بسمه تعالی

درس محاسبات عددی

دکتر سیاوش بیات

دانشکده مهندسی برق

پروژه درس محاسبات عددی

محمد حسین مسلمی / 97102463

سید محمد هاشمی کرویی / 97102644

محمدحسین مسلمی

سید محمد هاشمی کرویی

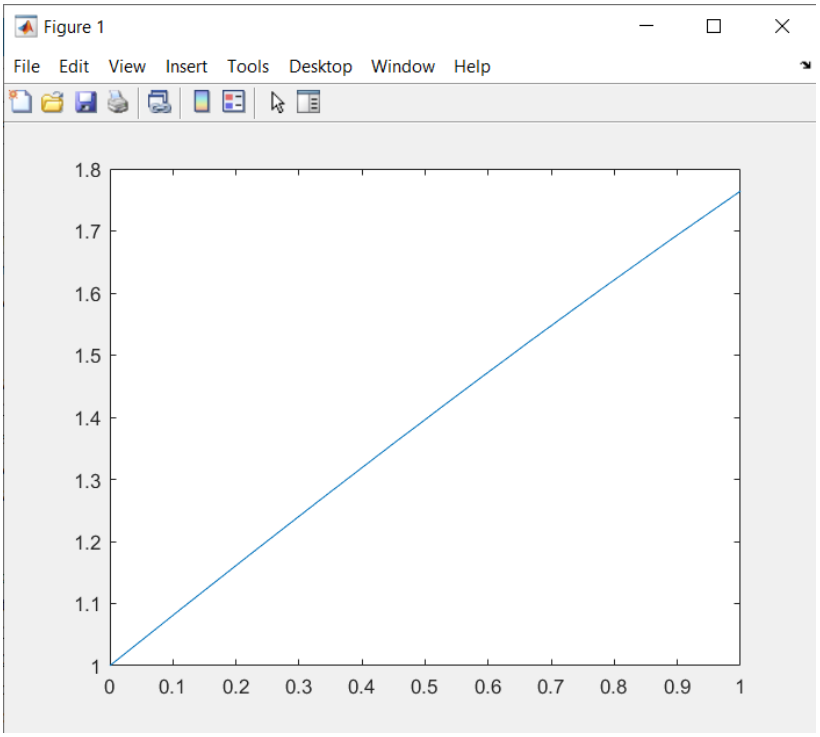
گزارش پروژه

1-الف)این تابع در solver1.m پیاده سازی شده است.تست آن به صورت زیر است:

تکه کد تست:

```
f={@(g1,g2,t) g2 , @(g1,g2,t) t-g1};
h=0.1;
ttotal=1;
g0=[0 ; 1];
g = solver1(f,h,g0,ttotal);
tplot= linspace(0,ttotal,ttotal/h);
plot(tplot,g(2,:));
a=ttotal/h;
```

حاصل:



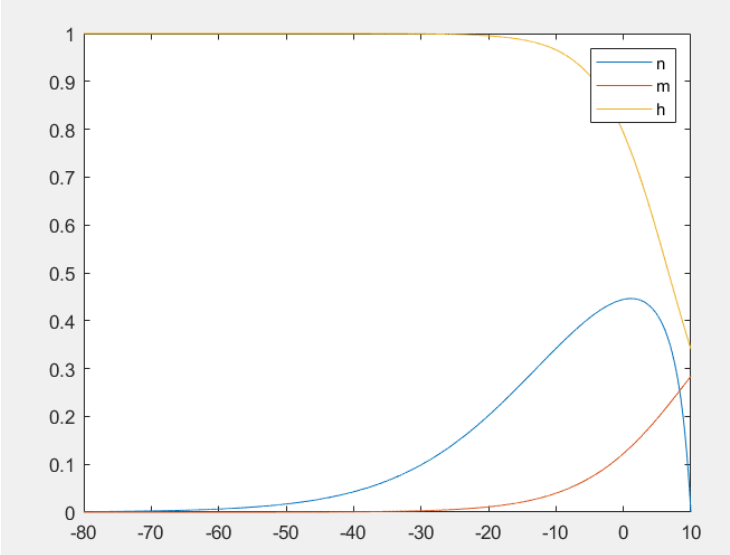
1-ب) این تابع در solver2.m پیاده سازی شده است.

