

Numerical Computations

Computer Assignment 1

Professor : Jamal Kazazi

Shahaboddin Sheybani

810101454

2023

Q1)

Main Order:

\*در سوال اول برای انجام خواسته سوال ابتدا باید آرایه ای از اعداد از کابربر دریافت کرده و پس از حذف هر عضو که تکرار شده باشد،عناصر باقی مانده را به باینری تبدیل می کنیم.سپس با استفاده از تابع xor که نوشته شده است xor هر دو عدد را محاسبه می کنیم و در پایان عدد باینری حاصل را به دسیمال تبدیل می کنیم و در خروجی چاپ میکنیم.

خط 1 و 2 : حذف دستورات قبلی و داده های ذخیره شده

خط 3 : دریافت تعداد اعداد دنباله از کاربر و ذخیره در n

خط 4 : تعریف کردن ماتریس صفری A با ابعاد تعداد عناصر دنباله

حط 5 تا 7 : گرفتن اعداد دنباله از کاربر به ترتیب و ریختن در A

خط 8 : تعریف کردن ماتریس B که اعداد تکراری A در آن حذف شده با استفاده از تابع نوشته شده در انتهای فایل

خط 9 تا 11 : بررسی تهی بودن B و نشان دادن صفر در خروجی

خط 12 تا 15 : چاپ کردن xor هر دو عدد موجود در دنباله با استفاده از تابع نوشته شده در انتهای فایل

Decimal To Binary Function :

خط 17 : تعریف تابع تبدیل دسیمال به باینری با خروجی یک عدد

خط 18 و 19 : تعریف خروجی و شمارنده n برابر با صفر

خط 20 : حلقه while تا زمانی ک مقدار عدد وارد شده بزرگتر مساوی یک است

خط 21 : ذخیره باقی مانده a بر2 در k

خط 22 : آپدیت مقدار خروجی با افزودن توان های ده در k

خط 23 : آپدیت a برابر حاصل جزصحیح تقسیم a بر 2

خط 24 : افزایش مقدار شمارنده n

خط 25 و 26 : پایان تابع و حلقه while

Binary To Decimal Function :

خط 28 : تعریف تابع باینری به دسیمال یک عدد با یک خروجی

خط 29 و 30 : تعریف متغیر موقتTemp و شمارنده n برابر 0

خط 31 : حلقه while تا زمانی که مقدار عدد باینری صفر نشود

خط 32 : ذخیره باقی مانده a بر10 در k

خط 33 : آپدیت مقدار temp با افزودن توان های 2 در k

خط 34 : آپدیت مقدار a برابر با جزصحیح تقسیم a بر10

خط 35 : افزایش مقدار شمارنده n

خط 36 تا 38 : برابر قرار دادن خروجی تابع با حاصل نهایی temp و پایان تعریف تابع و حلقه while

XOR Function :

خط 40 : تعریف تابع Xor با دریافت دو عدد و خروجی یک عدد

خط 41 : مقداردهی اولیه صفر برای خروجی

خط 42 و 43 : تبدیل دو عدد دریافتی به باینری

خط 44 تا 48 : یافتن عدد بزرگتر و ذخیره در max

خط 49 : صفر قرار دادن شمارنده n

خط 50 : حلقه while تا وقتی max صفر نشده باشد

خط 51 : آپدیت متغیر max با تقسیم به ده کردن

خط 52 : افزایش مقدار n که برابر با تعداد ارقام max می باشد

خط 55 : حلقه for برای n بار و انجام xor

خط 56 و 57 : برابر قرار دادن باقی مانده a وb در kaوkb

خط 58 و 59 : در صورت برابری دو رقم سمت راست دو عدد باینری مقدار m را برابر صفر در نظر میگیریم

خط 60 و 61 : در صورت عدم تساوی دو رقم سمت راست مقدار m برابر 1 می شود

خط 64 : آپدیت مقدار خروجی(باینری) با افزودن توان های 10 در m

خط 65 و 66 : آپدیت مقدار a و b برابر با جزصحیح تقسیمشان بر 10

خط 68 : تبدیل خروجی باینری به دسیمال

خط 69 : پایان تابع

Repeated Elements Remover Function:

خط 71 : تعریف تابع یکتاساز با دریافت یک آرایه و خروجی یکتا

خط 72 : مقداردهی اولیه آرایه خروجی برابر با تهی

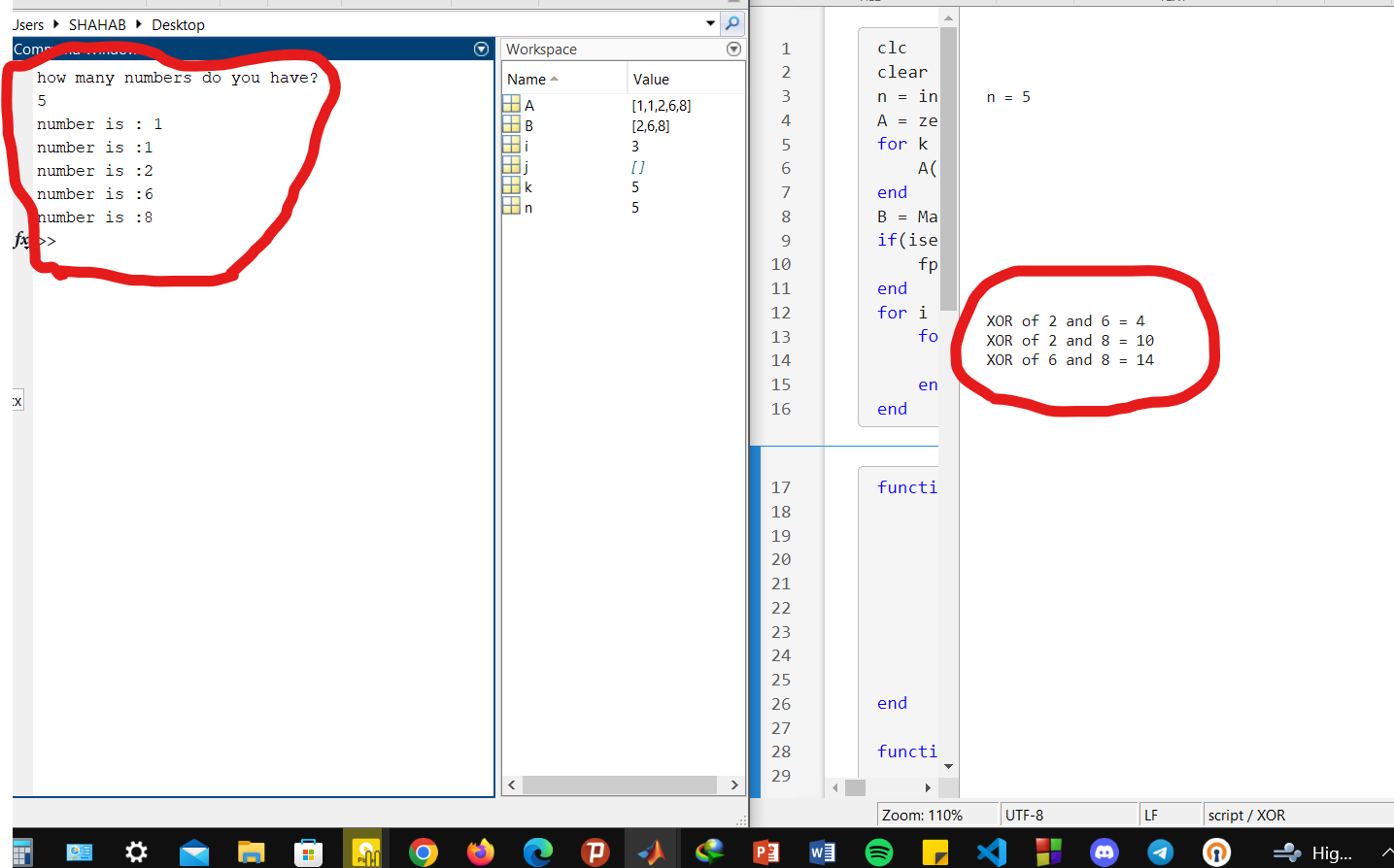
خط 73 : حلقه for از 1 تا طول آرایه داده شده

خط 74 : تابع) sum(arr(i) == arr تعداد دفعات عنصر arr(i) در کل تابع را به دست می آورد و در صورت نبود تکراری خروجی آن یک می شود

خط 75 : در صورت خروجی 1 بودن خط قبلی مقدار خروجی را آپدیت کرده و عنصر iام آرایه را به خروجی که مقدار اولیه تهی داشت می افزاید

خط 76 و 77 و 78 : پایان if و for و تابع نوشته شده

Result :



Q2a)

Main Orders :

در این سوال باید با استفاده از روش حذفی گاوس ماتریس را تبدیل به بالامثلثی کرده که این کار با دراوردن نسبت بین دو سطر مورد نیازمان انجام می دهیم و سپس آن نسبت از سطر را از سطر مورد نظر کم می کنیم تا به صفر برسیم.سپس از آخرین عنصر شروع کرده و جواب هارا به دست می آوریم.

خط 1 و 2 : دستورات اولیه حذف دیتای قبلی و دستورات قبلی

خط 3 و 4 : تعریف ماتریس های ضرایب و معلومات داده شده

خط 5 : برابر قرار دادن خروجی تابع gauss که در اینجا یک ماتریس 1 در 3 می باشد که به ترتیب ضرایب t^2 و t و 1 می باشند.

خط 6 و 7 : تعریف متغیر t و V(t)

خط 7 : مقدارهی دهی اولیه صفر برای تابع V

خط 8 و 9 : حلقه for برای تغییر از یک تا n(طول ماتریس معلومات)

خط 10 : آپدیت تابع V با افزودن ضرب توان های t در عناصر خروجی

خط 12 : محاسبه مقدار سرعت در ثانیه 10

خط 13 : رسم نمودار V(t) به دست آمده

Gauss Function :

خط 14 : تابع گاوس با دریافت دو ماتریس یک خروجی خواهد داشت

خط 15 : برابر قرار دادن متغیر n با طول ماتریس معلومات

خط 16 : ترکیب دو ماتریس A وb با قرار دادن آنها کنار هم

خط 17 : تعریف ماتریس جواب به صورت ماتریس صفری n عضوی

خط 18 و 19 : دو حلقه for برای تغییر ماتریس ها و بالامثلثی شدن

خط 20 : محاسبه نسبت عنصر مورد نظر برای صفر شدن به عنصر بالایی خود

خط 21 : کم کردن نسبت به دست آمده در سطر بالایی از سطر مورد نظر برای صفر شدن عنصر دلخواه

خط 22 و 23 : پایان دو حلقه for

خط24تا 26 : ماتریس B را همانند ستون آخر ماتریس ترکیب A,B می کنیم

خط 27 : حلقه for برای به دست آوردن ماتریس مجهولات که از طول آرایه شروع کرده و با کم کردن تک تک i ها پاسخ را به دست می آوریم

