CMO - Assignment - 1

Nirbhay sharma
SR: 24806
MTech - AI

September 13, 2024

1 Question 1 - Convex and Coercive function

1.1 Convex and strict convex

To test the convexity of f_1, f_2, we simply follow the definition of convex function as follows

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

$\forall x_1, x_2 \in [-2, 2]$ and $\forall \lambda \in [0, 1]$
For strict convex the inequality becomes strict as follows

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda)f(x_2)$$

We implement the above in the function and the output is reported in the next part

1.2 Convex, strict convex, x*, f(x*)

Based on the above algorithm, f_1, f_2 are convex, however, only f_2 is strict convex and f_1 is not strict convex

algorithm to find x^*: we find $x^*, f(x^*)$ using gradient descent algorithm with constant alpha, the algorithm is as follows

start with an $x = x^0$ then update x as $x^1 = x^0 - \alpha f'(x)$, the algorithm indeed converge and we get $x^*, f(x^*)$ for both f_1, f_2 as shown in Fig. 1

Uniqueness of x^*: No, the x^* is not unique, as we can start from another point and land at different x^* which has same value of $f(x^*)$

```
$ python que1.py
convex, f1: True, f2: True
strict convex: f1: False, f2: True
f1: x* = 0.4, f(x*) = 0.008
f2: x* = -5.299, f(x*) = 0.01
f3_roots: [-2.003, -0.406, 0.689, 2.417]
f3_minima: [-1.421, 1.799]
f3_maxima: [0.147]
```

Figure 1: convexity, strict convexity, $x^*, f(x^*)$ of f_1, f_2, stationary points, roots for f_3

1.3 Coercivity, stationary points and roots

checking coercivity: To check for coercivity we can do the following, $\lim_{||x|| \to \infty} f(x) = \infty$ i.e as $||x|| \to \infty \implies f(x) > M$

finding roots of f_3: To find roots, we can use the following algorithm, iterate over the x axis in some range, and try to find $x : |f(x)| < \epsilon$

finding local maxima: we call x^* a local maxima where $f(x^*) \geq f(x)\ \forall x \in B_\delta(x^*)$
finding local minima: we call x^* a local minima where $f(x^*) \leq f(x)\ \forall x \in B_\delta(x^*)$
algorithm: we implement the same algorithm, iterate over a range of x and for each x check its neighborhood for the above condition to declare it minima or maxima, The results are presented in Fig. 1

2 Question 2 - Gradient Decent

The function given is of the form

$$f(x) = \frac{1}{2}x^T Ax + b^T x : x, b \in R^5, A \in R^{5\times 5}$$

2.1 Constant Gradient descent

In this algorithm we update the x^k as follows with α_k as constant throughout

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x)$$

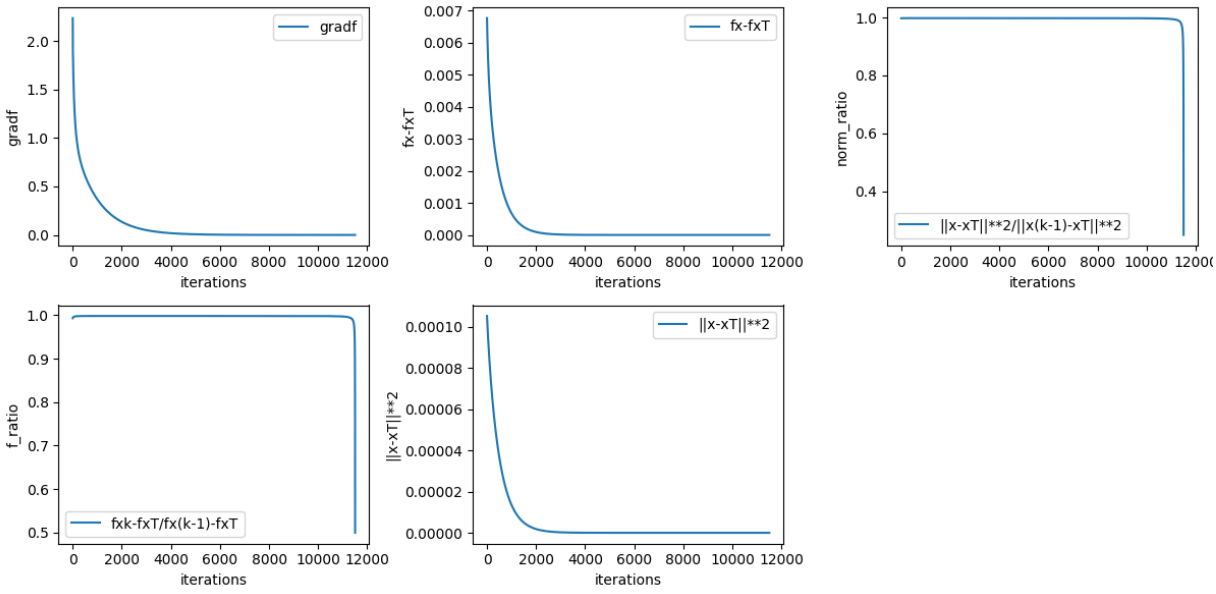

Figure 2: plots for Constant gradient descent

2.2 Diminishing Gradient descent

In this algorithm we update the x^k as follows with $\alpha_k = \frac{\alpha_0}{k+1}$ with each iteration

