

گزارش تمرین سری دوم

شبیه سازی رایانه ای در فیزیک

علیرضا رضایی

97100762

## فهرست

3	مقدمه ای بر تمرین 2.5 و 2.6
3	تمرین 2.5
4	نتایج
23	تمرین 2.6
24	نتایج
41	تمرین 2.7
42	نتایج
47	تمرین 3.1
48	نتایج
51	نتایج عددی

## مقدمه ای بر تمرین 2.5 و 2.6

برای این دو تمرین چون خیلی شبیه به هم بودند یک مازول ساختیم که کلاسی به اسم کانفیگ دارد که تنظیمات مقدماتی ای مانند اندازه ی صفحه و تعداد نقاط اولیه و... را در خود دارد و تابعی به نام مین لوپ که مجموعه ای از توابع را ورودی می گیرد و آنها را بطور رندوم روی تمام نقاط فضا اعمال می کند و بعد از هر مرحله اعمال توابع روی کل نقاط ، نقاط را می کشد.

کلاسی برای مدیریت نقاط و کلاسی هم برای مدیریت دکمه ی شروع با نقاط جدید در کد پیاده سازی شده است.

نقاط شروع نه تنها هر بار بطور رندوم انتخاب می شوند بلکه از تابع توزیع های متفاوتی هم می آیند که در کد با کامنت گذاری مشخص شده اند.

برای کشیدن نقاط از پای گیم استفاده شده است.

توضیحات جزئی به صورت کامنت گذاری در کد قرار گرفته اند.

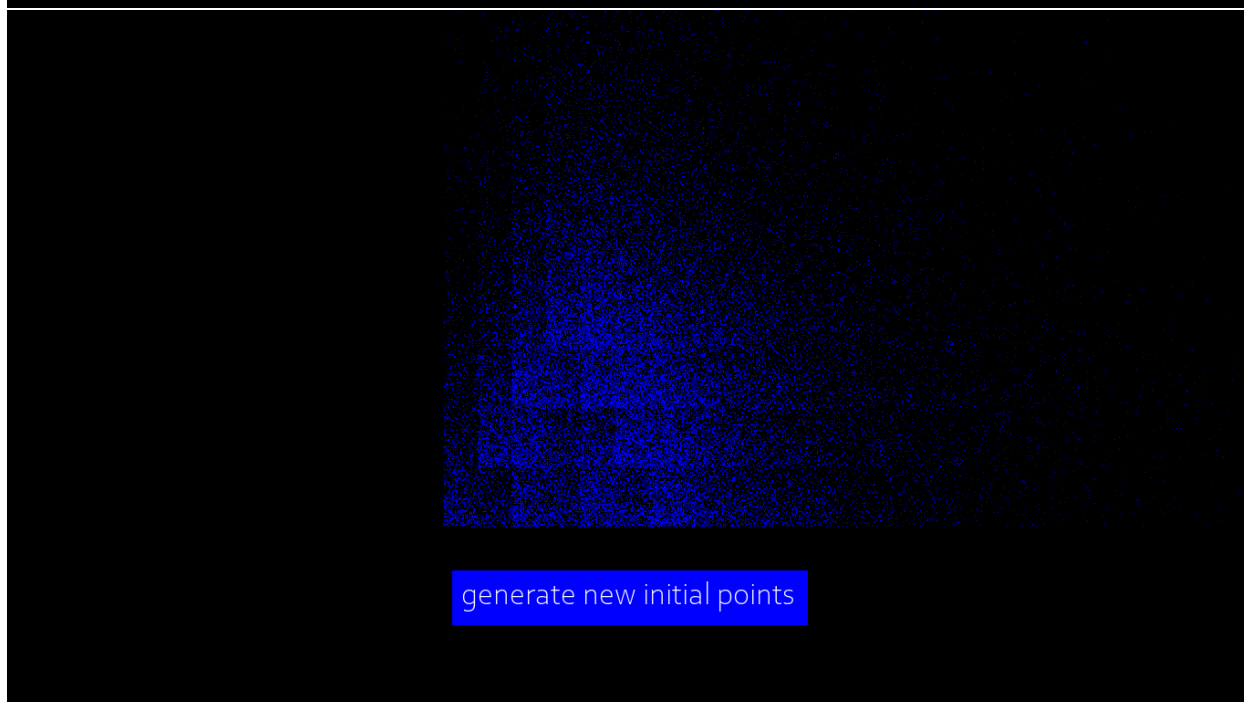
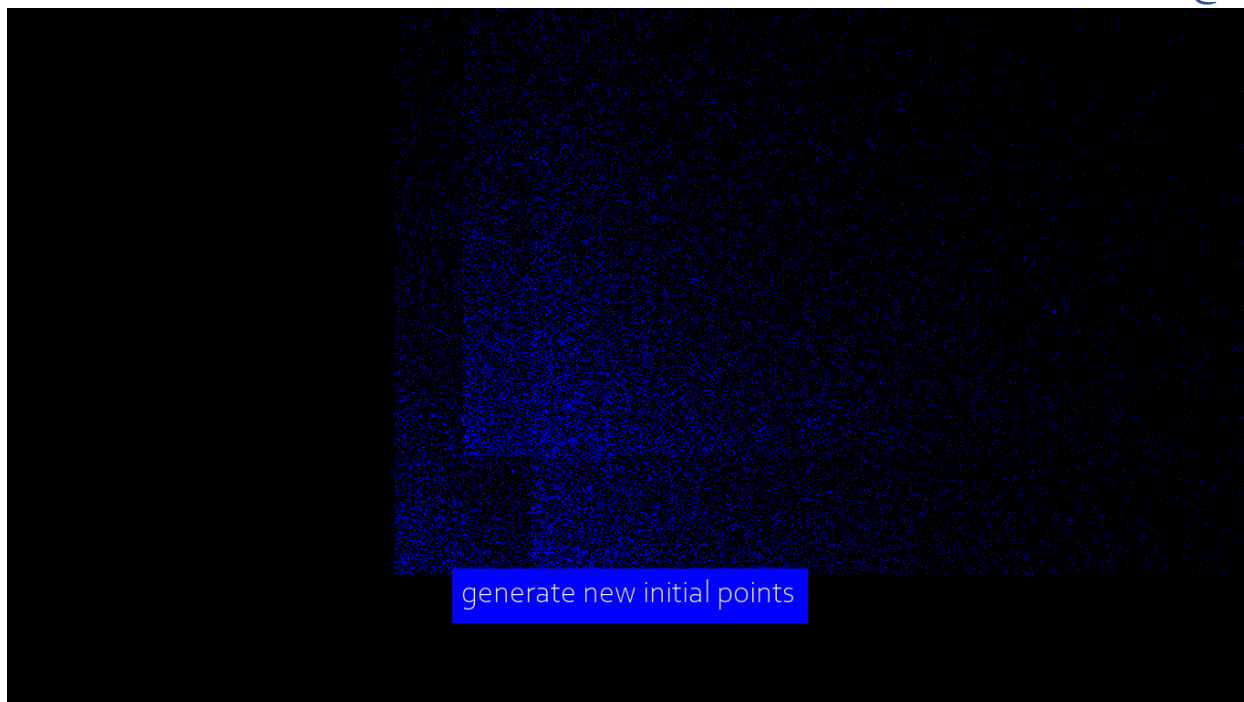
برای این دو تمرین یک کلیپ هم ضبط شده که در پوشه ی اصلی موجود است که دیدن آن خالی از لطف نیست. (فقط دقت شود که این فیلم در قسمت برگ سرخس از توابع ویرایش نشده ضبط شده بود که در کد نهایی و عکس هایی که از نتیجه قرار دادیم از کد های ویرایش شده استفاده شده است.)

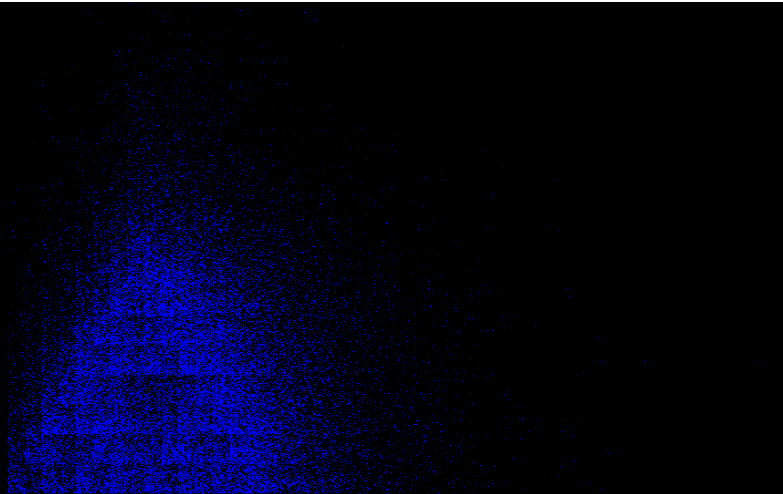
## تمرین 2.5

برای این تمرین به سه تابع نیاز داریم:

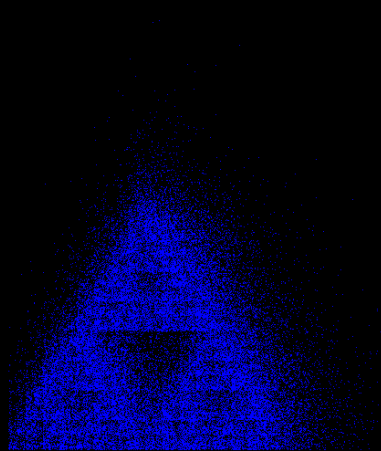
- 1- همه ی نقاط را با نسبت 0.5 کوچک (اسکیل) می کنیم تا مثلث پایین سمت چپ تشکیل شود.
- 2- همه ی نقاط را با نسبت 0.5 کوچک کرده و به اندازه ی ربع ضلعی که می خواهیم ضلع مثلثمان باشد با راست و با اندازه ی سینوس 60 درجه ضربدر نصف اندازه ی ضلع مثلث به بالا انتقال می دهیم تا مثلث بالایی تشکیل شود.
- 3- همه ی نقاط را با نسبت 0.5 کوچک کرده و به اندازه ی نصف ضلع مثلثی که می خواهیم به سمت راست انتقال می دهیم تا مثلث پایین سمت راست تشکیل شود.

این سه تابع را به تابع مین لوپ از مازولی که ساختیم پاس می دهیم و بقیه ی کار ها (که در مقدمه توضیح داده شده) انجام می شود.