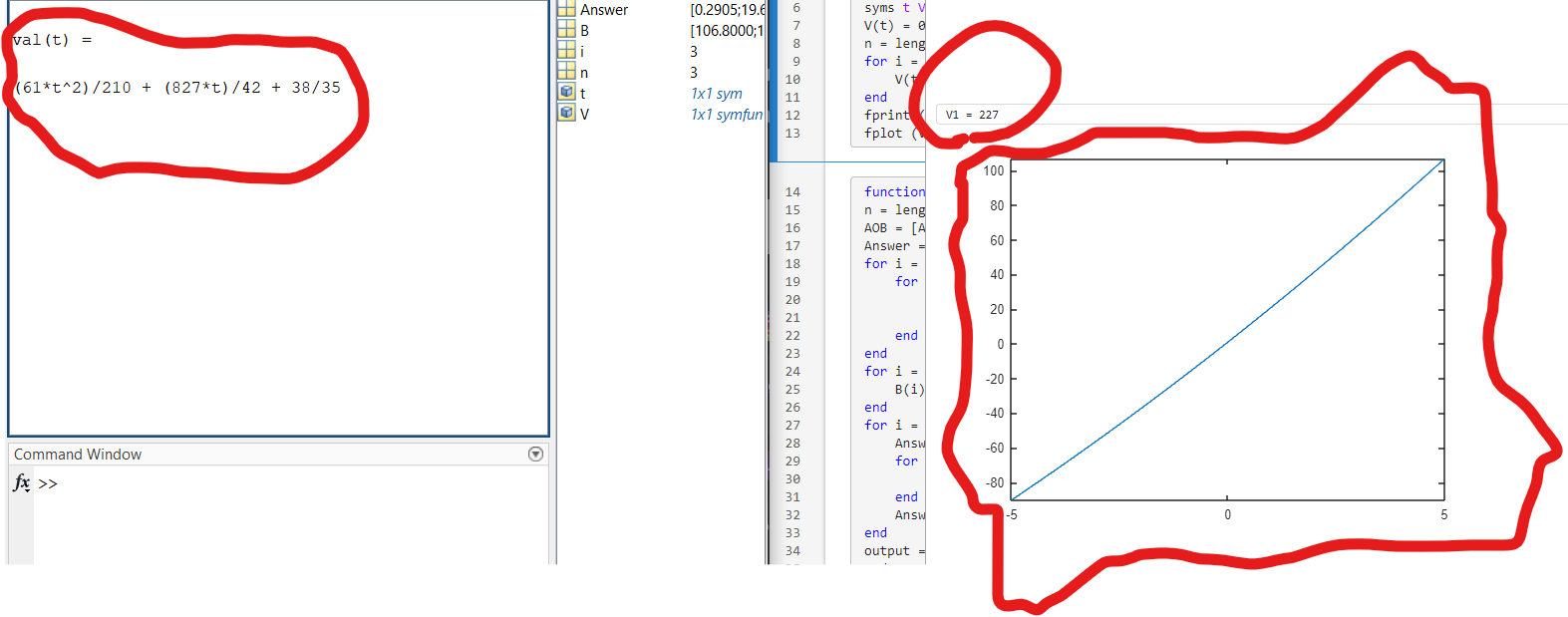
خط 28 : ابتدا عنصر i خروجی را برابر مقدار آن عنصر در ماتریس معلومات قرار می دهیم

خط 29 تا 31: مقدار ضرب ماتریس معلومات در پاسخ به دست آمده را با این FOR از پاسخ مرحله iام کم می کنیم

خط 32 : برای محاسبه نهایی خروجی هر کدام از عناصر answer را بر درایه نظیرش روی ماتریس معلومات بالامثلثی تقسیم میکنیم

خط 34 و 35 : خروجی را برابر ماتریس Answer قرار می دهیم و پایان

Result :



Q2b)

Main Orders :

طبق روش ژاکوبی که روشی تکراری است x مرحله nام را با استفاده از سایر متغیر های مرحله n-1ام به جواب می رسیم که روش محاسبه در میان خطوط قرار داده شده.

خط 1 تا 4 : دستورات حذف دیتا و دستورات قبلی و مقداردهی A وb

خط 5 : مساوی قرار دادن Answer که ماتریس ضرایب توان هایt می باشد با خروجی تابع Jacobi

خط 6 : تعریف متغیر n برابر با طول ماتریس معلومات

خط 7 : تعریف متغیر های t,V(t)

خط 8 : مقداردهی اولیه v(t) برابر صفر

خط 9 تا 11 : آپدیت مقدار تابع با افزودن عناصر خروجی ضرب در توان های t زیرا خروجی برابر ضرایب t^2 , t , 1 هستند

خط 12 : مقدار v2 را برابر V(10) قرار می دهیم

خط 13 : دستور رسم تابع V(T)

Jacobi Function :

خط 14 : تعریف تابع ژاکوبی با دریافت دو آرایه و تعداد تکرار

خط 15 : تعریف متغیر n برابر با طول ماتریس B

خط 16 و 17 : تعریف دو آرایه x و x جدید به طول n

خط 18 : حلقه for بیرونی برای انجام دادن محاسبات و به دست آوردن جواب ها بعد از برای مثال 20 بار

خط 19 تا 21 : آرایه جواب جدید به صورت زیر تعریف شده است

اختلاف عنصر نظیر عنصر jام در ماتریس معلومات و حاصل ضرب سایر ریشه ها در ضریب نظیرشان