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x)$$

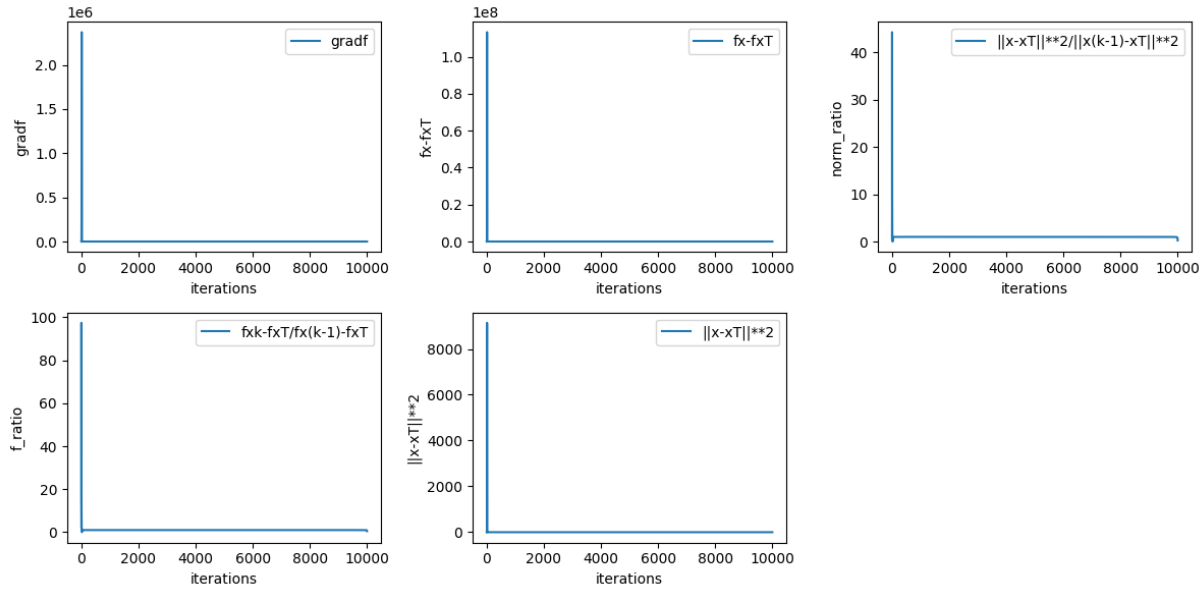

Figure 3: plots for Diminishing gradient descent

2.3 Inexact line search

In this algorithm we update the x^k as follows with α_k approximated using armijo-wolfe conditions

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x)$$

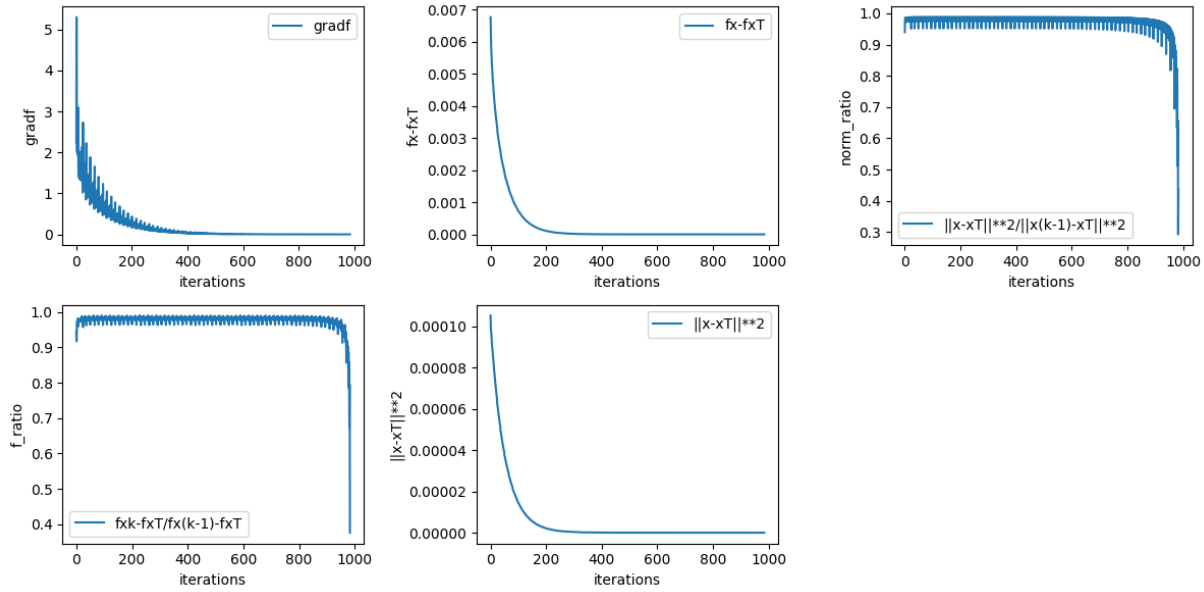

Figure 4: plots for Inexact Line search

2.4 Exact Line search

In this algorithm we update the x^k as follows with $\alpha_k = \frac{-\nabla f(x)^T p_k}{p_k^T A p_k} = \frac{||\nabla f(x)||^2}{\nabla f(x)^T A \nabla f(x)}$,

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x)$$

given the oracle, we find out $\nabla f(x)^T A \nabla f(x)$ as follows

$$f(x) = \frac{1}{2} x^T A x + b^T x$$

$$\nabla f(x) = Ax + b \implies \nabla f(0) = b$$

$$f(-\nabla f(x)) = \frac{1}{2} \nabla f(x)^T A \nabla f(x) - b^T \nabla f(x)$$

$$\nabla f(x)^T A \nabla f(x) = 2(f(-\nabla f(x)) + b^T \nabla f(x))$$

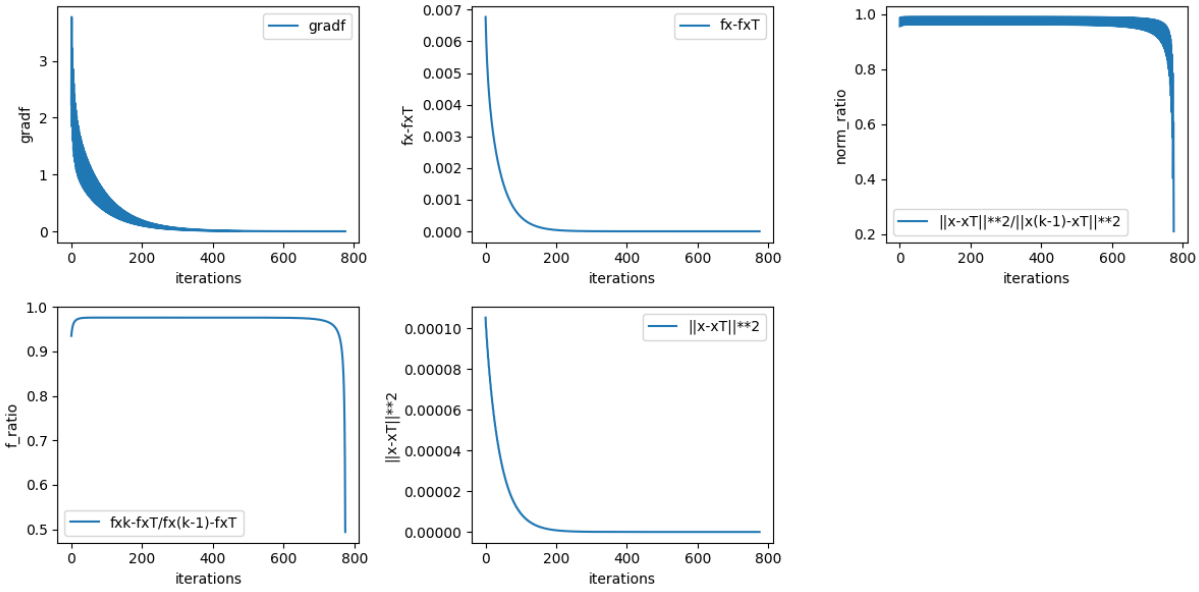

Figure 5: plots for Exact Line search

2.5 Results for Gradient Descent

The $x^*, f(x^*), T$ (number of iterations) are reported as the output of the code below in Fig. 6

```
$ python que2.py
ConstGradientDescent: x = [-4.03128275e-05 -5.00000000e-04 -1.00000000e-03 -2.00000000e-03
 -9.99990008e-03], f(x) = -0.006770156413271692, Iterations = 11509

DiminishingGradientDescent: x = [-4.03128275e-05 -5.00000000e-04 -1.00000000e-03 -1.98871635e-03
 -6.27461458e-03], f(x) = -0.006076199755554205, Iterations = 10000

InexactLineSearch: x = [-4.03146208e-05 -5.00000000e-04 -1.00000000e-03 -2.00000000e-03
 -9.99922787e-03], f(x) = -0.006770156383921725, Iterations = 986

ExactLineSearch: x = [-4.03110161e-05 -5.00000000e-04 -1.00000000e-03 -2.00000000e-03
 -9.99913118e-03], f(x) = -0.006770156375987994, Iterations = 777
```

Figure 6: $x^*, f(x^*), T$ (number of iterations)

The plots for $||\nabla f(x)||_2, f(x_k) - f(x_T), \frac{f(x_k)-f(x_T)}{f(x_{k-1})-f(x_T)}, ||x_k - x_T||_2^2, \frac{||x_k - x_T||_2^2}{||x_{k-1} - x_T||_2^2}$ for part 1, 2, 3, 4 are provided in Fig. 2, Fig. 3, Fig. 4, Fig. 5 respectively

2.6 Discussion on Results

For part 2, when run for 10000 iterations, the $f(x^*)$ was not matching with the $f(x^*)$ in part 1, possible reason is slow convergence rate of diminishing gradient descent as compared to constant gradient descent.

The exact line search has much faster convergence rate than any other algorithm and the number of iterations required to converge for each algorithm follow this order $T_4 < T_3 < T_1 < T_2$

3 Question 3 - Perturbed Gradient Descent

3.1 stationary points of f(x,y)

$$f(x,y) = e^{xy}$$

$$\nabla f(x,y) = [ye^{xy}, xe^{xy}]^T = 0 \implies x, y = (0,0)$$

$$\nabla^2 f(x,y) = \begin{bmatrix} y^2e^{xy} & (1+xy)e^{xy} \\ (1+xy)e^{xy} & x^2e^{xy} \end{bmatrix} \implies \nabla^2 f(0,0) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \implies \lambda = \pm 1 \tag{1}$$

The hessian matrix is neither PSD nor NSD, hence the result is inconclusive and the point $(x,y) = (0,0)$ is saddle point.

3.2 Staying on the line y = x

we can write line $y = x$ as the following set $L = \{(x_0, x_0) : x_0 \in R\}$

assume we started with the point on the line $x = y$ as $Z^0 = (\beta_0, \beta_0) \in L$ and run gradient descent as follows

$$Z^1 = Z^0 - \alpha_0 \nabla f(Z^0)$$

$$\nabla f(Z^0) = \nabla f(\beta_0, \beta_0) = [\beta_0 e^{\beta_0^2}, \beta_0 e^{\beta_0^2}]^T$$

$$Z^1 = [\beta_0, \beta_0]^T - \alpha_0 [\beta_0 e^{\beta_0^2}, \beta_0 e^{\beta_0^2}]^T = [\beta_0(1 - \alpha_0 e^{\beta_0^2}), \beta_0(1 - \alpha_0 e^{\beta_0^2})]^T = [\beta_1, \beta_1]^T \in L$$

$$\beta_1 = \beta_0(1 - \alpha_0 e^{\beta_0^2})$$

Now assume for any $Z^k = [\beta_k, \beta_k]^T \in L$, we can easily show $Z^{(K+1)} = [\beta_{k+1}, \beta_{k+1}]^T \in L$ where $\beta_{k+1} = \beta_k(1 - \alpha_k e^{\beta_k^2})$, hence on starting with point on $y = x$ line, one stay on the line $x = y$

3.3 Contour plot of f(x,y)

The countour plot of $e^{xy} : |x| < 1, |y| < 1$ is shown in Fig. 7

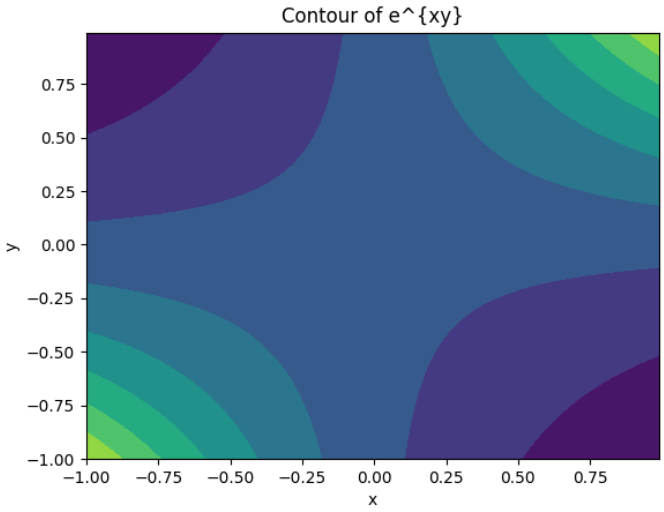

Figure 7: Contour of e^{xy}

3.4 Gradient descent with fixed step size

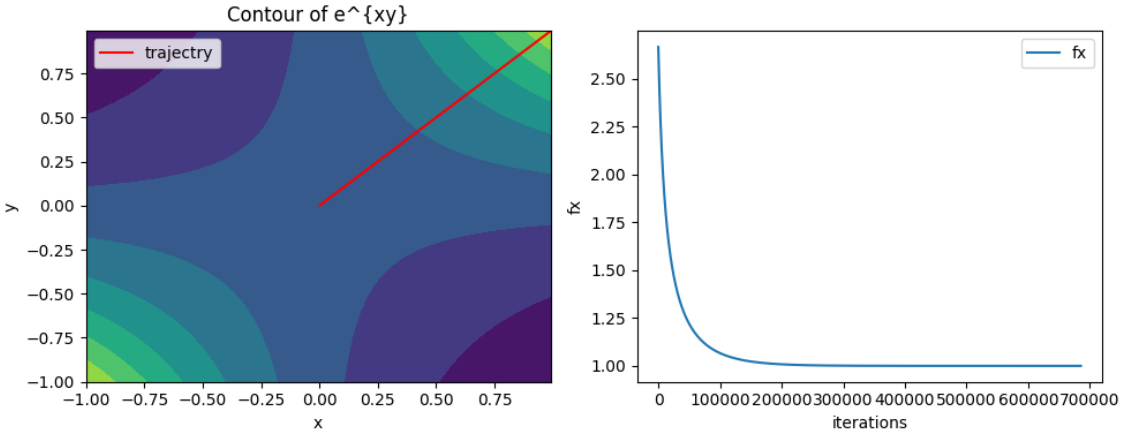

Figure 8: trajectry on contour plots and fx value

3.5 Gradient descent with diminishing step size

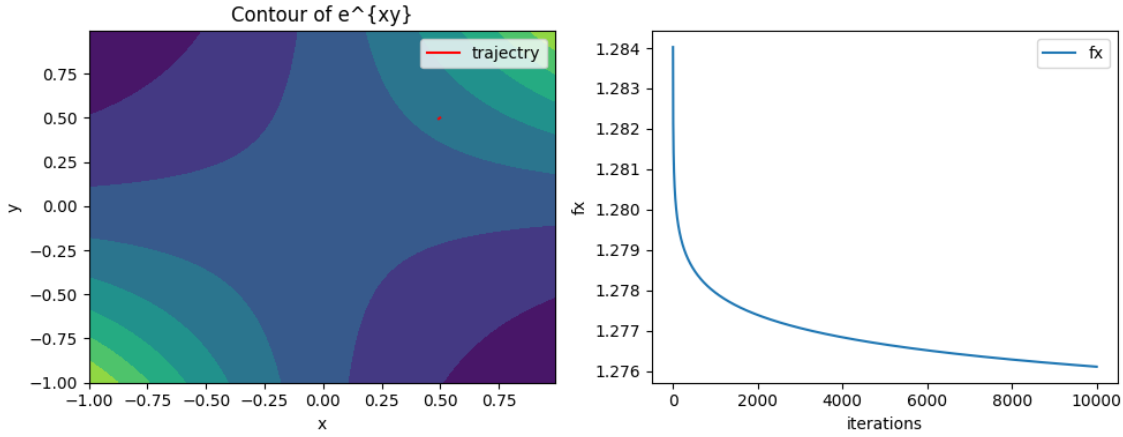

Figure 9: trajectry on contour plots and fx value

3.6 perturbed Gradient descent with fixed step size and var

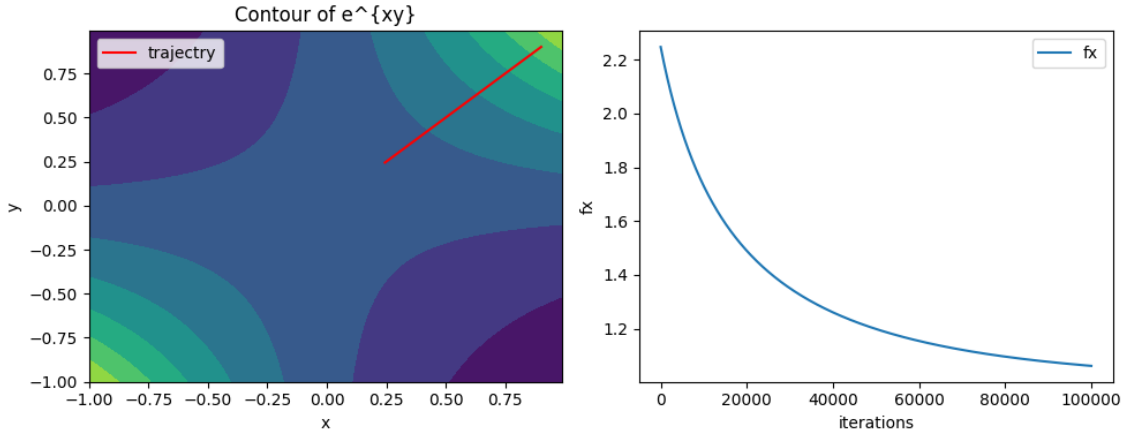

Figure 10: trajectry on contour plots and E[fx] value

3.7 perturbed Gradient descent with fixed step size and decreasing var

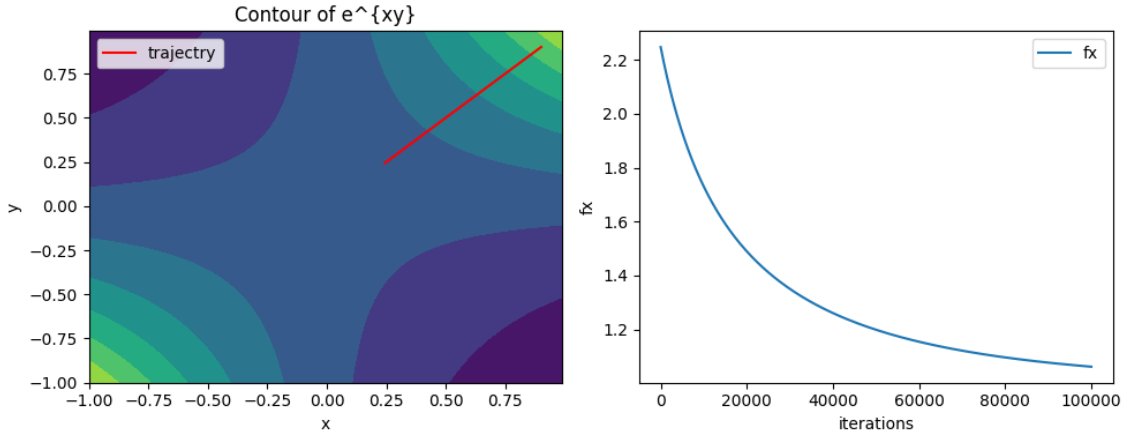

Figure 11: trajectry on contour plots and E[fx] value

3.8 perturbed Gradient descent with decreasing step size and fixed var

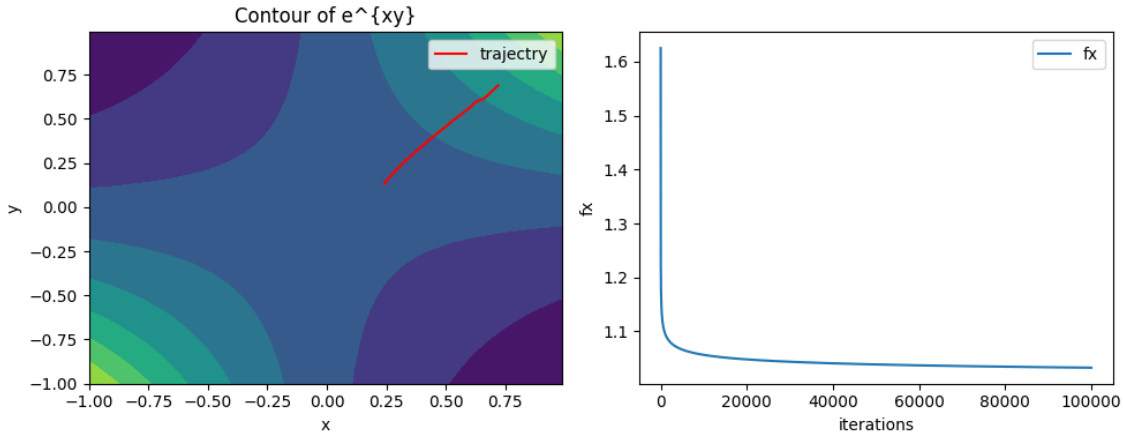

Figure 12: trajectry on contour plots and E[fx] value

3.9 perturbed Gradient descent with decreasing step size and decreasing var

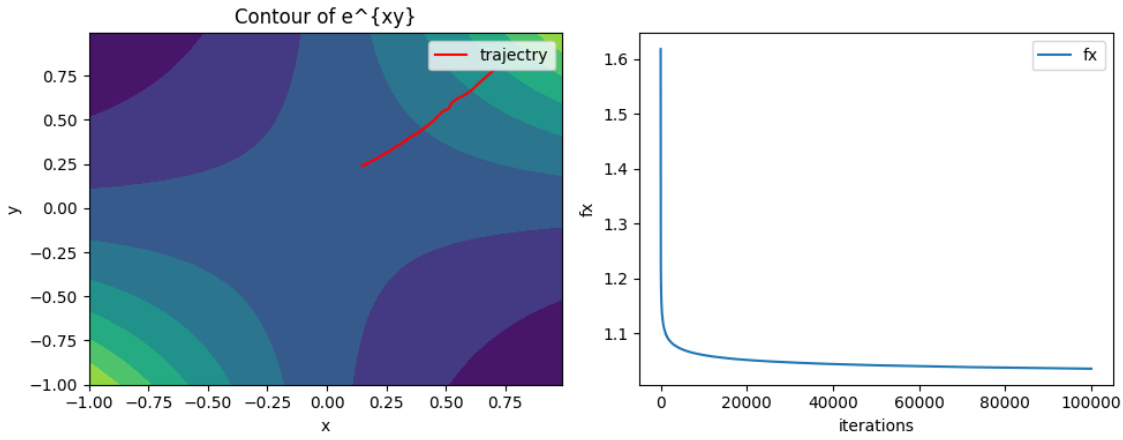

Figure 13: trajectry on contour plots and E[fx] value

3.10 relation between $E[f(\theta^{t+1})]$ and $E[f(\theta^t)]$

