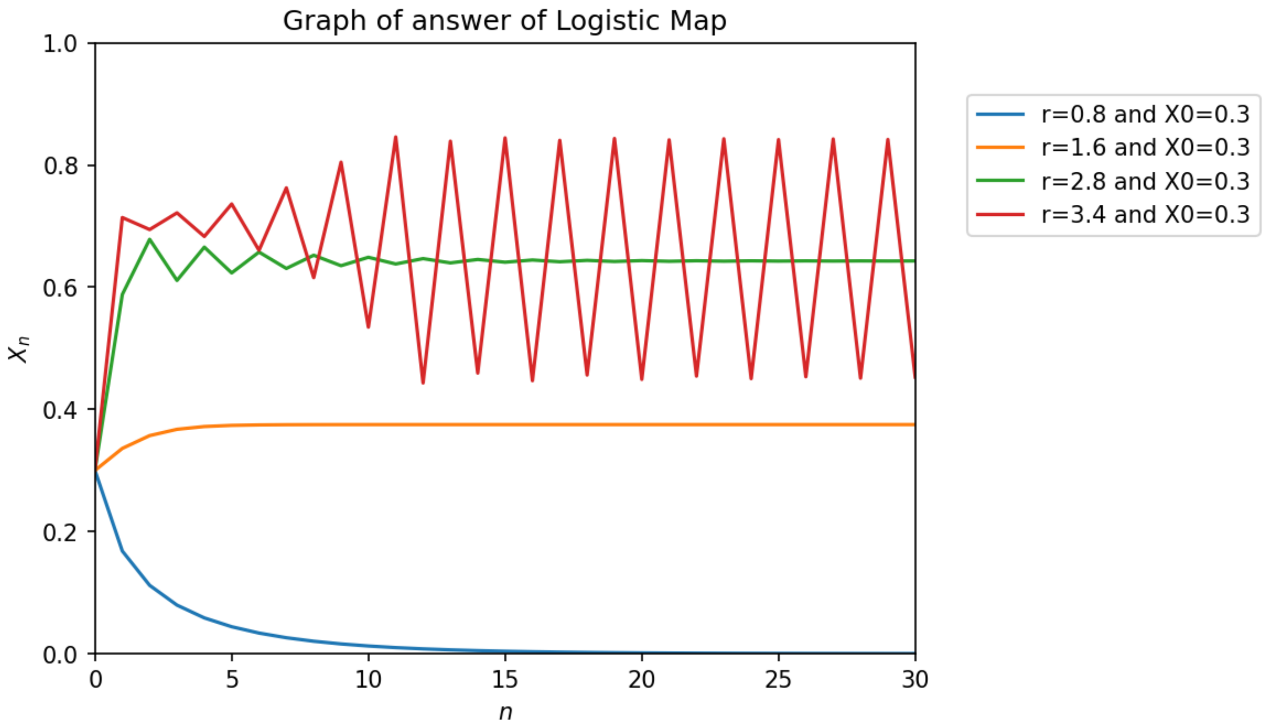
گزارش کار تمرین دهم شبیه‌سازی رایانه‌ای در فیزیک

علی اکرامیان – 99100563

در این تمرین من ابتدا یک شرایط اولیه x0 و یک تعداد نهایی N تعریف می‌کنم. حال تابع f(x,r) که همان نگاشت لوجستیک است را تعریف می‌کنم. پس از آن تابع اصلی برنامه که نگاشت را حل می‌کند به نام sol(N,x0,r) دارم که تعداد نهایی و مکان اولیه و مقدار r را می‌گیرد و مرحله به مرحله حل کرده و در هر بار حلقه حل می‌کند و نقطه‌ی آن را می‌کشد. طریقه‌ی حل نیز به این صورت است که چون رابطه‌ی X(n+1) را داریم بر حسب X(n) پس به راحتی با دادن شرایط اولیهx0 و انداختن تابع f(X[i],r) در یک حلقه می‌توان تا هر N دلخواه حل کرد! حال در بخش اول کد من صرفا به ازای چند مقدار r این تابع را با شرایط اولیه‌ی یکسال اجرا کرده‌ام تا نقش r در نگاشت لجستیک مشخص شود! در تصویر زیر واضح است که به ازای r های مختلف رفتارهای جالب و متفاوتی نشان می‌دهد!



