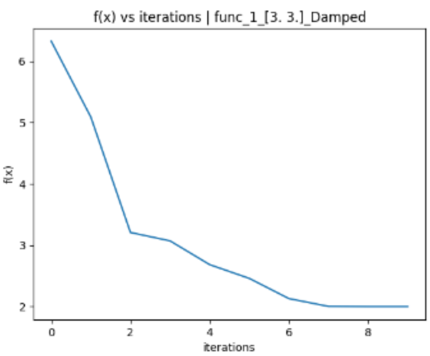

d_f(x) vs iterations | func_1_[3. 3.]_Backtracking
|d_f(x)|
iterations

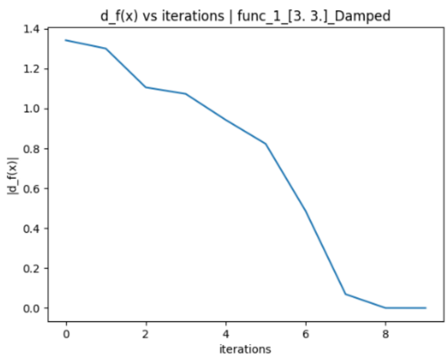

func_1_[3. 3.]_Backtracking
x_2
x_1

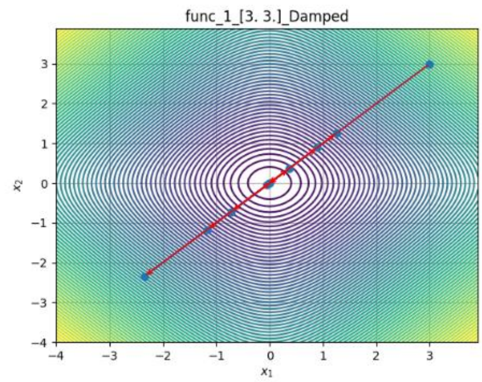

f(x) vs iterations | func_1_[3. 3.]_Bisection
f(x)
iterations

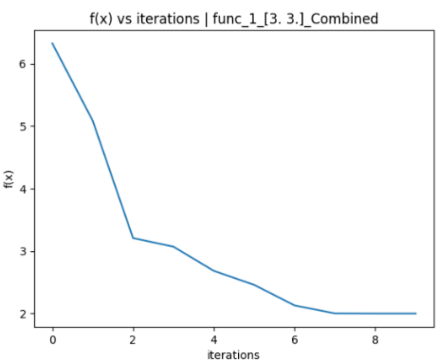

d_f(x) vs iterations | func_1_[3. 3.]_Bisection
|d_f(x)|
iterations

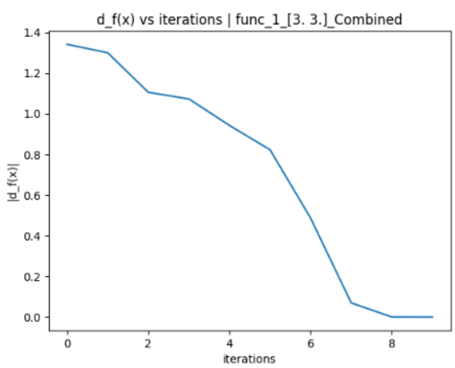

func_1_[3. 3.]_Bisection
x_2
x_1

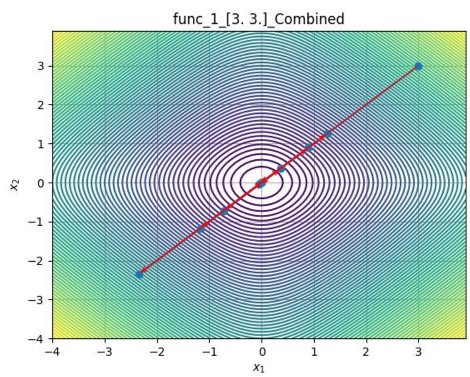

f(x) vs iterations | func_1_[3. 3.]_Damped
f(x)
iterations

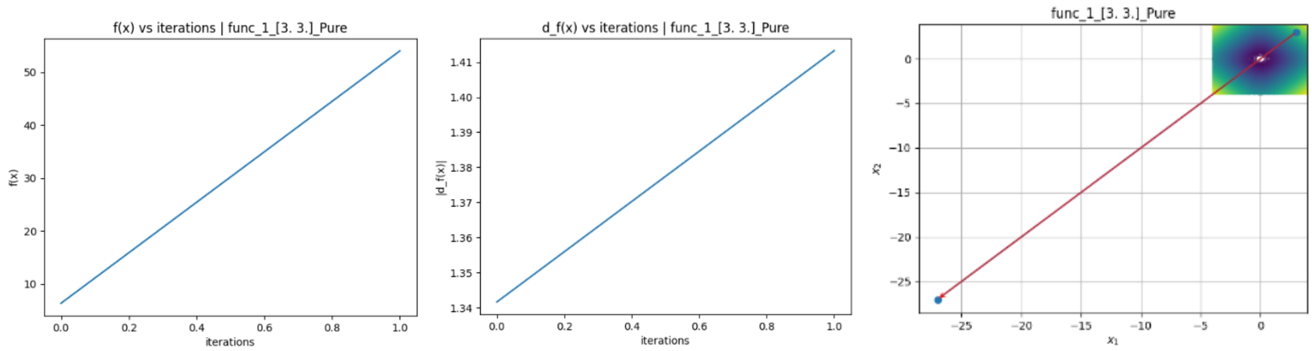

d_f(x) vs iterations | func_1_[3. 3.]_Damped
|d_f(x)|
iterations

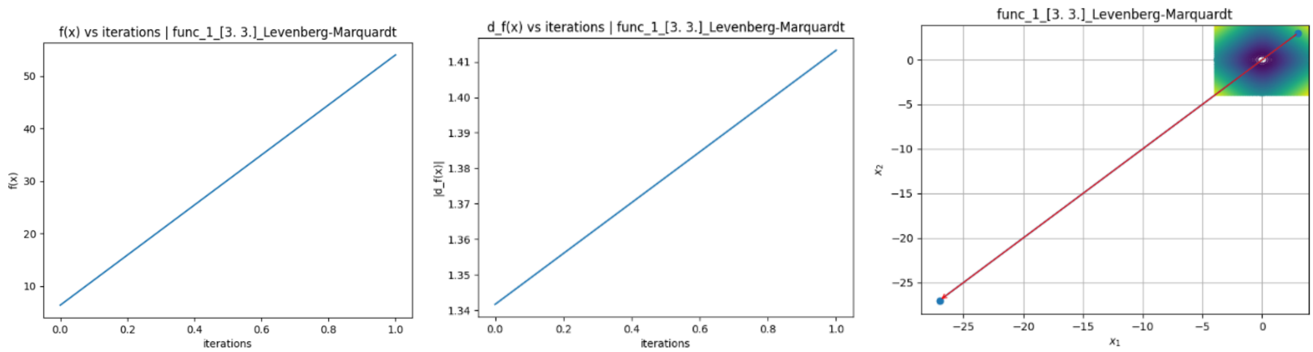

