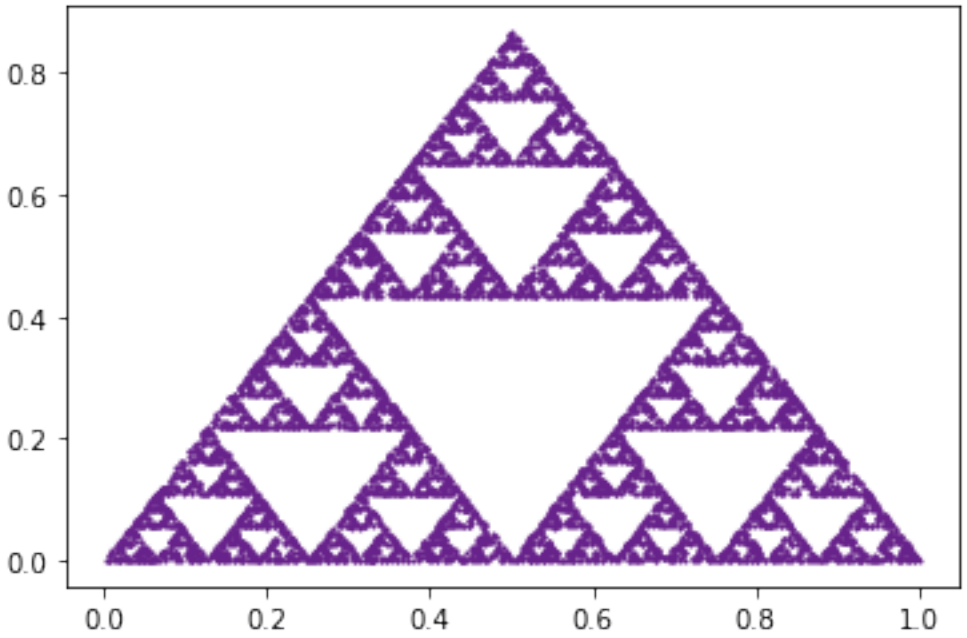
گزارش کار تمرین سری دوم شبیه سازی های فیزیکی

زهرا اکبری – 98100612

تمرین اول: مثلث سرپینسکی به روش تصادفی

تعاریف وابع انتقال،دوران و انقباض و همینطور چهار سه تابع اصلی رسم کننده مثلث سرپینسکی از حل روش تعینی در سری قبلی آورده شده‌اند. در اینجا فقط با تولید یک عدد تصادفی بین 1 تا 3، تعداد زیادی نقطه وارد لوپی از اعمال مکرر توابع با ترتیبی تصادفی میشوند و سپس در صفحه رسم میشوند. با افزایش تعداد نقاط به مثلث سرپینسکی نزدیک میشویم. در ادامه نتیجه با 10000 نقطه آورده شده است.محور های فاصله عمودی و افقی را نشان میدهند.



شکل 1 سرپینسکی با 10000 نقطه

تمرین دوم: فرکتال سرخس به روش تصادفی

روش رسم سرخس مشابه تمرین قبلی ست، تنها باید توابع را به شکل درست مشخص کنیم. این بار نقاط تصادفی را در مستطیلی به طول و عرض 4 و 1 انتخاب میکنیم مبدا مختصات وسط ضلع پایینی این مستطیل اولیه است که ابتدای شاخه سرخش روی آن قرار میگیرد.

در اینجا به جز توابع انتقال، دوران و انقباض که از قبل تعریف کرده بودیم به تابع جدیدی نیاز داریم که در دو راستای x و y به اندازه ای متفاوتی شکل را منقبض کند. این تابع با نام retractxy مشخص شده که دو ضریب انقباض میگیرد.

```
def contractxy(coordinates,r,k):
    newcoordinates = []
    cntr_xy = np.array([[r,0],[0,k]])
    for i in range(len(coordinates)):
        p = np.matmul(cntr_xy, coordinates[i])
        newcoordinates.append(p)
    return(newcoordinates)
```

در ادامه توابع سازنده سرخس را توضیح میدهیم

اولین تابع، سازنده بخش اصلی فرکتال یا برگ بزرگ اصلی در میانه است. که از یک انتقال به بالا، دوران به اندازه 4 درجه و انقباض به اندازه‌ی 0.3 و 0.4 در راستای افقی و عمودی تشکیل شده است.

```
def function1(coordinates):
    temp = rotation(coordinates,theta1)
    temp = contractxy(temp,(0.9),0.8)
    temp = translation(temp, 0,1)
    return(temp)
```

دومین تابع توصیف کننده برگ های سمت راست مشابع تابع قبلی ست صرفا اندازه دوران بزرگتر و انقباض بیشتری رخ داده تا برگ های کوچک تری ایجاد شوند.

```
def function2(coordinates):
    temp = rotation(coordinates,theta2)
    temp = contractxy(temp,(0.3),0.4)
    temp = translation(temp, 0,1)
    return(temp)
```