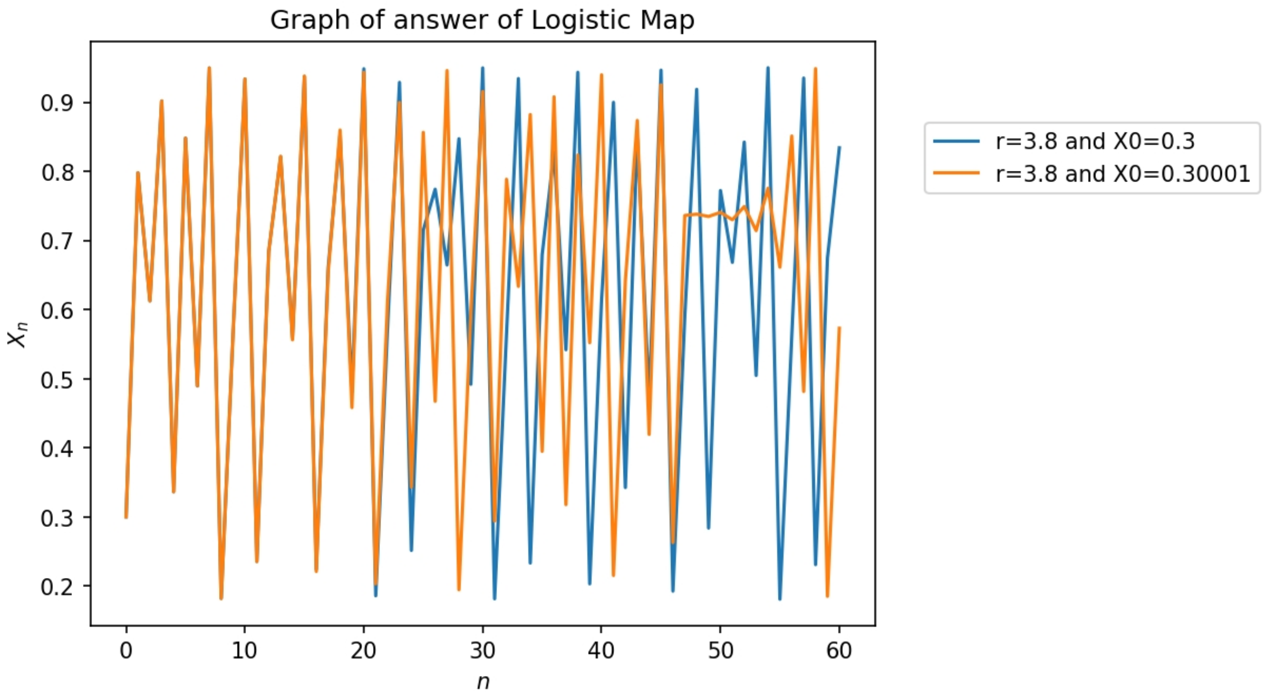
در ضمن من نگاشت لجستیک را به صورت آشناتر زیر گرفته ام در صورتی که در نگاشت گفته شده سر کلاس یک ضریب 4 تفاوت دارند که یعنی اسکیل r ها 4 برابر می‌شود. من در ابتدا با این شکل آشناتر کار کرده‌ام و سپس در انتها شکل‌ها را با این r گفته‌شده سر کلاس باز رسم کرده‌ام.
شکلی که من در ابتدا با آن کار کرده‌ام:

$$X_{n+1} = r\,X_n(1 - X_n)$$

و شکل گفته شده سر کلاس نیز به صورت زیر است:

$$X_{n+1} = 4\,r\,X_n(1 - X_n)$$

حال در قسمت بعد من حساسیت به شرایط اولیه را به ازای r های بزرگ نشان داده‌ام. که دو شرط اولیه‌ی 0.3 و 0.3 به علاوه‌ی delta داده‌ام که delta را نیز 0.0001 گرفته‌ام و شکل هر دو را رسم کرده ام و مشخص است که تا یک جایی منطبق اند ولی از یک جایی به بعد کاملا غیر قابل پیش‌بینی و دور از هم رفتار می‌کنند که یکی از نشان‌دهنده‌های وجود آشوب در سیستم است.
شکل را می‌بینیم:



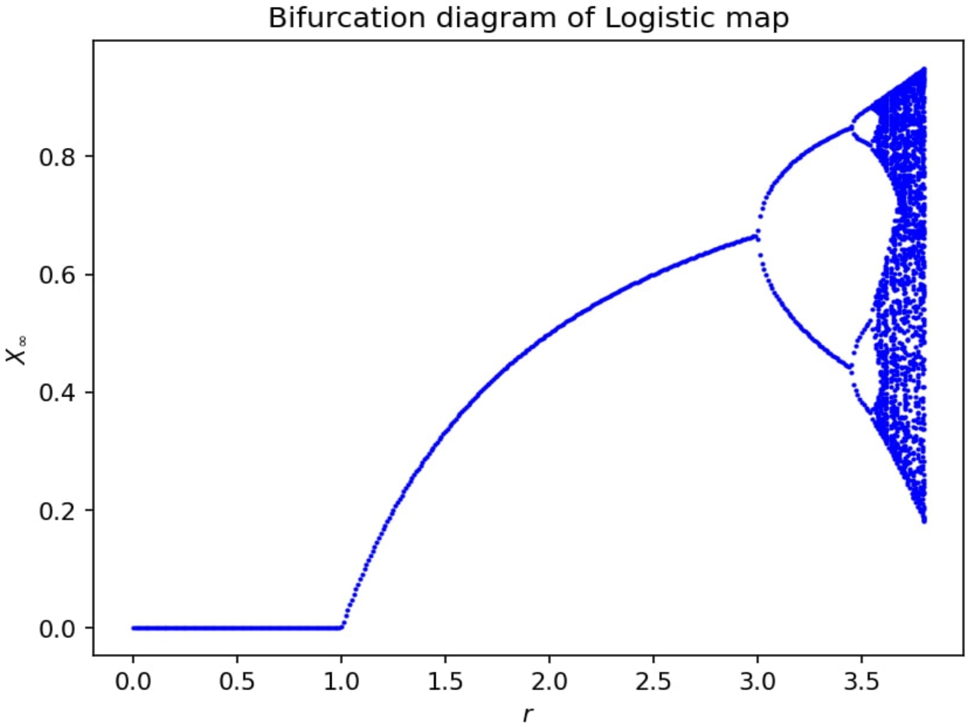
که در شکل واضح است به ازای این r داده شده تا تقریبا 20 مرحله n=20 روی هم منطبق اند ولی از بعد از 20 شروع به زیاد شدن اختلافات می‌کنند و در قدم‌های بیشتر دیگر رفتار هر کدام کاملا مستقل است.

در بخش بعد من نمودار دو شاخگی را رسم کرده‌ام که یک محور آن r و محور دیگر آن X∞ است. من شرایط اولیه را مانند بخش‌های قبل گرفته‌ام و یک N تعریف کرده‌ام که تا آن نگاشت را حل می‌کند و سپس یک Nt دارم که از Nt تا N را در نظر می‌گیرد که همان X∞ است (این کار را می‌کنیم که نوسانات اولیه در قدم‌های کمتر حذف شود و عملا جواب نهایی یا X∞ را درنظر می‌گیریم). حال به سراغ کد می‌رویم. تابع sol مقداری تغییر کرده است تا بتواند مکان‌هایی که در آن در حال نوسان و شاخه به شاخه شدن است را پیدا کند. تابع همان است فقط یک آرایه‌ی Y دارم که عددهای آرایه‌ی X است که از Nt تا N فقط درون آن است. حال چون می‌خواهیم محل‌هایی که مثلا دو تا مکان است یا چهار مکان یا بیشتر را پیدا کنیم ولی دقیقا مثل هم نیستند اعداد پس یک error تعریف می‌کنم که آن را 0.001 می‌گیرم تا با این دقت اعداد آرایه‌ی Y را اسکن کند و اعداد مجزا با این دقت را تحویل دهد. یک حلقه دارم که آن را از اینترنت کمک گرفتم که از تابع آماده‌ی np.isclose استفاده می‌کند و کاری که من می‌خواهم را انجام می‌دهد و خروجی آن که distinct است یک آرایه‌ی دوتایی یا چهارتایی و .... است که مکان‌هایی است که در r های بزرگ بین آن‌ها نوسان دارد به طور مرتب. حال یک حلقه میزنم روی این تعداد مکان‌های مجزا که بین آن‌ها نوسان دارد و آن‌ها را به ازای همان r رسم می‌کنم (یک نقطه به ازای هر کدام می‌گذارم).