func_1_[3. 3.]_Damped
x_2
x_1


f(x) vs iterations | func_1_[3. 3.]_Combined
f(x)
iterations

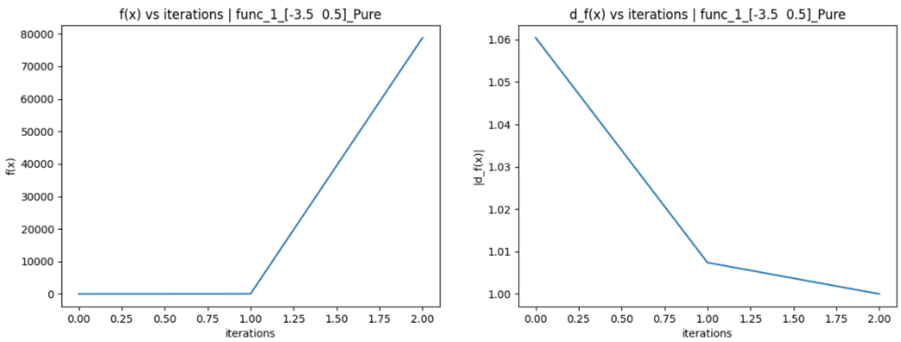

d_f(x) vs iterations | func_1_[3. 3.]_Combined
|d_f(x)|
iterations

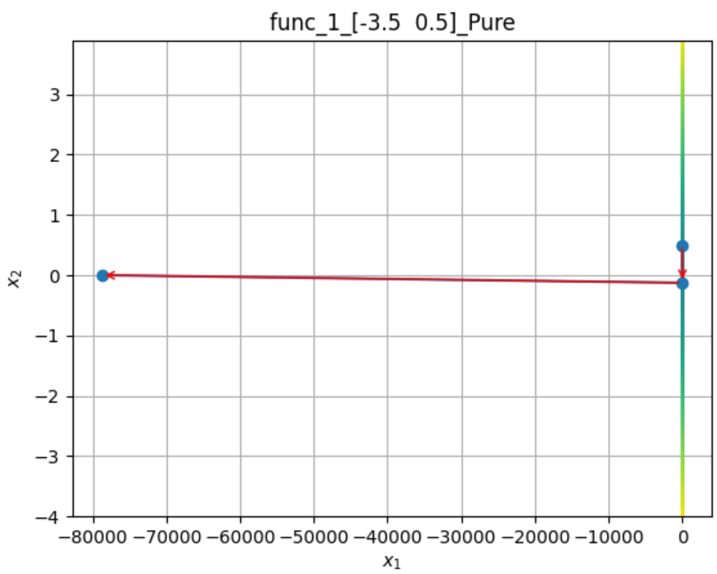

func_1_[3. 3.]_Combined
x_2
x_1

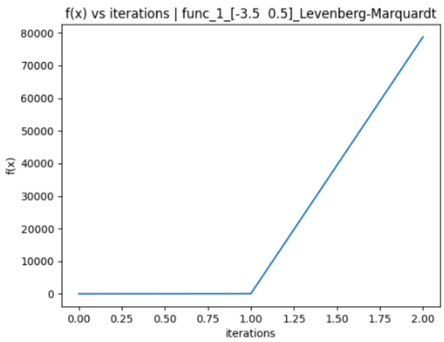

f(x) vs iterations | func_1_[3. 3.]_Pure
f(x)
iterations
d_f(x) vs iterations | func_1_[3. 3.]_Pure
|d_f(x)|
iterations
func_1_[3. 3.]_Pure
x_2
x_1

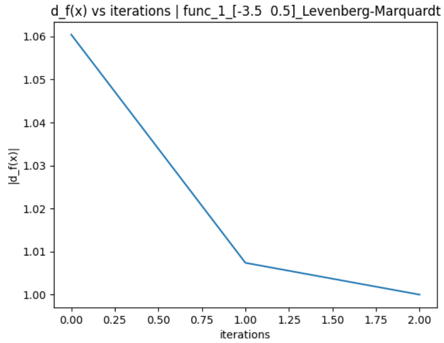

f(x) vs iterations | func_1_[3. 3.]_Levenberg-Marquardt
f(x)
iterations
d_f(x) vs iterations | func_1_[3. 3.]_Levenberg-Marquardt
|d_f(x)|
iterations
func_1_[3. 3.]_Levenberg-Marquardt
x_2
x_1