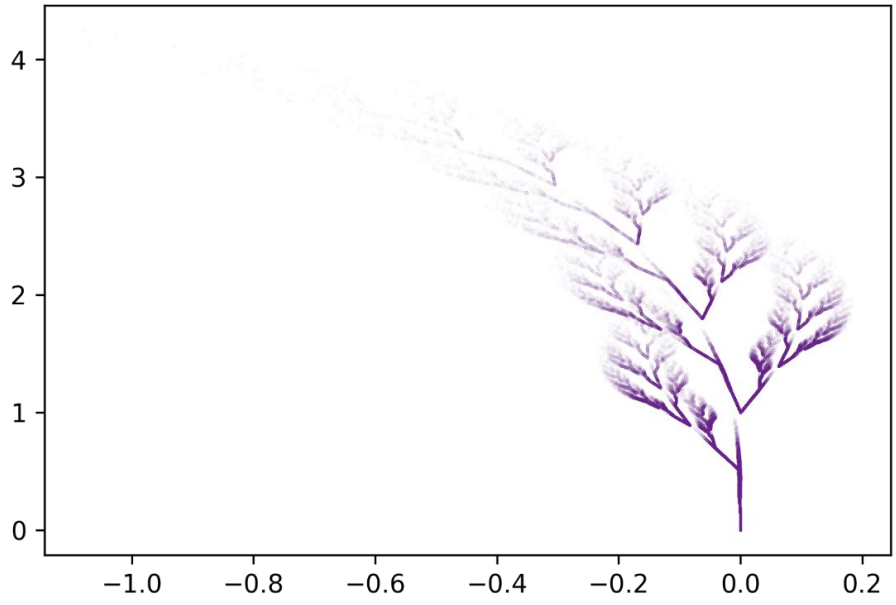
سومین تابع برای برگ های سمت چپ مشابه تابع بالاست اما باید دقت شود نیاز داریم تعقر برگ های شاخه های کوچک سمت چپ رو به بالا باشد. برای این ضریب انقباض در راستای افقی را منفی قرار میدهیم. و دقت میکنیم این قرینگی قبل از دوران رخ دهد تا هنوز ابتدای شاخه روی x=0 باشد.

```
def function3(coordinates):
    temp = contractxy(coordinates,-(0.3),0.4)
    temp = rotation(temp,theta3)
    temp = translation(temp, 0,0.5)
    return(temp)
```

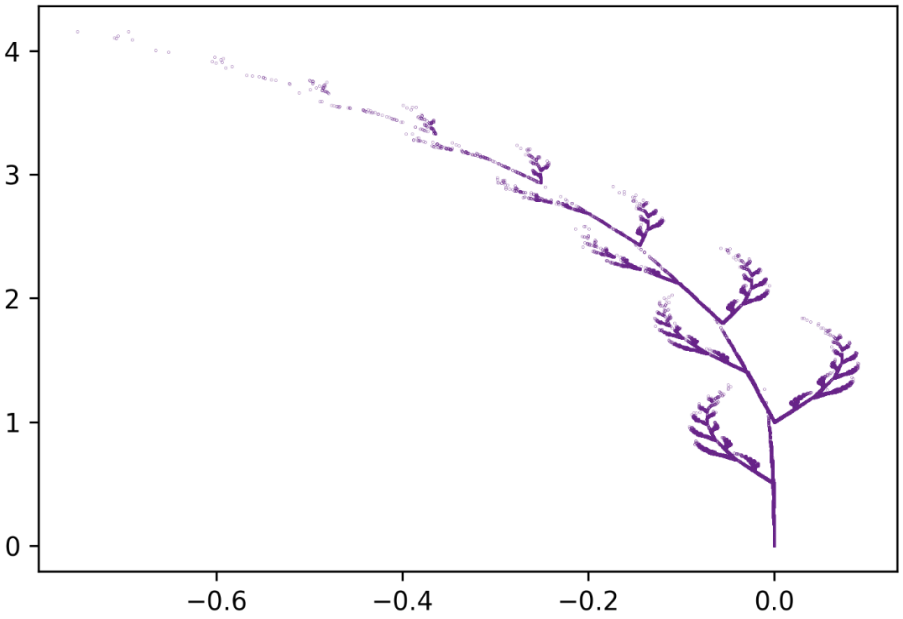
تابع آخر برای رسم شاخه‌ی سرخس است. در این تابع نقاط مستیطل داده شده به مقدار زیادی در راستای افقی کوچک میشوند و به زاویه مشابه تابع اول(بخش اصلی) یعنی 4 درجه دوران میابند. میزان انقباض در راستای عمودی بستگی به میزان انتقال در تابع اول دارد تا فضای خالی پایینی کامل پر شود.

```
def function4(coordinates):
    temp = rotation(coordinates,theta1)
    temp = contractxy(temp,(0.01),0.25)
    #temp = translation(temp, 0,0.5)
    return(temp)
```

در نتایج برای 12000 نقطه‌ی زیر محور ها فاصله در راستای افقی و عمودی اند.