assume $f \in C^1$ according to the update equation

$$\theta^{(t+1)} = \theta^{(t)} - \alpha_t(\nabla f(x) + \zeta^{(t)})$$

using taylor theorem, we get

$$f(\theta^{(t+1)}) = f(\theta^{(t)}) + \nabla f(\theta^{(t)})^T(\theta^{(t+1)} - \theta^{(t)}) + o(||\theta^{(t+1)} - \theta^{(t)}||)$$

$$\theta^{(t+1)} - \theta^{(t)} = -\alpha_t(\nabla f(x) + \zeta^{(t)})$$

$$||\theta^{(t+1)} - \theta^{(t)}|| = \alpha_t||(\nabla f(x) + \zeta^{(t)}|| \leq \alpha_t||(\nabla f(x)|| + ||\zeta^{(t)}||$$

$$f(\theta^{(t+1)}) = f(\theta^{(t)}) - \alpha_t \nabla f(\theta^{(t)})^T(\nabla f(\theta^{(t)}) + \zeta^{(t)}) + o(\alpha_t||(\nabla f(x)|| + \alpha_t||\zeta^{(t)}||)$$

$$f(\theta^{(t+1)}) = f(\theta^{(t)}) - \alpha_t||\nabla f(\theta^{(t)})||^2 - \alpha_t \nabla f(\theta^{(t)})^T\zeta^{(t)}) + o(\alpha_t||(\nabla f(x)||) + o(\alpha_t||\zeta^{(t)}||)$$

$$E[f(\theta^{(t+1)})] = E[f(\theta^{(t)})] - E[\alpha_t||\nabla f(\theta^{(t)})||^2] - E[\alpha_t \nabla f(\theta^{(t)})^T\zeta^{(t)})] + E[o(\alpha_t||(\nabla f(x)||) + o(\alpha_t||\zeta^{(t)}||)]$$

$$E[\alpha_t \nabla f(\theta^{(t)})^T\zeta^{(t)})] = 0$$

$$E[f(\theta^{(t+1)})] = E[f(\theta^{(t)})] - \alpha_t||\nabla f(\theta^{(t)})||^2 + o(\alpha_t||(\nabla f(x)||) + o(\alpha_t||\zeta^{(t)}||)$$

thereby establishing relation between $E[f(\theta^{t+1})]$ and $E[f(\theta^t)]$

3.11 Discussion on results for perturbed gradient descent

Perturbed gradient descent indeed helps in to move outside saddle point as shown in Fig. 12 and Fig. 13, it helps to move outside saddle point, which was initially stuck in Fig. 9

4 Question 4 - Golden section search and Fibonacci sequence

4.1 Extrema of $f(x)$

Extremas are the points where $f'(x) = 0$

$$f(x) = x(x-1)(x-3)(x+2)$$

$$f'(x) = (2x-1)(2x^2-2x-6) = 0$$

$$x = \frac{1}{2}, \frac{1 \pm \sqrt{13}}{2} = 0.5, 2.302, -1.302$$

now we do second derivative test to see which one of them is minima or maxima

$$f''(x) = 12x^2 - 12x - 10$$

$$f''(0.5) = 12(0.5)^2 - 12 * 0.5 - 10 = -13 < 0$$

$$f''(2.302) = 12(2.302)^2 - 12 * 2.302 - 10 = 25.96 > 0$$

$$f''(-1.302) = 12(-1.302)^2 + 12 * 1.302 - 10 = 25.96 > 0$$

hence maxima $= \frac{1}{2}$, minima $= \frac{1 \pm \sqrt{13}}{2}$

4.2 Golden section search and fibonacci search

The algorithm was implemented and the plots for $f(a_t), f(b_t), (b_t - a_t), \frac{(b_t - a_t)}{(b_{t-1} - a_{t-1})}$ for Golden and fibonacci search are presented in Fig. 15, Fig. 16 respectively.

