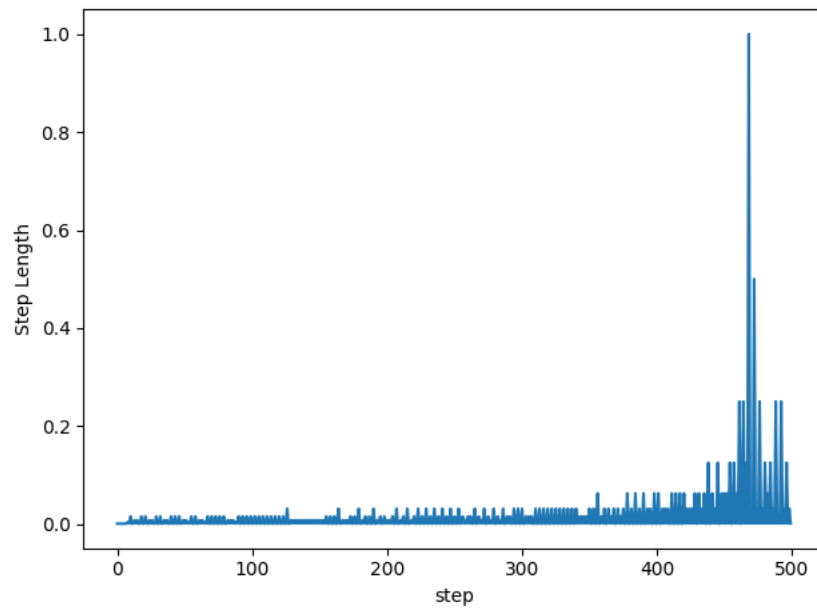


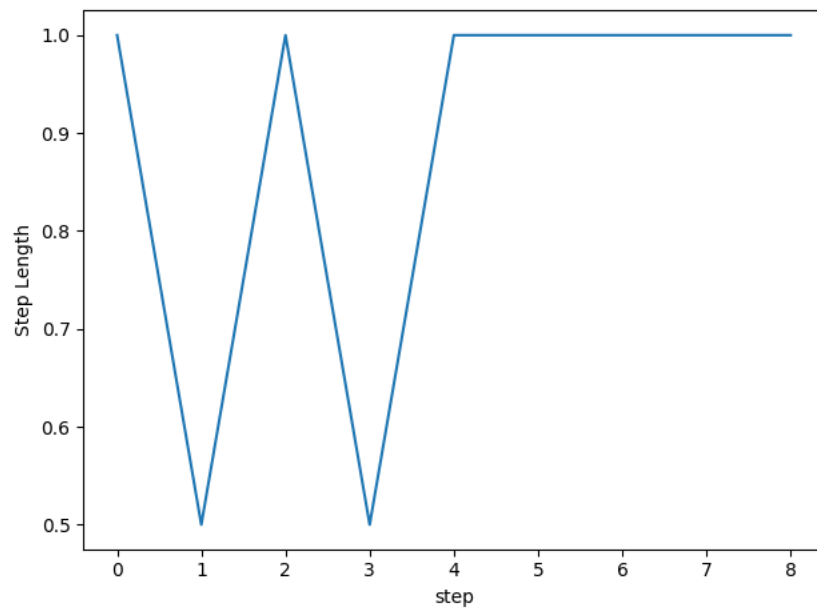
-٨
نقطه شروع (1.2 , 1.2)

Gradient Descent:

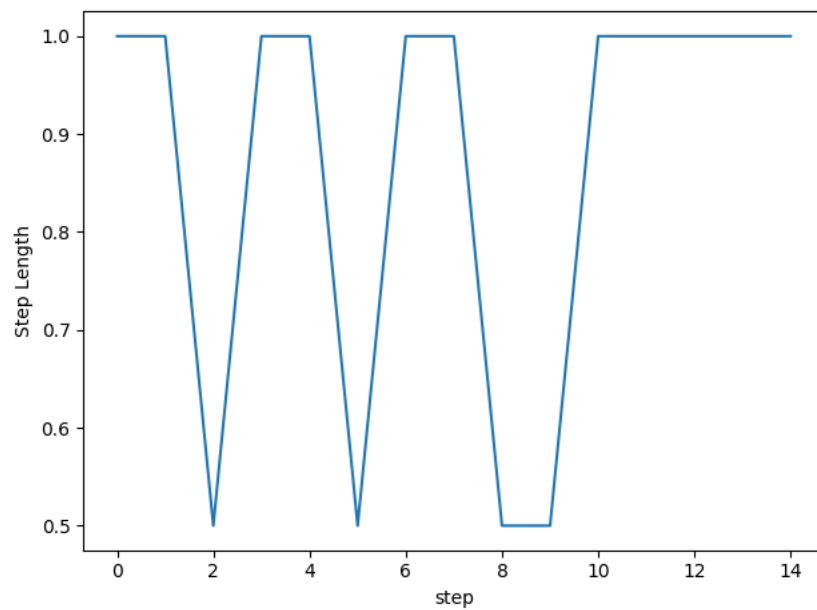


$x = \begin{bmatrix} 1.00427107 \\ 1.00857804 \end{bmatrix}$

Newton:



Quasi Newton BFGS:

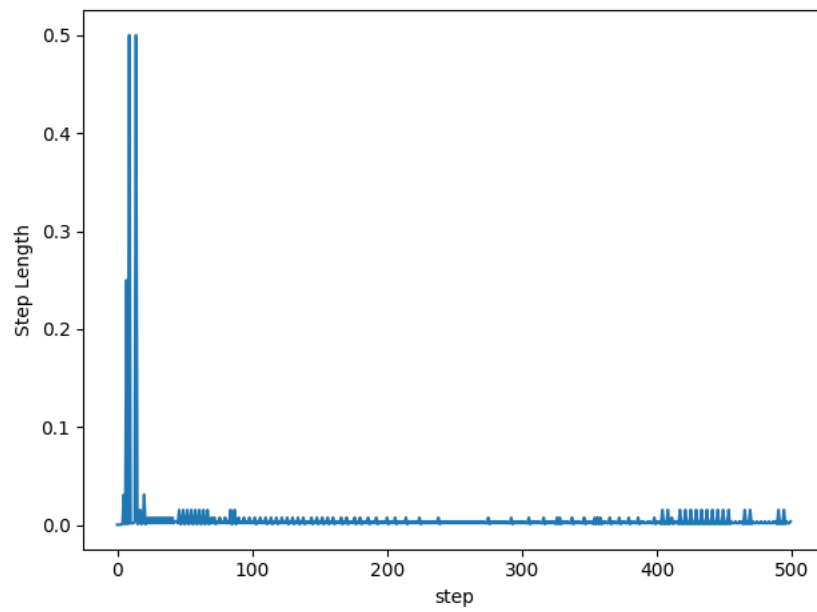


همانطور که از شکل‌ها مشخص است gradient descent در ۵۰۰ قدم نتوانسته جواب کاملاً بهینه را بیابد. همچنین در ابتدا طول قدم‌ها بسیار کوچک بوده و در بازه‌ای افزایش می‌یابد، که به طور کلی در بسیاری از زمان‌ها نزدیک به ۰ است.

Newton و quasi newton بسیار بهتر عمل کرده و جواب بهینه را به ترتیب در ۸ و ۱۴ قدم بدست آورده‌اند. و میزان تغییر step length در هر دو شکل مشهود است، که به طور کلی در بسیاری از زمان‌ها نزدیک به ۱ است.

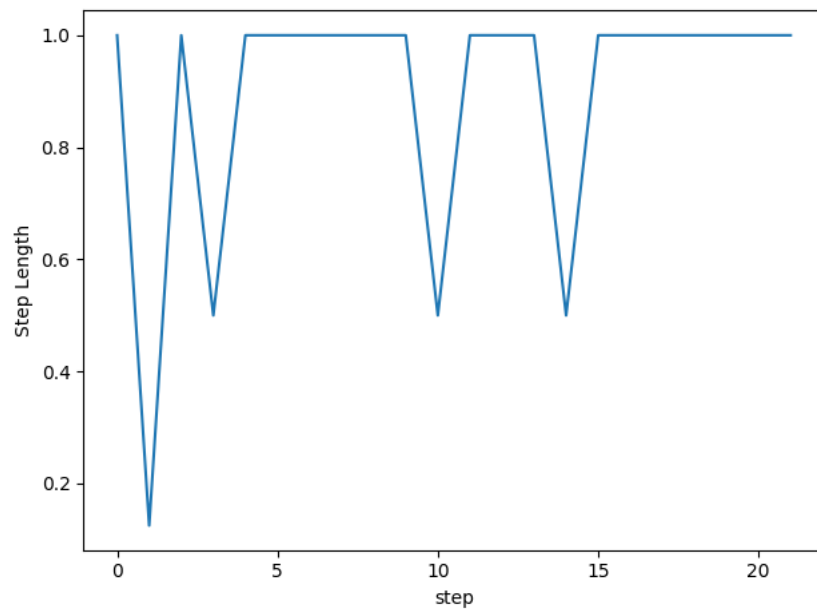
نقطه شروع (-1.2 , 1)