شکل 2 فرکتال سرخس با 12000 نقطه اولیه شکل



فرکتال سرخس با 10000 نقطه اولیه 3 n

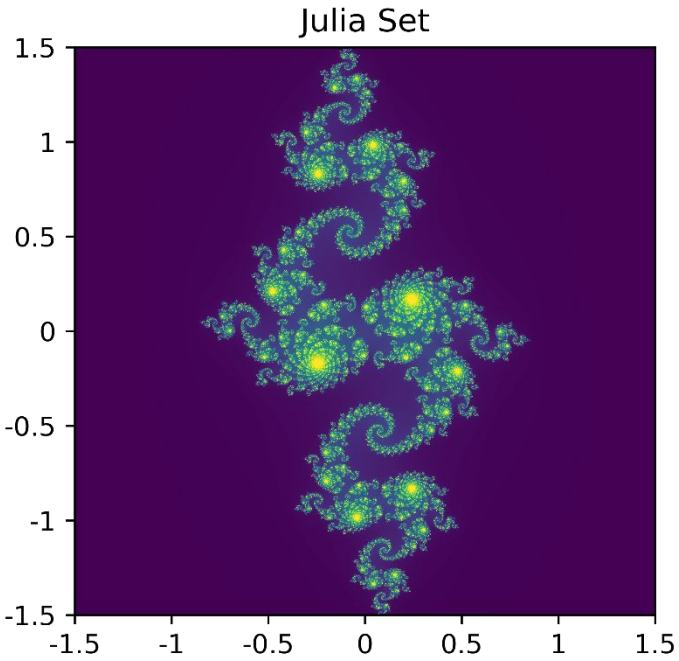
تمرین سوم: مجموعه جولیا

برای رسم مجموعه جولیا تابع آن را تعریف میکنیم.

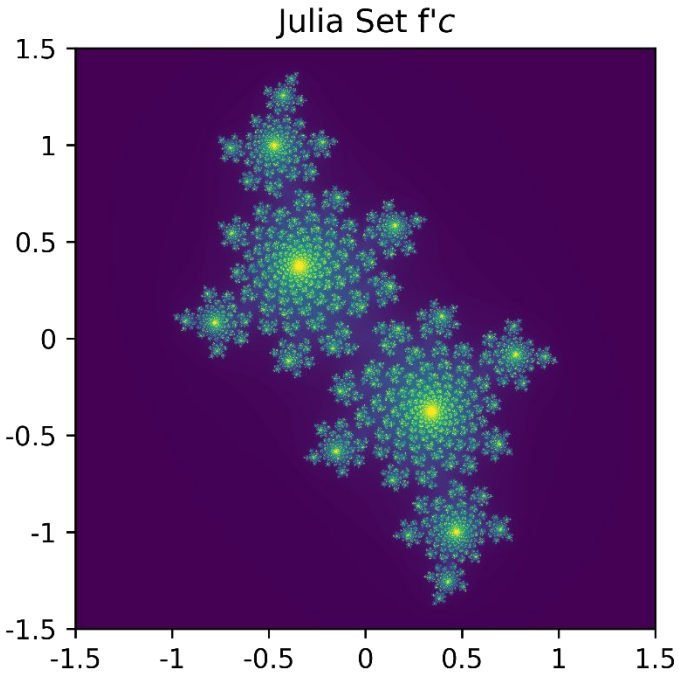
```
#defining function f= z^2+c
def function(coordinates):
    coordinates = coordinates*coordinates + c
    return(coordinates)
```

با استفاده از دو لوپ تو در تو صفحه را مش بندی میکنیم و برای نقاط صفحه مختلط تابع را تا زمانی که از کره واحد خارج نشده اند اجرا میکنیم(کمتر از 200 بار بررسی میشود.) تعداد اعمال تابع روی هر خانه از صفحه در متغیر counter ثبت میشود. با استفاده از نسبت این متغیر به 200 میتوانیم به هر خانه رنگی نسبت دهیم. با استفاده از imshow از کتابخانه matplotlip این نتایج را به تصویر میکشیم.

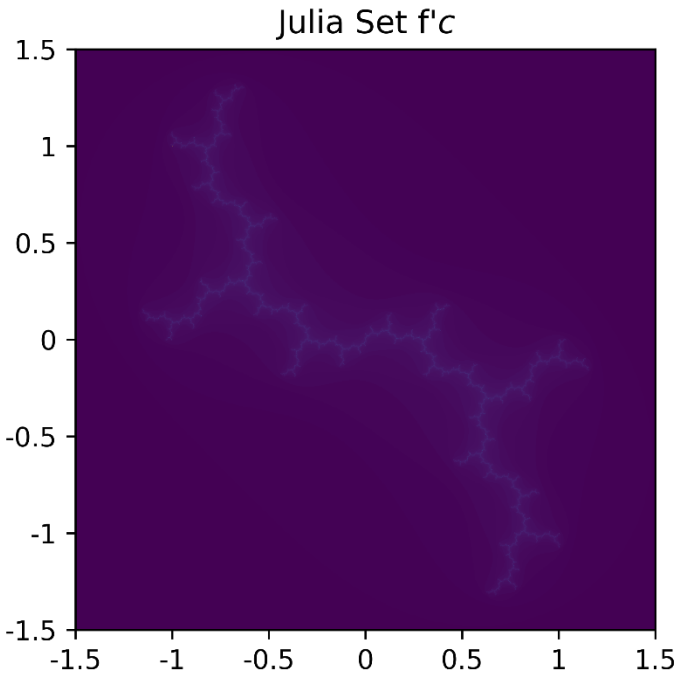
نتایج برای برخی ثوابت c در ادامه آورده شده‌اند.در برخی شکل ها، طیف رنگی فقط دو رنگ دارد به نظر میرسد علت آن بالا بردن تعداد ماکسیموم چک کردن بیرون رفتن از دایره واحد است. ولی برای اینکه تمام شکل ها ثابت مشخصی داشته باشند عدد 200 برایشان تغییر داده نشده است. محور های افقی و عمودی فاصله را نشان میدهند.



