

**بهینه سازی محدب**

**گزارش پیاده سازی سوال ۳**

**استاد درس: دکتر هادی امیری**

**دانشجو: مینو احمدی**

**810897032**

**دانشکده علوم مهندسی، دانشگاه تهران**

Miinouahmadii@gmail.com

برای ران کردن کد ها از google colab استفاده کنید. با تشکر

**سوال سوم قسمت الف :**

در این قسمت من ابتدا گرادیان تابع و سپس هسین ان را محاسبه کردم

بعد از تشکیل ماتریس هسین دیده میشود که تمام درایه های ان مثبت است پس این ماتریس مثبت نیمه معین است چون ماتریس هسین مثبت معین شد پس تابع ما کانوکس بوده است .

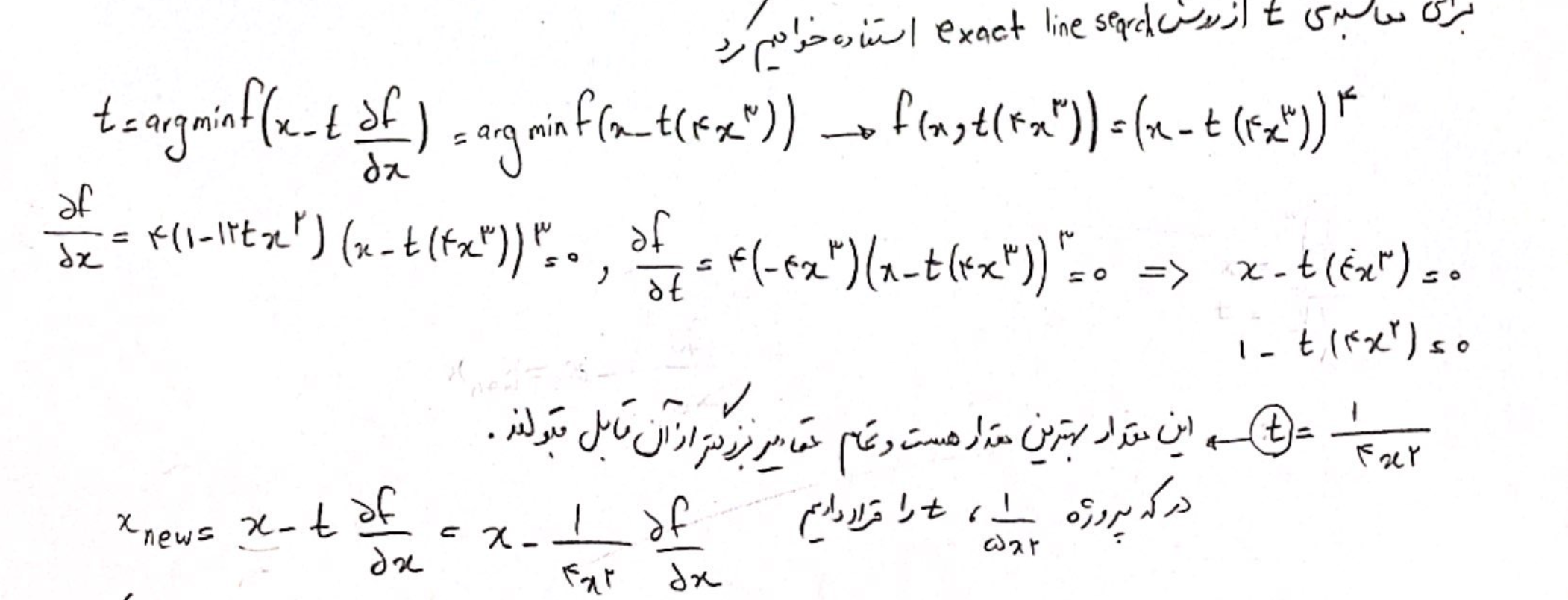
**سوال سه قسمت ب :**

در این سوال line search ما به صورت دقیق بوده به همین دلیل t ای که ارگمان تابع f را مینیموم کند را ابتدا پیدا کردیم و سپس با توجه به ان t مراحل گرادیان کاهشی را انقدر تکرار کردیم تا e   یا همان خطای ما کمتر از عدد مورد نظر صورت سوال شد . تفاوت اصلی این بخش و بخش بعدی در نحوه ی محاسبه ی t میباشد.

cur\_x = cur\_x - 1/(5 \* cur\_x\*\*2) \* df(prev\_x) #Grad descent

اینجا ما ابتدا سوال را به روش تحلیلی حل کردیم و t حاصل شده برابر با 1/(4\*x^2) شد .