The interval and number of iterations for golden and fibonacci search is presented in the Fig. 14, As output by both algorithms, the minima point is approximately lies around 2.302, which is indeed a minima in the interval $[1, 3]$ as proved earlier

```
$ python que4.py
Golden section search: interval: [2.302711416340475, 2.302742627813805, 2.302761917565162, 2.3027931290384918], Iterations: 21

Fibonacci search: interval: [2.302775628793678, 2.302775628793678, 2.3027756287936785, 2.302775628793678], Iterations: 77
```

Figure 14: Interval and number of iterations for golden and fibonacci search algorithm

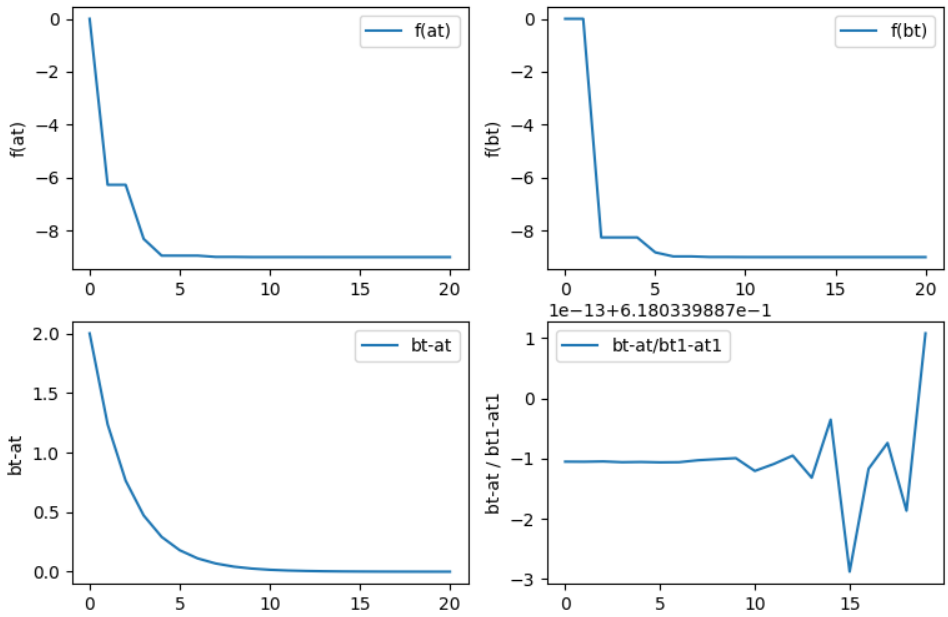

Figure 15: plots for Golden section search

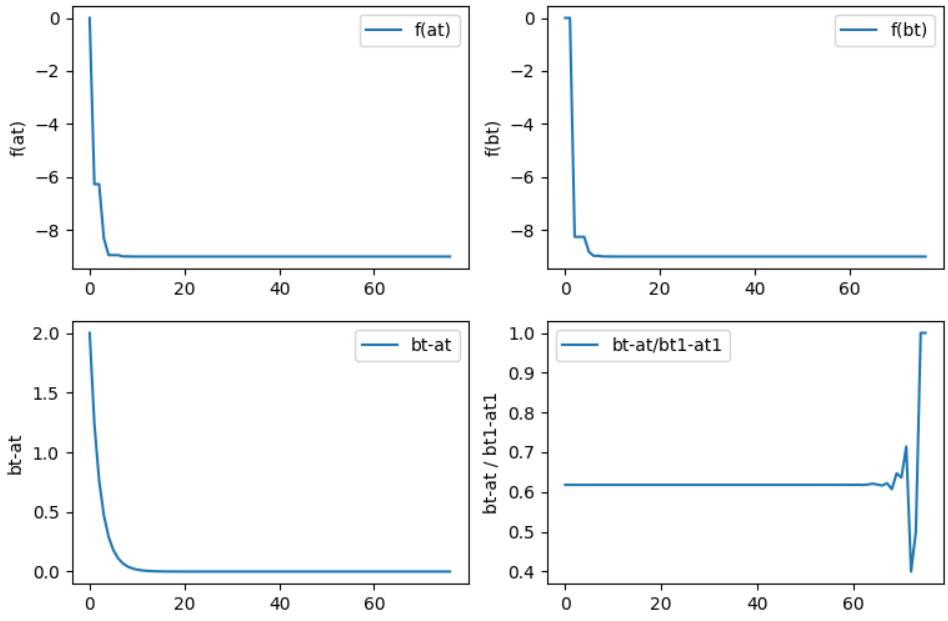

Figure 16: plots for Fibonacci search