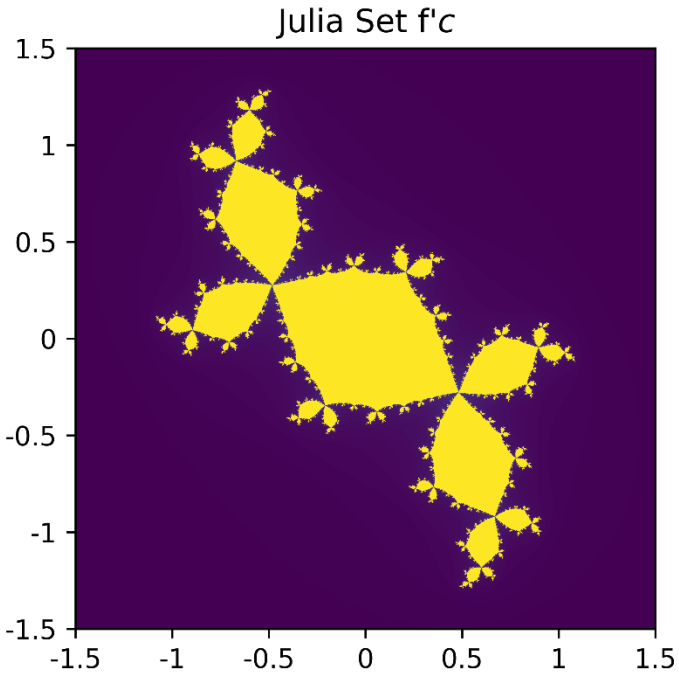
شکل 4 مجموعه جولیا برای c = -0.8 -0.16i



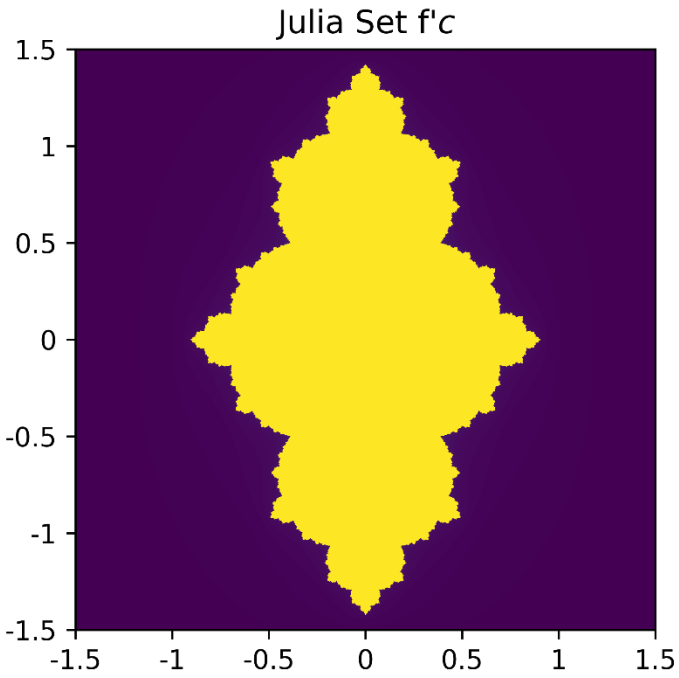
شکل 4 مجموعه جولیا برای c = -0.4 -0.6i



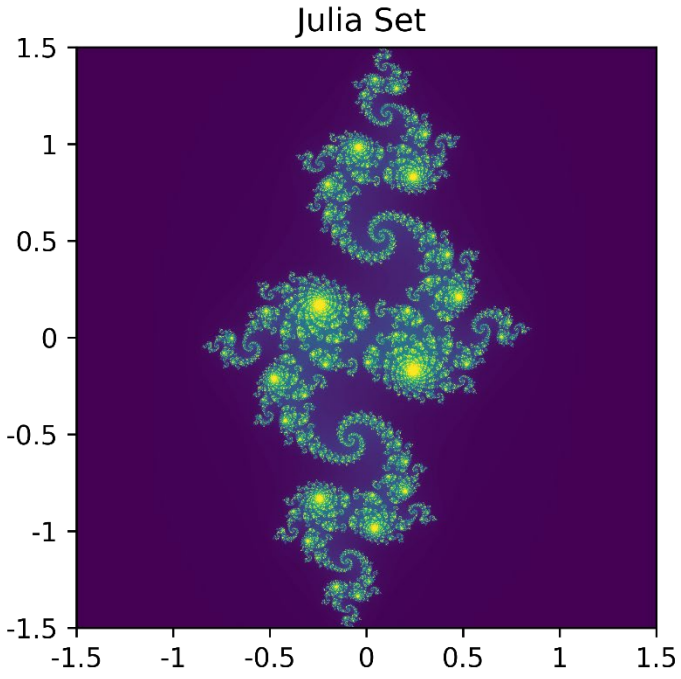
شکل 5 مجموعه جولیا برای c = -0 -i



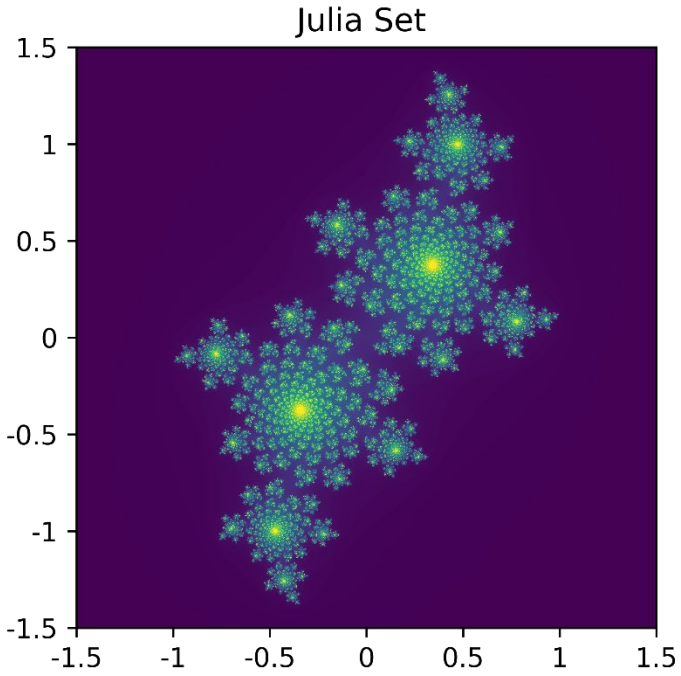
شکل 6 مجموعه جولیا با c= −0.12−0.75 i



شکل 7 مجموعه جولیا برای c =-0.6



شکل 8 مجموعه جولیا برای $c= -0.8+0.16i$



شکل 9 مجموعه جولیا برای $c = -0.4+0.6i$

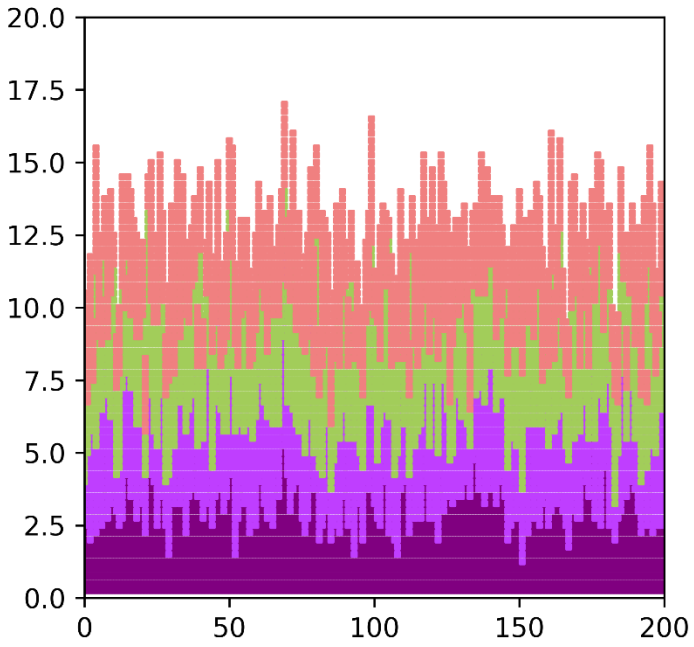
تمرین چهارم: ول‌نشست

برای شبیه سازی ول نشست روی محور افقی با 200 خانه ابتدا به تعداد t که نماینده زمان یا تعداد قدم هاست در لوپی عدد تصادفی انتخاب میکنیم و به اندازه خانه مربوط ار محور افقی با اندیس این عدد تصادفی یکی اضافه میکنیم. این توصیف کننده پدیده ول نشست است.

```
colorlist= ['purple','#BF3EFF','#A2CD5A','#F08080']
for i in range(t):
    color_i = (i//2500) %
4                                   #dedicating different colors fro,
colorlist for each 2500 squares
    x = rnd.randint(0,199)                                          #choosing
a random number betweeen 0 and 199 (h imdex goes from 0 to 199)
    h[x] += 1                                                       #adding 1
to a random index of h
    plt.scatter(x,h[x]/4, s=[2.9], marker='s', color =
colorlist[color_i])      #plotting by scattering small squares
```

برای رسم این پدیده میتوان از تابع histogram از کتابخانه matplotlib استفاده کرد ولی چون در تمرین های آینده مثل کنار نشست نمیخواهیم میله هایی بدون توجه به قدمی که در آن هستیم رنگ شوند از قرار دادن مربع های کوچک در مختصات مربوط استفاده میکنیم. اندازه این مربع ها با مقیاس محور های افقی و عمودی و همینطور ضریب ¼ در رسم ارتفاع طوری تنظیم شده که کمترین تداخل بین دو مربع همسایه کناری وجود داشته باشد و همزمان فاصله ای هم نداشته باشند. با این وجود به علت فضای کم تعداد کمی از ستون ها در هم فرورفته اند ولی این مقدار در به تصویر کشیدن ول‌نشست تداخلی ایجاد نمیکند. برای بهتر مشخص شدن ناهمواری ها از چهار رنگ در هر 2500 قدم استفاده

شده است. در شکل زیر محور عمودی تعداد ذره ها و محور افقی اندیس خانه است.

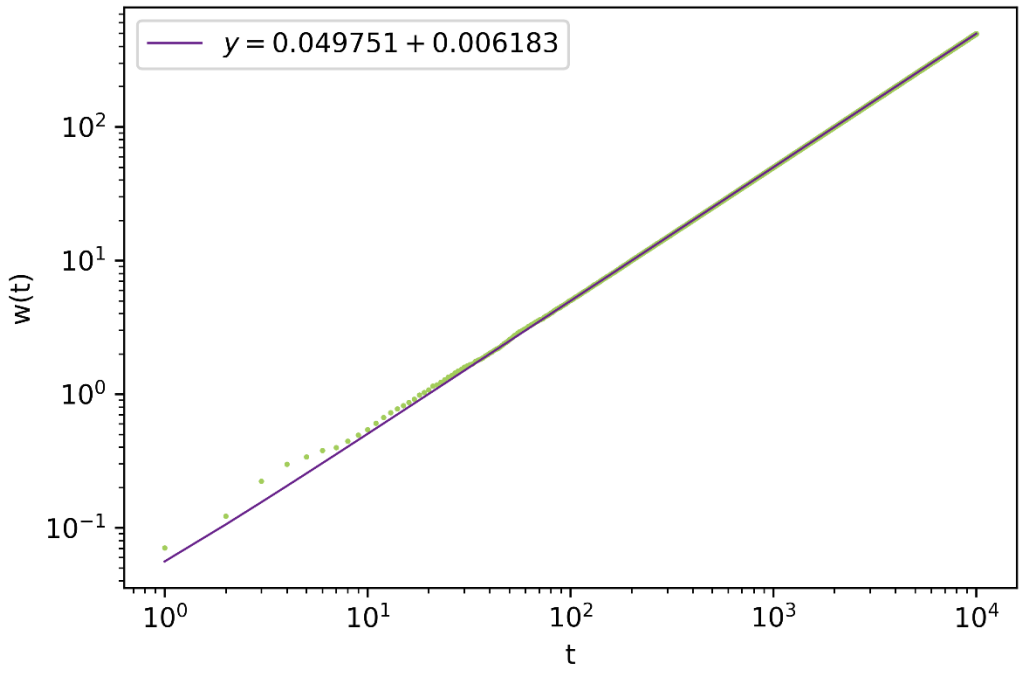


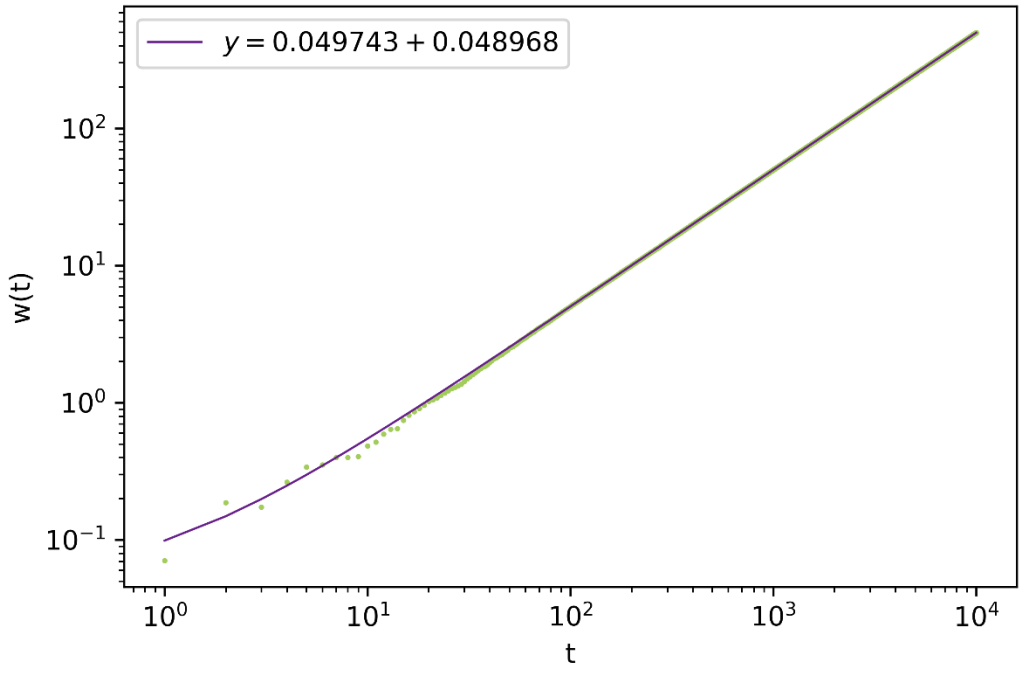
شکل 10 ولنشست پس از t = 10000

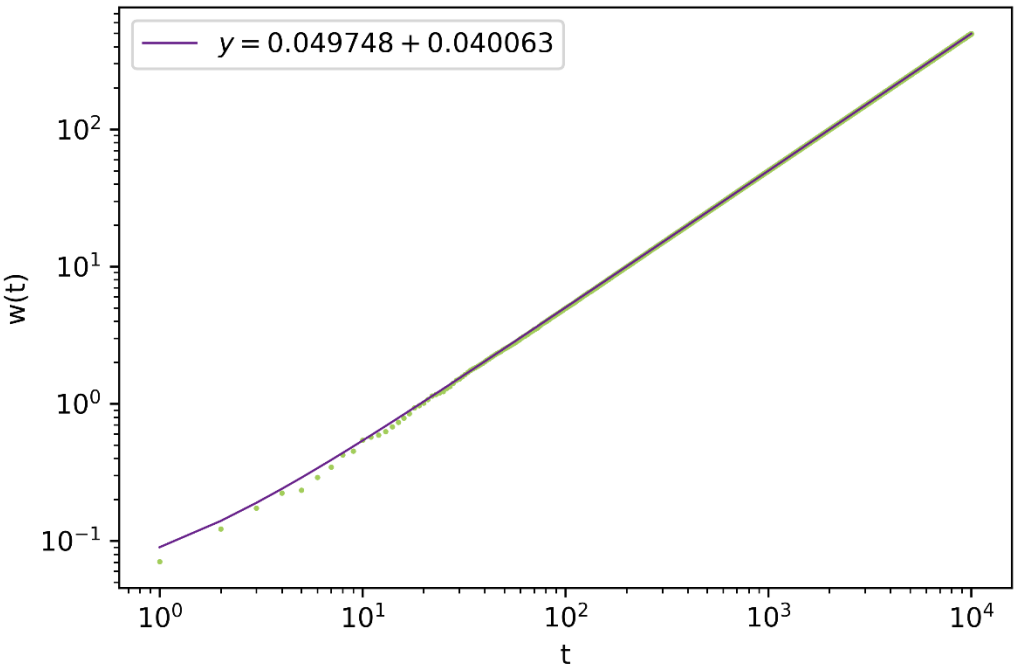
در ادامه متوسط ارتفاع و ناهمواری (انحراف از میانگین) را در همان لوپ قبلی محاسبه میکنیم.

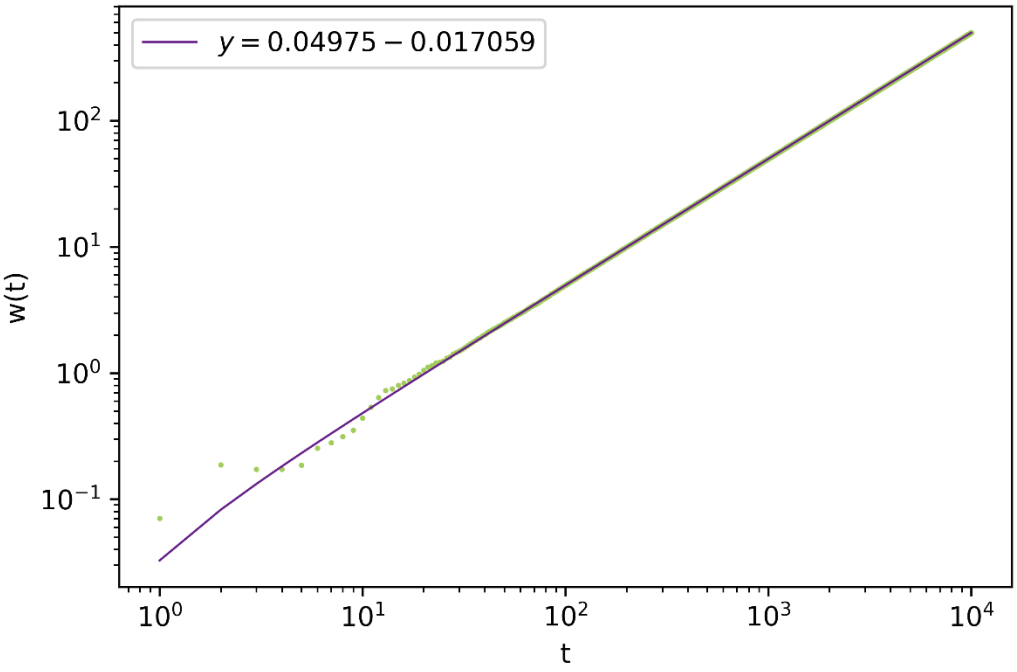
```
h_mean2 = (i/L)*(i/L)                                    #calculating
standard deviation
    for j in range(L):
        h2_sum = h2_sum+ (h[j])^2
    h2_mean = h2_sum/L
    w[i] = np.sqrt(h2_mean-h_mean2)
```

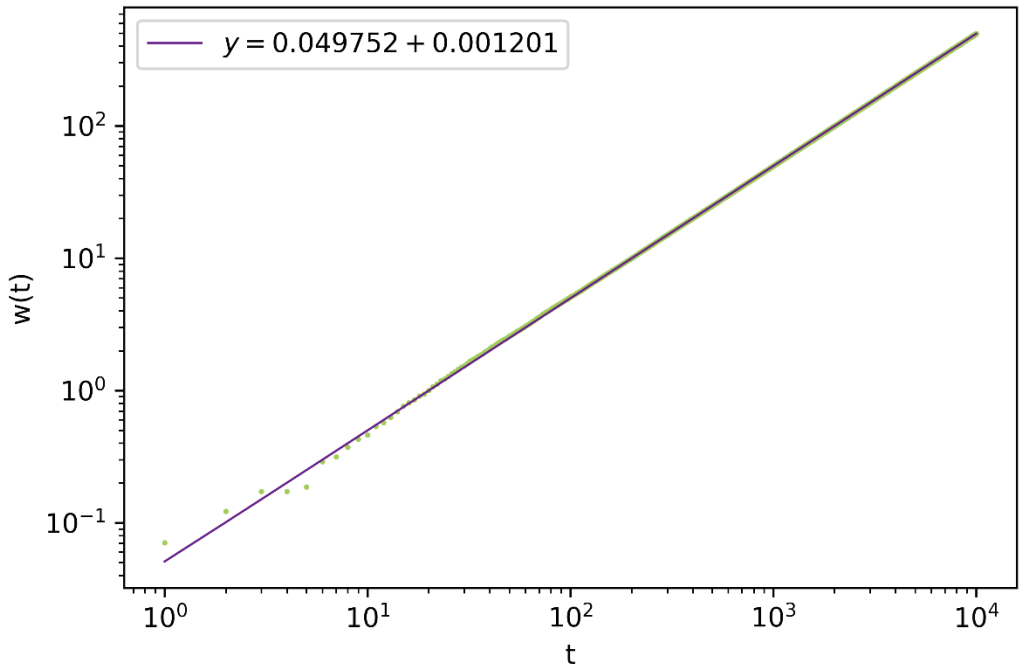
در بلاک بعدی از کد انحراف از میانگین در زمان های به قدم 1 که در آرایه w قرار گرفته اند. در بلاک بعدی از کد انحراف معیار بر حسب زمان رسم در مقیاس لگاریتمی رسم میشود تا شیب آن β به دست آورده شود. در شکل های زیر داده ها برای 5 بار تکرار شدند تا بتوانیم میانیگن شیب و خطای آن را مشخص کنیم.


$y = 0.049751 + 0.006183$
w(t)
t
10^{-1}
10^{0}
10^{1}
10^{2}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}


$y = 0.049743 + 0.048968$
w(t)
t
10^{-1}
10^{0}
10^{1}
10^{2}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}


$y = 0.049748 + 0.040063$
w(t)
t
10^{-1}
10^{0}
10^{1}
10^{2}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}


$y = 0.04975 - 0.017059$
w(t)
t
10^{-1}
10^{0}
10^{1}
10^{2}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}



طبق این داده ها $\beta = 0.049749 \pm 10^{-6}$ خواهد بود.

چند نکته:

- دربعضی نمودار های بالا ابتدای خط فیت شده انحنای کمی دیده میشود. با اینکه با چند دستور از کتابخانه های مختلف فیت کردن خط به روش کمترین مربعات بر داده ها رامتحان کردم باز این ناهمواری دیده میشود که فکر میکنم به علت یک به یک نبودن مقیاس دو محور و تراکم کم نقاط در این ناحیه از خط است. در بخشی که تراکم نقاط بالاست خط صافی رسم شده و مشکلی وجود ندارد.
- در هم فرورفتگی های شکل ول‌نشست را میتوان با فیکس کردن تعداد پیکسل های محور های نمودار به اندازه تعداد خانه های محور افقی در اندازه‌ی هر مربع کوچک نسبت به dpi مشخص شده و همینطور استفاده از آرگومان zorder (به نرتیب رنگ های درون لیست) حل کرد در ادامه نتایج استفاده از این دو روش آورده شده اند.در شکل زیر محور

عمودی تعداد ذره ها و محور افقی اندیس خانه است.