1-ج)این بردار در بردار از هندل فانکشن f پیاده سازی شده است.

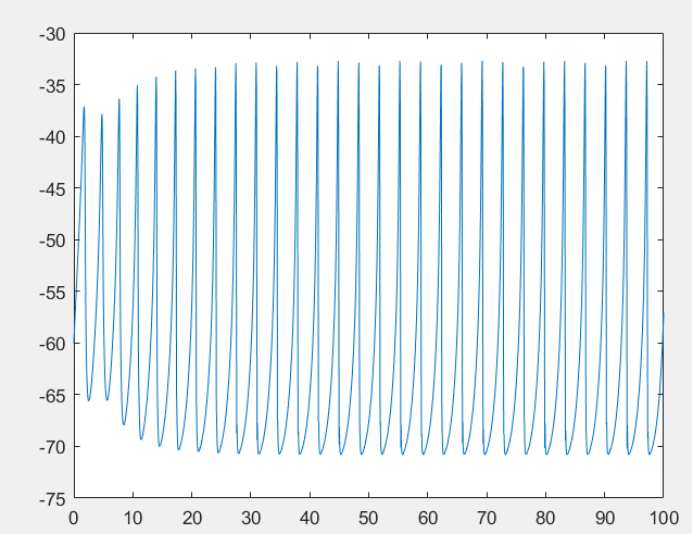
1-د)توابع m و n و h برحسب v به دست آوردیم و در نهایت آنها را رسم کردیم:

$$\frac{dn}{dt}=0\rightarrow\beta_n(u)n=\alpha_n(u)(1-n)\rightarrow n(v)=\frac{\alpha_n(u)}{\alpha_n(u)+\beta_n(u)}$$

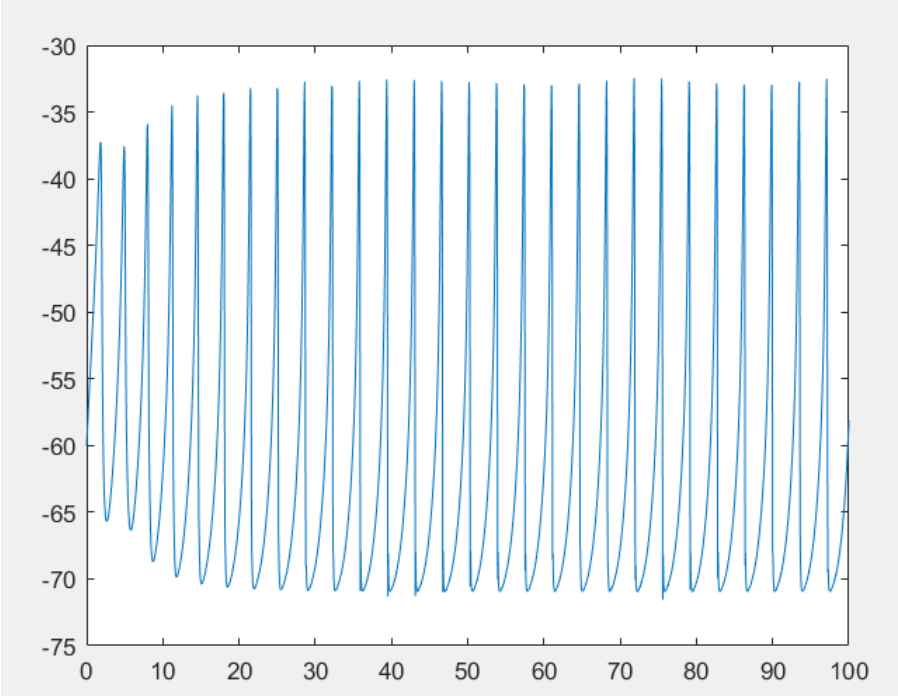
و به طور مشابه برای m و h .