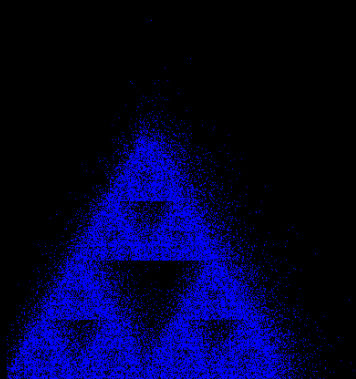




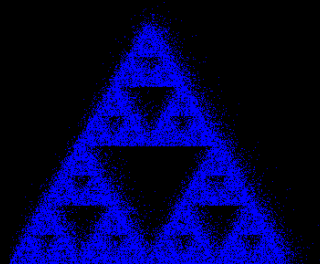
generate new initial points



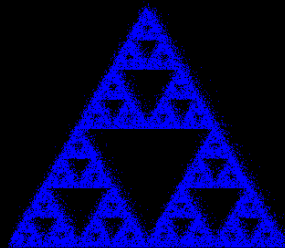
generate new initial points



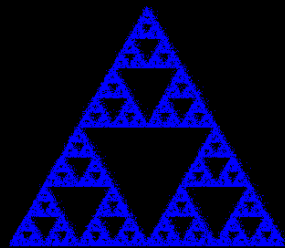
generate new initial points



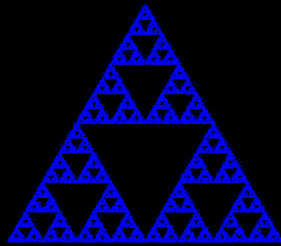
generate new initial points



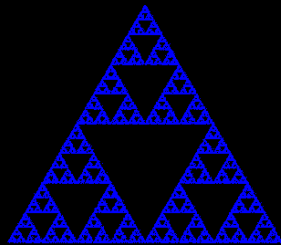
generate new initial points



generate new initial points

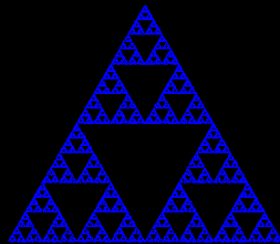


generate new initial points

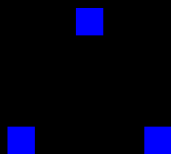


generate new initial points

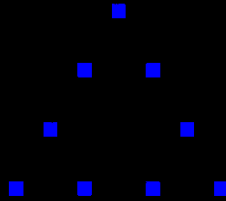




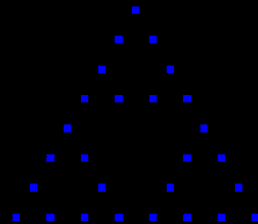
generate new initial points



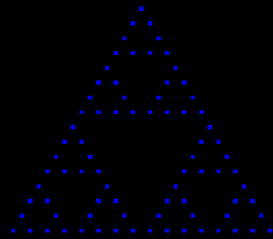
generate new initial points



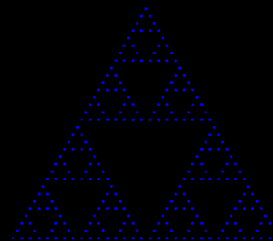
generate new initial points



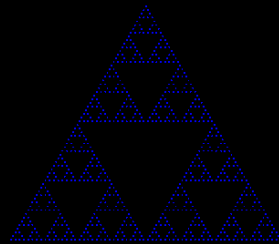
generate new initial points



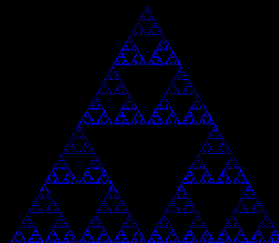
generate new initial points



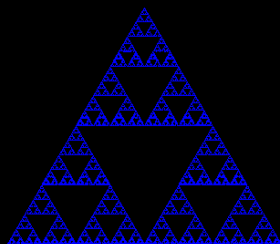
generate new initial points



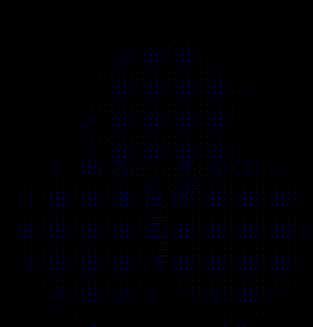
generate new initial points



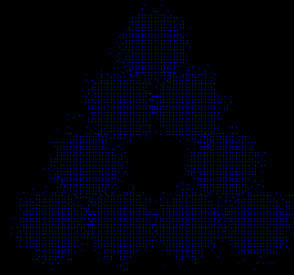
generate new initial points



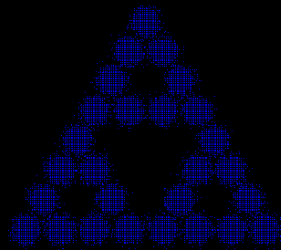
generate new initial points



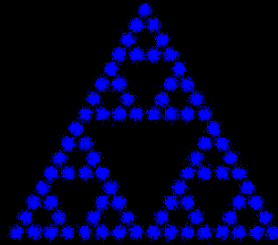
generate new initial points



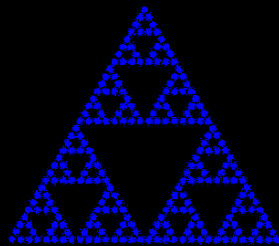
generate new initial points



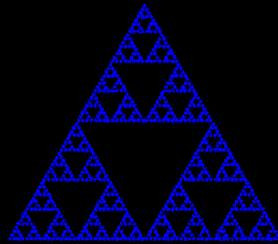
generate new initial points



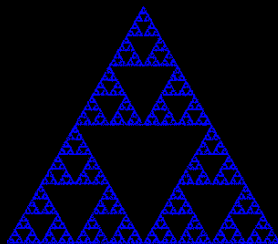
generate new initial points



generate new initial points

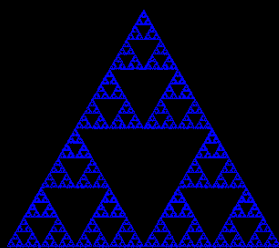


generate new initial points

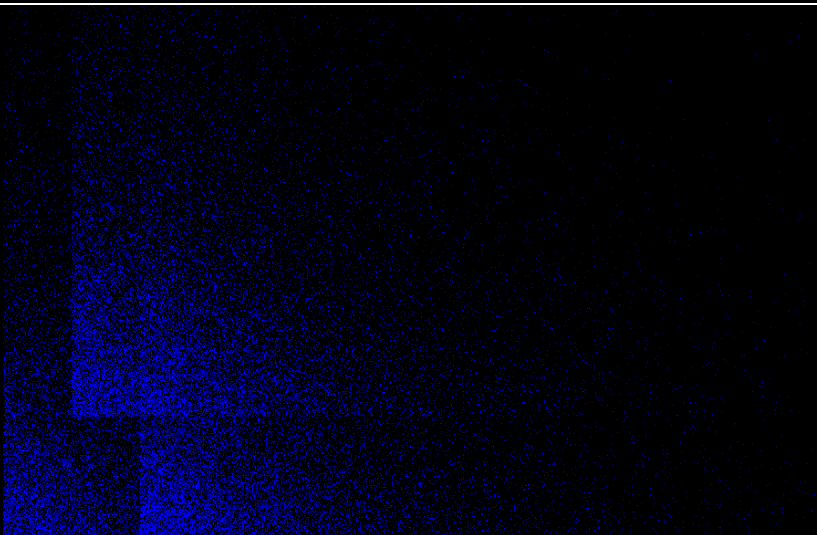


generate new initial points

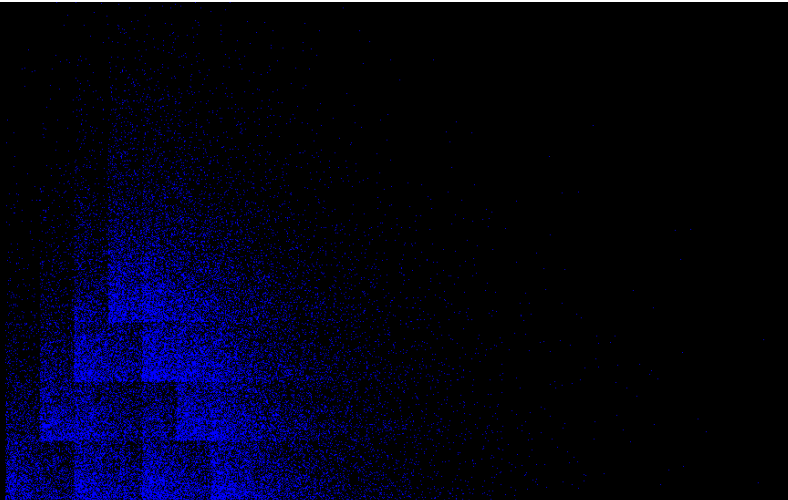