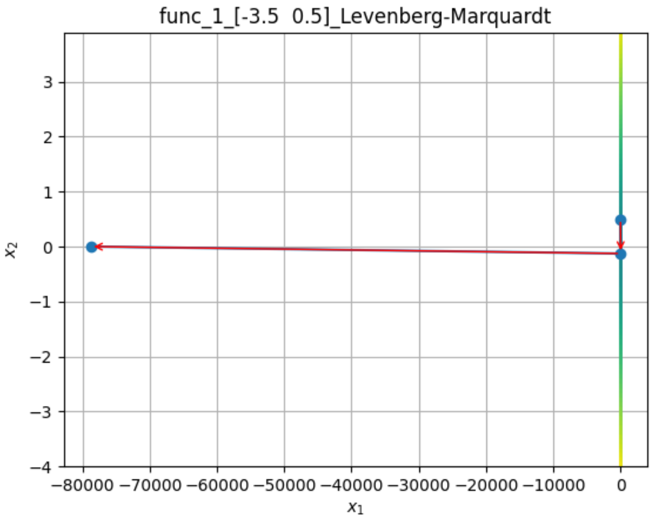
Test case 16


f(x) vs iterations | func_1_[-3.5 0.5]_Pure
f(x)
iterations
d_f(x) vs iterations | func_1_[-3.5 0.5]_Pure
|d_f(x)|
iterations


func_1_[-3.5 0.5]_Pure
x_2
x_1


f(x) vs iterations | func_1_[-3.5 0.5]_Levenberg-Marquardt
f(x)
iterations


d_f(x) vs iterations | func_1_[-3.5 0.5]_Levenberg-Marquardt
|d_f(x)|
iterations


func_1_[-3.5 0.5]_Levenberg-Marquardt
x_2
x_1

6. Supplementary

Trid Function

- Stationary points for **d**=2:

$$\nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \\ \vdots \\ \partial f(\bar{x})/\partial x_d \end{bmatrix} = \begin{bmatrix} 2(x_1 - 1) - x_2 \\ 2(x_2 - 1) - x_1 - x_3 \\ \vdots \\ 2(x_d - 1) - x_{d-1} - x_{d+1} \end{bmatrix} = \mathbf{0}$$