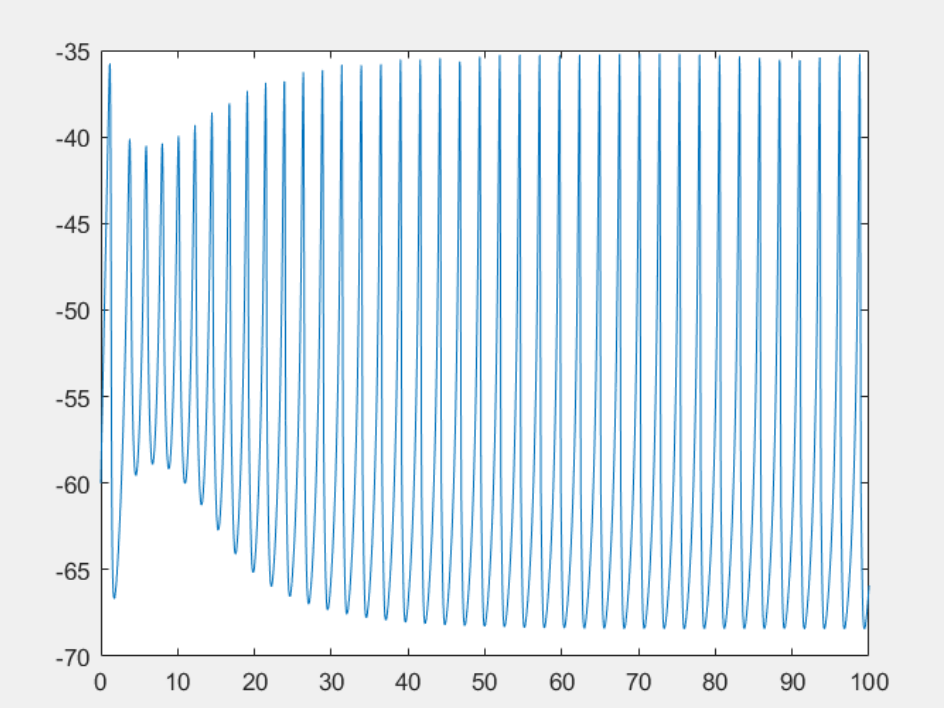
1-ه)این قطعه کد در قسمت ه درفایل اصلی متلب قرار دارد.



در حالت I=0

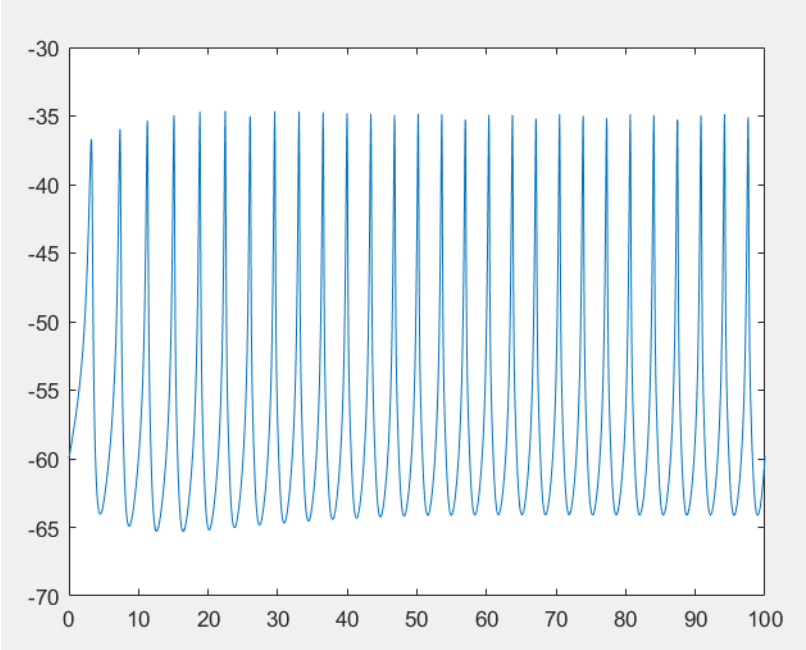


در حالت I=9

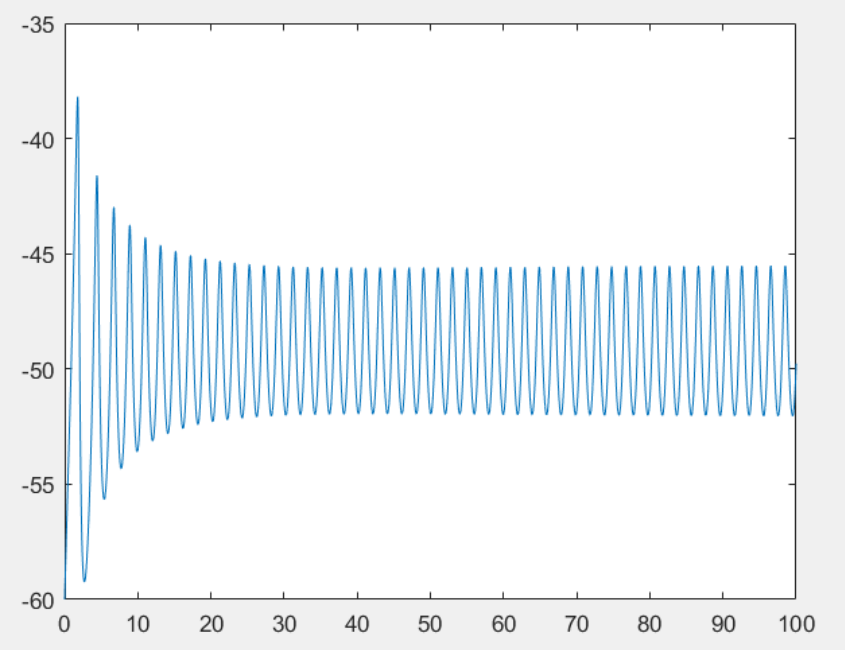


در حالت I=20