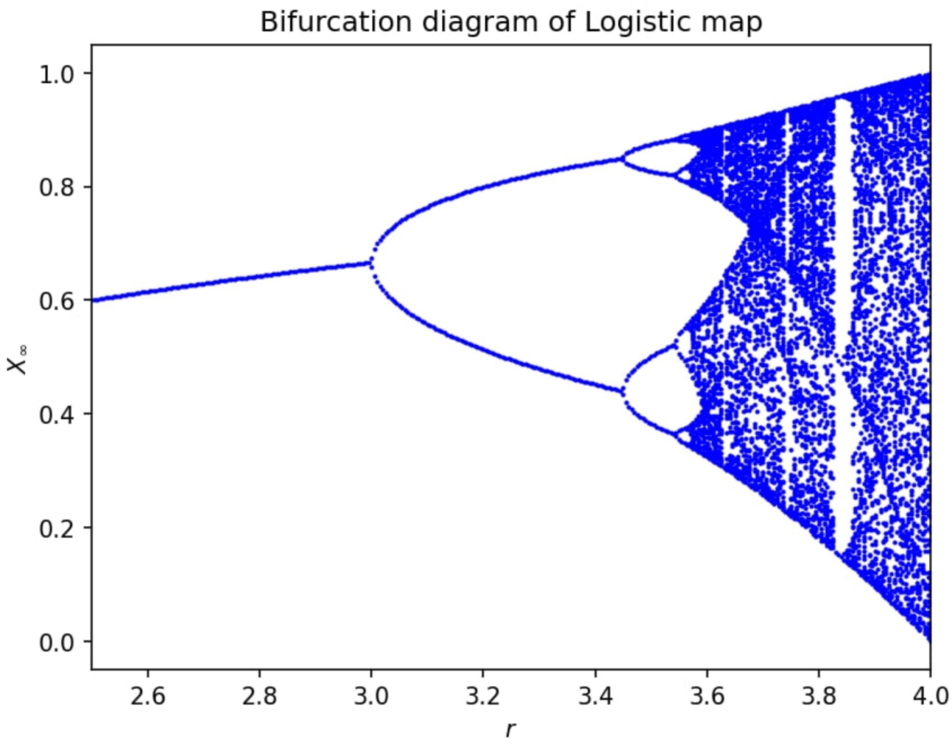
حال یک حلقه‌ی دیگر دارم که به ازای r های بین 0 تا r_max می‌تواند حل و رسم کند. من در این حلقه r را 0.002 در هر مرحله زیاد می‌کنم و تقریبا ران تایم آن 15 دقیقه است اگر تا r_max های کمتری حل شود یا این عدد 0.002 به مثلا 0.005 یا بیشتر تبدیل شود طبعا زمان کمتری می‌برد ولی شکل به خوبی این نمی‌شود. من دو بار این کار را انجام دادم و یک بار از r بین 0 تا 4 رسم کردم و یک بار در بازه‌ی 2.6 تا 4 که حالت زوم‌شده‌ی قبلی است.

شکل‌ها را می‌بینیم:

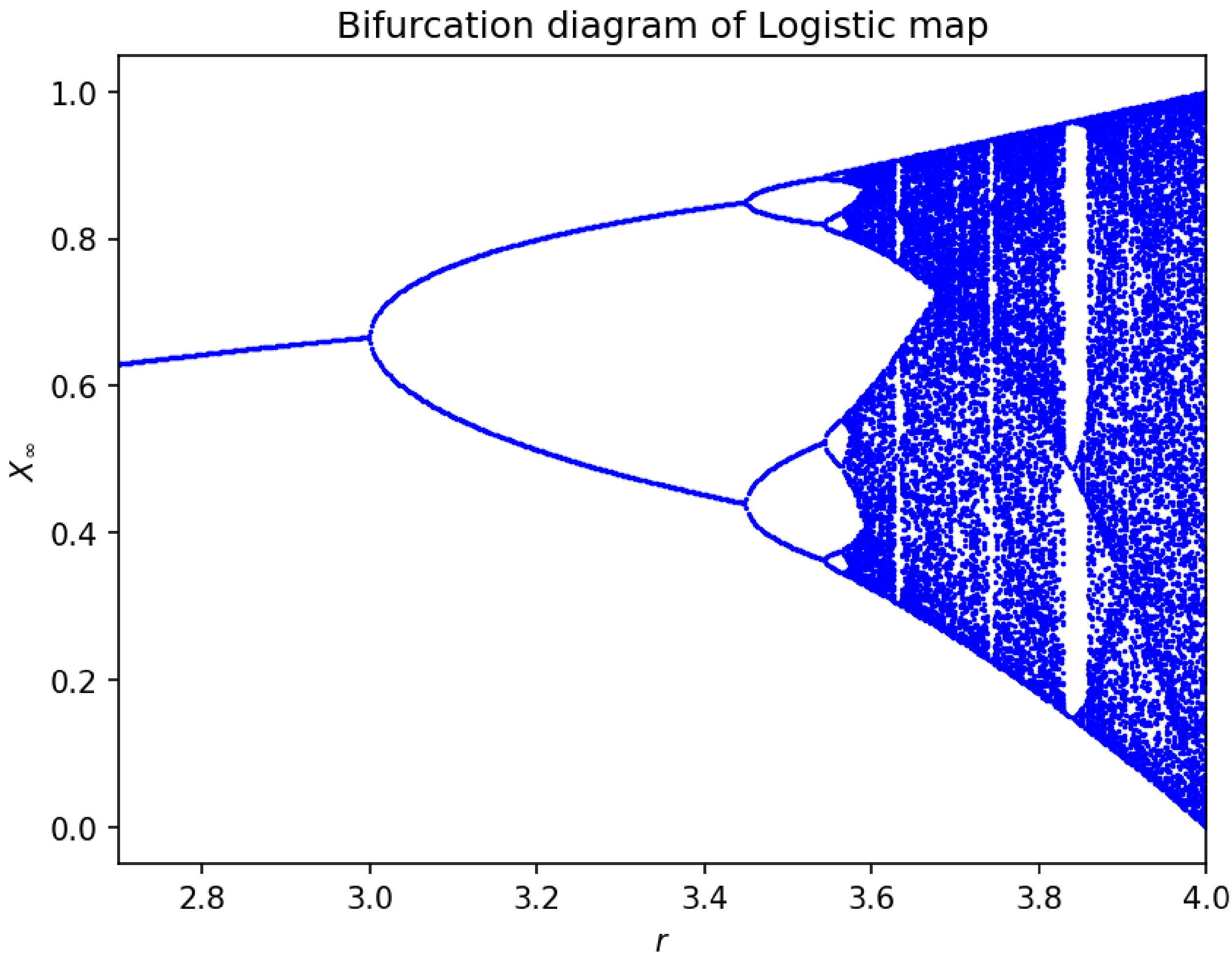
در بازه‌ی 0 تا 4 و افزایش r مساوی 0.005:



در بازه‌ی 2.6 تا 4 و افزایش r مساوی 0.005:



حال افزایش را 0.002 می‌کنم و می‌کشم:



که همه‌ی جزئیات در ان مشخص است از جمله دوشاخگی ابتدایی در r=3 و بعد از آن و پنجره‌ها و غیره که بسیار زیباست!

حال در بخش بعدی من ثوابت فیگنبام را محاسبه کرده‌ام که یکی نسبت r هایی است که در آن‌ها دوشخاگی‌ها رخ می‌دهد و یکی دیگر مقیاس بندی X∞ هاست. ثوابت فیگنبام به صورت زیر هستند که من اول δ را محاسبه‌کردم و سپس α را. تعاریفشان نیز به صورت زیر است:

$$\delta = \lim_{n\to\infty}\left(\frac{r_n - r_{n-1}}{r_{n+1} - r_n}\right) \qquad \alpha = \lim_{n\to\infty}\left(\frac{d_n}{d_{n+1}}\right)$$

حال پس نیاز داریم هایی که در آن‌ها چندشاخگی رخ می‌دهد را پیدا کنیم. تابع sol را تغییر می‌دهم به طوری که آرایه‌ی distinct را بازگرداند چون به آن نیاز داریم.سپس یک آرایه‌ی cut تعریف می‌کنم که چندشاخگی‌های 2 و 4 و 8 و 16 در آن است. حال یک حلقه روی تمام این آرایه می‌زنم و حلقه‌ی حل با r های مختلف را داخل آن می‌گذارم. دقت r را نیز 0.001 می‌گذارم (چون رسم ندارد ران تایمش کوتاه می‌شود!) و می‌گویم اگر طول آرایه‌ی distinct که معرف مکان‌های نوسانی سیستم بود مثلا از 1 به 2