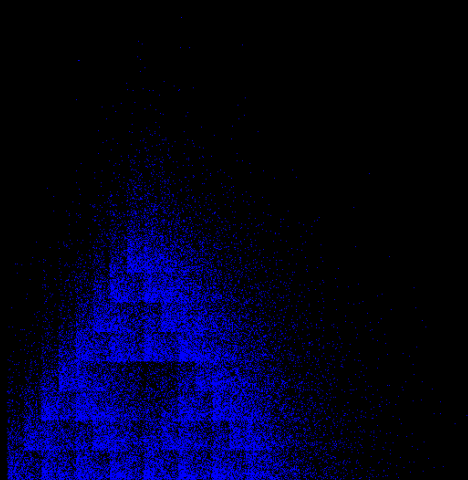
generate new initial points



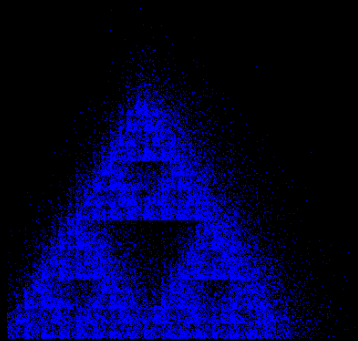
generate new initial points



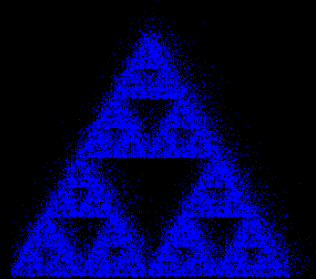
generate new initial points



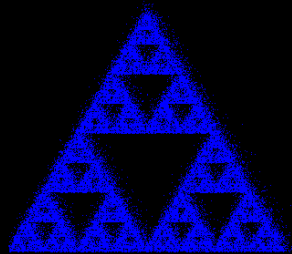
generate new initial points



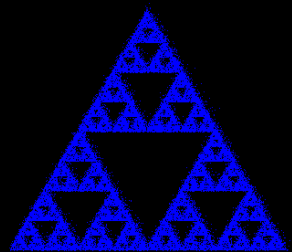
generate new initial points



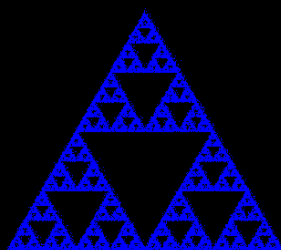
generate new initial points



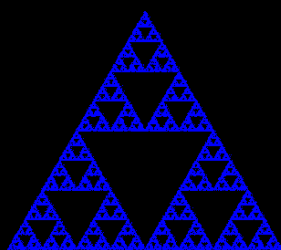
generate new initial points



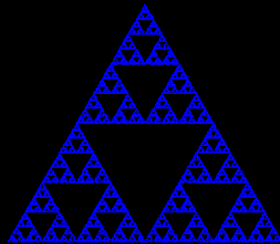
generate new initial points



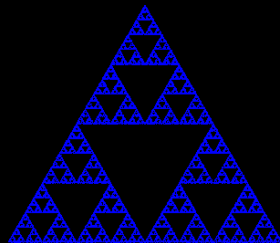
generate new initial points



generate new initial points



generate new initial points



generate new initial points

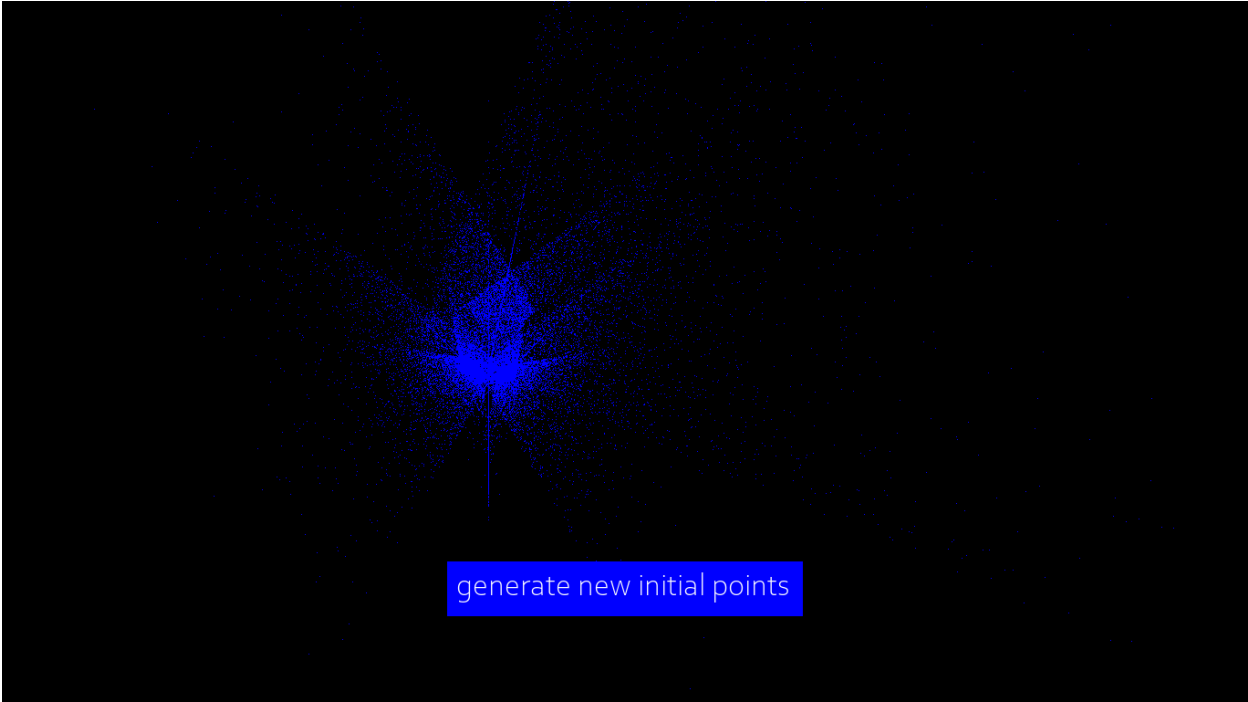
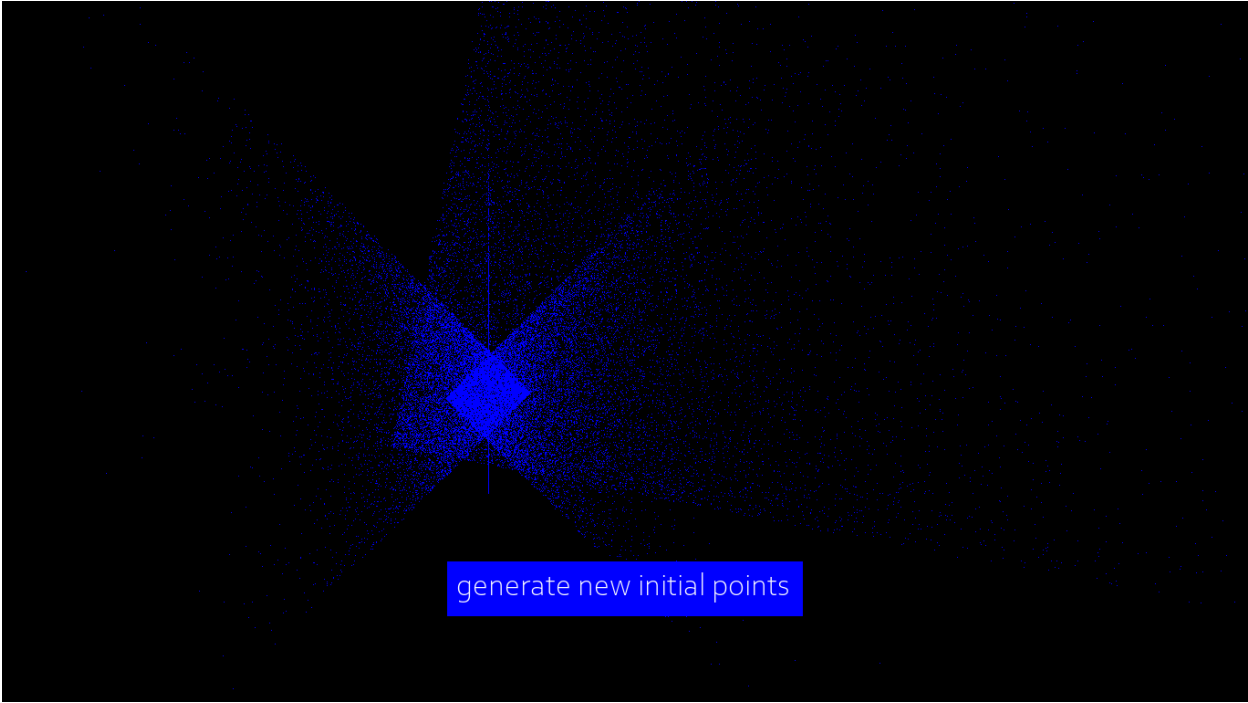
## تمرین 2.6

برای این تمرین به چهار تابع نیاز داریم:

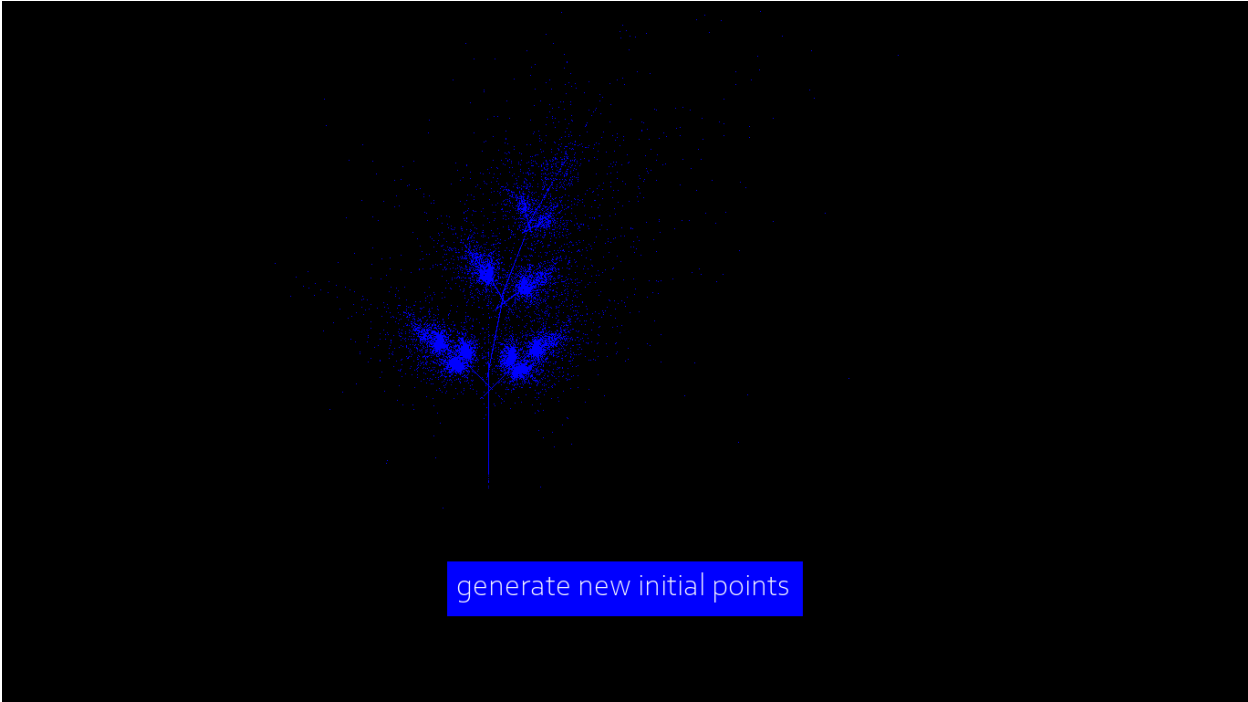
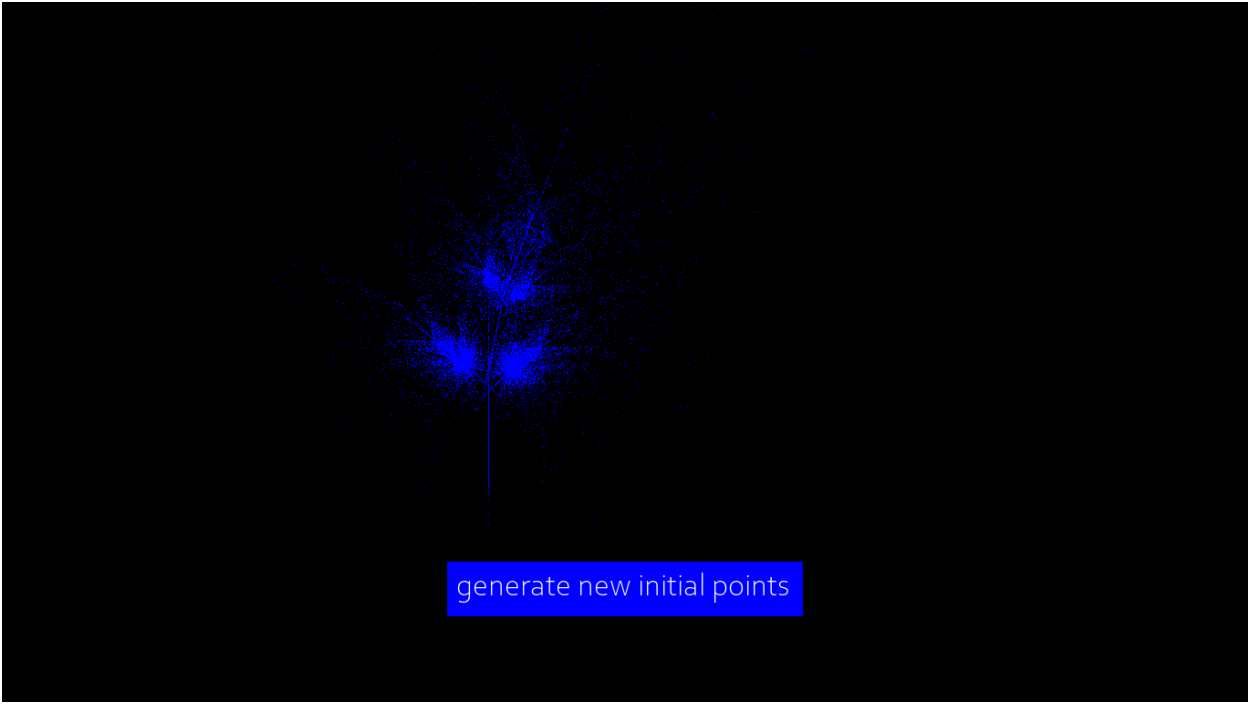
- 1- همه ی نقاط را در راستای محور ایکس کاملاً منقبض می کند و روی محور ایگرگ می آورد و در راستای ایگرگ هم به مقدار زیادی نقاط را منقبض می کند تا شاخه ی پایینی را تشکیل دهد.
- 2- همه ی نقاط را هم در راستای ایکس و هم در راستای ایگرگ (در راستای ایکس کمی بیشتر) منقبض می کند و سپس به مقدار کمی در جهت چرخش عقربه های ساعت می چرخاند و کمی به بالا انتقال می دهد تا شکل اسکیل شده را کمی به بالا و راست ببرد.
- 3- همه ی نقاط را نسبت به محور ایگرگ بازتاب می دهد و به مقدار خیلی زیادی ( در راستای ایکس کمی بیشتر) منقبض می کند و به مقدار زیادی (45 درجه) در جهت خلاف حرکت عقربه های ساعت می چرخاند و کمی به بالا انتقال می دهد تا برگ سمت چپ را تشکیل دهد.
- 4- همه ی نقاط را به مقدار خیلی زیادی ( در راستای ایکس کمی بیشتر) منقبض می کند و به مقدار زیادی (45 درجه) در جهت حرکت عقربه های ساعت می چرخاند و کمی به بالا انتقال می دهد تا برگ سمت راست را تشکیل دهد.