We can find roots by equating system of equations to 0 (for d = 2):

- $2(x_1 - 1) - x_2 = 0$
  - $x_1 = \frac{x_2+2}{2} = \frac{x_2}{2} + 1$ . . . (1)
  - $x_2 = 2x_1 - 2$ . . . (2)

- $2(x_2 - 1) - x_1 - x_3 = 0 \Rightarrow 2x_2 - x_1 - x_3 = 2 \Rightarrow 2x_2 - x_1 = 2$ (for d = 2, $x_3 = 0$)
  - Substituting (1) in above:
    - $\Rightarrow 2x_2 = x_1 + 2 = \frac{x_2+2}{2} + 2$
    - $x_2 = 2$ . . . (3)
  - Substituting (3) in (2):
    - $x_1 = 2$

For d =2, the stationary points are (2,2):

- Validating minima by analysing Hessian:

$$H = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Hessian is positive definite. Hence, the stationary point (2,2) is a minima.

Three Hump Camel Function

- Stationary points:

$$\nabla f(\bar{x}) = \begin{bmatrix} \partial f(\bar{x})/\partial x_1 \\ \partial f(\bar{x})/\partial x_2 \end{bmatrix} = \begin{bmatrix} 4x_1 - 4.2{x_1}^3 + {x_1}^5 + x_2 \\ x_1 + 2x_2 \end{bmatrix} = \mathbf{0}$$

- We can find roots by equating system of equations to 0.
  - $4x_1 - 4.2{x_1}^3 + {x_1}^5 + x_2 = 0$ . . . (1)
  - $x_1 + 2x_2 = 0 \Rightarrow x_1 = -2x_2$ . . . (2)

- Substituting (2) in (1):
  - $4(-2x_2) - 4.2(-2x_2)^3 + (-2x_2)^5 + x_2 = 0$
  - $\Rightarrow -7x_2 + 33.6{x_2}^3 - 32{x_2}^5 = 0$
  - $\Rightarrow x_2(-7 + 33.6{x_2}^2 - 32{x_2}^4) = 0$ ... (3)
  - $x_2 = 0$ (root) ... (4)
  - Plugging (4) in (2):
  - $x_1 = 0$
  - First pair of stationary points (0,0)
- Solving the other part of eq. (3), i.e., $(-7 + 33.6{x_2}^2 - 32{x_2}^4) = 0$
  - Substituting $y = {x_2}^2$ in the above equation:
  - $-32y^2 + 33.6y - 7 = 0$ ... (5)
  - $y = 0.286$ and $y = 0.763$ ... (6)
  - We know $y = {x_2}^2 \Rightarrow x_2 = \pm\sqrt{y}$ ... (7)
  - Plugging the values from eq. (6) to eq. (7):
  - $x_2 = \pm 0.5347$ and $x_2 = \pm 0.8734$ ... (8)
  - Plugging (8) in (2):
  - For $x_2 = 0.5347,\ x_1 = -1.0694$
  - For $x_2 = -0.5347,\ x_1 = 1.0694$
  - For $x_2 = 0.8734,\ x_1 = -1.7468$
  - For $x_2 = -0.8734,\ x_1 = 1.7468$

- Stationary points in the form of (x_1, x_2) are:
  - $(0,0),\ (-1.0694, 0.5347),\ (1.0694, -0.5347),\ (-1.7468,\ 0.8734),\ (1.7468, -0.8734)$
- Validating minima by plugging values of stationary points in the Hessian:
- $H = \begin{bmatrix} 4 + 5{x_1}^4 - 12.6{x_1}^2 & 1 \\ 1 & 2 \end{bmatrix}$
- Plugging the stationary points in the above Hessian matrix, we find that Hessian is positive definite for the stationary points: $(0,0),\ (-1.7468, 0.8734),\ (1.7468, -0.8734)$
- Therefore, the minima for this function are: $(0,0),\ (-1.7468, 0.8734),\ (1.7468, -0.8734)$