Gradient Descent:

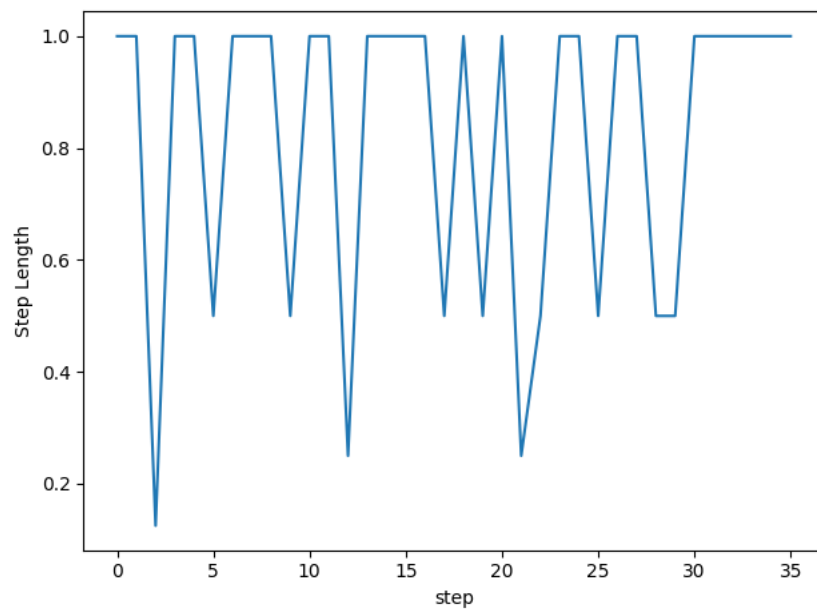


$x = \begin{bmatrix} 0.89897199 \\ 0.80745907 \end{bmatrix}$

Newton:



Quasi Newton BFGS:



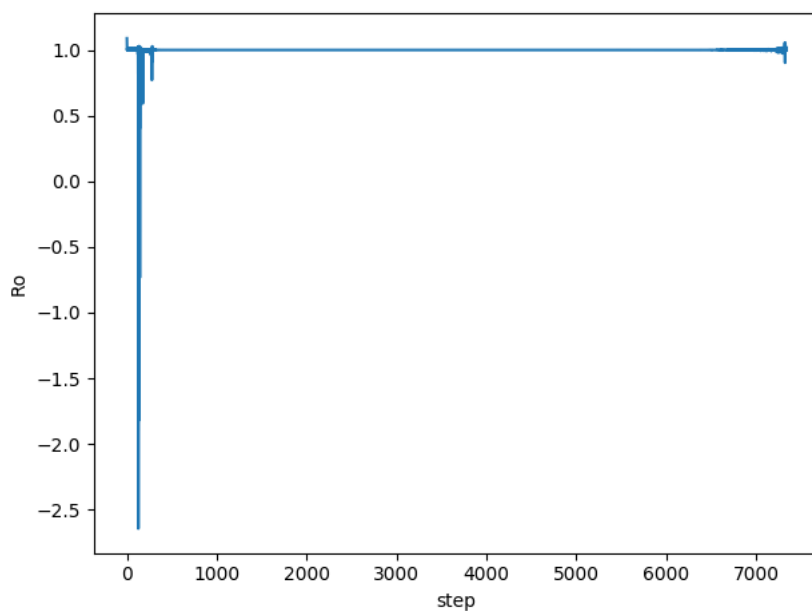
همانطور که از شکل‌ها مشخص است gradient descent در ۵۰۰ قدم نتوانسته جواب کاملاً بهینه را بیابد. همچنین در ابتدا طول قدم‌ها بزرگ بوده و به تدریج کاهش می‌یابد، که به طور کلی در بسیاری از زمان‌ها نزدیک به ۰ است.

Newton و quasi newton بسیار بهتر عمل کرده و جواب بهینه را به ترتیب در ۲۰ و ۳۵ قدم بدست آورده‌اند. و میزان تغییر step length در هر دو شکل مشهود است، که به طور کلی در بسیاری از زمان‌ها نزدیک به ۱ است.

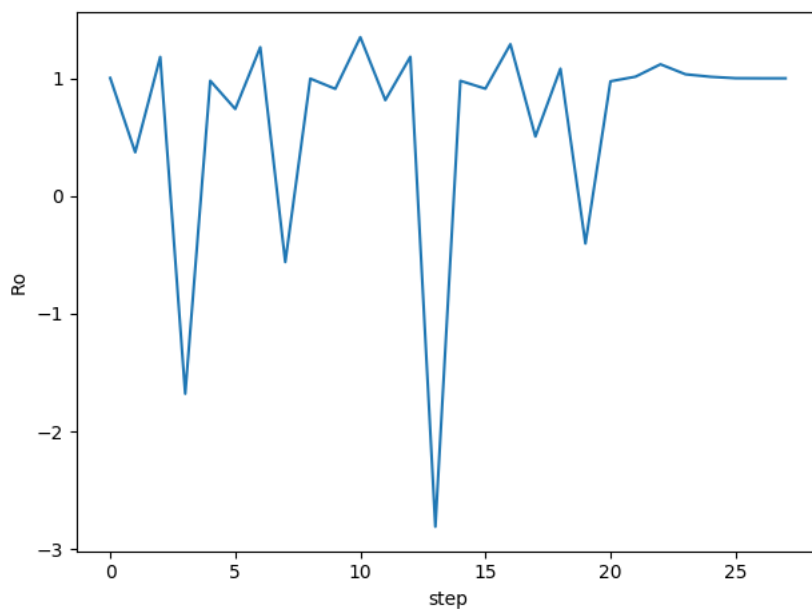
برآورد کلی : newton بهترین دقیقترین عملکرد را دارد به این معنی که در کمترین تعداد گم جواب را بدست می‌آورد.

۹- الف) بله. مهمترین دلیل آن این است که تابع، تابع ساده‌ای است و میتوان دستی از آن مشتق گرفت و در تمام مراحل از آن مشتق حساب شده استفاده کرد که بار محاسباتی ای نخواهد داشت. علاوه بر این از آنجا که ابعاد ماتریس‌ها در این مورد ۲ در ۲ می‌باشد. به طور کلی بار محاسباتی بالایی نخواهد داشت و بنابراین استفاده از همین مناسب است.

(ب)
Cauchy

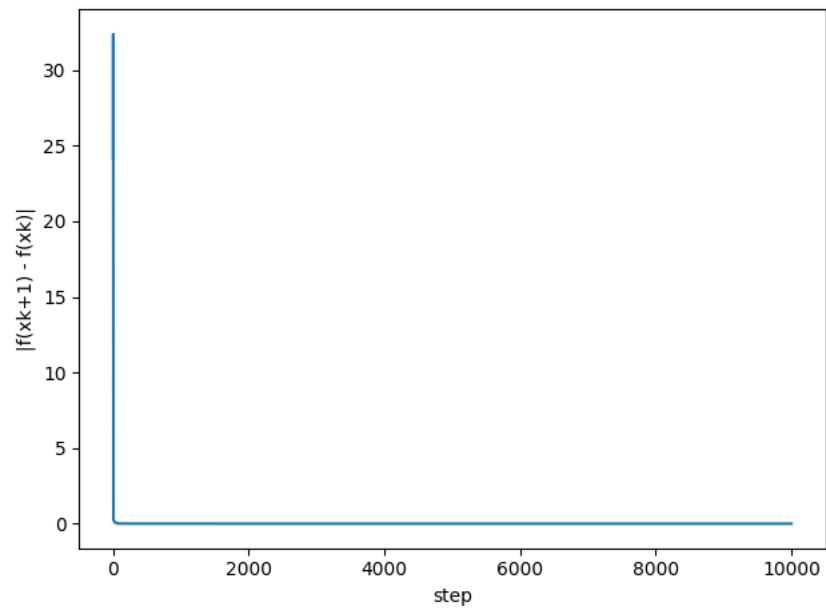


Dogleg



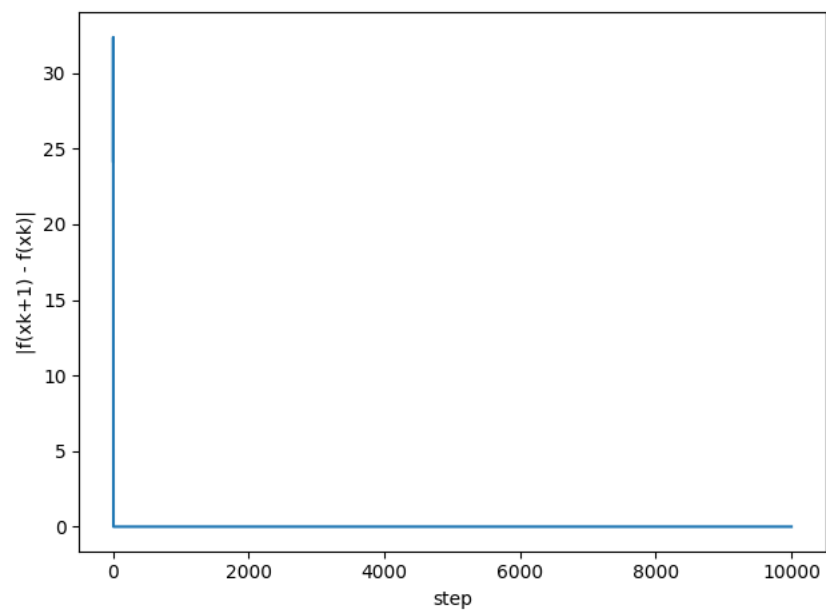
N = 100

با استفاده از Cauchy point



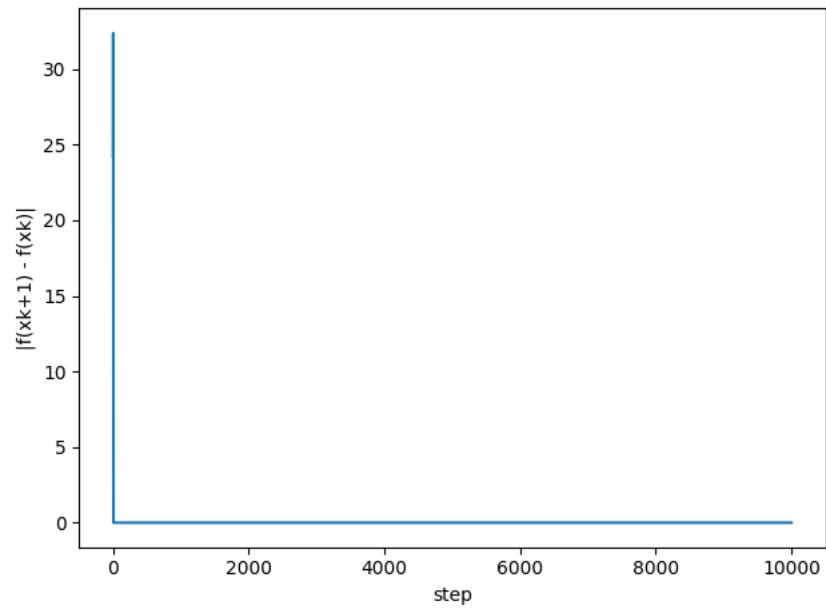
$$f(x) = 0.3325593500624111$$

با استفاده از dogleg



$$f(x) = 8.636386187867405$$

با استفاده از BFGS



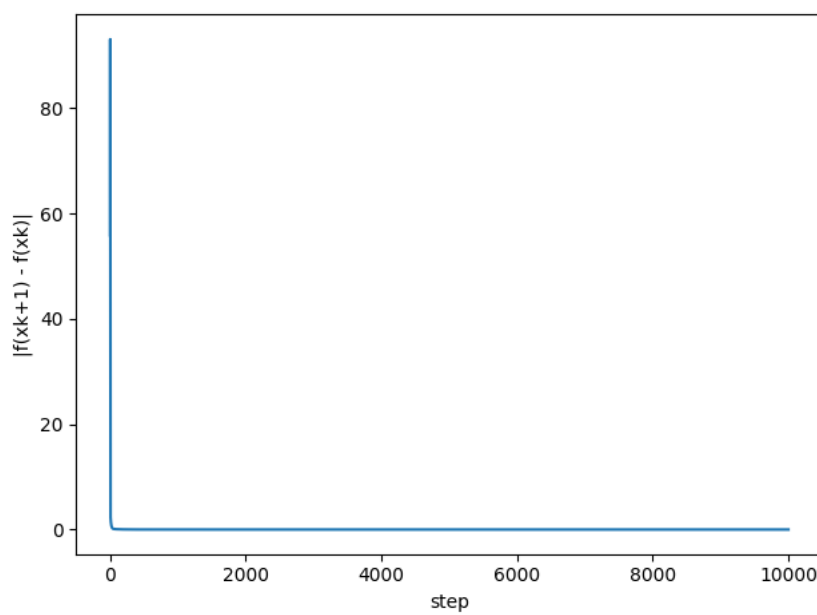
$$f(x) = 9.947380187619055$$

استفاده از cvxpy به عنوان toolbox

```
Optimal objective value using cvxpy: 0.012665003850926837
```