این چهار تابع را به تابع مین لوپ از ماژولی که ساختیم پاس می دهیم و بقیه ی کارها (که در مقدمه توضیح داده شده) انجام می شود.

برای این سوال سه تابع کمکی هم ساختیم که به ما ماتریس های چرخش و انقباض و بازتاب را می دهد.





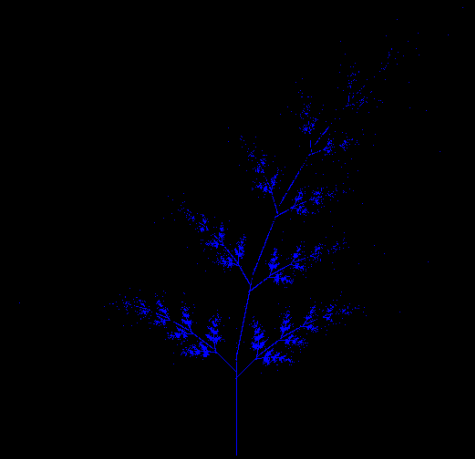




generate new initial points



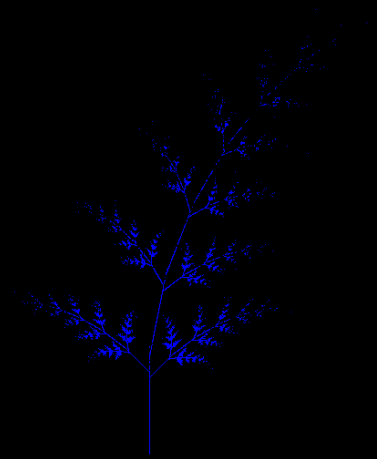
generate new initial points



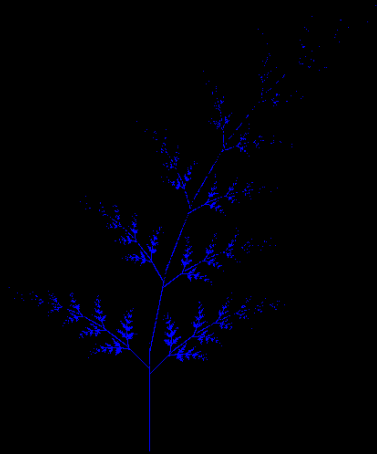
generate new initial points



generate new initial points



generate new initial points



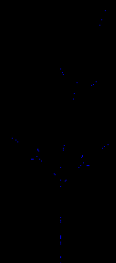
generate new initial points

1.  $\mathbf{p}_1$   
2.  $\mathbf{p}_2$   
3.  $\mathbf{p}_3$   
4.  $\mathbf{p}_4$   
5.  $\mathbf{p}_5$   
6.  $\mathbf{p}_6$   
7.  $\mathbf{p}_7$   
8.  $\mathbf{p}_8$   
9.  $\mathbf{p}_9$   
10.  $\mathbf{p}_{10}$

generate new initial points

1.  $\mathbf{p}_1$   
2.  $\mathbf{p}_2$   
3.  $\mathbf{p}_3$   
4.  $\mathbf{p}_4$   
5.  $\mathbf{p}_5$   
6.  $\mathbf{p}_6$   
7.  $\mathbf{p}_7$   
8.  $\mathbf{p}_8$   
9.  $\mathbf{p}_9$   
10.  $\mathbf{p}_{10}$

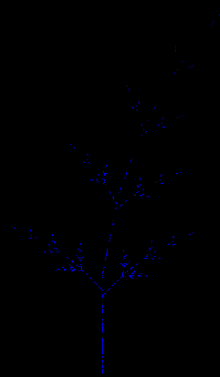
generate new initial points



generate new initial points



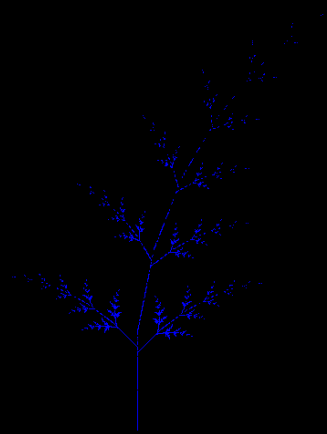
generate new initial points



generate new initial points



generate new initial points

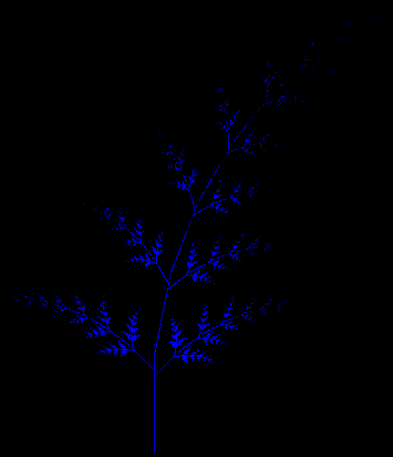


generate new initial points



generate new initial points

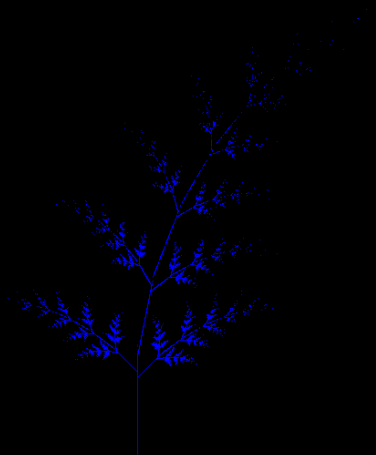




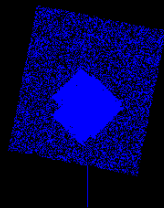
generate new initial points



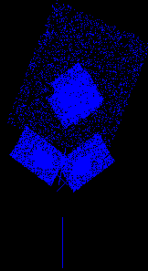
generate new initial points



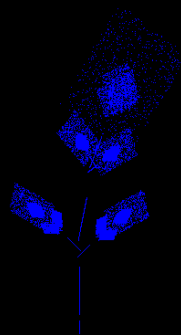
generate new initial points



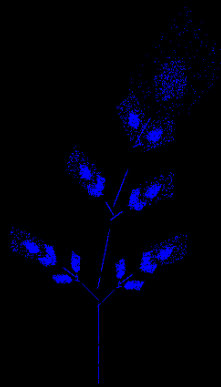
generate new initial points



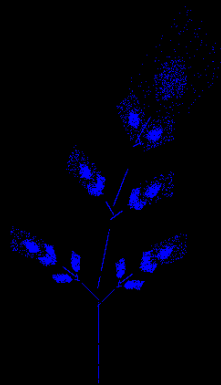
generate new initial points



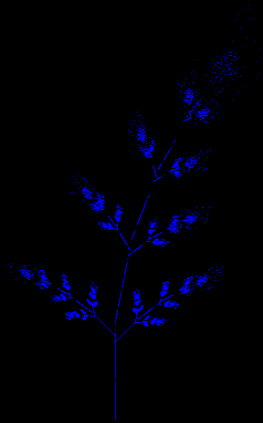
generate new initial points



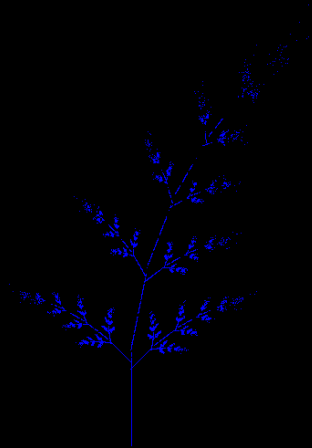
generate new initial points



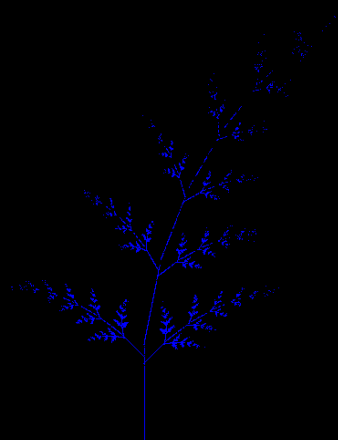
generate new initial points



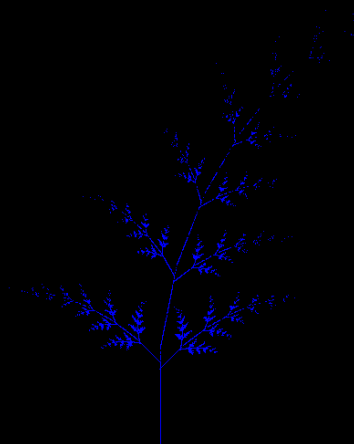
generate new initial points



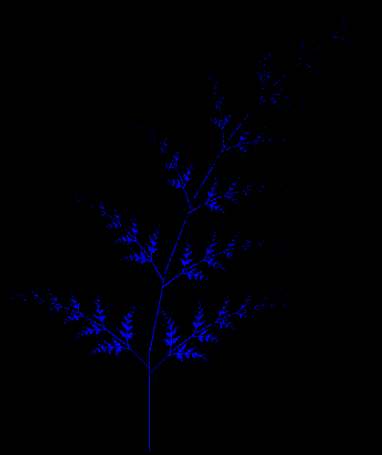
generate new initial points



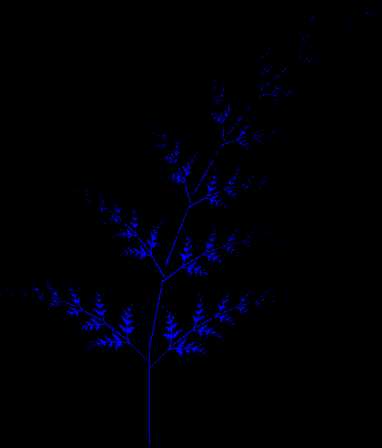
generate new initial points



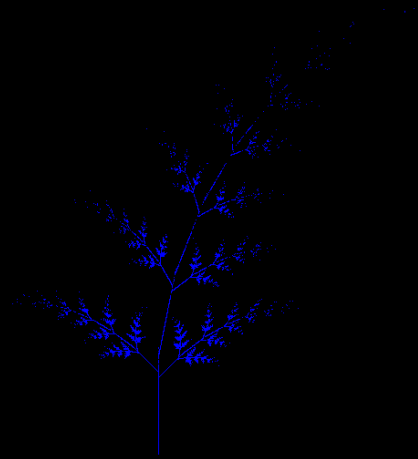
generate new initial points



generate new initial points



generate new initial points



generate new initial points



## تمرین 2.7

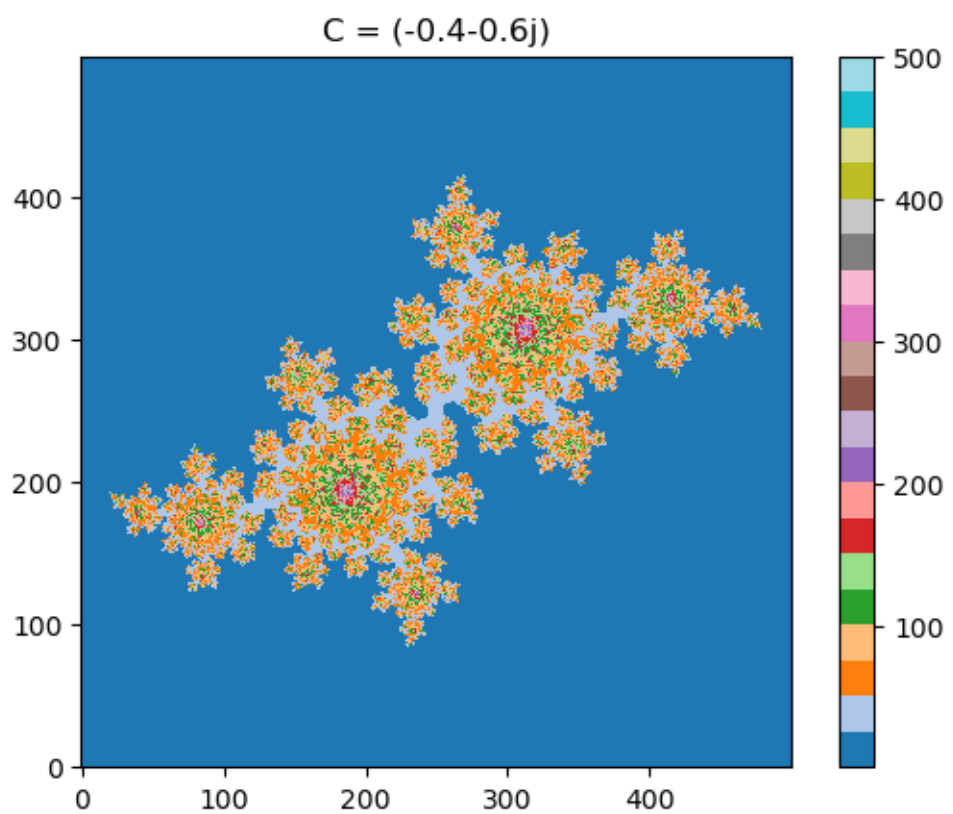
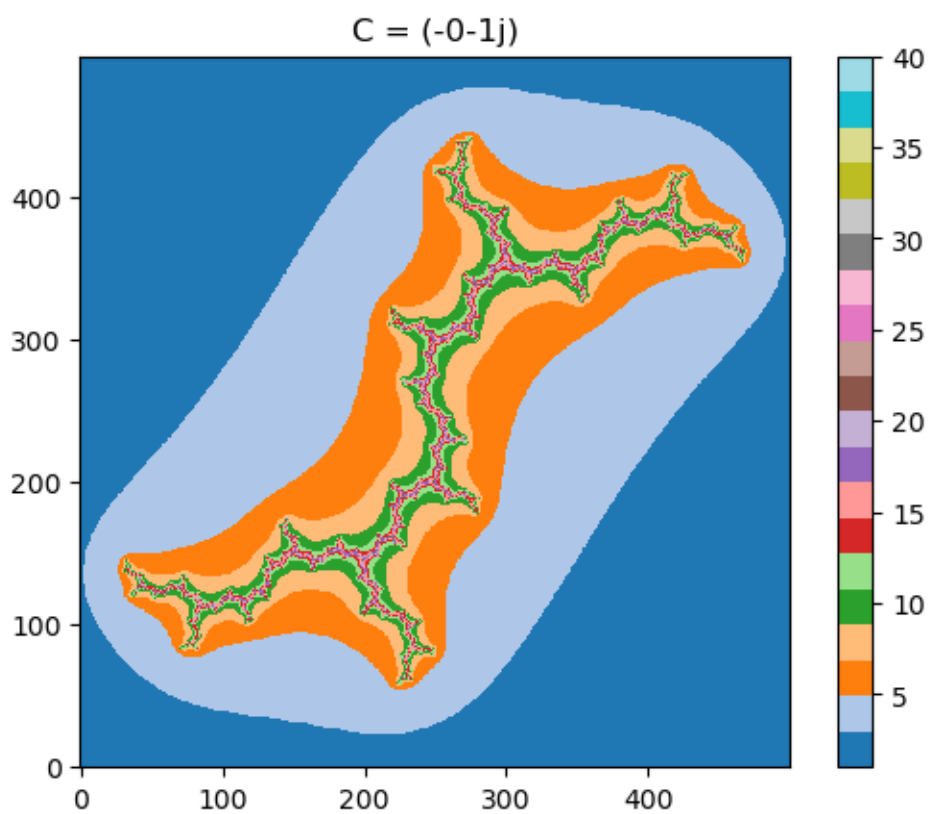
در این تمرین تابعی برای محاسبه ی قدمی که بعد از آن هر نقطه دیگر کاملاً از محدوده خارج می شود ساخته ایم و تابعی دیگر هم برای کشیدن نتایج که به توضیح خاصی احتیاج ندارد.

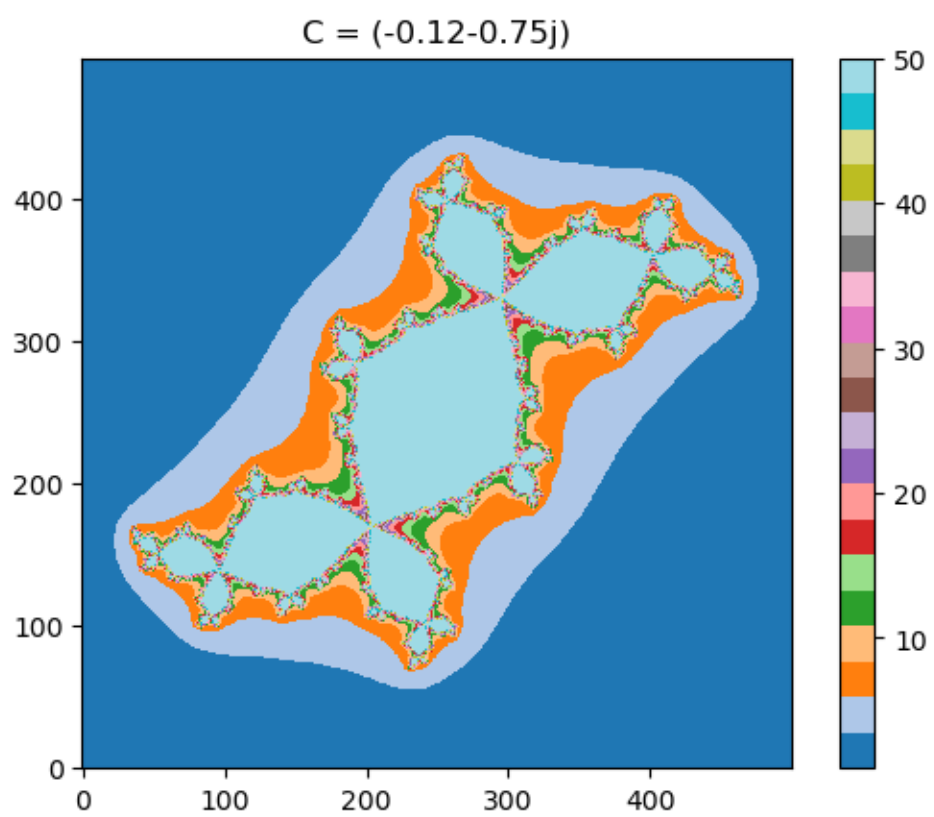
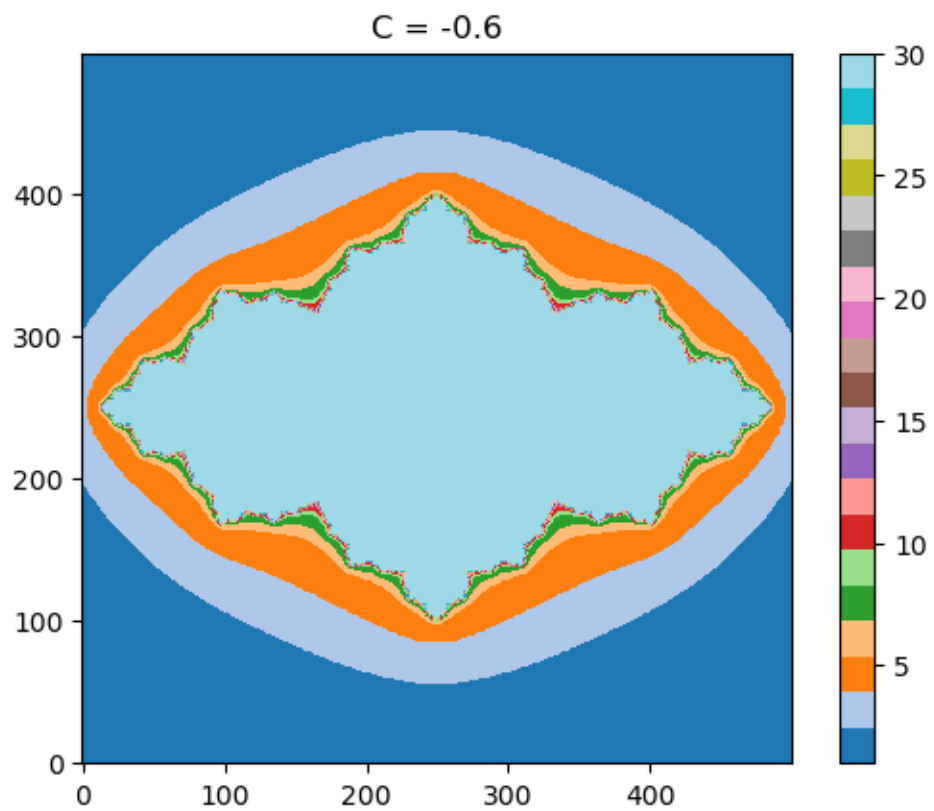
در تابع محاسبه گر محاسباتمان را بصورت ماتریسی با نامپای انجام داده ایم که سرعت انجام کار را بالا ببرد. ابتدا از نقطه ی قبلی نقطه ی جدید را محاسبه می کنیم و فاصله ی آن را از مبدا می سنجیم و اگر بیشتر از مقداری شد که مشخص کرده ایم (مکس دیستنس) در ماتریس جامپ اوت به درایه ی متناظر با این نقطه یک واحد اضافه می کنیم و اگر از آن مقدار بیشتر نشد یک بار دیگر کل این مرحله را تکرار می کنیم و این کار را تا تمام شدن کل مراحل (استپ) هایی که در نظر داریم جلو برویم ادامه می دهیم و میبینیم هر نقطه از کدام مرحله دیگر به بیرون از فضا می رود.