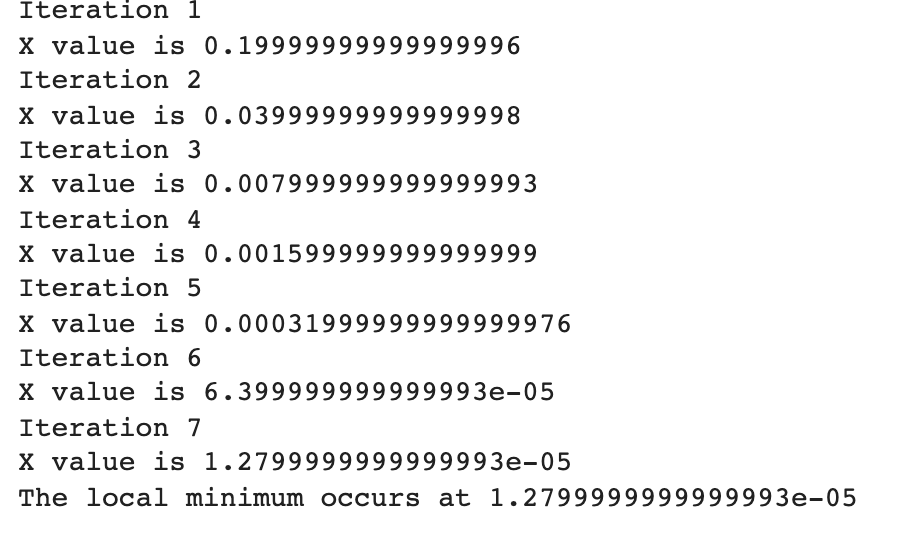
حل تحلیل در زیر امده :



این t بهترین t حاصل شده با روش خط دقیق است . جواب های کوچکتر از ان نیز کاملا قابل قبپلند فقط هرچه کوچکتر شود در واقع به تعداد ایتریشن های بیشتری نیاز داریم .

در اینجا هربار مقدار t ما با توجه به نسبتی که با x دارد تغییر میکند.

در اینجا الگوریتم به پایان رسیده و جواب حاصل میشود با این راه ما توانستیم با ۷ مرحله به جواب برسیم. خروجی های هر مرحله در زیر امده است :



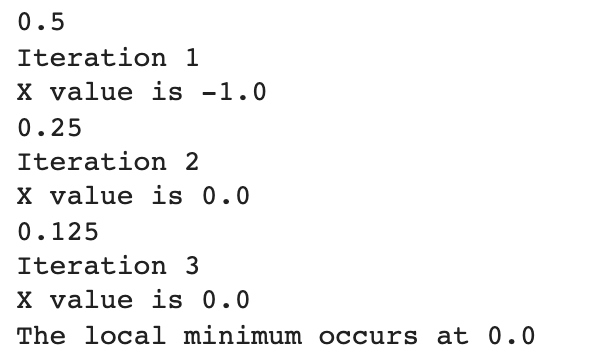
**سوال سوم قسمت پ :**

در این قسمت روش محاسبه t مورد نظر ما با روش بازگشتی محاسبه شده برای این کار نیاز به بتای داده شده در سوال داریم یعنی هر بار t ما در بتا ضرب شده و سپس برای t مرحله بعد دوباره عدد مرحله ی قبلی در بتا ضرب میشود .

rate \*= beta

cur\_x = cur\_x - rate \* df(prev\_x) #Grad descent with backtracking line search

در نتیجه خروجی با سه بار ایتریشن حاصل میشود خروجی هر مرحله در زیر امده است :



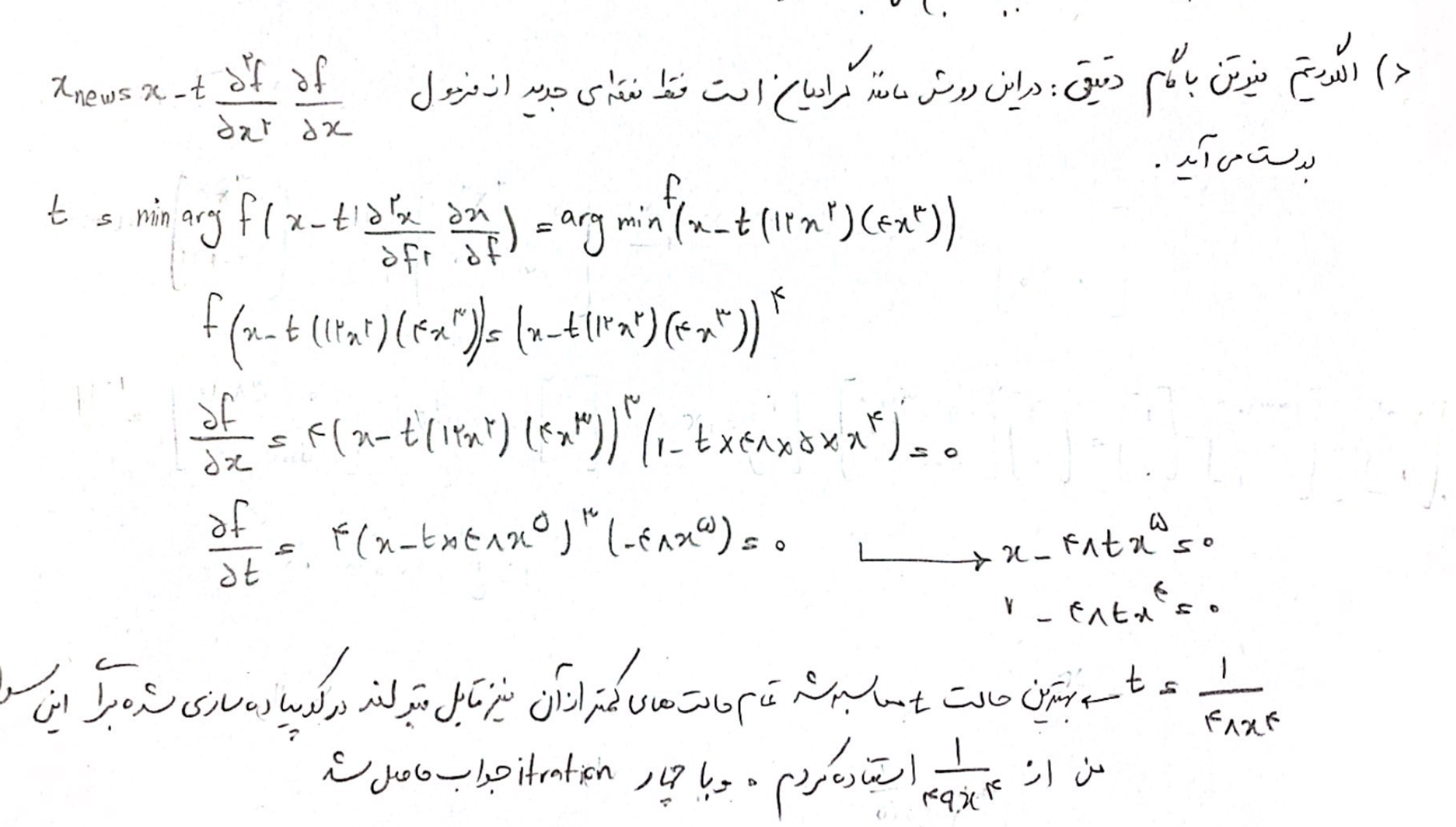
**سوال سوم قسمت اخر :**

در اینجا از روش نیوتن استفاده شده است که تنها تفاوت ان در رابطه ی ایکس جدید با قدیم است که ضریب t تغییر کرده است در واقع مشتق دوم تابع بر حسب ایکس نیز به ضریب t اضافه شده است .

cur\_x = cur\_x - 1/(49 \* cur\_x\*\*4) \* df2(prev\_x) \* df(prev\_x) #Newton's algorithm with exact line search

با توجه به انکه ضریب داخل f تغییر میکند برای پیدا کردنt ای که ارگمان تابع را مینیموم کند از روش تحلیلی استفاده کردیم و با مشتق گیری رابطه بین t و x مشخص شد در واقع t=1/(48\*x^4) است با توجه به اینن موضوع تمام مقادیر کوچکتر از t نیز مناسبند فقط تعداد ایتریشن ها تغییر خواهد کرد .

حل تحلیل پیدا کردن t در زیر امده است :



با این روش خروجی جواب ها با ۴ ایتریشن حاصل شد . میتوانید خروجی هارا مشاهده کنید :