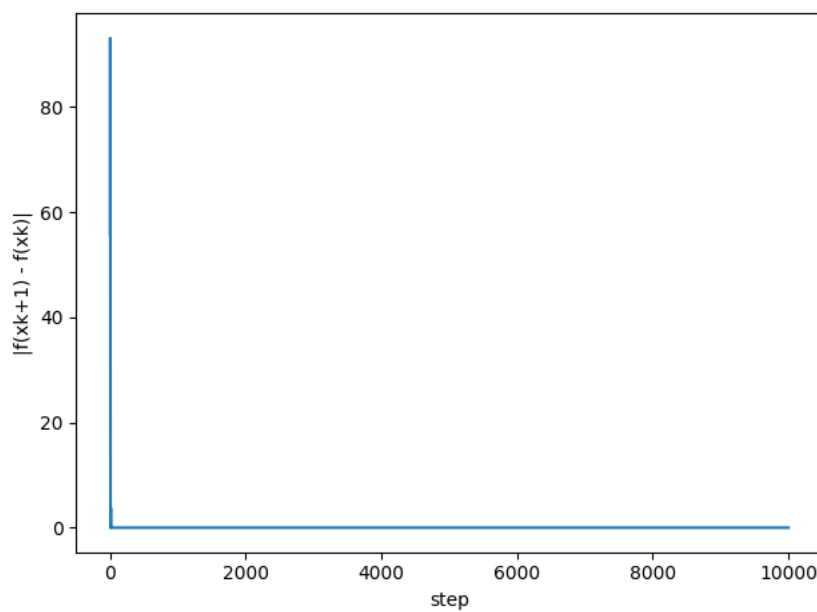
N = 400

با استفاده از Cauchy point



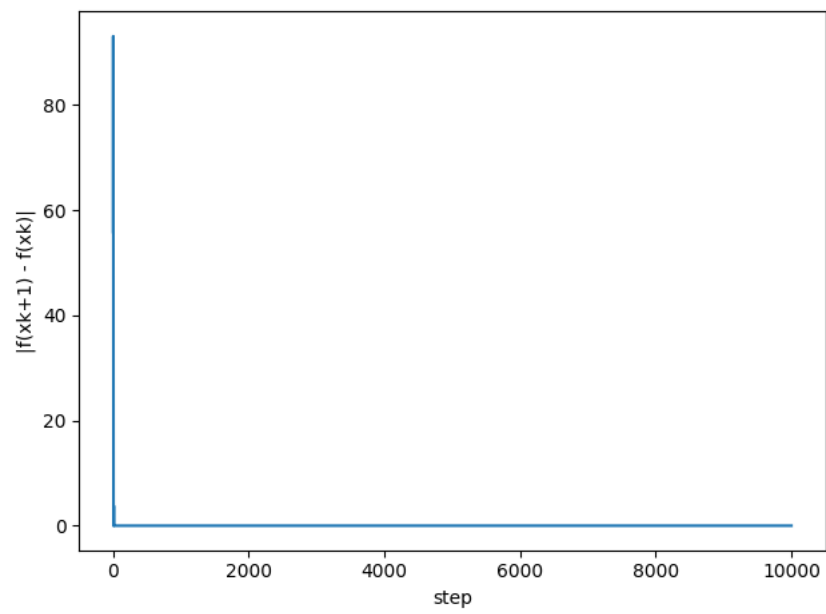
$$f(x) = 1.395711862837176$$

با استفاده از dogleg



$$f(x) = 38.26551133055435$$

با استفاده از BFGS



$$f(x) = 40.26846196055435$$

استفاده از cvxpy به عنوان toolbox

```
Optimal objective value using cvxpy: 0.09486875487885982
```

بررسی:

در بین الگوریتم‌های استفاده شده Cauchy بیشترین سرعت را داشته پس از آن dogleg و پس از آن BFGS . همچنین مقادیر بدست آمده از طریق Cauchy بهینه تر می‌باشند و پس از آن dogleg و BFGS قرار دارند. و در کل الگوریتم آماده‌ی cvxpy چه از نظر سرعت و چه از نظر دقت بهترین جواب را داده است.