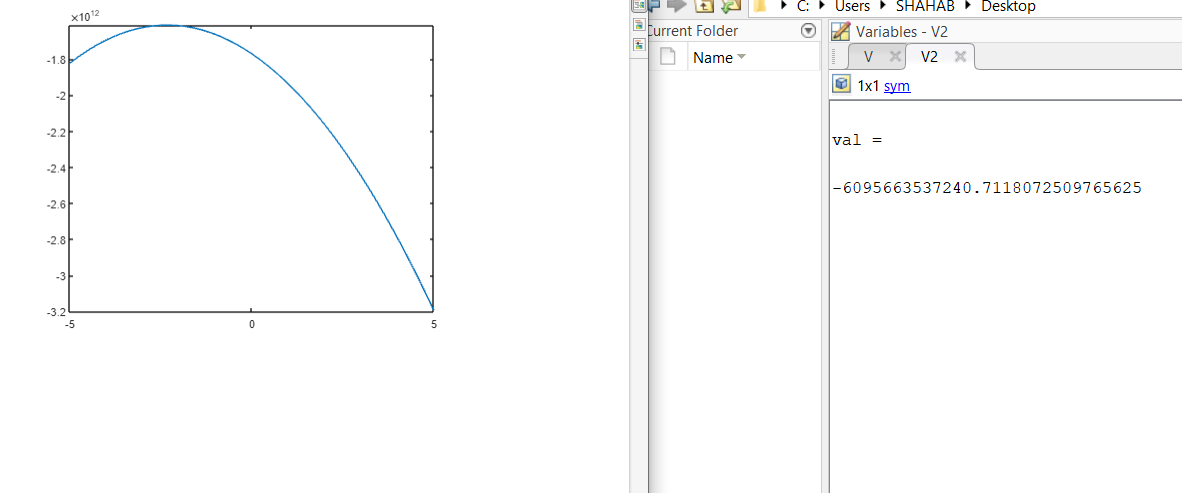
دستور A[1:J-1 ;J+1:n] این کار را انجام می دهد که همه به جز همان عنصر jام را به دست آورده و محاسباتی انجام می دهد.

خط 22 : ماتریس جواب های قبلی برابر ماتریس جواب های جدید قرار داده می شود و این فرآیند به تعداد تکرار گفته شده انجام می شود

خط 23 تا 24 : خروجی تابع برابر ماتریس جواب های جدید (x\_new) قرار داده می شود و پایان تابع

با توجه به خروجی این روش همگرا نمی باشد و نیازمند محورگیریست

Result :



Q2c)

\*در این سوال باید با روش لاگرانژ ضرایب را به دست آوریم

که در آن Ln از رابطه (x-x0)(x-xn-1)(x-xn+1)…/(xn – x0)(xn – ..) به دست می آید.

F(x) = L1\*F(1) + L2\*F(2) + ….

Main Orders :

خط 1و2 : حذف دیتا و دستورات قبلی

خط 3و4 : تعریف داده های صورت سوال در دو ماتریس

خط 5 : تعریف تابع v(x) که به صورت متغیر تعریف شده است

خط 6 : برابر قرار دادن v(X) با خروجی تابع لاگرانژ که فراخوانی می شود

خط 7 : محاسبه مقدار سرعت در ثانیه 10 و ذخیره در v3

خط 8 : دستور رسم نمودار V(x)

Lagrange Function :

خط 10 : تعریف تابع لاگرانژ که با دریافت دو ماتریس اعداد یک تابع را در خروجی نمایش می دهد

خط 11 : برابر قرار دادن n با طول آرایه اعدادمان

خط 12 : تعریف متغیر x و V(x)

خط 13 : مقداردهی اولیه تابع برابر صفر

خط 14 : حلقه for که به ازای تعداد داده هایمان ضرایب را به دست بیاورد

خط 15 : مقداردهی اولیه ضریب L برابر با 1

حط 16 : حلقه For دوم که برای L مرحله iام آن را در x-a / x0 – a ضرب کند

خط 17 : شرطی که مقدارx همان مرحله را انجام ندهد

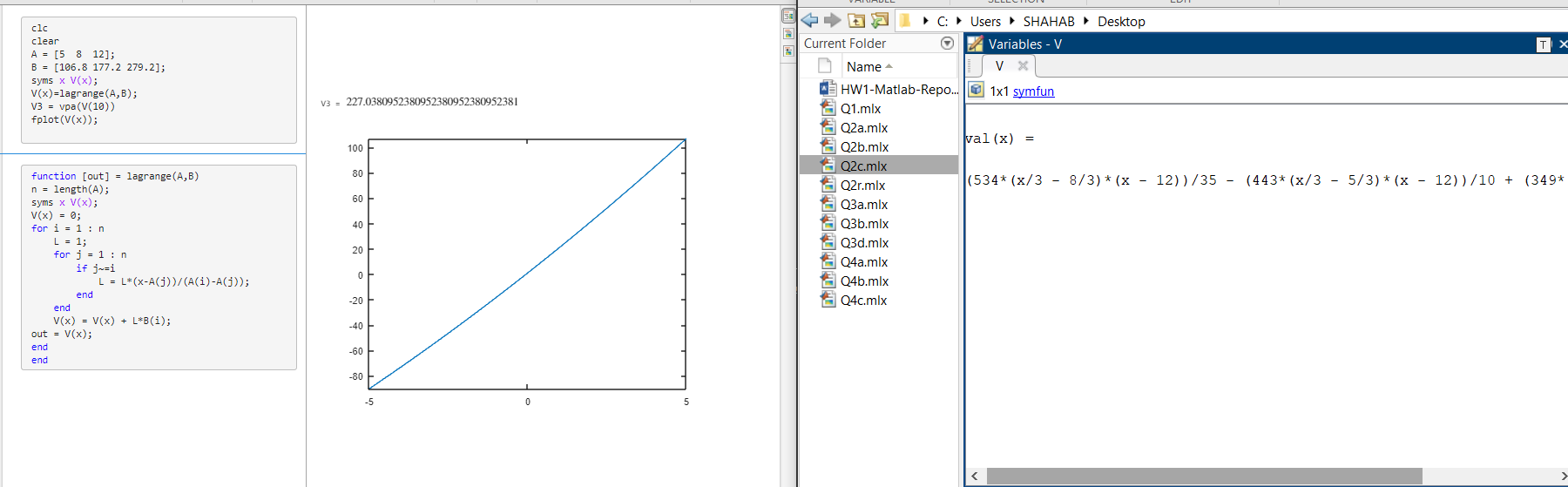
خط 18 : آپدیت مقدار L

خط 19 و 20 : پایان دو حلقه for

خط 21 : آپدیت مقدار V(x) با افزودن همان ضریب Li\*Fi

خط 22 : برابر قرار دادن خروجی با V(x)

Result :



Q2h)

در هر یک از قسمت های a,b,c بعد از انجام خواسته های آن بخش نمودار تابع رسم شده است و خروجی آن در هر قسمت مشخص شده است

Q2v)

در هر یک از قسمت های a,b,c بعد از انجام خواسته های آن بخش مقدار سرعت در t = 10 به دست آمده است و خروجی آن در هر قسمت نوشته شده است

Q2r)

خط 1 تا 4 : دستوران اولیه و تعریف ماتریس AوB که چون 4 نقطه داریم باید ماتریس هایی به طول 4 داشته باشیم

خط 5 تا 13 : همان دستورات بخش a که در قسمت main وارد شد در اینجا هم وارد شده است و در انتها مقدار V(10) محاسبه شده است و نمودار رسم شده است

خط 16 تا 24 : همان دستورات بخش b که در قسمت main وارد شد در اینجا هم وارد شده است و در انتها مقدار V(10) محاسبه شده است و نمودار رسم شده است

خط 27 تا 32 : ماتریس های A,B جدیدمان مشخص شده و همان دستورات بخش c که در قسمت main وارد شد در اینجا هم وارد شده است و در انتها مقدار V(10) محاسبه شده است و نمودار رسم شده است

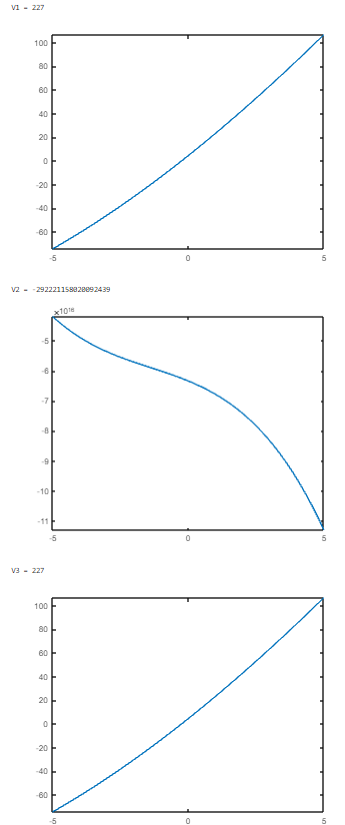
خط 34 تا 55 : تابع روش حذفی گاوس

خط 57 تا 68 : تابع روش ژاکوبی

خط 70 تا 84 : تابع روش لاگرانژ

توضیحات توابع در قسمت مربوط به خودشان نوشته شده است

Result :



Q3a)

Main Orders :

خط 1 و 2 : دستورات حذف دیتا و دستورات قبلی

خط 3 : تعریف x تحت عنوان متغیر

خط 4 : تعریف تابع f1

خط 5 : دستور subplot و رسم به شکل 5\*3 )دو آرگومان اول) و نمودار شماره 1 همین تابع f1 می باشد

خط 6 : دستور رسم تابع f1

خط 7 : نام نموداری که رسم کردیم تا در خروجی چاپ شود

خط 9 : حلقه for که به ازای n های مختلف نمودار را رسم کند

خط 10 : تعریف تابع func برابر با f2(x,i) i همان نقش n را بازی می کند

خط 11 : شماره ادامه نمودارها با دستور subplot که نمودار ها باید از 2 به بعد باشد پس در حلقه i-2 قرار می دهیم

خط 12 : دستور رسم تابع func

خط 13 و 14 : نام نموداری که رسم شده به همراه شماره n زیگما

خط 17 : حلقه for که به ازای n های مختلف نمودار را رسم کند

خط 18 : تعریف تابع deltaF برابر با قدرمطلق f2 – f1

خط 19 : شماره ادامه نمودارها با دستور subplot که نمودار ها باید از 9 به بعد باشد پس در حلقه i+5 قرار می دهیم

خط 20 : دستور رسم تابع deltaF به ازای هر i

خط 21 و 22 : نام نموداری که رسم شده به همراه شماره n زیگما

Sigma Function :

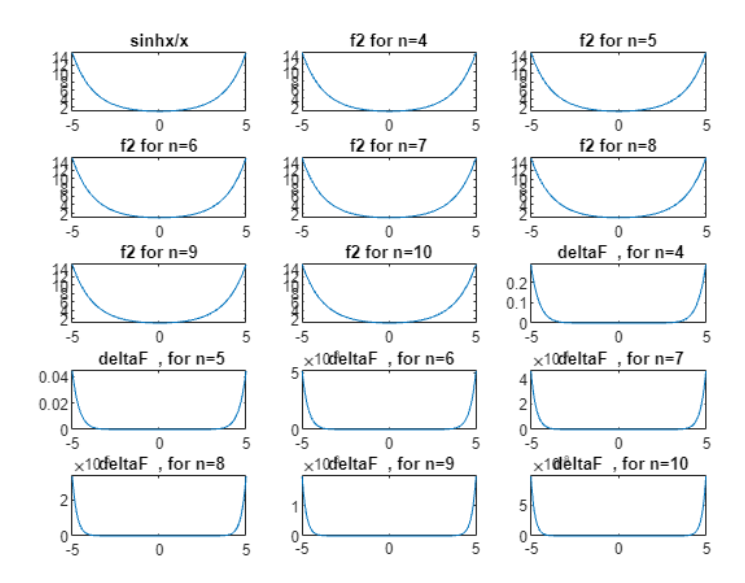
خط 24 : تعریف اولیه تابع که یک x و یک n در آرگومان دارد

خط 25 : مقداردهی اولیه خروجی برابر با صفر

خط 26 تا 28 : حاصل سیگما را از 0 تا n محاسبه می کند و مقدار خروجی را آپدیت می کند

خط 29 : دستور پایان تابع نوشته شده

Result :



Q3b)

خط 1 و 2 : حذف دیتا و دستورات قبلی

خط 3 : تعریف متغیر x

خط 4 : تعریف تابع f1

خط 6 : متغیر limit\_f1 را برابر خروجی تابع limit(f1,0) قرار می دهیم

خط 7 : چاپ مقدار حد تابع f1

خط 9 : حلقه for که به ازای n های مختلف مقدار حد تابع f2 را محاسبه کند

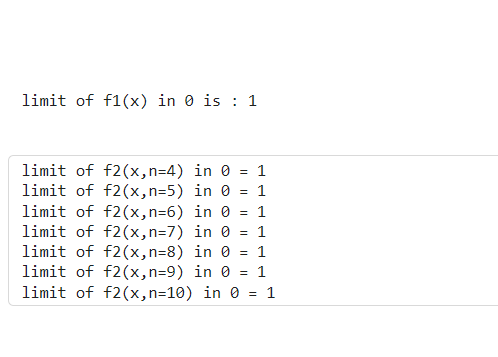
خط 10 : متغیر limit\_f2 را برابر خروجی تابع limit(f2,0) قرار داده

خط 11 : چاپ مقدار حد در هر حلقه به ازای n های مختلف

خط 12 : پایان حلقه for

خط 14 تا 19 : تعریف تابع f2 که در قسمت قبل توضیح داده شد

Result :



Q3c)

با افزایش مقدار n نمودار f2 به نمودار f1 نزدیک تر میشود زیرا f2 برابر بسط تیلور تابع sinhx/x می باشد و دقت بالا رفته است

Q3d)

Main orders :

خط 1 و 2 : دستورات حذف دیتا و دستورات قبلی

خط 3 : تعریف متغیر x

خط 4 : تعریف تابعی که میخواهیم ریشه آن را به دست آوریم

خط 5 : تابع new که همان نیوتون رافسون باشد را محاسبه میکنیم به ازای ورودی خطا = 0.0005 و نقطه شروع 10 و تابع f

Newton Raphson Function :

خط6 : تعریف اولیه تابع نیوتون با دریافت مقدار خطا ، نقطه اولیه ، تابع و متغیر

خط7 : مقدار previous را برابر نقطه اولیه قرار می دهیم

خط8 : خروجی تابع را برابر اختلاف نقطه قبلی و مقدار تابع به مقدار مشتقشان در نقطه اولیه قرار می دهیم

Subs(F,x) مقدار F در x را محاسبه می کند

خط 9 : حلقه while را نوشته و تا وقتی اختلاف نقطه قبلی و جدید از خطا بیشتر باشد محاسبات ادامه پیدا می کند

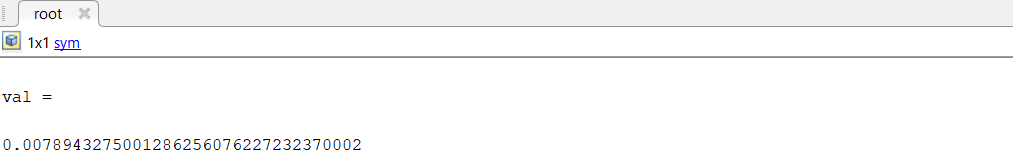
خط10 : نقطه قبلی برابر خروجی قرار می گیرد

خط 11 :از این دستور استفاده کرده تا با 16 رقم اعداد را داشته باشیم

خط 12 : خروجی برابر اختلاف مورد نظر در قسمت بالا قرار گرفته و این فرآین درون حلقه تکرار می شود

خط 13 و 14 : پایان حلقه while و تابع

خط 16 تا 21 : تعریف تابع f2 یا همان سیگما که توضیحات آن در قسمت اول قرار دارد

Result : 

Q4a)

Main Orders :

* مقدار os ها برابر (9.5 – 5)/0.05 یعنی 90 حالت می باشد پس باید ماتریس ها 90 تایی تعریف شوند

خط 1و2 : حذف دیتا و دستورات قبلی

خط 3 تا 7 : مقداردهی اولیه ضرایب ثابت

خط 8 : تعریف یک ماتریس که مقدار d ها را به ازای هر os ثبت کند

خط 9 : مقداردهی شمارنده i

خط 10 : تعریف یک ماتریس که مقادیر os ها در آن ثبت شود

خط 11 : یک حلقه for که مقدار های مختلف os را در تابع قرار دهد و با شروع از 5 و تغییر 0.05ی تا 9.5 حرکت خواهد کرد

خط 12 : تعریف تابع f که متغیر آن d باشد و با روش نابجایی به جواب برسیم

خط 13 : عنصر iام ماتریس d را برابر خروجی تابع نابجایی قرار می دهیم و درواقع تابع را با شروع از نقاط a,b و تکرار برای تابع f فراخوانی میکنیم

خط 14 : عنصر i ماتریس svalues را هم برابر os آن مرحله قرار داده

خط 15 : شمارنده را آپدیت می کنیم

خط 16 : پایان حلقه for

خط 17 : دستور plot را برای دو ماتریس svaluesوd و نقاط متناظرشان رسم کرده

خط 18 تا 20 : مشخص کردن نام محور های مختصات و نام نمودار

InPlace Function :

خط 21 : تعریف تابع نابجایی که با دریافت دو نقطه شروع و تابع F و تعداد تکرار ریشه یک خروجی خواهد داشت

خط 22 : حلقه for که با شروع از 1 و تا مقدار تکرار فرآیند زیر را تکرار کند

خط 23 و 24 : در ابتدا مقدار ai,bi را برابر مقدار ورودی تابع قرار داده

خط 25 : ریشه مرحله iام هم طبق فرمول روش نابجایی که از رابطه زیر به دست می آید را می نویسیم xn = af(b) – bf(a) / f(b) – f(a)

خط 26و27 : حال در صورتی که مقدار تابع در نقطه ri دقیقا برابر صفر شد از حلقه خارج می شویم چون ریشه دقیق را داریم

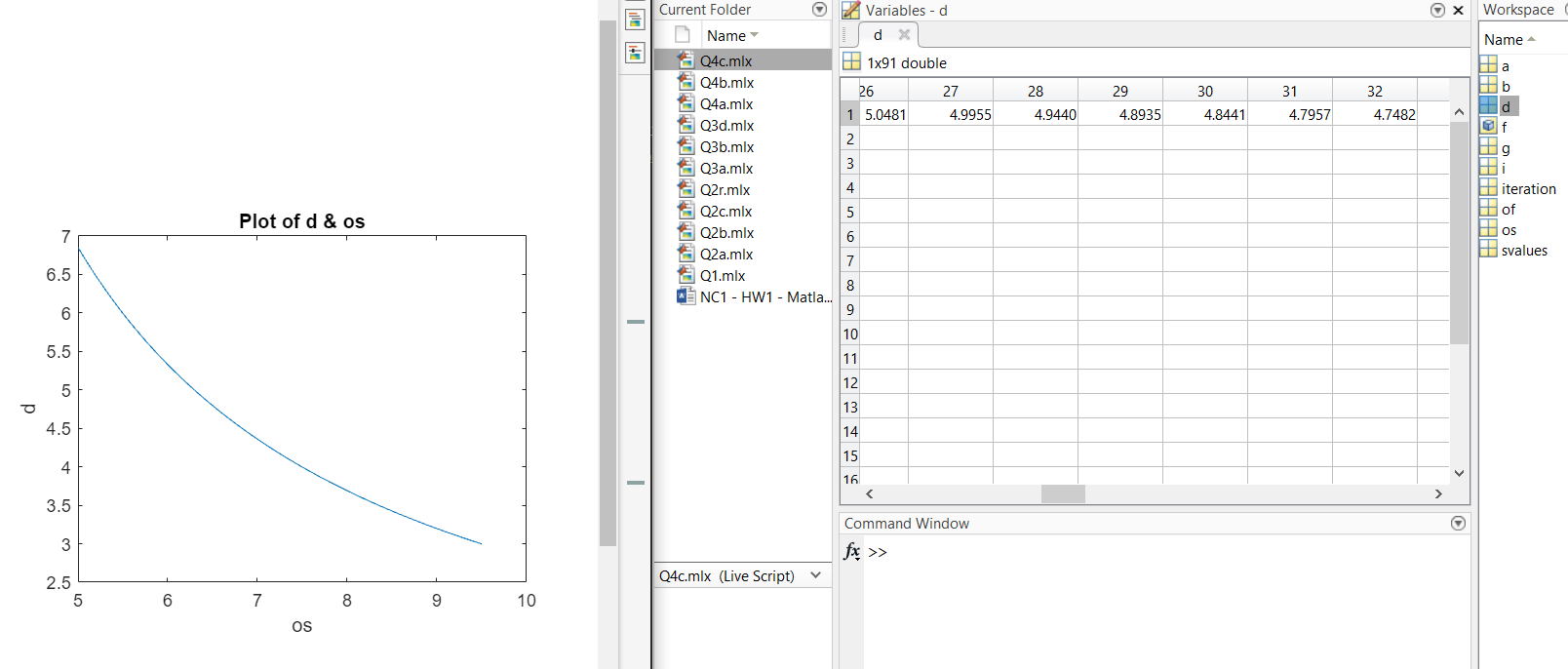
خط 28 و 29 : طبق فرمول هایی که داشتیم اگر مقدار f(xn)\*f(a) مثبت بود ریشه در بازه (ri,bi) قرار دارد پس مقدار a را برابر ri قرار داده

خط 30 و 31 : در غیر این صورت یعنی اگر حاصل آن ضرب منفی بود ریشه در بازه(a,ri) قرار دارد پس مقدار b برابر ri می شود

خط 32 تا 35 : پایان حلقه ها و تابع و اینکه خروجی برابر ri قرار می گیرد و خروجی پس از تعداد مشخصی تکرار به دست می آید

Result :

مقادیر d به ازای os های مختلف در ماتریس ذخیره شده و نمودار آنها برحسب یکدیگر رسم شده است



Q4b)

Main Orders:

فرق روش دوبخشی با نابجایی فقط در فرمول محاسبه ریشه مرحله n ام می باشد و سایر موارد کاملا یکسان است یعنی توضیحات قسمت دستورات اصلی در این بخش کاملا شبیه بخش قبلی می باشد

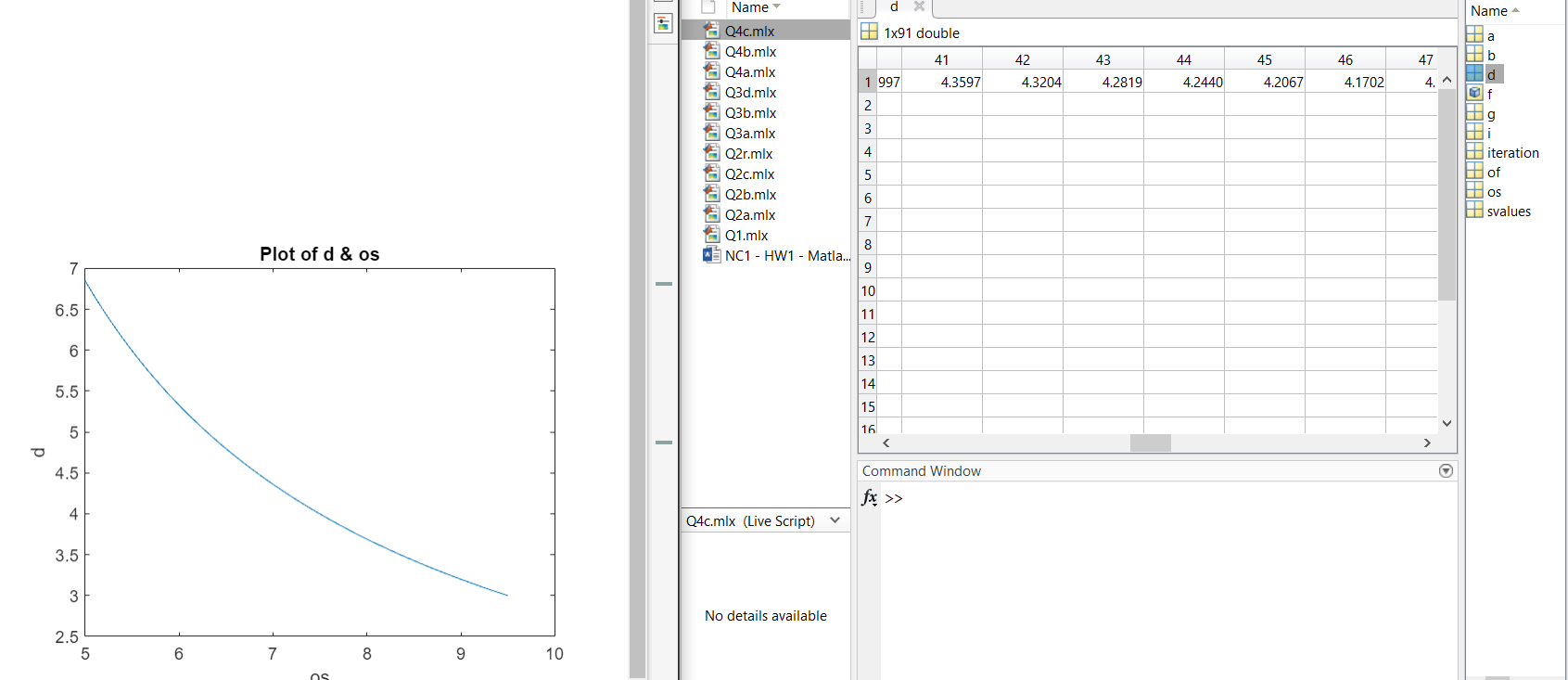
HalfRoot Function:

تابع روش دوبخشی نیز به مانند تابع روش نابجایی می باشد و توضیح جدیدی نمی توان در این قسمت اضافه کرد و همه چیز مشابه است

فقط در خط 25 چیزی که تغییر می کند نحوه محاسبه xn یا همان ri ما می باشد

که در روش دوبخشی از رابطه Xn = (ai + bi) / 2 به دست می آید

Result :



Q4c)

خط 1 تا 15 : اجرای حل تابع به روش نابجایی که توضیحات آن در قسمت خودش به طور کامل گفته شده است

خط 16 تا 26 : اجرای حل تابع به روش دوبخشی که توضیحات آن در قسمت خودش به طور کامل گفته شده است

خط 28 : d را برحسب سایر موارد پارامتری میکنیم که مستقیم به ریشه برسیم و تابعی برحسب s می نویسیم

خط 29 : تعریف ماتریسی که مقادیر d را ذخیره کند

خط 30 : تعریف شمارنده i برابر 1

خط 31 : تعریف ماتریسی که همه مقادیر s را در آن ذخیره کنیم و در هنگام رسم نمودار استفاده کنیم

خط 32 : حلقه for برای تغییر مقادیر s در بازه مشخص

خط 33 : مقادیر را تا 16 رقم ذخیره کنیم

خط 34 : ماتریس d3 را هر مرحله با افزودن مقدار D(s) آپدیت میکنیم

خط 35 : آپدیت شمارنده i

خط 36 : آپدیت ماتریس svalues با افزودنs های جدید

خط 39 و 40 :تعریف ماتریس های مقادیر خطا در هر روش با کم کردن ماتریس مقادیر دقیق d از d های دیگر روش

خط 41و42 : رسم نمودار خطاها با محوریت s

\*در ادامه تابع های استفاده شده در قسمت قبل نوشته شده است که توضیحات آن موجود می باشد.

مودار خطای دو روش رسم شده که دارای مقیاس کوچکیست Result :