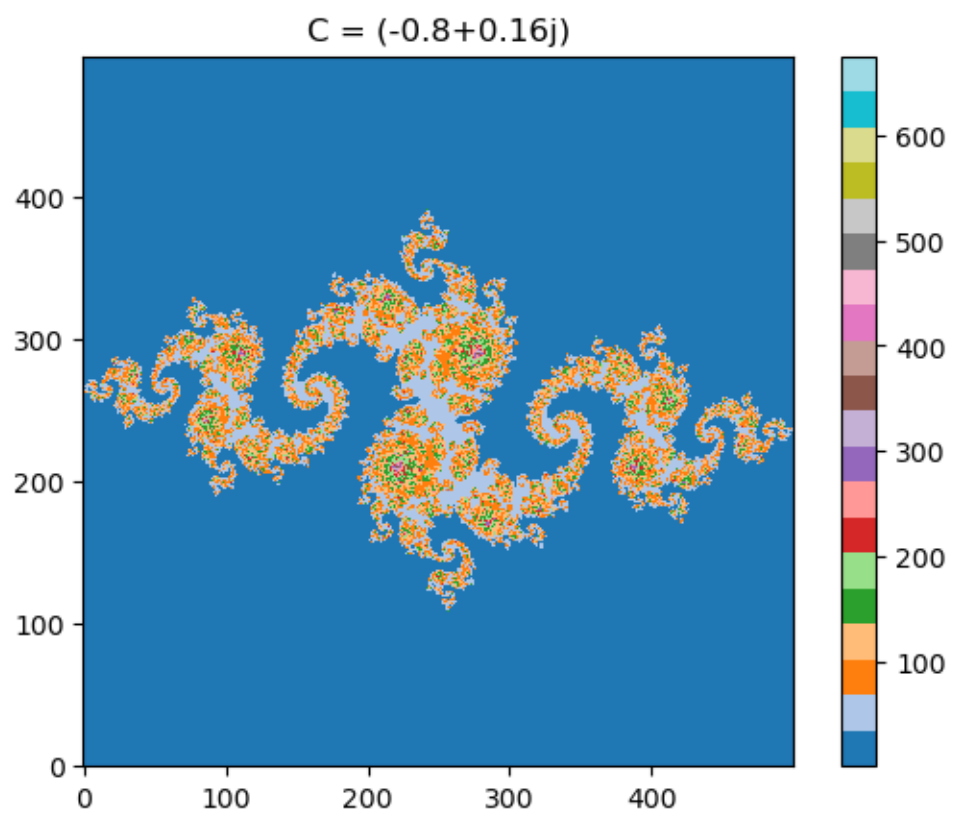
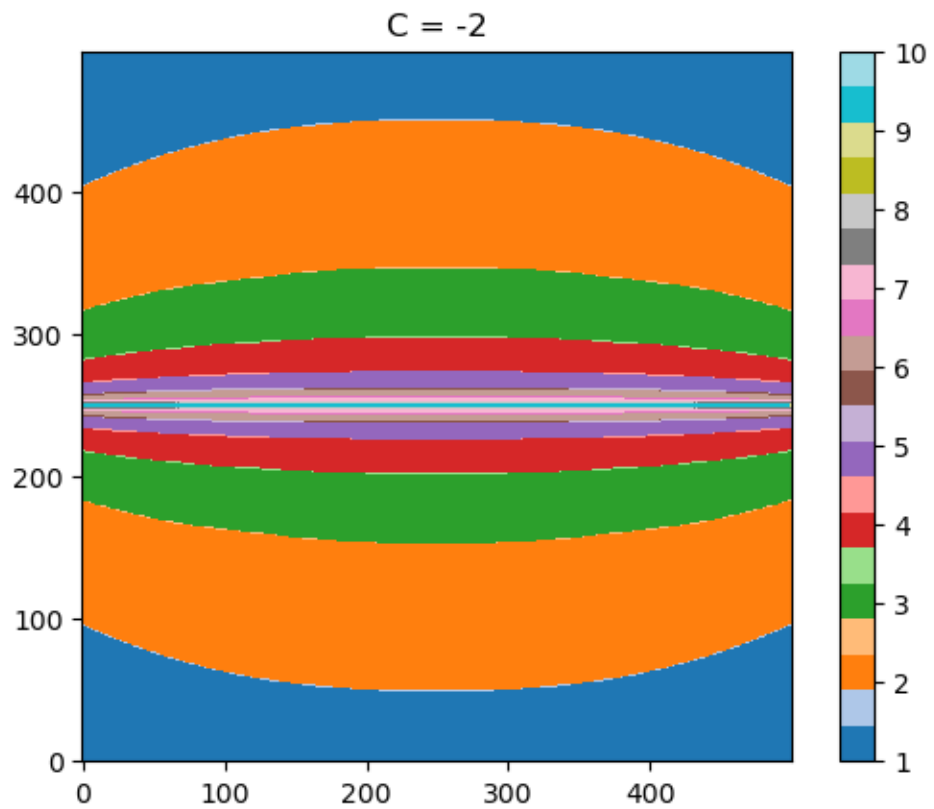
حال به کمک کالر مپ کتابخانه ی مت پلات لیب به هر نقطه از فضا متناسب با مرحله ای که از صفحه بیرون رفت (و یا کلاً نرفت) رنگی نسبت می دهیم و رسمش می کنیم.

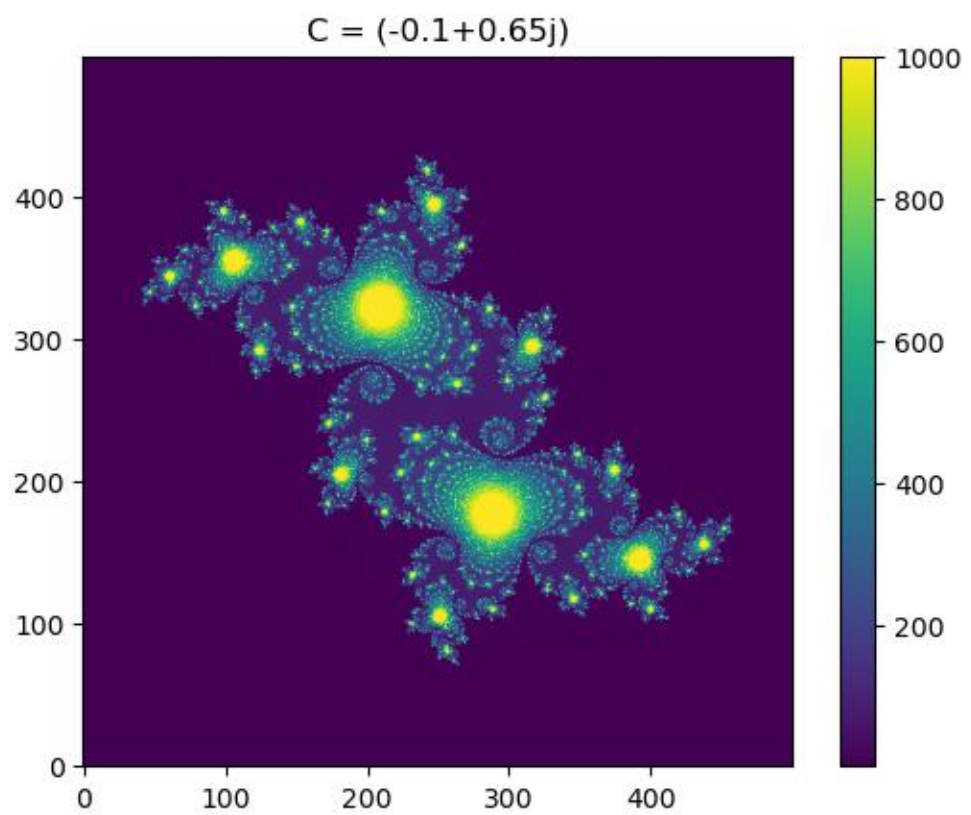
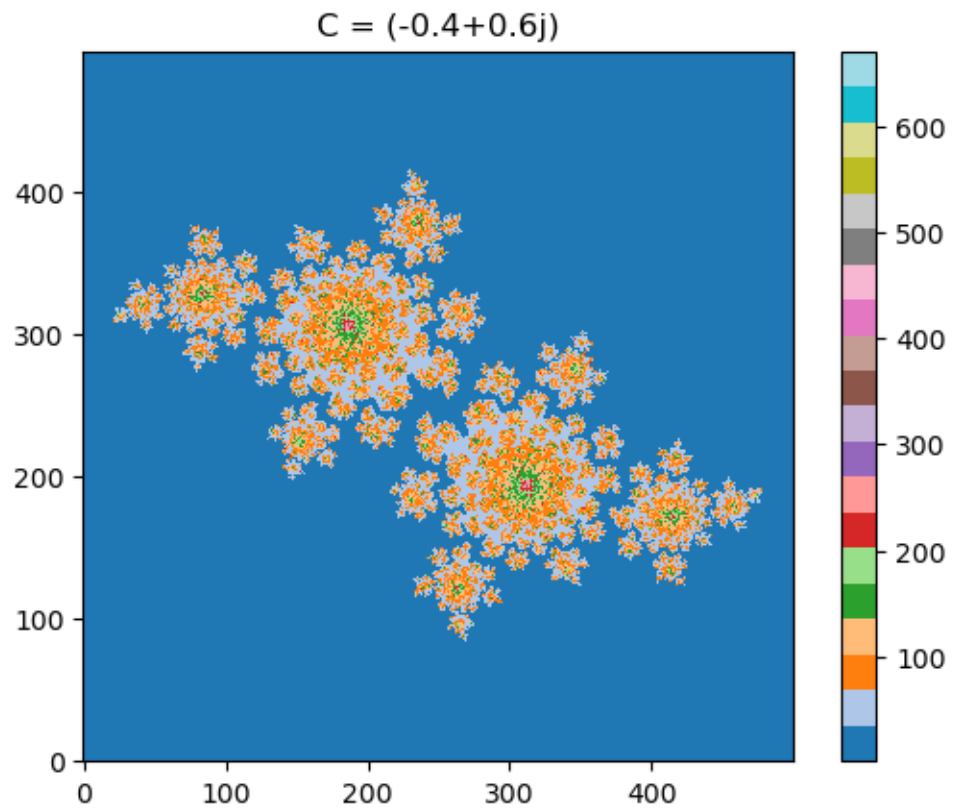
این کار را به ازای مقادیر ثابت مختلف انجام می دهیم و اشکال را رسم می کنیم.