تبدیل شد آن‌جا حل قطع شده و آن r را در آرایه‌ی cut_r بیاندازد و سپس همین کار را برای 2 به 4 و ... نیز انجام دهد. حال پس ما یک آرایه‌داریم که cut_r است و r هایی که در آن‌ها چندشاخگی رخ داده است را داریم. با این error و گام r که من گرفتم توانستم 4 تا ازین r های چندشاخگی را بدست بیاورم. با توجه به این که برای بدست آوردن δ نیاز داریم که سه تا ازین r های متوالی داشته باشیم و من چهارتا دارم پس می‌توانم دو بار δ را بدست آورده و میانگین بگیرم. همین کار را انجام می‌دهم و میانگین δ را نیز پرینت می‌کنم در آخر. خروجی‌های کد نیز همین ثابت δ و آرایه cut_r است که محل چندشاخگی‌ها را نشان می‌دهد.

خروجی کد را می‌بینیم:

```
Bifacturation r s:  [2.997 3.449 3.545 3.566]
Feigenbaum Constant =  4.6398809523809526
```

که باید 4.66 باشد حال خطای نسبی را محسابه می‌کنم:

(4.66 − 4.64) / 4.66 = 0.4 %

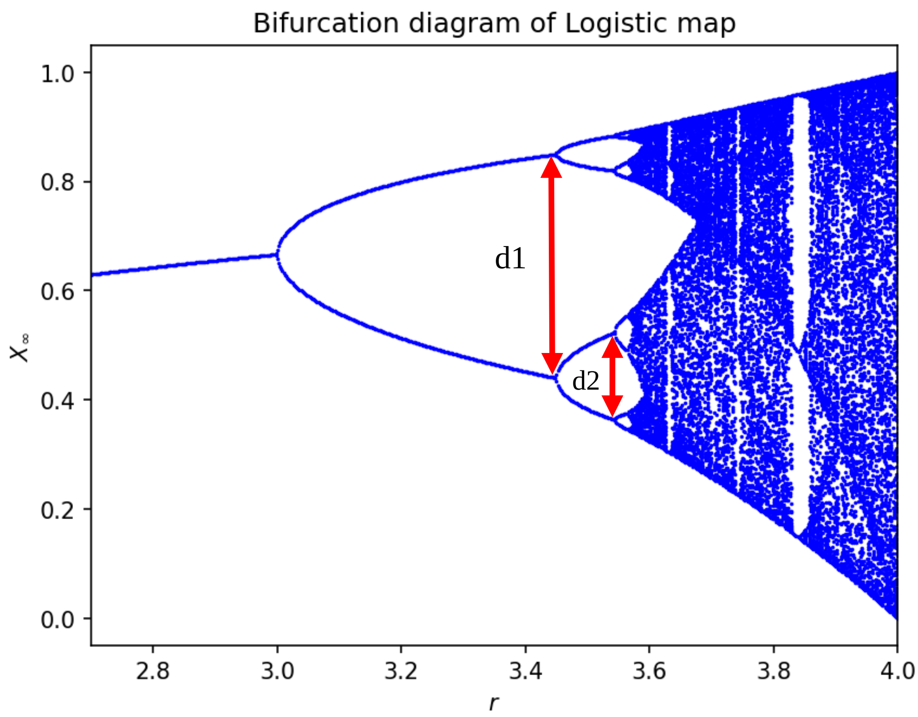
که مقدار قابل قبولی است!

حال ثابت α را بدست می‌آوریم.تنها تغییر در کد در این بخش نیز این است که حلقه‌ی حل به ازای r های متفاوت را یک تغییر می‌دهم که اگر طول آن distinct برابر با 4 یا 8 شود (دو بار این حلقه را گذاشتم تا یک بار به ازای 4 و یک باز به ازای 8 قطع شود) حل قطع شده و آرایه‌ی قبلی آن که فاصله‌ی دو نقطه‌ای که دو شاخگی‌شان (یا چهارشاخگی‌شان) از هم است را به من بدهد. سپس طبق رابطه α را محسابه می‌کنم. این بار فقط یک بار حساب کردم و ثابت را تا مقدار قابل قبولی نزدیک به واقیت بدست آوردم.