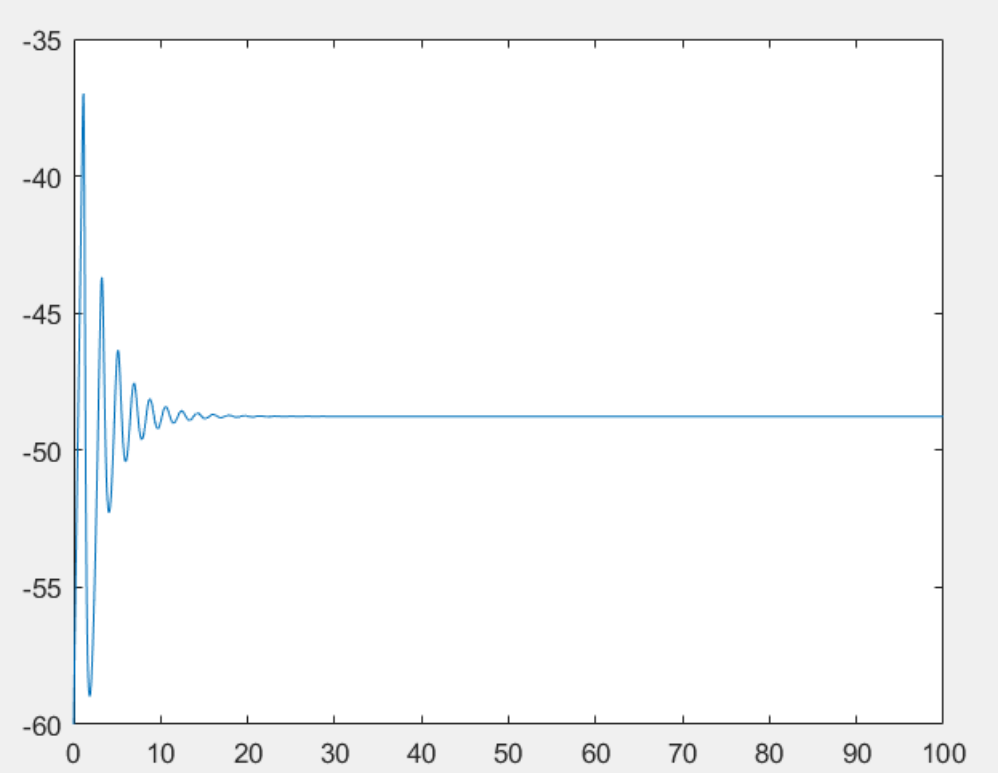
1-و) این مساله با روش رونگ-کوتا پیاده سازی شده در تابع solver2 حل شده است.



در حالت I=0



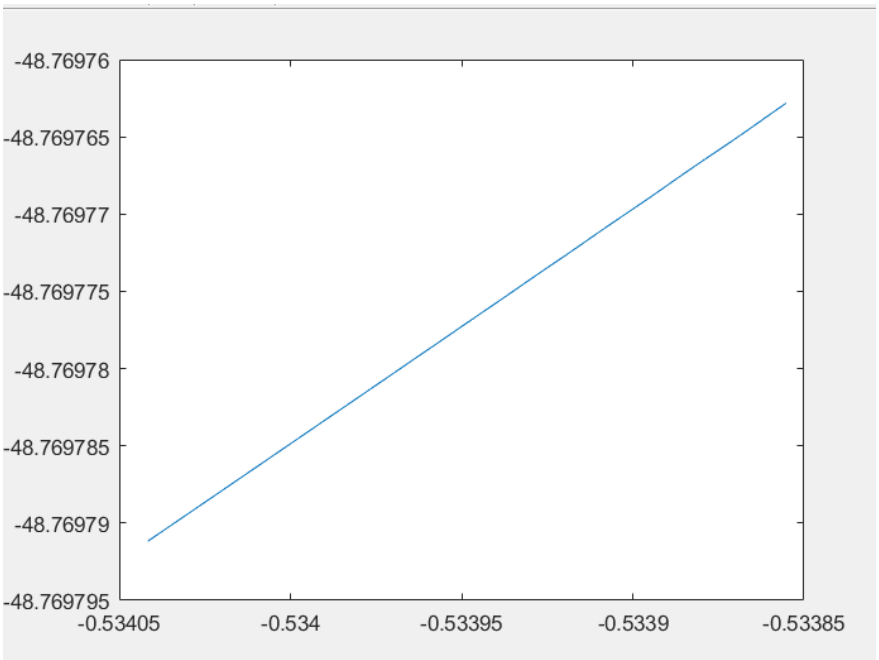
در حالت I=9



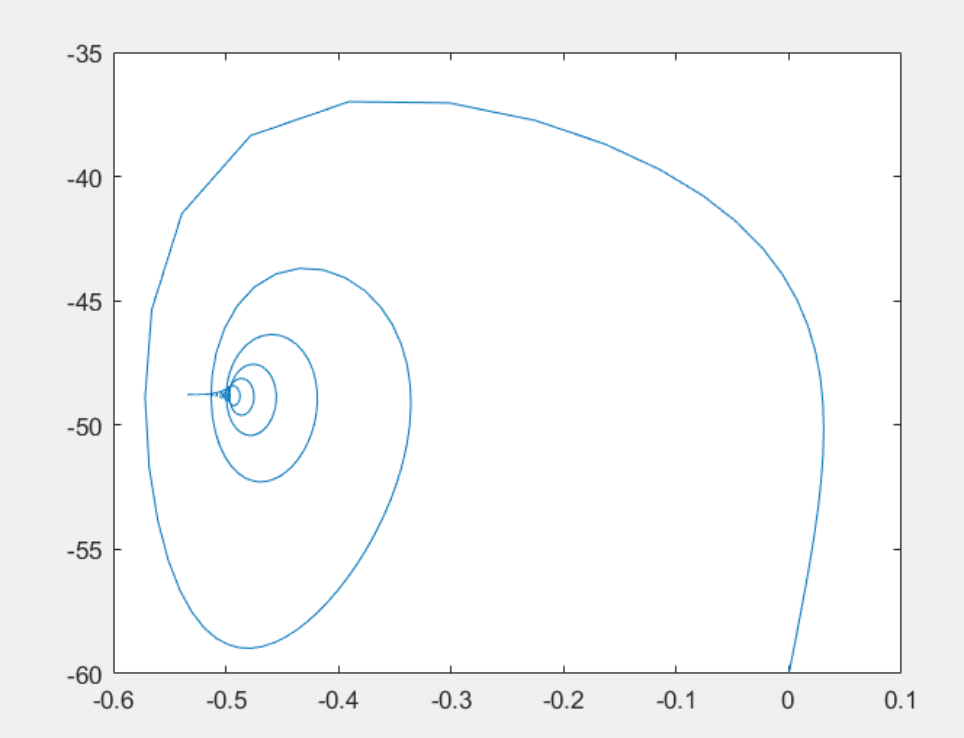
در حالت I=20

به نظر خطا به علت خطای انباشته ی بیشتر در روش اویلر است که وقتی زمان و تعداد محاسبات زیاد می شود از مقدار واقعی فاصله می گیرد.

1-و) در قسمت و کد پیاده سازی شده است.در بازه ی خواسته شده برای i=20 در شکل زیر رسم شده است:



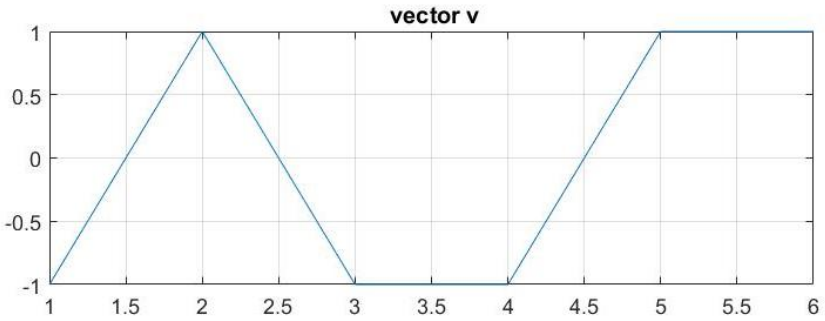
و برای کل بازه برای همان مقدار **i** در شکل زیر رسم شده است:

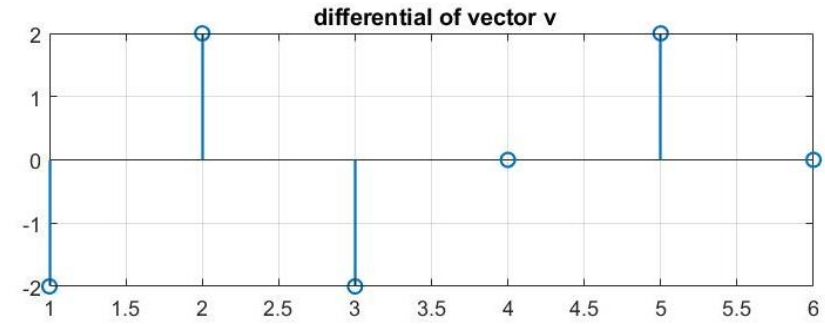


در هر دو رسم از روش رونگ-کوتا برای حل استفاده شده است

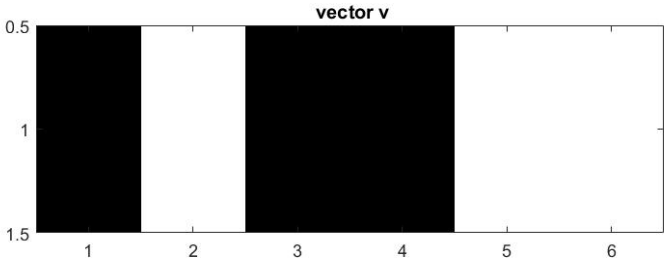
فاز دو

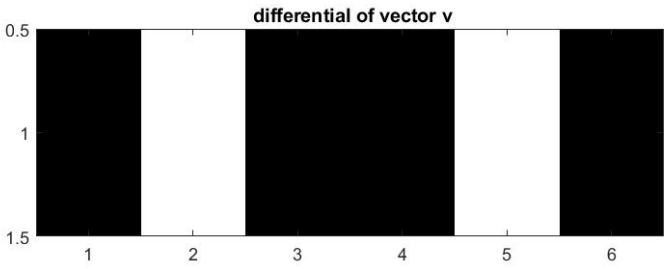
الف) و ب) در حالت گسسته مشتق در هر نقطه برابر با تفاضل مقدار آن نقطه با مقدار نقطه قبلی است که با ضرب بردار v به طول n در ماتریس مربعی $W_{n\times n}$ به صورتی که داده شده مقدار مشتق در هر نقطه بدست میآید به اینصورت که مثلا در نقطه صفر آرایه اول سطر اول w برابر 1 و آرایه آخر سطر اول آن برابر 1- است لذا با ضرب این سطر در v ، مشتق v در نقطه صفر بدست می آید . حال با circular shift دادن سطر اول w و قرار دادن آن در سطرهای بعدی عمل مشتق گیری در هر نقطه را بدست میآوریم و حاصل به صورت زیر است :





ج) حال به شیوه ای که در صورت سوال گفته شده هر دو بردار را binary کرده و با استفاده از imagesc و همینطور colormap(gray) نشان میدهیم که به صورت زیر است :





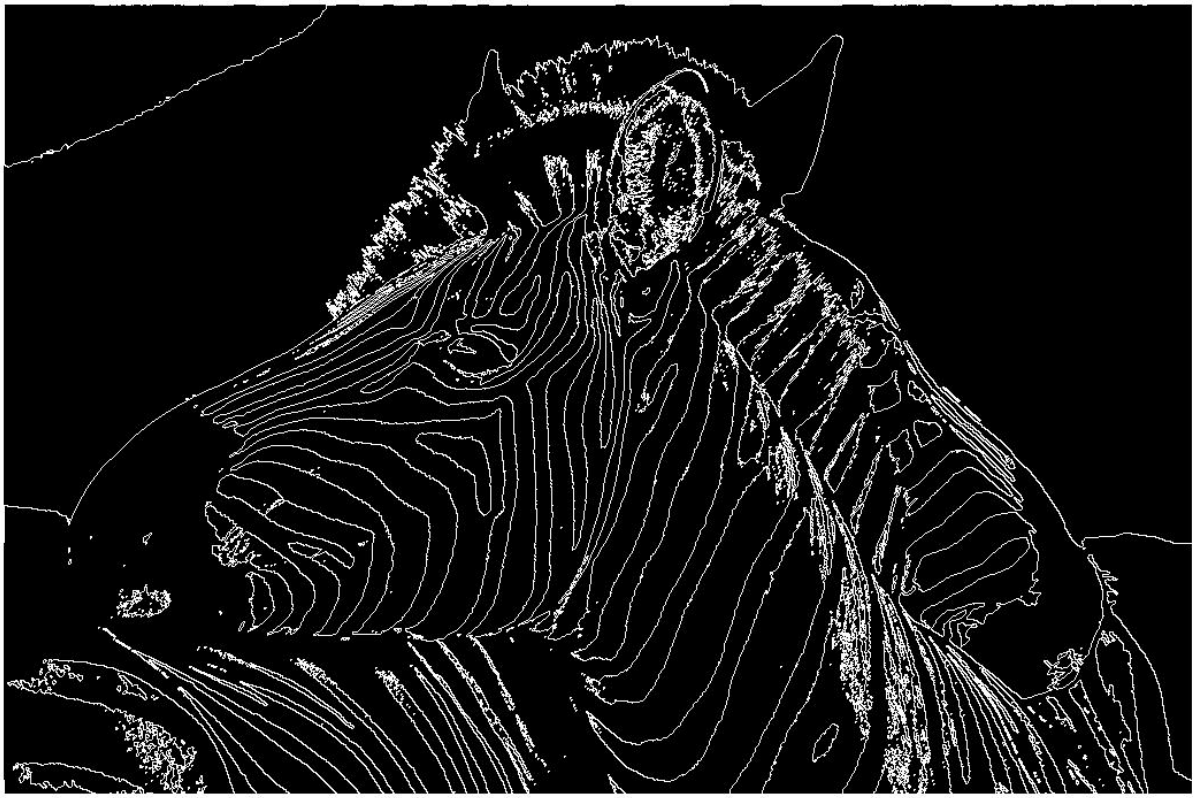
عمل مشتق گیری به این صورت اتفاق می افتد که در لبه های تصویر یا به عبارتی دگیر مکانهایی که بردار v از مقدار صفر به یک میرود یک مشتق مثبت یا همان رنگ سفید ایجاد میشود و همانطور که مشاهده میشود در نقاطی مثل 2 در بردار v که از صفر به یک رفته ایم یا به عبارتی مقدار مشتق مثبت میشود و لبه داریم یک پالس سفید داریم.

د) پس از بارگذاری عکس داده شده و سپس باینری کردن آن داریم :

ه) حال با استفاده از مقداری استفاده از امکانات متلب ماتریس مربعی w1 با مرتبه n که برابر تعداد ستون های بردار عکس است میسازیم سپس هر سطر بردار عکس را transpose کرده و در w1 ضرب کرده و حاصل را دوباره transpose میکنیم که سطری میشود و در بردار horizontal در زیر هم قرار میدهیم .

و) در این قسمت مثل قبل ماتریس w2 با مرتبه m میسازیم که m تعداد سطر های ماتریس عکس است و بر خلاف حالت قبل چون به صورت ستونی عمل میکنیم نیازی به transpose نیست . و در نهایت حاصل را به صورت گفته شده در ماتریس vertical قرار میدهیم .

ی) در این قسمت برآیند دو ماتریس verticalو horizontal با به اینصورت حساب میکنیم که برابر جزر مجموع توان دوهای دو ماتریس است و حاصل را با imshow رسم میکنیم که برابر زیر است :



ز) کاری که ما در اینجا کردیم این بود که از تصویر از دو جهت عمودی و افقی مشتق گرفتیم و سپس برآیند این دو را حساب

کردیم که حاصل برابر لبه های تصویر است . علت آن اینست که در لبه های تصویر مقدار ماتریس آن تغییر میکند ولی مکانهایی که لبه نیستند مقدار ثابت دارند و مشتق آنها برابر صفر است و سیاه نشان داده می شوند اما در لبه ها چون مقدار تغییر کرده مشتق ناصفر میشود و به صورت لبه نشان داده میشود . روش ما در اینجا به اینصورت بود که در دو جهت عمودی و افقی مشتق گرفتیم اما روش های دیگری هم وجود دارد مثلا میتوان در هر نقطه با استفاده از kernel های مختلف گرادیان گرفت و مشتق را حساب کرد و انتخاب روش بستگی به میزان دقت و خطای مطلوب ما و سرعت و مقدار محاسبات مورد نیاز و کابرد آن دارد .