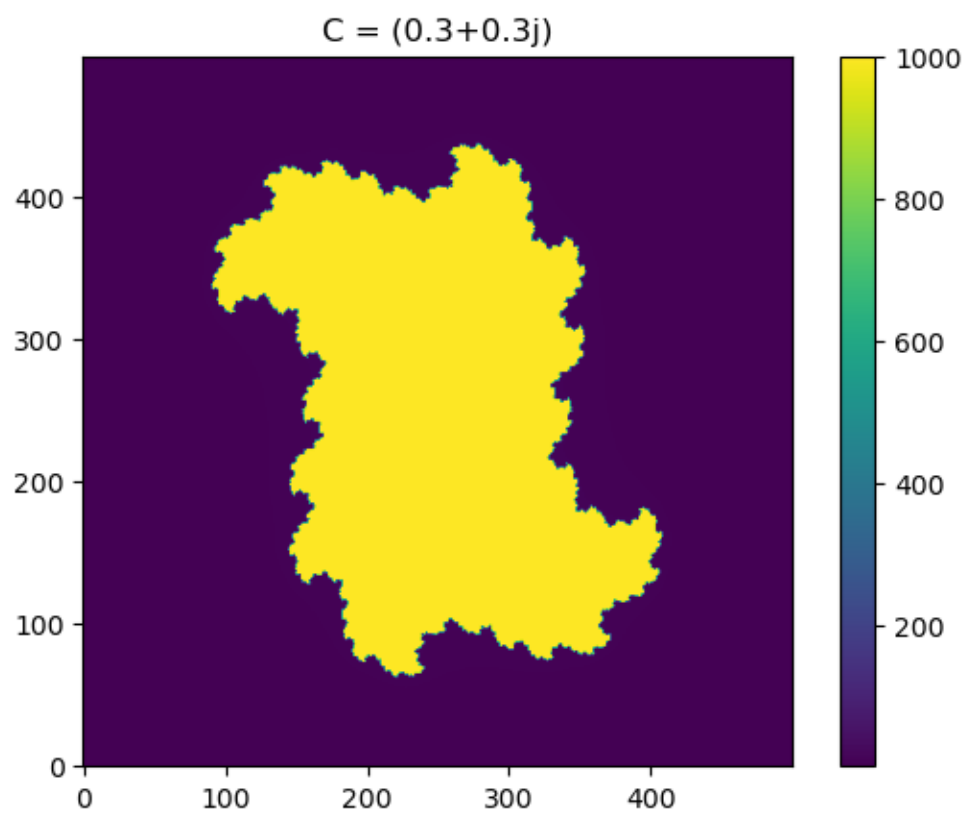
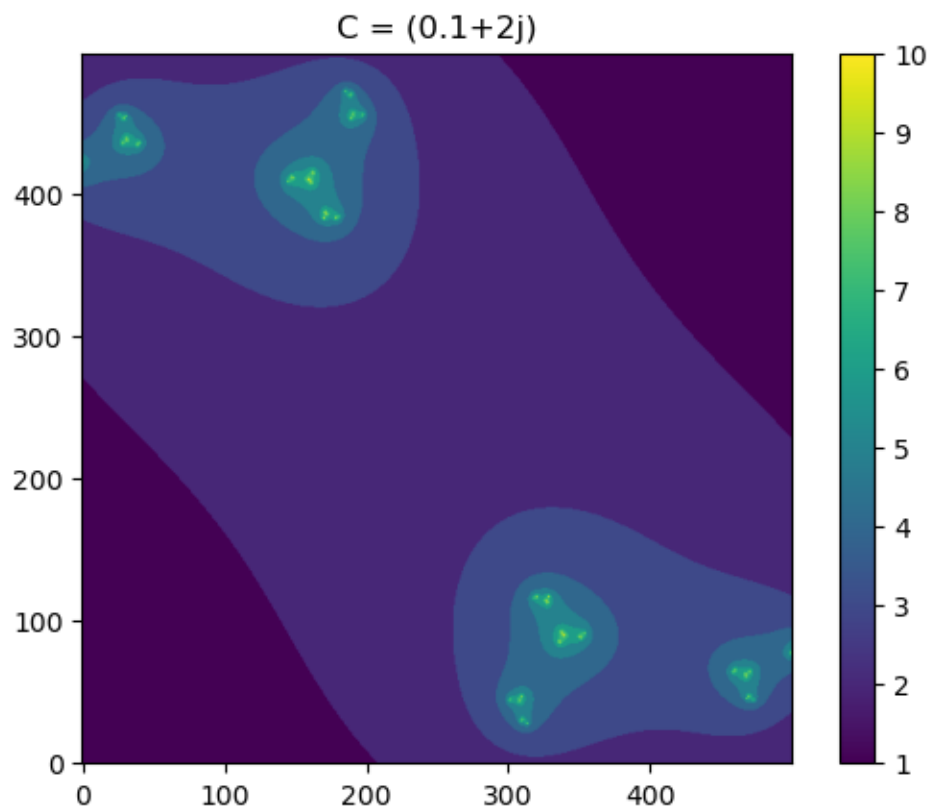
نتایج











### تمرین 3.1

در این تمرین فضا را به دویست قسمت تقسیم میکنیم و آرایه ای به همین طول می سازیم.

در هر مرحله به طور رندوم یکی از این خانه ها را انتخاب کرده و به عددش یکی اضافه می کنیم به این معنی که لایه ای روی آن نشسته.

یک شبکه (گرید) متناظر با این نقاط و ارتفاع لایه ای که در داخلشان است می سازیم و ستون ها را همان نقاط فضا و سطر ها را ارتفاع هر نقطه می گیریم.

خانه های این شبکه سه مقدار 0 یا 1 یا 2 را می توانند اختیار کنند.

عدد صفر که عدد پیشفرضشان است به معنی رنگ سفید و عدم وجود لایه در آن ارتفاع است.

عدد یک که به معنی رنگ آبی است در یک بازه ی زمانی به لایه هایی که در آن بازه می نشینند اختصاص می یابد.

عدد 2 هم که به معنی رنگ زرد است وضعیتی مشابه عدد یک و رنگ آبی دارد.

کلاس فیگ این کار ها را برای ما انجام می دهد و شکل نهایی را (در صورت خواست ما) رسم می کند.

پس از انجام شدن این مراحل کار به انجام محاسبات می کشد که با تابع دو کلکیولیشن انجامشان می دهیم.

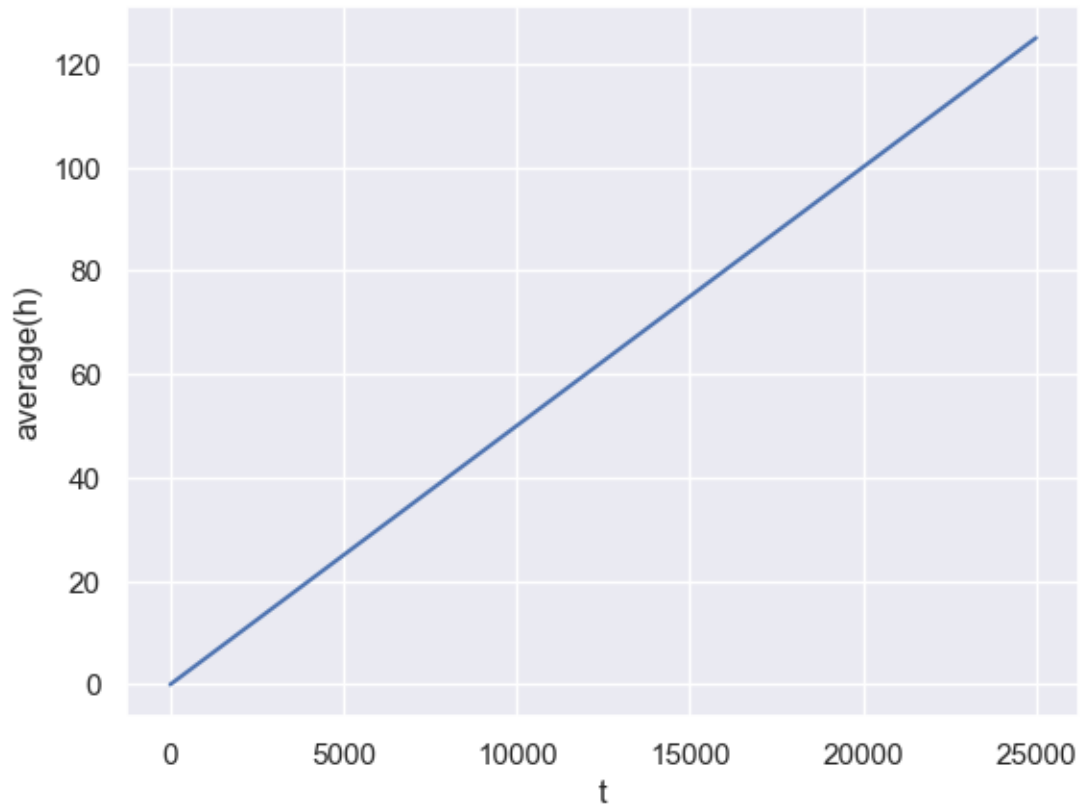
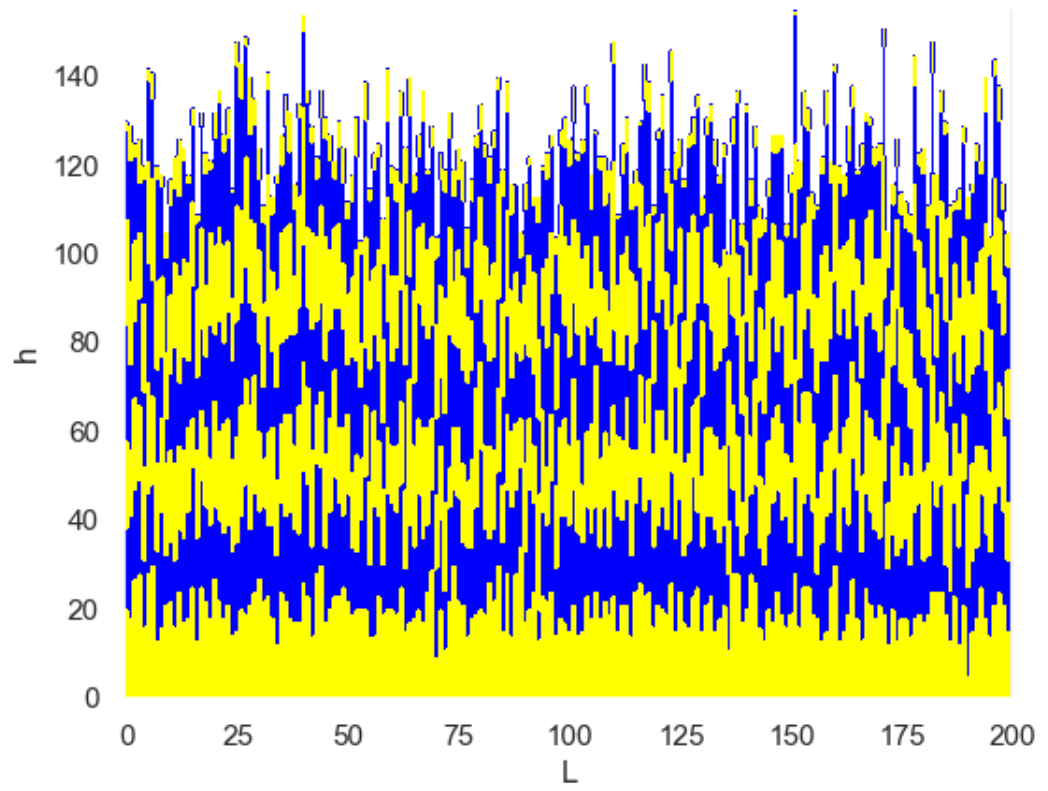
در این تابع نمودار متوسط ارتفاع لایه ها نسبت به زمان (که در همان کلاس فیگ در یک متغیری ذخیره شده اند) را رسم می کنیم.

سپس تابع واریانس را نسبت به زمان (که در همان کلاس فیگ در یک متغیری ذخیره شده اند) رسم می کنیم.

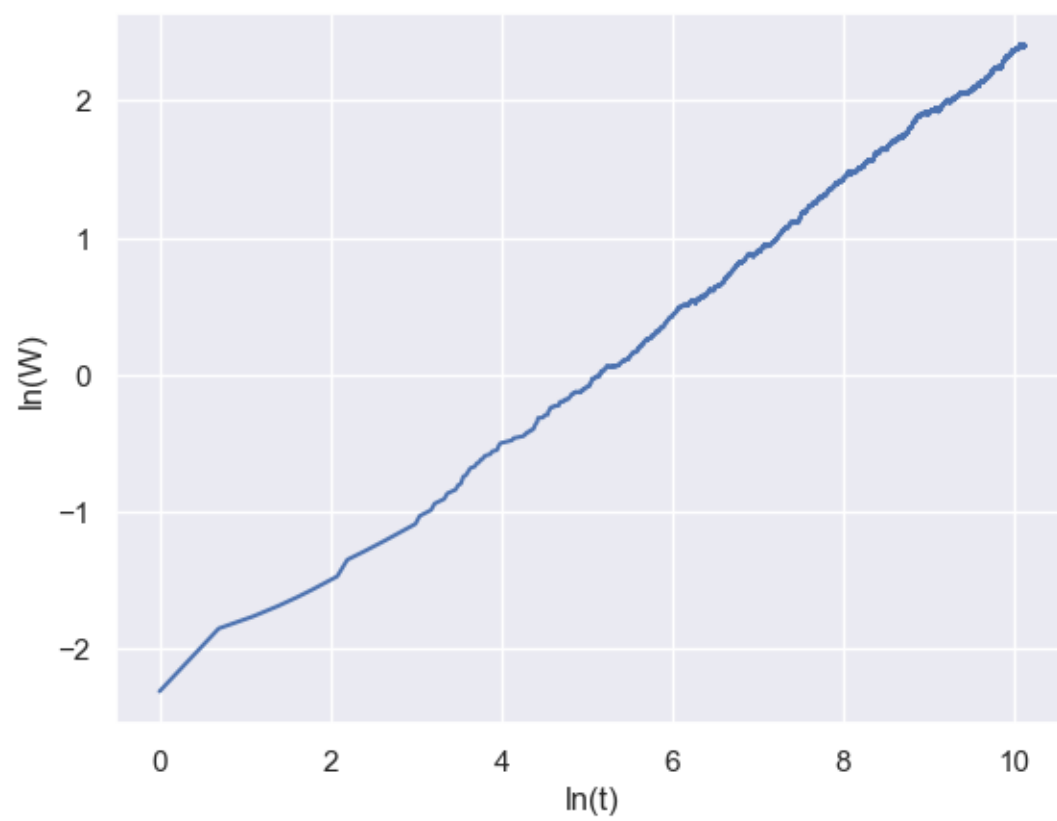
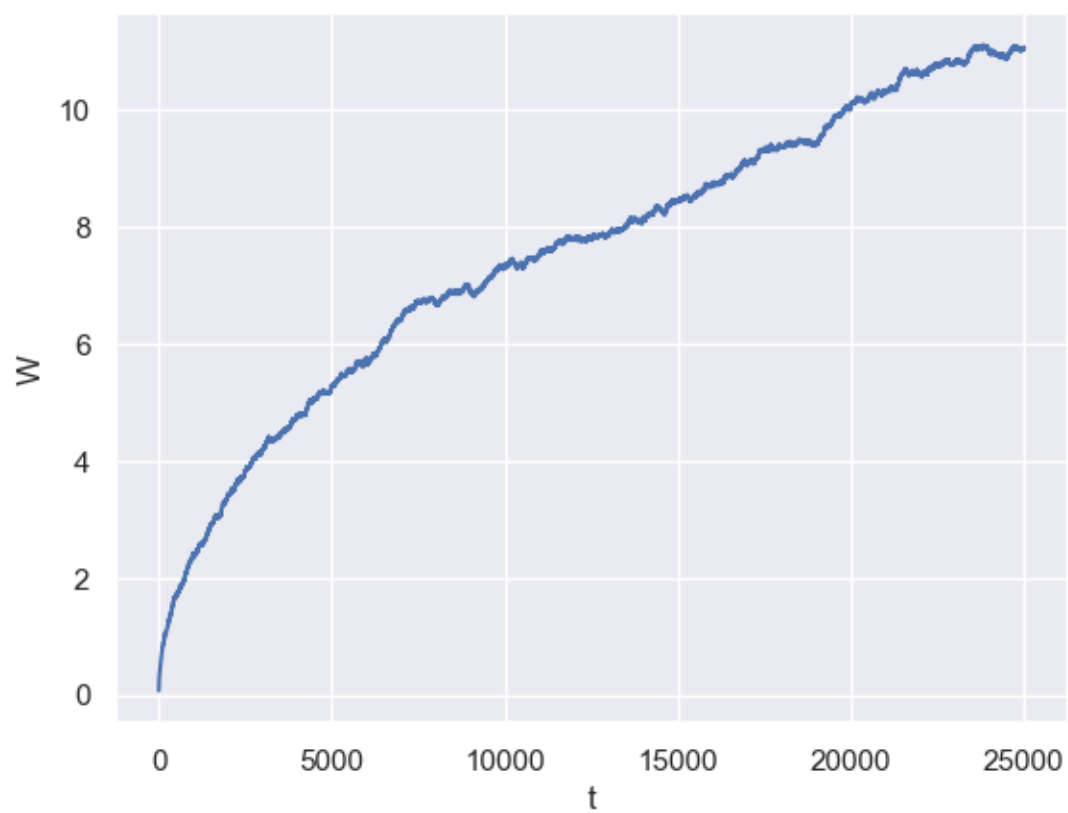
سپس لگاریتم واریانس را نسبت به زمان رسم میکنیم که تقریبا شکلی خطی دارد پس با کمک کتابخانه ی ساکیت لرن خطی بر آن فیت کرده و شیب و عرض از مبدا آن را بدست می آوریم که شیب همان پارامتر بتا پی است که سواز از ما خواسته.

این تابع را ده بار به ازای ده آبجکت فیگ مختلف (این آبجکت را ده بار صدا میزنیم و چون متغیر هایی تصادفی دارد هر بار یک نتیجه ی متفاوت به ما می دهد) اعمال می کنیم و نتایج خروجی که همان بتا ها هستند را ذخیره میکنیم و در نهایت از آنها میانگین گرفته که بتای میانگین را به ما می دهد و واریانس آنها هم معیاری از خطا به ما می دهد.

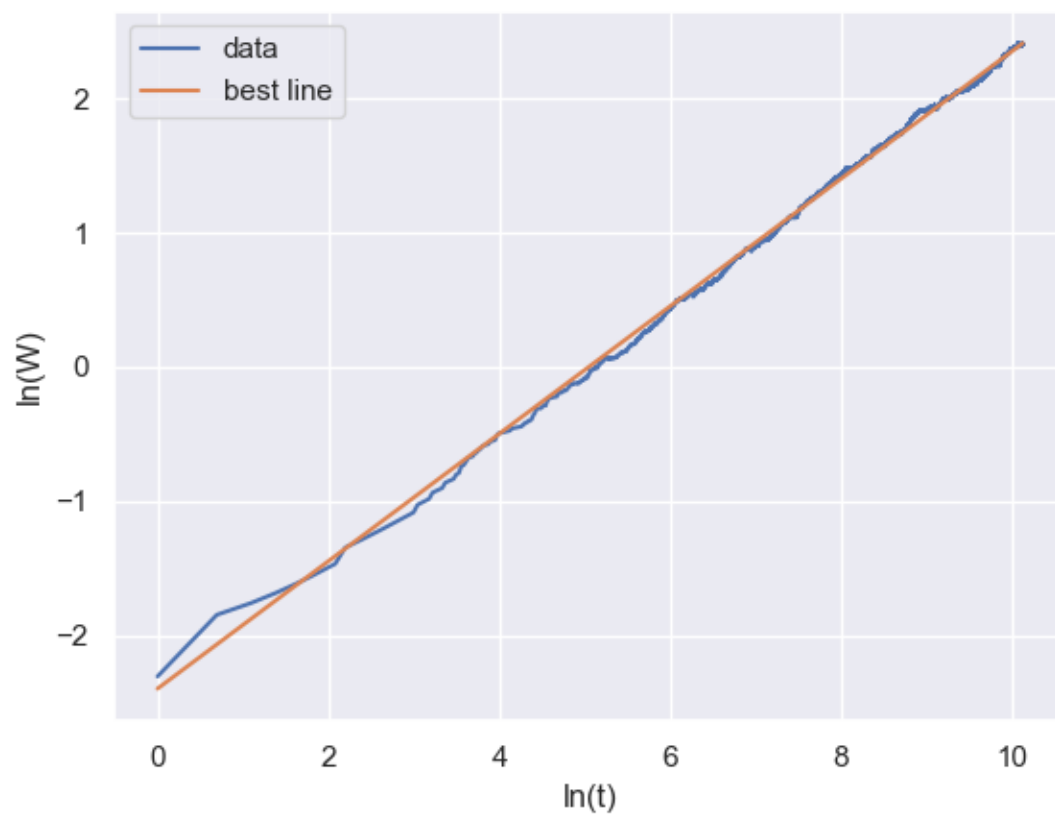
نتایج

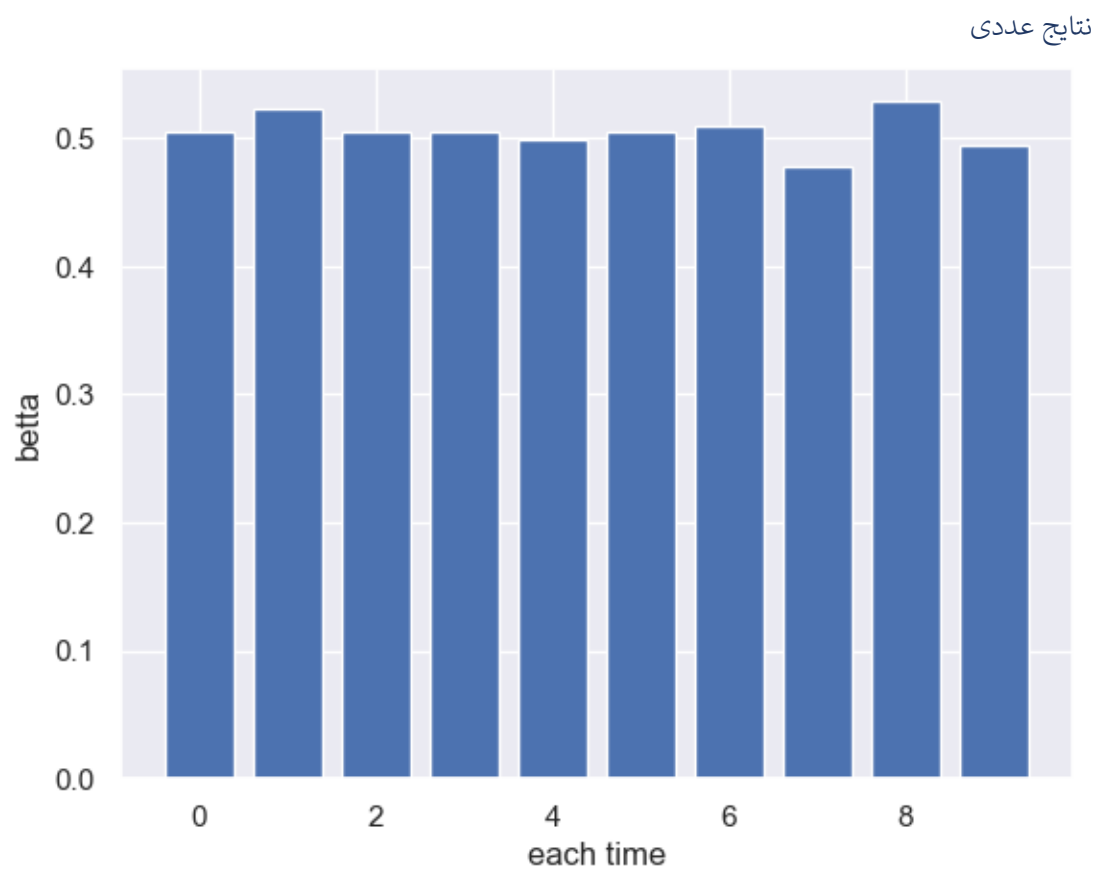






$$\ln(W) = 0.4748\ln(t) + (-2.3987)$$
$$r^2 = 0.9969$$





میانگین بتا ها = 0.5049

واریانس بتا ها = 0.0002