خروجی کد نیز α است که می‌بینیم:

```
Feigenbaum Constant α =  -2.5224769328366423
```

در شکل زیر واضح‌تر می‌گویم که کدام فاصله‌ها را برداشته‌ام:

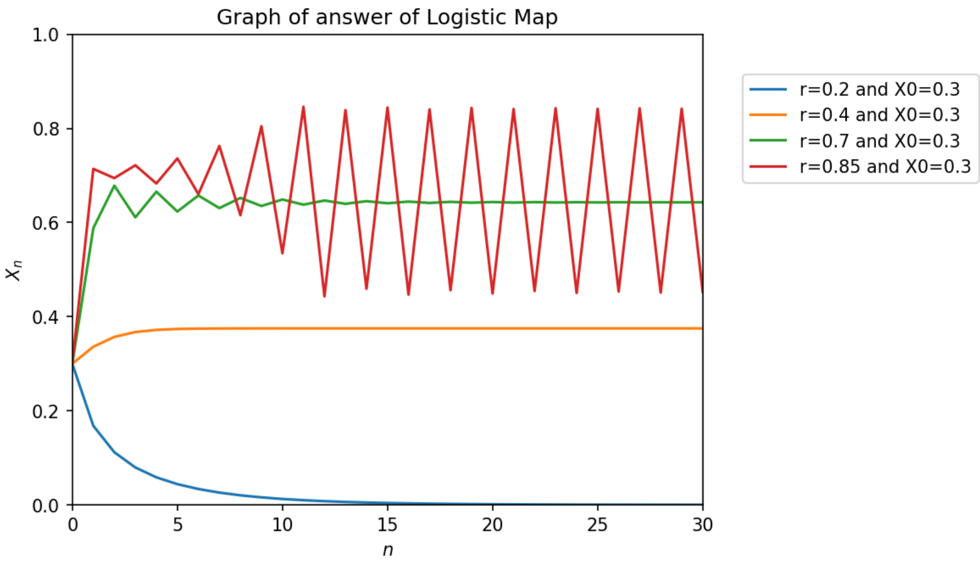


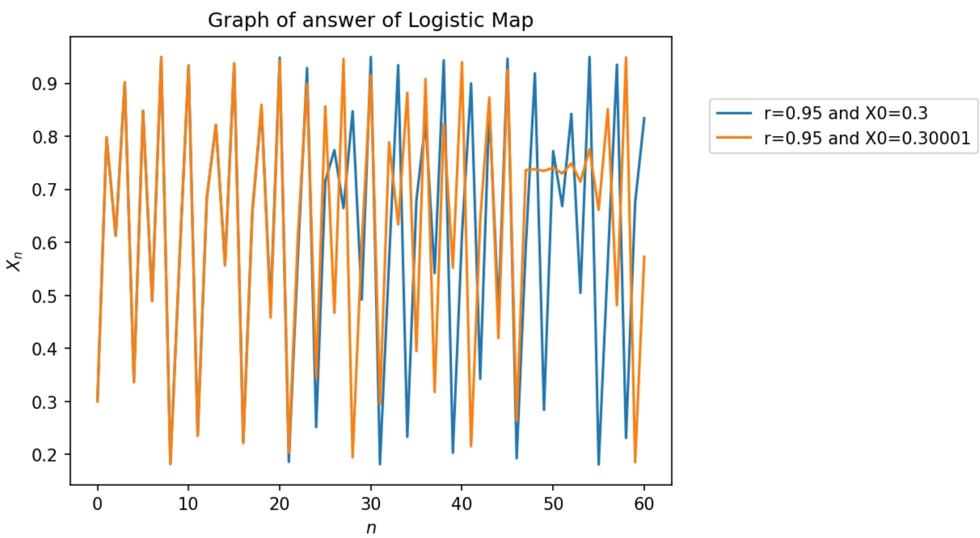
که علامت منفی نیز به خاطر برعکس شدن چاق و لاغر در شکل است.
پس ثوابت را این‌گونه بدست آوردم:

α = -2.522

δ = 4.64

در آخر نیز نگاشت لجستیک را به صورتی که سر کلاس گفته شده بود گرفتم و همه‌ی شکل‌ها را به ازای آن r ها بدست آوردم. بدیهی است که صرفا r ها بر 4 تقسیم شده و تفاوت دیگری ندارد. ثوابت نیز چون نسبت هستند تغییری نمی‌کنند (چون صورت و مخرجشان با یک ضریب اسکیل می‌شوند!). همه‌ی شکل‌هایشان را یک‌جا می‌آورم.





و نمودارهای چندشاخگی: